REDES NEURAIS PROFUNDAS PARA AUXÍLIO À TOMADA DE DECISÃO NO MERCADO DE AÇÕES

Gustavo Luiz Godoy Bichara

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Civil, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Civil.

Orientador: Alexandre Gonçalves Evsukoff

Rio de Janeiro
Março de 2019

REDES NEURAIS PROFUNDAS PARA AUXÍLIO À TOMADA DE DECISÃO NO MERCADO DE AÇÕES

Gustavo Luiz Godoy Bichara

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Examinada por:

_____

Prof. Alexandre Gonçalves Evsukoff, D.Sc.

_____

Profª. Beatriz de Souza Leite Pires de Lima, D.Sc.

_____

Prof. Jose Manoel de Seixas, D.Sc.

_____

Profª. Elaine Maria Tavares Rodrigues, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2019

Bichara, Gustavo Luiz Godoy

Redes Neurais Profundas para Auxílio à Tomada de Decisão no Mercado de Ações / Gustavo Luiz Godoy Bichara. – Rio de Janeiro: UFRJ/COPPE, 2019.

XIV, 104 p.: il.; 29,7 cm.

Orientador: Alexandre Gonçalves Evsukoff

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Civil, 2019.

Referências Bibliográficas: p. 86-92.

1. Aprendizado de Máquina. 2. *Deep Learning*. 3. *Ensemble*. 4. Previsão de Ações. I. Evsukoff, Alexandre Gonçalves. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Civil. III. Título.

Aos meus pais, Eliane e Mauro, por todo amor, trabalho e vida dedicados aos três filhos. Sem eles, essa história não existiria.

Aos meus irmãos, Junior e Renato, por todas as conversas e discussões que me fazem refletir e crescer como pessoa e profissional.

À minha noiva Roberta, pelos sonhos que edificamos diariamente.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por me permitir enxergar.

Aos meus pais, que sempre me acolheram nos momentos de dificuldade, me auxiliaram e me incentivaram a seguir em frente. Sempre foram duros nas minhas falhas e compreensivos nas minhas faltas.

À minha noiva, que sempre esteve ao meu lado me incentivando, ouvindo minhas incompreensíveis explicações sobre *machine learning*, e por ter contribuído na revisão deste trabalho.

Ao meu orientador Alexandre Evsukoff por ter acreditado no meu potencial, por todo conhecimento transmitido e pelo direcionamento ao longo das disciplinas e deste projeto de pesquisa.

À professora Beatriz Lima, que me recebeu muito bem desde o primeiro dia, sempre muito solícita, me auxiliou, orientou e ensinou muito durante todo o curso.

Ao amigo Carlos Salvador, por toda troca de conhecimento e experiências que contribuíram para me tornar uma pessoa melhor, acadêmica e espiritualmente.

Aos meus companheiros de jornada Carlos Eduardo Covas, Carlos Eduardo dos Anjos e Manuel Vargas. Muito aprendi com cada um deles.

Aos demais professores e colegas de COPPE que em algum momento me ofereceram o que há de mais precioso para os seres humanos: uma fração de seu tempo.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

REDES NEURAIS PROFUNDAS PARA AUXÍLIO À TOMADA DE DECISÃO NO MERCADO DE AÇÕES

Gustavo Luiz Godoy Bichara

Março/2019

Orientador: Alexandre Gonçalves Evsukoff

Programa: Engenharia Civil

Este trabalho apresenta a aplicação de uma rede neural profunda composta de Redes Neurais Convolucionais e Redes Neurais Recorrentes (*Long Short-Term Memory*) para extração semântica de notícias em língua portuguesa e processamento de indicadores técnicos da ação do Banco do Brasil ON (BBAS3). Aliado a isso, faz também o uso de filtros de entropia, baseados na entropia de Shannon, e combinação de modelos para criar um sistema de apoio à tomada de decisão de investimento. Ao longo da pesquisa, faz-se uma discussão das vantagens no uso de diferentes técnicas de Processamento de Linguagem Natural e do treinamento com janelas deslizantes. Os testes realizados indicam um elevado potencial para a aplicação conjunta destas técnicas, uma vez que atingem boa assertividade nas previsões dos movimentos da ação e conseguem retornos financeiros superiores aos benchmarks do mercado no período analisado.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DEEP LEARNING FOR DECISION SUPPORT IN STOCK MARKET

Gustavo Luiz Godoy Bichara

March/2019

Advisor: Alexandre Gonçalves Evsukoff.

Department: Civil Engineering

This work presents a Deep Learning application composed by Convolutional Neural Networks and Recurrent Neural Networks (Long Short-Term Memory) for semantic extraction from portuguese language news and technical indicators processing from Banco do Brasil stock (BBAS3). Furthermore, it applies entropy filters and ensemble methods to create an investment decision support system. Throughout this research, we discuss the advantages of using different techniques of Natural Language Processing and sliding window training. The results of experiments indicate a high potential for the joint application of these techniques, since they reach good assertiveness in stock movements predictions and achieve higher financial returns than market benchmarks during the period.

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

A capacidade de prever eventos futuros sempre instigou os seres humanos, principalmente quando isso permitiu a obtenção de uma vantagem competitiva. Seja ao fazer previsão de demanda para um determinado mercado, previsão do tempo para uma região ou previsão da evolução de cotações de ações, o que se espera é obter um bom suporte à decisão que conduza a resultados positivos, em termos de atendimento às demandas, otimização de produção ou realização de lucros.

No universo de indicadores econômicos e financeiros, uma boa acurácia nas previsões de inflação, crescimento do PIB, taxas de juros, cotações de ações, taxas de câmbio, entre outras, ajudam a nortear decisões de investimentos tanto de empresas quanto de governos. Observando-se mais especificamente o mercado de capitais, há grande interesse das empresas na evolução das cotações de ações, derivativos, câmbio e índices, pois isso influencia sobremaneira em captação de recursos para realização e proteção de seus investimentos, mediante a emissão de novas ações, debêntures, *hedge* ou pelo uso de outros mecanismos financeiros [1]. Some-se a isso os investidores - pessoas físicas e jurídicas - que buscam no mercado de capitais boas opções de investimento para aumento patrimonial, renda extra, formação de previdência ou apenas como capital especulativo, criando-se, desta forma, o ambiente ideal para negociação de ativos com elevada liquidez e grande variedade de relação risco/retorno.

Neste cenário, os participantes do mercado que estiverem de posse das melhores previsões do movimento de ações terão vantagens sobre os demais participantes. E para isso, os diversos agentes do mercado utilizam alguns mecanismos que auxiliam no seu processo de tomada de decisão:

- Análise Fundamentalista: trata de estudos de fundamentos econômicos do mercado, setor de atuação da empresa, indicadores econômicos e financeiros, análise de balanços, fluxos de caixa e demonstrativos de resultados com o objetivo de identificar o verdadeiro valor de um ativo e se este está sendo

negociado com valores depreciados ou apreciados [1]. É muito usado por grandes investidores, como fundos de pensão, grandes empresas, bancos, fundos de investimento e também por pessoas físicas.

- Análise Técnica: engloba uma ampla variedade de formas de estudar as oscilações dos preços dos ativos usando gráficos que explicitam os comportamentos de preços, volumes e indicadores de momento e tendência que auxiliam na tomada de decisão de compra e venda [2]. Tem entre os maiores usuários os pequenos investidores, *traders*, analistas de investimentos e profissionais de mesa de operações.

Dadas as suas características, não se justifica a aplicação da análise fundamentalista no curto prazo, uma vez que se baseia em indicadores econômicos divulgados mensal ou trimestralmente e dados das empresas disponibilizados trimestralmente. Neste contexto, a análise técnica apresenta maior vantagem ao usar as flutuações de preços e volume financeiro diários ou *intraday* para gerar seus próprios indicadores.

Além disso, as oscilações de curto prazo estão mais associadas às informações, privilegiadas ou não, que cada indivíduo possui sobre uma ação [3], enquanto as estratégias de longo prazo são motivadas por uma visão de valor da empresa [4]. Já a Hipótese dos Mercados Eficientes (*Efficient Market Hypothesis* - EMH) estabelece que os preços das ações refletem toda informação disponível até aquele momento e, por isso, estabelece que é impossível obter retornos superiores à média do mercado por meio de previsões, uma vez que os mercados não reagem a novas informações [5][6]. Em consonância com a EMH, a Teoria do Passeio Aleatório (*Random Walk Hypothesis - RWH)* afirma que a evolução dos preços das ações segue um passeio aleatório e, consequentemente, não pode ser previsto [7].

Entretanto, com o uso de técnicas de *machine learning*, como redes neurais artificiais, aliadas à análise técnica, estudos publicados apresentaram boa assertividade na previsão dos movimentos do mercado [8][9], bem como maiores retornos e maior estabilidade nas previsões dos movimentos das ações usando redes neurais profundas e notícias publicadas nos meios de comunicação [10]-[16].

Esses resultados devem-se ao crescente avanço dos algoritmos de *machine learning*, da capacidade de processamento das máquinas e da facilidade de aquisição de dados. Entre os algoritmos mencionados, destacam-se as Redes Neurais Recorrentes (*Recurrent Neural Networks- RNN*), em especial as redes *Long Short-Term Memory* (LSTM), que se sobressaem no processamento de séries temporais [17][18], e as Redes Neurais Convolucionais (*Convolutional Neural Network – CNN*), que vêm apresentando resultados diferenciados na obtenção da semântica de textos [19]-[22].

Outros trabalhos também obtiveram resultados expressivos usando diferentes tipos de algoritmos, como classificação Bayesiana [23][24], Máquinas de Vetores de Suporte [25]-[28] e Máquinas de Boltzmann Restritas [29]

Diante disso, este trabalho busca aprofundar a pesquisa e aplicação de técnicas de *machine learning* em previsão dos movimentos das ações, variando técnicas de pré-processamento dos dados e formas de treinamento, bem como visa a aplicar filtros baseados na entropia de Shannon aos resultados e também verificar as vantagens do uso da combinação de modelos.

1.1. Motivação

Atualmente, no Brasil, o mercado de ações se desenvolve em torno da B3 (Brasil, Bolsa, Balcão), resultado da fusão entre a Bolsa de Valores de São Paulo, a Bolsa de Mercadorias e Futuros e a Companhia de Custódia e de Liquidação Financeira de Títulos, o que a torna a 5ª maior bolsa de mercado de capitais do mundo, com cerca de 360 empresas listadas, mais de dois mil papeis negociados, 1,2 milhões de negócios diários e mais de 12 bilhões de reais movimentados diariamente em 2018 [30].

Em vista disso, identificar uma boa oportunidade de negócios torna-se uma tarefa árdua, apesar de lucrativa, para investidores individuais. Devido à grande quantidade de índices, ativos e derivativos, muitos dados são gerados a partir das operações realizadas diariamente, necessitando-se, então, de um modelo robusto e com capacidade para

processar estes dados, com a finalidade de auxiliar no processo de identificação de oportunidades e tomada de decisão.

No entanto, séries temporais financeiras são séries não-lineares e não-estacionárias, e, por isso, de difícil previsão, não sendo adequadamente representadas por modelos lineares. Com isso, opta-se pelo uso de modelos não-lineares, com capacidade de captar tal evolução temporal, o que direciona aos algoritmos de *machine learning*, mais especificamente às redes CNN e LSTM.

Diante da capacidade dos algoritmos de Redes Neurais Profundas em processar grande quantidade de dados e entregar bons resultados [10]-[16], estes se apresentam como uma opção interessante para um trabalho aplicado ao mercado brasileiro. Sendo assim, a presente pesquisa usa como base o modelo SI-RCNN aplicado por VARGAS, *et al* [16] na previsão dos movimentos das ações da Chevron, apresentado na Figura 1.

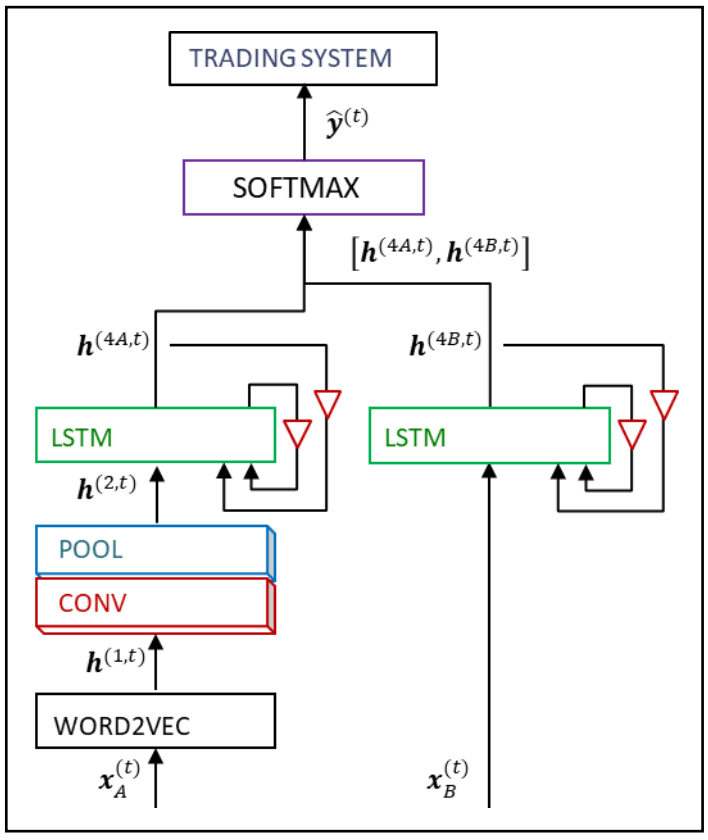

Figura 1 – Modelo SI-RCNN aplicado por VARGAS, *et al* [16]. (Adaptado de VARGAS, *et al* [16])

O modelo SI-RCNN fez uso de notícias em inglês vetorizadas em *word2vec-cbow* com 300 dimensões, que são enviadas para um conjunto de três filtros convolucionais ([3x64], [4x64], [5x64] e deslocamento 1), *pooling* (tamanho 2 e deslocamento 2), concatenados e enviados uma LSTM de 128 unidades. Paralelamente, um conjunto de 7 indicadores técnicos é enviado para uma LSTM de 128 unidades. As saídas dessas duas redes LSTM intermediárias são, então, concatenadas, passam por uma camada totalmente conectada e seguem para uma saída binária ([1,0] = compra e [0,1] = venda) com função de ativação *softmax* [16].

Com isso, pode-se verificar a aplicabilidade desses métodos à realidade do mercado nacional: preços com elevada volatilidade e notícias em língua portuguesa. Adicionalmente, busca-se implementar algumas melhorias ao modelo, com foco na obtenção de maior confiabilidade nos resultados.

1.2. Objetivos

Ante o exposto, este trabalho visa a desenvolver um estudo de *machine learning* para previsão de movimentação dos preços de ações no mercado financeiro, com o objetivo de obter uma saída que identifique a oscilação provável da cotação de um ativo (positiva, neutra ou negativa) e seu grau de certeza para o dia seguinte ao período analisado, auxiliando assim a tomada de decisão.

Uma vez que o volume de ativos e dados disponíveis requerem elevada capacidade computacional para sua execução, optou-se pelo desenvolvimento de um modelo com escopo reduzido, contendo apenas uma ação e dados diários, mas observando sempre características que permitam sua escalabilidade para muitos papéis, histórico maior que 10 anos ou intervalo de dados em horas ou mesmo em minutos. Dessa forma, o trabalho de filtragem automática de oportunidades seria uma consequência da adoção em larga escala deste modelo, o que está fora do escopo deste trabalho.

Para se chegar a um resultado de elevada confiabilidade, este trabalho busca transitar por algumas técnicas de pré-processamento, treinamento e pós-processamento, de modo

a identificar as mais adequadas a este tipo problema. Sendo assim, objetiva-se responder aos seguintes pontos principais:

- Verificar a aplicabilidade do modelo SI-RCNN a uma ação do mercado brasileiro;
- Verificar se há vantagens no uso de notícias (apenas título) textuais em língua portuguesa sobre o modelo sem notícias;
- Avaliar as diferenças de um treinamento dinâmico de janela deslizante (*sliding window*) sobre o treinamento estático usado no modelo SI-RCNN (apresentados e discutidos na seção 6.4);
- Aplicar e analisar as potencialidades do *ensemble* de modelos sobre o uso de modelos individuais;
- Identificar se a aplicação da entropia de Shannon aos resultados dos modelos conduz a maior assertividade nas operações, condicionando a execução de uma operação a um elevado grau de certeza na saída da rede.

Além disso, dentre as técnicas de pré-processamento textual, o trabalho faz uso de dois algoritmos de vetorização de palavras, testa a presença ou ausência de *stopwords* e faz pequenas variações na filtragem de palavras por frequência, visando a identificar se alguma delas apresenta vantagens sobre as demais, sendo estes considerados objetivos intermediários.

1.3. Estrutura do Trabalho

No Capítulo 1 faz-se uma introdução ao problema em questão, apresenta-se as motivações para o estudo, os objetivos e os trabalhos publicados que possuem relação e contribuíram com o desenvolvimento desta pesquisa. A Figura 2 ilustra o fluxo de trabalho seguido e a complexidade do modelo.

O Capítulo 2 apresenta o Mercado Financeiro, seus conceitos gerais e as particularidades do Mercado de Ações. Em seguida, fala-se sobre a Hipótese dos

Mercados Eficientes e como a Análise Fundamentalista e a Análise Técnica confrontam essa hipótese. Este capítulo é base para a apresentação dos indicadores usados no pré-processamento dos dados.

No Capítulo 3 é feita uma explanação sobre o Processamento de Linguagem Natural, seus desafios e formas de realizar o tratamento do texto e a representação vetorial de palavras. Posteriormente, são apresentados os dois métodos de vetorização de palavras que são usados na presente pesquisa. Esse corpo teórico fundamenta toda a linha de pré-processamento das notícias.

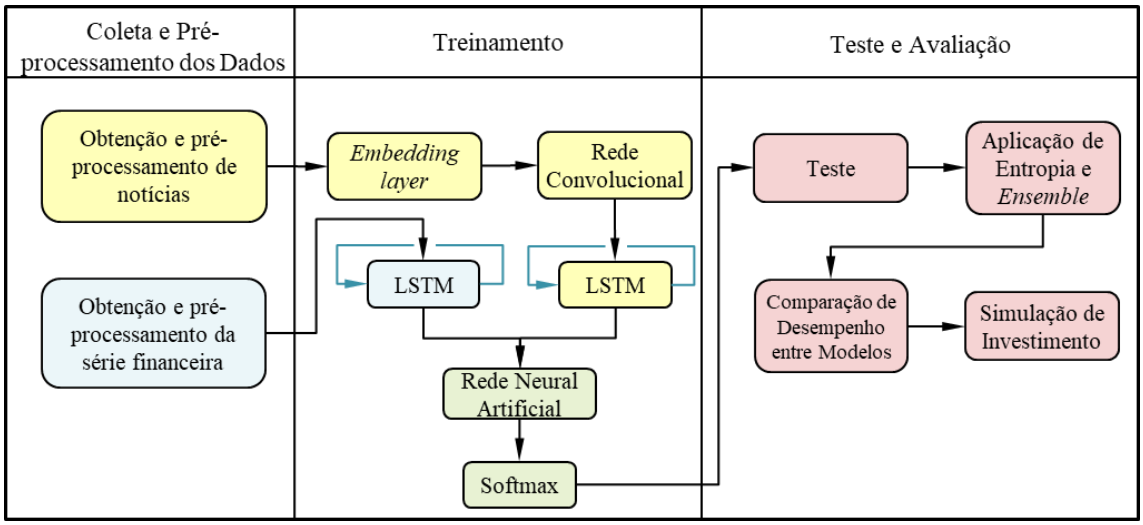

Figura 2 – Diagrama simplificado do modelo de previsão para auxílio à tomada de decisão. (Elaborado pelo autor)

Em seguida, o Capítulo 4 traz alguns conceitos do ramo conexionista do aprendizado de máquina, essencial para a construção do modelo da coluna de treinamento da Figura 2. Inicialmente, são apresentadas as Redes Neurais Artificiais, sua forma de treinamento e suas limitações, para depois se introduzir as Redes Neurais Recorrentes e sua facilidade em trabalhar com séries temporais. Por fim, faz-se uma explanação sobre as Redes Neurais Convolucionais, sua importância para capturar a semântica no tratamento de textos e sua capacidade de fazer uma precisa análise de sensibilidade.

No Capítulo 5 são introduzidos os conceitos da Teoria da Informação, da Entropia de Shannon e como estes se aplicam aos problemas de *machine learning*. Em seguida,

apresenta-se a técnica de *Ensemble* de Modelos, sua aplicação e as principais vantagens sobre o uso de modelos individuais. Estes conceitos são aplicados na etapa final do trabalho, após os testes, como forma de buscar maior assertividade do modelo.

Já o Capítulo 6 permeia todo o diagrama da Figura 2, trata em detalhes a construção do modelo e dos cenários usados no presente trabalho, mostrando desde a obtenção dos dados e seu pré-processamento, passando pela arquitetura do modelo e formas de treinamento e chegando aos cenários analisados e às medidas de avaliação de resultados.

Por fim, o Capítulo 7 apresenta os resultados dos cenários executados e discute-se as vantagens de cada abordagem que foi seguida no trabalho. O Capítulo 8 traz a conclusão acerca das questões colocadas previamente no Capítulo 1, bem como propostas de trabalhos futuros.

Complementando o trabalho, são apresentadas as Referências Bibliográficas usadas para embasar a pesquisa: o Anexo A com informações adicionais sobre os cenários que fizeram parte da etapa inicial do estudo e, em seguida, foram descontinuados ao longo da evolução dos resultados, e o Anexo B que apresenta em maiores detalhes como ocorre o treinamento de vetores de palavras nos métodos GloVe e *word2vec*.

2. Mercado Financeiro

Em linhas gerais, o Sistema Financeiro Nacional (SFN) é formado por um conjunto de entidades e instituições que promovem a intermediação financeira entre agentes econômicos. Este sistema é constituído pelos Órgãos Normativos (CMN, CNSP e CNPC), cuja função é estabelecer regras gerais para o funcionamento do SFN, pelos Órgãos Supervisores (BCB, CVM, Susep e Previc), que se destinam a trabalhar para que os integrantes do sistema financeiro sigam as regras definas pelos Órgãos Normativos, e pelos Órgãos Operadores (Bancos, corretoras de valores, bolsas de valores, entre outros), que são as entidades que fazem a intermediação financeira, trabalhando diretamente com os demais agentes econômicos [31].

As interações entre as entidades do SFN e os demais agentes econômicos formam o que se convencionou chamar de Mercado Financeiro. Este mercado é o ambiente onde se realiza o intercâmbio de ativos financeiros e se determinam seus preços, por meio da transferência de recursos financeiros dos agentes superavitários para os deficitários [32].

O Mercado Financeiro se desenvolve de forma segmentada, em quatro subdivisões [32]:

- Mercado Monetário: onde ocorrem as operações de oferta de moedas e taxas de juros de curtíssimo e curto prazos para garantir a liquidez da economia;
- Mercado de Crédito: onde ocorrem as operações de empréstimos, arrendamentos e financiamentos de curto e médio prazos entre superavitários e deficitários;
- Mercado de Capitais: tem por finalidade suprir o financiamento de projetos e empresas a médio e longo prazos, onde são realizadas as ofertas públicas de ações, securitização de recebíveis, mercado de bônus, debêntures, operações de *hedge* entre outras;
- Mercado Cambial: onde são feitas as operações de conversão de moedas.

Dentre as atividades desenvolvidas no mercado de capitais, destacam-se as operações com ações, onde, inicialmente, empresas realizam suas ofertas públicas de ações (IPO)

para financiar seus projetos, aquisições ou operações e, posteriormente, essas ações passam a ser negociadas em um mercado secundário. No Brasil, essas operações são realizadas na B3.

2.1. O Mercado de Ações

Por definição, uma ação é um título negociável, emitido por uma Sociedade Anônima, que representa a propriedade da menor fração do seu capital e que dá direito, ao seu titular, de participar da vida da empresa.

Após a realização de um IPO, as ações de uma empresa passam a ser negociadas na Bolsa de Valores e são acessíveis ao público em geral, seja pessoa física ou jurídica, e seguem algumas regras de negociação, sendo as principais - e que são importantes para o escopo deste trabalho – apresentadas a seguir [33]:

- Os negócios são fechados apenas quando comprador e vendedor estão de acordo com o preço, ou seja, os volumes e preços de compra e venda são exatamente iguais;

- Existem dois tipos de ordem: limitada, quando se deseja um preço específico pelos ativos, seja na compra ou na venda, sem garantia de execução imediata; e ordem a mercado, quando se deseja a execução imediata da operação, ou seja, comprar no melhor preço de venda ou vender no melhor preço de compra;

- Uma ordem limitada de compra ou de venda é definida por uma quantidade de ações e um preço de negociação. Ofertas de compra abaixo dos preços de venda ficam disponibilizados em um livro de ofertas, da mesma forma que as ofertas de venda acima dos preços de compra (Tabela 1).

- Uma ordem limitada de compra ou de venda entrará no final da fila na sua faixa de preço no livro de ofertas. Ou seja, no livro de ofertas da Tabela 1, a maior oferta de compra é 53,36 e há 200 ações sendo demandadas. Um agente que

deseje comprar 100 ações a 53,36 entrará no final da fila neste preço e só terá sua ordem executada se houver 300 ações oferecidas na venda a esse preço ou se os compradores da frente da fila retirarem suas ordens.

- Diariamente a bolsa de valores realiza os leilões de abertura e de fechamento do mercado no qual aceita ordens de compra e venda durante um período (15min na abertura e 5min no fechamento) que não são executados imediatamente e são acumulados de forma a compor os preços de abertura e de fechamento, respectivamente.

Tabela 1 – Exemplo de um livro de ofertas para a ação BBAS3 (Fonte dos dados: B3. Tabela elaborada pelo autor)

LIVRO DE OFERTAS					
Ativo	**Último**	**Variação**	**Hora**	**Volume**	**Negócios**
BBAS3	53,39	**0,81%**	10:34:10	38.10 M	1814
COMPRA			**VENDA**		
Qnt. Ofertas	**Qntd Ações**	**Preço**	**Preço**	**Qntd Ações**	**Qnt. Ofertas**
2	200	53,36	53,37	600	2
1	500	53,35	53,38	500	1
2	1700	53,34	53,39	1200	4
2	1600	53,33	53,40	1300	4
8	3800	53,32	53,41	2200	4
5	1400	53,31	53,42	4000	7
9	5600	53,30	53,43	4000	7

Essas regras básicas são restrições importantes que devem ser consideradas em qualquer algoritmo que se preste a fazer previsão de ações, seja para operações automatizadas ou para auxílio à tomada de decisão.

Mediante as operações realizadas diariamente na bolsa de valores, as ações possuem, no término do dia, 7 (sete) dados essenciais, sendo 4 (quatro) de preço: abertura, fechamento, máxima e mínima; e três de volume: número de negócios, volume de ações negociado e volume financeiro negociado. A partir desses dados, muitos indicadores técnicos e fundamentalistas são elaborados e usados por analistas do mercado.

A Figura 3 apresenta o gráfico diário com a evolução de preços das ações ordinárias do Banco do Brasil (BBAS3) ao longo de um período de 7 meses, onde é possível visualizar o comportamento do preço, bem como o resumo dos principais dados obtidos ao final de um dia de operações.

Os indicadores de que este estudo trata usam os preços de abertura, fechamento, máxima e mínima dos dados diário, não considerando as interações *intraday*. Ou seja, não há nenhuma informação sobre em que preço o ativo estava a cada momento ao longo do dia, com exceção da abertura e do fechamento do mercado.

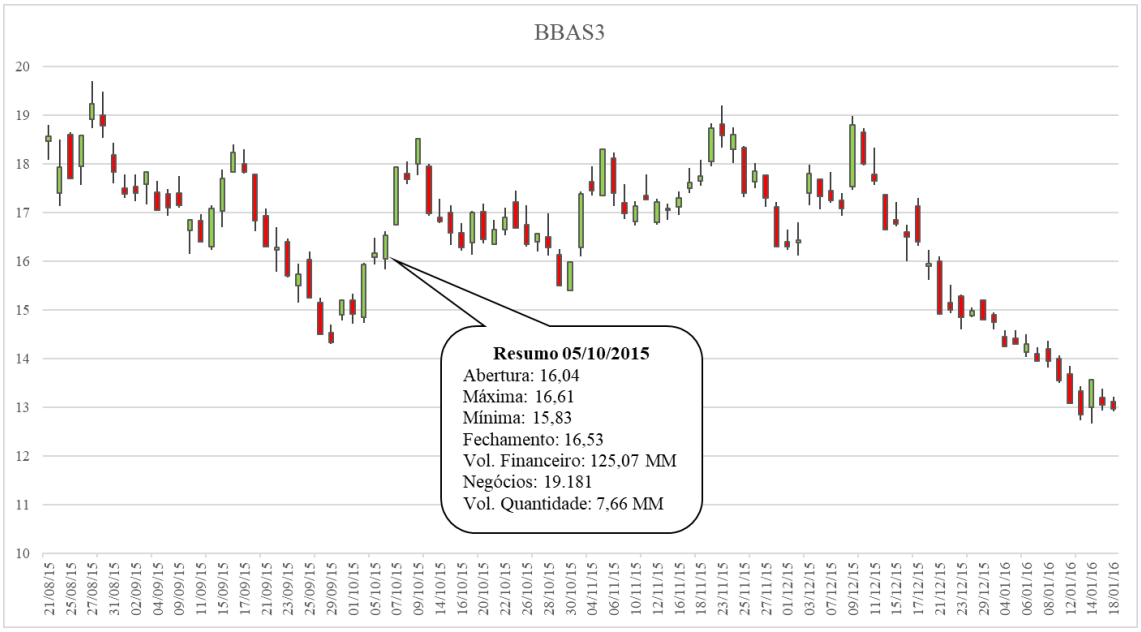

Figura 3 – Exemplo de um gráfico diário para a ação BBAS3 com um resumo dos 7 (sete) dados principais para um dia de operação. (Fonte dos dados: B3. Gráfico elaborado pelo autor)

Consequentemente, em uma análise de dados diários históricos, o uso de ordens limitadas impossibilita saber se a operação seria executada ou não, impedindo uma avaliação do modelo. Sendo assim, a única maneira de garantir que uma ordem seja executada a um preço conhecido, sem enviesar o resultado do modelo, é com a realização das operações a mercado exclusivamente nos leilões de abertura e de fechamento.

2.2. Hipótese dos Mercados Eficientes

A teoria do *Random Walk* popularizada por MALKIEL em 1973 [7] teve por base a formulação de Eugene Fama de 1965 [5], na qual este concluiu, com base em dados empíricos, que as movimentações nos preços das ações são imprevisíveis, seguindo uma distribuição probabilística. Mais tarde, em 1970, FAMA [6] propôs a Hipótese dos Mercados Eficientes em que estabelece que os preços das ações refletem toda informação disponível até aquele momento, podendo ser de três formas:

- Hipótese fraca: os preços negociados refletem toda informação histórica disponível publicamente até o momento;
- Hipótese semiforte: os preços refletem as informações históricas e se ajustam rapidamente para refletir as novas informações públicas disponíveis;
- Hipótese forte: os preços refletem as informações públicas e também as informações privilegiadas obtidas por pequenos grupos monopolistas.

Com base em seus estudos empíricos, FAMA [6] estabeleceu que é impossível para um investidor obter resultados superiores ao mercado em geral por um longo período de tempo, e que a única maneira possível de obter retornos superiores seria ao acaso.

O presente trabalho tem a contribuir para essa discussão, uma vez que faz uso de algoritmos de Redes Neurais Profundas para o processamento de notícias com o objetivo de prever as movimentações futuras das ações e, consequentemente, obter resultados superiores aos do mercado em geral.

2.3. Análise Fundamentalista

A análise fundamentalista é um campo de estudos que busca encontrar os fundamentos econômicos e financeiros por trás do valor dos ativos, partindo da premissa que o preço da ação pode estar muito apreciado ou depreciado com relação ao seu valor, o que indicaria uma boa oportunidade de compra ou de venda.

Para essa análise, são considerados os dados do mercado em geral, como perspectivas de PIB, câmbio, taxas de juros e inflação, o setor de atuação da empresa, bem como suas perspectivas de crescimento, e indicadores econômicos de uma região, para identificar o ambiente em que a empresa está atuando. Após isso, faz-se uma análise detalhada dos balanços trimestrais, demonstrativos de fluxos de caixa e demonstrativos de resultados, buscando-se identificar a evolução das receitas e das despesas, margens operacionais e líquidas, níveis de endividamento, perfil de dívida, dentre outros detalhes que vão auxiliar na identificação do valor do ativo [1][4].

Com base nesses dados, diversos indicadores podem ser gerados, como, por exemplo, as relações preço sobre lucro e preço sobre valor patrimonial, o retorno sobre o capital investido, o retorno sobre o patrimônio líquido e o *dividend yield* (dividendo pago por ação). Aliado a isso, há também os métodos de precificação da ação, como o do fluxo de caixa descontado e dos dividendos descontados [1][4].

Apesar de sua grande relevância e de ser amplamente usada pelos grandes fundos de investimento, corretoras, bancos e fundos de pensão, a análise fundamentalista não faz parte do escopo deste trabalho.

2.4. Análise Técnica

A análise técnica engloba uma ampla variedade de formas de estudar as oscilações dos preços dos ativos, partindo da premissa que o movimento dos mercados reflete o comportamento coletivo dos agentes econômicos - incluindo os momentos de euforia, desespero e tranquilidade - usando ferramentas que explicitam os comportamentos de preços, volumes e indicadores de momento e tendência, que auxiliam na tomada de decisão de compra e venda [34]. As duas principais formas dessa análise são o *Tape Reading* e a Análise Gráfica.

O *Tape Reading* foi uma técnica muito usada no início do século XX que se baseava na leitura da fita impressa (daí o seu nome) com as negociações executadas, para analisar a dinâmica dos preços, níveis de agressão dos compradores e vendedores, volumes

negociados e livro de ofertas para, a partir disso, determinar em qual direção está o fluxo do mercado.

Já na Análise Gráfica, usa-se um ou mais gráficos, como o da Figura 3, onde os 4 (quatro) principais valores de preço são identificados em barras ou *candlesticks*. A partir destes dados, são calculados e plotados indicadores que sinalizam linhas de suporte e resistência, tendência de alta ou de baixa, reversão de tendência ou zona de consolidação [2][34]. Pode-se incluir nos gráficos janelas adicionais com os valores de volumes movimentados e diversos indicadores, como na Figura 4.

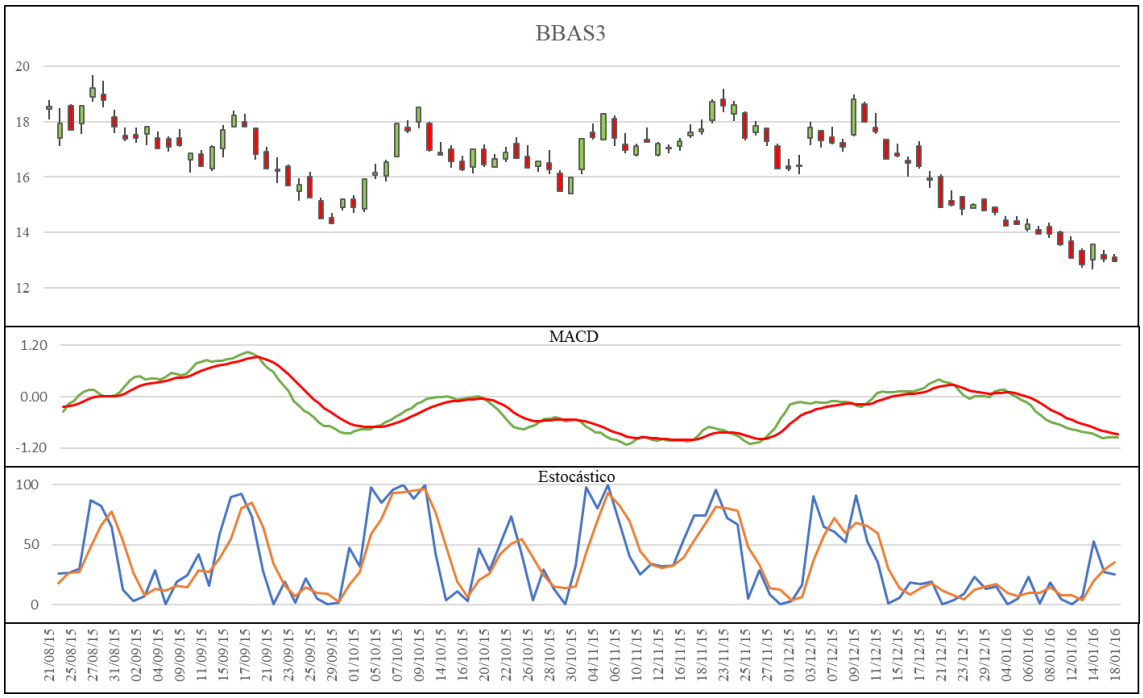

Figura 4 – Exemplo de um gráfico diário para a ação BBAS3 com indicadores MACD (*Moving Average Convergence-Divergence)* e Estocástico. (Fonte dos dados: B3. Gráfico elaborado pelo Autor)

2.4.1 Indicadores Técnicos

Existem diversas maneiras de se encarar o estudo e a aplicação da análise gráfica, podendo-se fazer uso de *price action* (ação do preço), *price levels* (Níveis de preço), Teoria de Dow, Teoria das Ondas de Elliot, Indicadores Técnicos ou uma combinação

dessas ferramentas. Neste contexto, o presente trabalho faz uso exclusivamente de indicadores como forma de auxiliar na tomada de decisão. Entre os principais indicadores utilizados na análise técnica, foram adotados nesta pesquisa os mesmos 6 (seis) indicadores usados por ZHAI *et al*. [10] e VARGAS *et al*. [14][16], com o intuito de dar continuidade ao trabalho realizado em VARGAS *et al*. [16] e verificar a validade destes parâmetros ao mercado brasileiro. São eles:

- Estocástico: oscilador que busca identificar zonas de sobrecompra e sobrevenda, anunciando possível reversão do preço. Ele é composto de duas linhas, sendo uma rápida (%K) e uma lenta (%D), calculadas da seguinte forma [34]:

$$\%K_t = \frac{(C_t - L_p)}{(H_p - L_p)} \times 100 \tag{2.1}$$

$$\%D_t = \frac{\%K_t + \%K_{t-1} + \%K_{t-2}}{3} \quad , \tag{2.2},$$

onde C_t é o preço de fechamento no dia t e L_p e H_p são o menor e o maior valores, respectivamente, que a ação atingiu no período p;

- *Momentum*: Taxa de aceleração do preço de uma ação. Indica o quão rápido um ativo sobe ou desce de preço:

$$Momentum_t = C_t - C_{t-p}, \tag{2.3}$$

onde C_t é o preço de fechamento no dia t e C_{t-p} é o preço de fechamento no dia $t - p$.

- RoC (*Rate of Change*): indica a velocidade com que uma variável muda durante um período de tempo:

$$RoC_t = \frac{C_t}{C_{t-p}} \times 100, \tag{2.4}$$

onde C_t é o preço de fechamento no dia t e C_{t-p} é o preço de fechamento no dia $t - p$.

- Williams %R: indicador de momento que mede zonas de sobrecompra e sobrevenda. Compara o preço de fechamento com a variação entre a máxima e a mínima em um período:

$$\%R_t = \frac{(H_p - C_t)}{(H_p - L_p)} \times 100, \tag{2.5}$$

onde C_t é o preço de fechamento no dia *t* e L_p e H_p são o menor e o maior valores, respectivamente, que a ação atingiu no período *p*;

- Oscilador Acumulação/Distribuição (A/D): um indicador de momento que tenta identificar oferta e demanda ao determinar se os investidores estão comprando (acumulando) ou vendendo (distribuindo) um certo ativo:

$$A/D_t = \frac{(BP_p - SP_t)}{2 \times (H_t - L_t)} \times 100, \tag{2.6}$$

onde BP_p é o poder dos compradores, representado pelo maior valor de abertura em um período *p*, SP_p é o poder dos vendedores, representado pelo menor valor de fechamento em um período *p*, e L_t e H_t são o menor e o maior valores, respectivamente, que a ação atingiu no dia *t*;

- Disparity *p*: mede a posição relativa do último fechamento com relação à média móvel de um período:

$$Disp_t = \frac{C_t}{SMA_p} \times 100, \tag{2.7}$$

onde C_t é o preço de fechamento no dia t e SMA_p é a média móvel simples dos fechamentos no período *p* .

Todos os indicadores ora apresentados necessitam da definição de um período de tempo para que sejam calculados. Em todos os casos foi adotada uma janela de 5 dias, conforme estabelecido por ZHAI *et al*. [10] e VARGAS *et al*. [14][16].

3. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (*Natural Language Processing – NLP*) compreende uma série de atividades que visam à extração de informações relevantes de uma base de dados não estruturados, com a finalidade de obter o entendimento de conteúdo textual não-trivial, desconhecido e potencialmente útil [35][36].

Uma das áreas de maior importância do NLP é a compreensão automática de textos em linguagem natural humana. Seu objetivo primordial é a obtenção de informações subjetivas, por meio de análise léxica, morfológica, sintática e semântica, a fim de criar conhecimento estruturado que possa ser utilizado para categorização, sumarização, análise de sentimento, reconhecimento de discursos, tradução e diversas outras aplicações [36].

Para que se possa aplicar informação não-estruturada em modelos preditivos, é necessário que se faça um tratamento prévio dessa informação, transformando-a em dados estruturados que sejam amigáveis ao processamento pelas máquinas [35]. Como esse processo busca encontrar uma boa representação das características dos documentos, deve-se calibrar a troca existente entre o volume de dados usados e o nível de semântica requerido versus uma extração de características que seja computacionalmente exequível e de uso prático. De acordo com o problema estudado, essa granularidade pode variar do nível do caractere, passando por palavras e termos e chegando até mesmo a conceitos inteiros [37].

As tarefas que desenvolvidas no NLP são diversas e devem ser aplicadas de acordo com a base de dados e o objetivo final do usuário. Por exemplo, para o desenvolvimento de um algoritmo gerador de texto completo, deve-se treinar a rede incluindo todos os caracteres especiais, pontuação e diferenciação de maiúscula e minúscula. No caso de uma extração semântica, o pré-processamento do texto é feito de maneira distinta do exemplo supracitado, sendo usual adotar as seguintes tarefas [37]:

- Tokenização: Separação dos textos em *tokens*, podendo ser por caractere, palavra, termos, conceitos ou sentenças. Este trabalho usa a tokenização por palavras;
- Remoção de pontuação, números e caracteres especiais: apesar de importantes para a compreensão humana, a remoção dessas características da base de dados pode otimizar o modelo. Muito usado juntamente com a tokenização por palavras;
- Remoção de *stopwords*: *Stopwords* são palavras de elevada frequência nos textos e que acrescentam pouca semântica às sentenças, como artigos definidos e indefinidos, pronomes, preposições, conjunções e verbos auxiliares.
- Remoção de palavras infrequentes: Palavras de baixa frequência por vezes são pouco representativas em bases de dados muito grandes, podendo ser removidas dos textos em prol do menor custo computacional. Além disso, estudos sugerem que usar apenas o top 10% das palavras mais frequentes não reduz a performance de classificadores [37].
- Vetorização de palavras: esta é a parte mais importante do processo, uma vez que as palavras devem ser representadas de forma numérica, para que os algoritmos possam processá-las e realizar operações matemáticas para, então, extrair informação.

No presente trabalho, a aplicação do NLP é indicada para uma extração semântica das notícias publicadas nos meios de comunicação, com o objetivo de identificar uma polaridade nessas notícias que indique um viés de alta ou de baixa para o mercado. A partir da base de notícias obtida, são aplicadas a tokenização das palavras, a normalização em letras minúsculas e a remoção de pontuação e de caracteres especiais. Além disso, foram considerados cenários com e sem *stopwords*, com remoção de palavras de baixa frequência, variando de 2 a 50 como frequência mínima na base de dados, e duas formas distintas de vetorização de palavras.

3.1. Representação Vetorial de Palavras

A representação vetorial de palavras, ou vetorização de palavras, é alvo de diversas pesquisas no meio acadêmico e na indústria, na busca por melhorias na efetividade dos modelos e por reduzir custos computacionais [35][37][38]. Sabe-se que o problema fundamental do processamento de linguagem é a maldição da dimensionalidade, que surge com a modelagem da distribuição de probabilidade conjunta entre muitas variáveis discretas [38].

Isso fica especialmente claro quando se usa a representação discreta, com vetor *one-hot*, onde todos os vetores são ortogonais entre si. Cada palavra é representada por um vetor de V dimensões com 1 em uma das posições e 0 nas demais, onde V representa o tamanho do vocabulário da base de dados (Figura 5). Ou seja, a cada palavra adicional no vocabulário, o tamanho da matriz de representação aumenta em $2V+1$ e, quando da modelagem da probabilidade conjunta de n palavras consecutivas, a quantidade de parâmetros passa a ser de $V^n - 1$ [38].

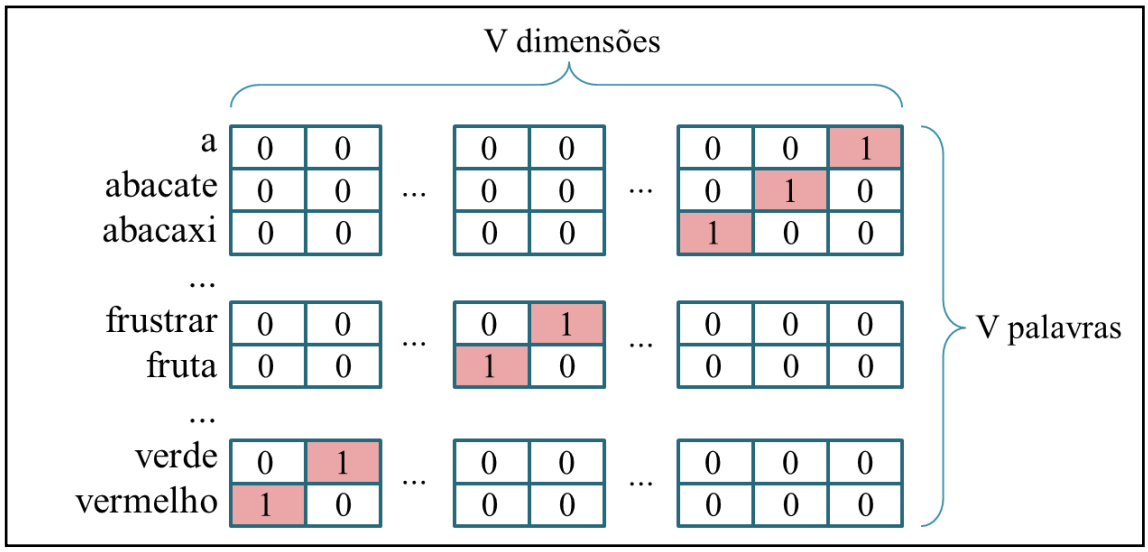

Figura 5 – Ilustração da representação *one-hot*, evidenciando a dimensão da matriz e a ausência de correlação entre as palavras. Uma vez que cada vetor é ortogonal a todos os demais, sua medida de similaridade dos cossenos é zero. (Elaborado pelo autor)

A título informativo, o dicionário da língua inglesa chega a 290 mil verbetes, enquanto o maior dicionário em língua portuguesa passa dos 300 mil verbetes [39]. Quando se inclui as conjugações verbais, nomes próprios e diferentes grafias da mesma palavra, esse número aumenta ainda mais. Por exemplo, o corpus usado para treinar as representações vetoriais em português passa de 1,3 bilhões de palavras totais para treinar um vocabulário de 935 mil palavras únicas [40].

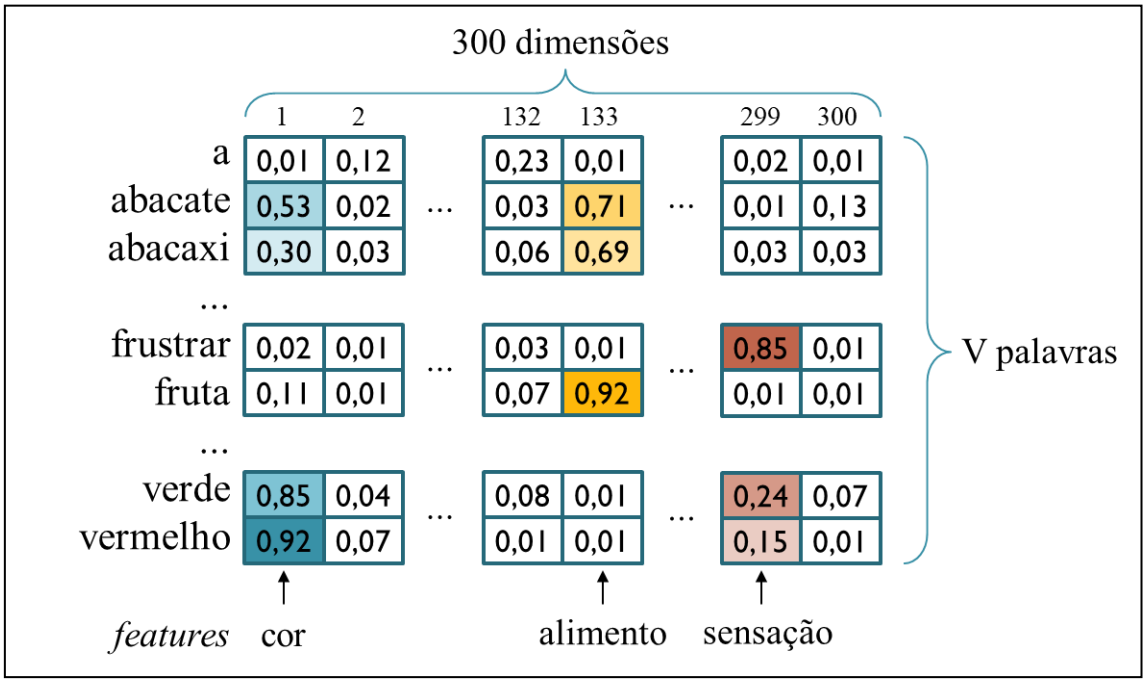

Figura 6 – Ilustração da representação distribuída de palavras. Cada dimensão representa uma *feature*, o que permite uma similaridade entre palavras que pertencem ao mesmo grupo. Por exemplo: fruta, abacate e abacaxi possuem valores elevados na mesma dimensão, que poderia estar representando o grupo dos alimentos. (Elaborado pelo autor)

Diante deste problema, BENGIO *et al*. [38] propuseram uma forma de representação distribuída das palavras, onde cada palavra é associada a um vetor no espaço $\mathbb{R}^m$, baseado em sua função de probabilidade conjunta. Dessa forma, cada palavra representa um ponto no espaço vetorial, de modo que palavras similares estarão propensas a apresentar vetores similares - medida pela similaridade dos cossenos, - o que permite obter maior uniformidade semântica e sintática nas palavras, bem como múltiplos graus de similaridade entre as palavras [41]. Ou seja, além de palavras similares estarem

agrupadas na mesma região, as distâncias entre palavras relacionadas são também similares. A Figura 6 ilustra como ficaria essa representação de palavras em um espaço vetorial de 300 dimensões.

É possível visualizar essa relação de forma fácil ao aplicar a técnica t-SNE, que permite que dados de elevada dimensionalidade sejam plotados em duas ou três dimensões [42]. A Figura 7 ilustra a visualização desse comportamento após a aplicação do t-SNE. Por exemplo, na relação masculino-feminino, a distância entre homem e mulher é aproximadamente a mesma distância que existe entre rei e rainha ou garçom e garçonete.

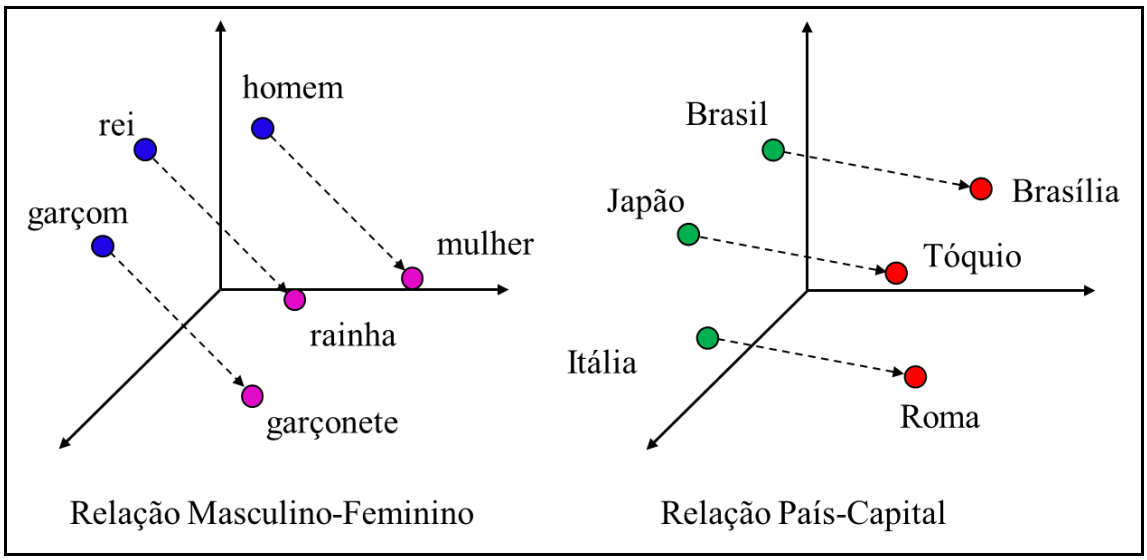

Figura 7 – Ilustração de uma visualização 3D da similaridade entre palavras relacionadas em uma representação distribuída de palavras. Exemplo da relação masculino-feminino e país-capital. (Elaborado pelo autor)

Posteriormente, MIKOLOV *et al*. [43] propuseram uma nova arquitetura simples e robusta, conhecida como *word2vec*, que treinada com uma quantidade massiva de dados obtinha performance superior a modelos complexos treinados com menor quantidade de dados. Em seguida, PENNINGTON *et al*. [44] apresentaram um modelo, batizado de GloVe, que captura diretamente as estatísticas globais do corpus usado no treinamento. Estes dois modelos de *word embedding* são amplamente usados e foram objeto de testes

no presente trabalho. Seus conceitos e formas de treinamento são apresentados no Anexo B.

Diante disso, o presente trabalho fez uso dos vetores de palavras GloVe e *word2vec skip-gram* em língua portuguesa, disponibilizados pelo repositório NILC - Núcleo Interinstitucional de Linguística Computacional da USP (Universidade de São Paulo) como base para a vetorização das notícias presentes no conjunto de dados [40]. Posteriormente, seus resultados são analisados e comparados, apresentando se há vantagem na escolha de um dos dois métodos para esta aplicação.

4. Redes Neurais Artificiais e Redes Neurais Profundas

As Redes Neurais Artificiais surgiram das buscas por replicar o funcionamento do cérebro humano, devido ao fato de este ser um sistema de processamento de informação altamente complexo, não-linear e com grande capacidade de aprendizado contínuo, de reconhecimento de padrões e de nuances perceptivas [45].

Pode-se dizer que o desenvolvimento das Redes Neurais teve sua origem no trabalho realizado por MCCULLOCH e PITTS em 1943 [46], com a publicação do modelo matemático do funcionamento de um neurônio. Posteriormente, em 1949, o primeiro método de treinamento de redes neurais artificiais foi proposto por HEBB [47], e, em 1957, ROSENBLATT [48] desenvolveu o modelo *perceptron*, que consistia em um modelo de unidades sensoriais conectadas a uma única camada de neurônios de McCulloch e Pitts, capaz de fazer o ajuste de parâmetros em função do erro de resposta e, dessa forma, aprender a reconhecer padrões [49].

Em 1960, WIDROW e HOFF [50] desenvolveram o modelo ADALINE (*Adaptive Linear Neuron*), cujo treinamento é realizado a partir de um algoritmo de aprendizado por ajuste de mínimos quadrados (*least mean square* – LSM). No entanto, em 1969, MINSKY e PAPERT [51] demonstraram que redes neurais de uma única camada, como o *perceptron*, não eram capazes de ajustar funções não-lineares, o que levou a um período de poucas pesquisas na área.

Apenas na década de 1980 as redes neurais voltaram a ter destaque com os trabalhos de Hopfield (1982), Kohonen (1982) e, principalmente, RUMELHART *et al.* (1986) [52] com a publicação de um algoritmo que permitia o ajuste de pesos em redes de mais de uma camada, sendo capaz de ajustar funções não-lineares, e o treinamento via retropropagação do erro (*backpropagation*). A partir deste ponto, as Redes Neurais Artificiais Perceptron Multicamadas (*Multilayer Perceptron* - MLP) se tornaram populares e foram empregadas na resolução de diversos tipos de problemas preditivos.

No entanto, as redes MLP treinadas com a retropropagação do erro via gradiente descendente possuem uma limitação de alcance desse treinamento, uma vez que os ajustes tendem a ser cada vez mais próximos de zero ao longo das camadas da rede. Isso faz com que redes de muitas camadas não consigam treinar adequadamente suas camadas mais profundas [53].

Com o aumento da capacidade computacional e da disponibilidade de grandes quantidades de dados, a busca pela criação de modelos de arquitetura mais profunda e capazes de generalizar problemas de maior complexidade se acirrou, e, no início dos anos 2000, foram publicadas diversas pesquisas que proporcionaram o adequado treinamento de redes de arquitetura profunda [54][55][56], também conhecidas como *Deep Learning*.

4.1. Redes Neurais Artificiais (ANN)

As Redes Neurais mais amplamente usadas são as MLP. Essas redes são arquiteturas rasas compostas por uma camada de entrada, responsável pelo recebimento dos dados, uma ou mais camadas escondidas, cujos neurônios irão extrair informação dos dados de entrada, e uma camada de saída, responsável pela previsão gerada pela rede.

Em seu processo de treinamento, as redes MLP seguem dois passos principais: *feedforward,* que corresponde à fase onde os dados são recebidos na camada de entrada, processados ao longo das camadas intermediárias e enviados para a camada de saída; e *backpropagation*, que ocorre após o resultado gerado pela rede e o cálculo do erro de predição, quando se faz o ajuste dos pesos dos neurônios no sentido inverso, ou seja, da camada de saída, passando pelas camadas intermediárias até a camada de entrada, conforme apresentado na Figura 8.

Durante o passo *feedforward*, os dados de entradas são processados por meio de multiplicação de matrizes e aplicação de função de ativação, de modo a capturar as não linearidades. No exemplo em questão, usa-se a função logística (σ). Sendo assim, as camadas intermediárias e de saída possuem a seguinte formulação matemática:

$$h_m = \sigma(W_{mn}^{(1)} x_n + b_1), \tag{4.1}$$

$$h_p = \sigma(W_{pm}^{(2)} h_m + b_2), \tag{4.2}$$

$$\hat{y}_k = \sigma(W_{kp}^{(3)} h_p + b_3), \tag{4.3}$$

onde x_n é o vetor do dado de entrada na rede, $W_{mn}^{(1)}$, $W_{pm}^{(2)}$ e $W_{kp}^{(3)}$ são as matrizes de pesos das camadas 1, 2 e 3, respectivamente, b_1, b_2 e b_3 são os vieses (*bias*) inseridos em cada uma das 3 camadas e $\hat{y}_k$ é a saída prevista pela rede.

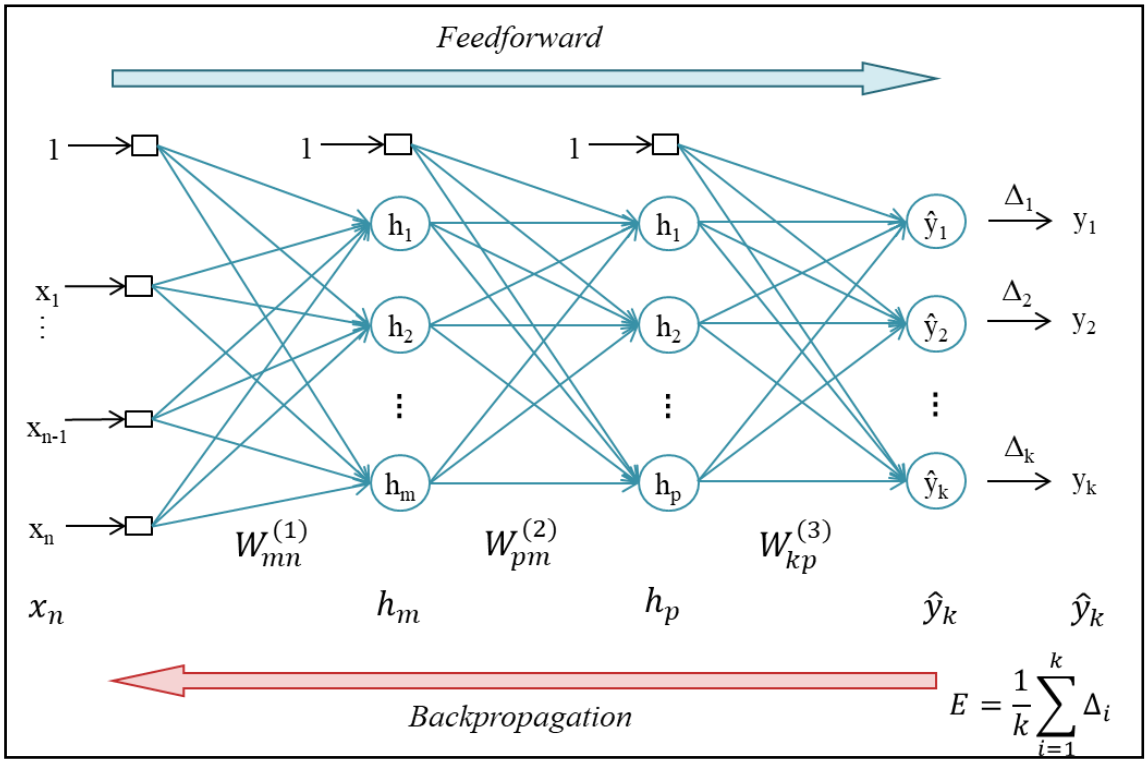

Figura 8 – Esquemático de uma Rede Neural Artificial MLP indicando seu passo *feedforward* e seu passo *backpropagation*.(Adaptado de SILVA *et al.* [49])

Combinando-se todas as equações, tem-se:

$$\hat{y}_k = \sigma(W_{kp}^{(3)}(\sigma(W_{pm}^{(2)}\sigma(W_{mn}^{(1)} x_n + b_1) + b_2)) + b_3), \tag{4.4}$$

Após o passo *feedforward*, para cada registro de entrada, calcula-se o erro de predição, neste caso representado pela função de erro médio quadrático, conforme a equação 4.5.

$$E = \frac{1}{2}\sum_{i=1}^{k} \left\| \hat{y}_i - y_i \right\|^2 , \tag{4.5}$$

Uma vez que a rede MLP busca prever o resultado correto, esta função erro deve ser minimizada. Para isso, o passo do *backpropagation* faz uso do método do gradiente descendente estocástico, onde os pesos são atualizados de acordo com uma taxa de aprendizado η e as derivadas parciais da função erro com relação aos pesos da rede:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \tag{4.6}$$

Aplicando-se as equações 4.4 e 4.5 em 4.6, e posterior regra da cadeia na equação 4.6, de modo a obter os ajustes dos pesos por camada, pode-se observar a aplicação de derivadas parciais sucessivamente em funções sigmoides. Dado que essas funções e suas derivadas são denotadas, respectivamente, por:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{4.7}$$

$$\sigma'(z) = \sigma(z)(1-\sigma(z)), \tag{4.8}$$

Observa-se pela equação 4.7 que a função sigmoide varia entre (0,1), já a equação 4.8 possui seu máximo quando $z = 0{,}5$, variando, então, entre *(0, 0,25]*. Com isso, ao se aplicar derivadas sucessivas ao longo de múltiplas camadas de uma rede MLP, quanto mais distante da saída, menor será o ajuste dos pesos da camada. Ou seja, pode-se concluir que as redes MLP com treinamento de gradiente descendente não podem assumir uma arquitetura profunda, sob pena de não treinar as camadas mais distantes da saída da rede.

4.2. Redes Neurais Recorrentes (RNN)

As redes neurais MLP clássicas possuem elevada capacidade de aprendizado e de resolução de problemas de *machine learning*. Porém, estes modelos possuem uma

arquitetura com entrada e saída de dados estáticas, o que dificulta seu uso no trato de séries temporais e demais tarefas que envolvam o sequenciamento de dados de entrada, como processamento de linguagem e vídeos. Diante disso, deve-se realizar um extenso pré-processamento dos dados, com remoção de tendência e sazonalidade, identificação e remoção de ciclos senoidais e ruídos não correlatos, para manter apenas as componentes não-lineares, que serão o foco do aprendizado e predição da rede MLP [57].

Para tratar esse tipo de problema, foram desenvolvidas as Redes Neurais Recorrentes (*Recurrent Neural Networks - RNN*), que são redes que não apenas recebem dados, processam e geram uma saída, mas que também guardam uma parte da informação referente ao dado recém processado [17]. A Figura 9 apresenta o esquema de uma RNN e sua visualização desdobrada, mostrando suas interações.

No entanto, foi demonstrado por BENGIO *et al.* [58] que há um problema com o treinamento das RNN pelo método do *backpropagation through time (BPTT)*, uma vez que há a dissipação do gradiente (*vanishing gradient*) ao longo dos passos de treinamento da rede, assim como ocorre com redes MLP de muitas camadas. Isso faz com que a rede não aprenda dependências de longo prazo, restringindo-se a aprender apenas relações de curto prazo.

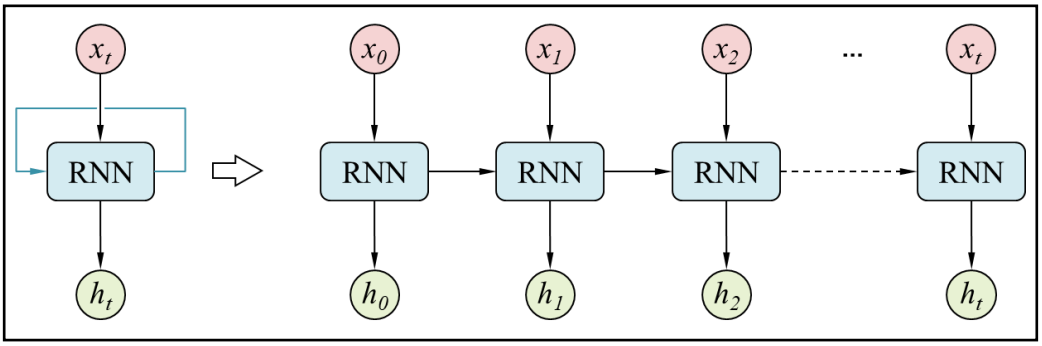

Figura 9 - À esquerda, um modelo de RNN apresentando a reutilização de informações pregressas a cada novo registro. À direita, uma ilustração do funcionamento da rede ao longo do tempo *(t_0, t_1, ..., t_t)* com a transferência de informações ao longo dos passos [17][59]. (Adaptado de OLAH [59])

Com isso, HOCHREITER e SCHMIDHUBER [18] estudaram o problema a fundo e propuseram uma nova arquitetura de rede, onde apresentam uma célula de memória capaz de aprender e esquecer dependências de longo prazo. Essas redes são chamadas de *Long Short-Term Memory* (*LSTM*). A Figura 10 apresenta uma célula dessa rede e seus componentes.

Cada célula de uma rede LSTM possui 3 entradas de dados e duas saídas, e seu interior possui quatro camadas de redes neurais que interagem entre si, conforme apresentado a seguir, onde t representa o registro atual, $t-1$ o registro imediatamente anterior e n o número de dimensões dos registros de entrada:

- x_t é o vetor de entrada de dados com n dimensões, onde cada registro adicional é informado à rede;
- c_{t-1} e c_t são as informações de memória de n dimensões que entram e saem do módulo, respectivamente, e são a chave para o registro de dependências de longo prazo entre os registros nas redes LSTM;

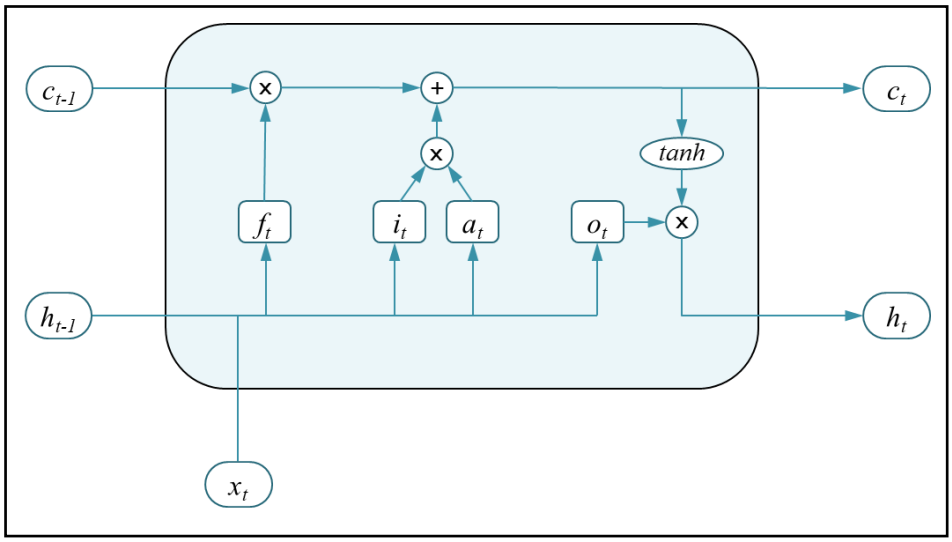

Figura 10 - Modelo de célula LSTM e sua recorrência. Cada célula possui 4 camadas de redes neurais MLP interagindo entre si, que são denominadas f_t (*forget gate*), i_t (*input gate*), a_t (*add*) e o_t (*output gate*) [18][59]. (Adaptado de OLAH [59])

- h_{t-1} e h_t são os vetores de saída de *m* dimensões da célula nos momentos $t-1$ e t, respectivamente.
- Os vetores x_t e h_{t-1} são concatenados na entrada da célula de memória e, em seguida, alimentam cada uma das 4 camadas de rede neural;
- f_t (*forget gate*) é a camada que processa os dados de entrada x_t e h_{t-1} e decide quanto da memória c_{t-1} será esquecida, aplicando-se a multiplicação de pesos (W_f) e uma função de ativação logística (σ).

$$f_t = \sigma(W_f .[h_{t-1}, x_t] + b_f) \tag{4.9}$$

- O resultado desse processo é um vetor com *n* dimensões e valores que variam entre 0 e 1. Aplica-se, então, o produto de Hadamard (elemento a elemento – $A_{ij} \times B_{ij}$) [60] entre este vetor de saída e o vetor de memória c_{t-1}. Elementos com valores próximos a 1 indicam que a memória c_{t-1} será preservada, enquanto valores próximos a 0 indicam que deve ser esquecida.
- As camadas i_t (*input gate*) e a_t (*add*) são as camadas que decidem o quanto da entrada atual será incluída na célula de memória. Essas camadas também aplicam pesos (W_i e W_a) às entradas e usam as funções logística (σ) e tangente hiperbólica (*tanh*) como ativação, respectivamente.

$$i_t = \sigma(W_i .[h_{t-1}, x_t] + b_i) \tag{4.10}$$

$$a_t = \tanh(W_a .[h_{t-1}, x_t] + b_a) \tag{4.11}$$

- Em seguida, os vetores de saída i_t e a_t são, então, multiplicados elemento a elemento e são adicionados ao resultado da operação de c_{t-1} com f_t, resultando no vetor de memória c_t.

$$c_t = c_{t-1} * f_t + i_t * a_t \tag{4.12}$$

- Na camada o_t os dados de entrada x_t e h_{t-1} são concatenados e multiplicados por pesos W_o e ativados como uma função logística e, posteriormente, combinado com a célula de memória c_t para formar a saída do módulo h_t.

$$o_t = \sigma(W_o .[h_{t-1}, x_t] + b_o) \tag{4.13}$$

$$h_t = o_t * \tanh(c_t) \tag{4.14}$$

Em uma RNN convencional, o processo de retropropagação do erro de um determinado instante é dependente do estado imediatamente anterior, conforme ilustrado na Figura 11. Isso resulta em uma dependência entre todos os passos de treinamento e uma extensa multiplicação de termos de valor pequeno (entre 0 e 1), causando a dissipação do gradiente em poucos passos de treinamento.

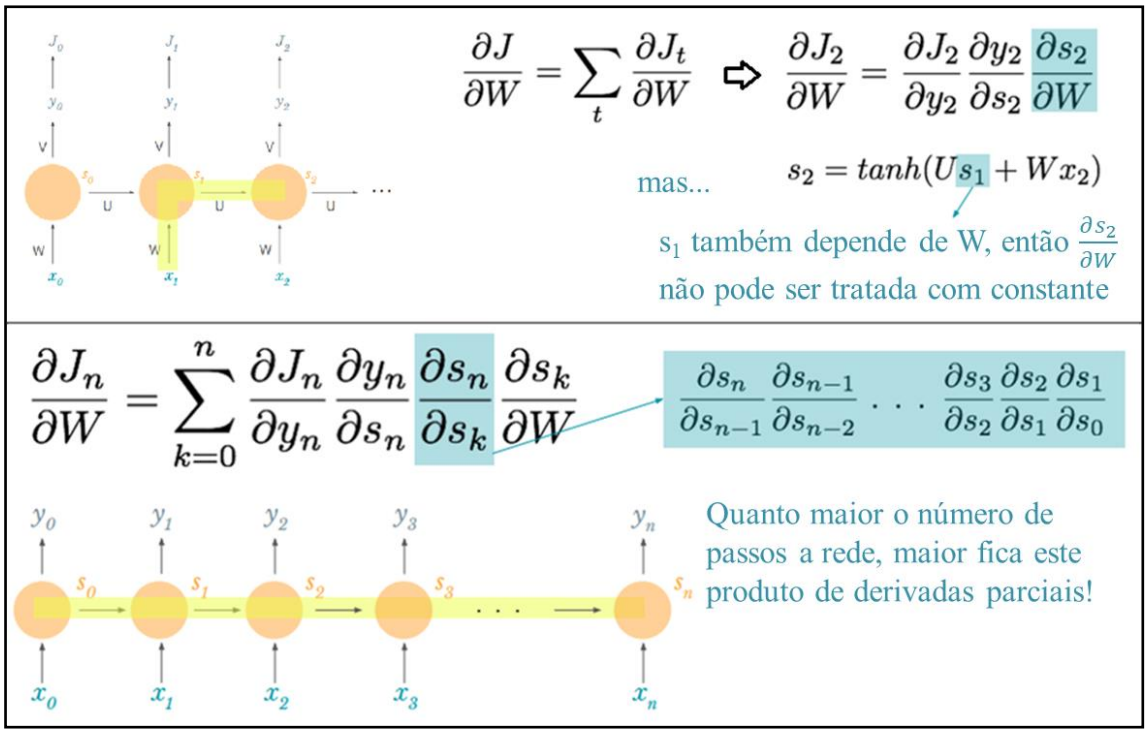

Figura 11 - Ilustração do BPTT, desenvolvimento da regra da cadeia e a dissipação do gradiente em uma RNN simples (Adaptado de [61]).

Já em uma rede LSTM, as informações de memória c_{t-1} e c_t não são alterados durante a evolução dos passos, mas apenas dentro da célula de memória. Dessa forma, quando se aplica o gradiente na equação 4.12, obtém-se:

$$\frac{\partial c_t}{\partial c_{t-1}} = f_t \tag{4.15}$$

que é um termo independente de c_{t-1}. Isso permite que a retropropagação do erro continue fazendo efeito ao longo do tempo.

4.3. Redes Neurais Convolucionais

As Redes Neurais Convolucionais (*Convolutional Neural Networks* - *CNN*) são tipos especiais de redes neurais que usam o conceito de convolução em uma de suas camadas, em vez de usar a multiplicação de matrizes [62]. Estas redes têm notável aplicação no reconhecimento de imagens, uma vez que são invariantes ao deslocamento dos dados, seja por rotação, translação, escala, achatamento ou espessura [63].

As redes Convolucionais são compostas basicamente por três camadas (Figura 12):

- Camada Convolucional: funciona como um filtro $N \times N$, que percorre a matriz de entrada realizando produto elemento a elemento, seguido de uma soma. A cada janela de aplicação do filtro, é gerado o valor de um elemento da matriz de saída. Em seguida, é aplicado um deslocamento (*stride*) s_f e o filtro segue até percorrer a matriz de entrada por completo.
- Camada de Ativação: Após a camada de convolução, é aplicada uma função de ativação para captar não linearidades. Em geral, usa-se a função ReLU (*Rectified Linear Unit – R(x) = Max (0, x)*), porém, outras, como tangente hiperbólica ou logística, também podem ser usadas.
- Camada de *Pooling*: também age como um filtro, tendo dimensão $M \times M$ e deslocamento (*stride*) s_p. No entanto, não executa multiplicação, apenas faz uma seleção entre elemento da janela $M \times M$ aplicada. Entre as aplicações de *pooling*,

as mais comuns são a escolha do valor máximo ou da média dos valores da janela.

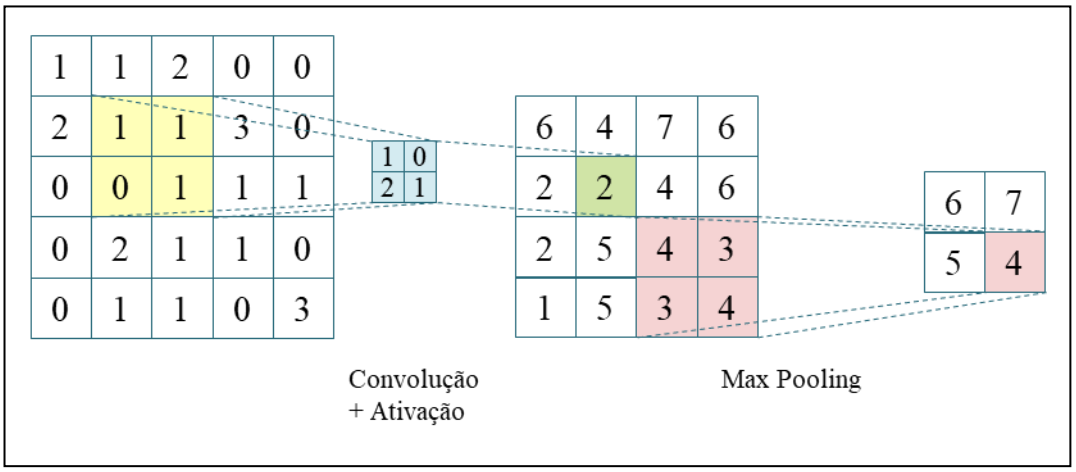

Figura 12 - Exemplo de uma Rede Convolucional com uma matriz de entrada 5x5, filtro convolucional de 2x2 e deslocamento 1, camada de ativação *ReLU* e *Max Pooling* 2x2, deslocamento 2. (Elaborado pelo autor).

A aplicação de Redes Convolucionais para extração de semântica de textos se mostrou muito efetiva, mesmo com o uso de arquiteturas simples. O procedimento adotado para o tratamento de sentenças é similar ao tratamento de imagens, como pode ser observado na Figura 13. Porém, esta arquitetura possui algumas pequenas particularidades [20]-[22]:

- Seja uma sentença com n palavras. Em uma representação vetorial de k dimensões, a matriz de entrada possui $n \times k$ dimensões;
- Aplicam-se filtros de dimensão $h \times k$ ($1 \leq h \leq n$) e, normalmente, deslocamento 1;
- Faz-se uma seleção de *max pooling* ao longo do tempo (ao longo da sentença);
- Concatena-se os resultados do *pooling*, aplica-se a uma camada de rede neural totalmente conectada e função de ativação.

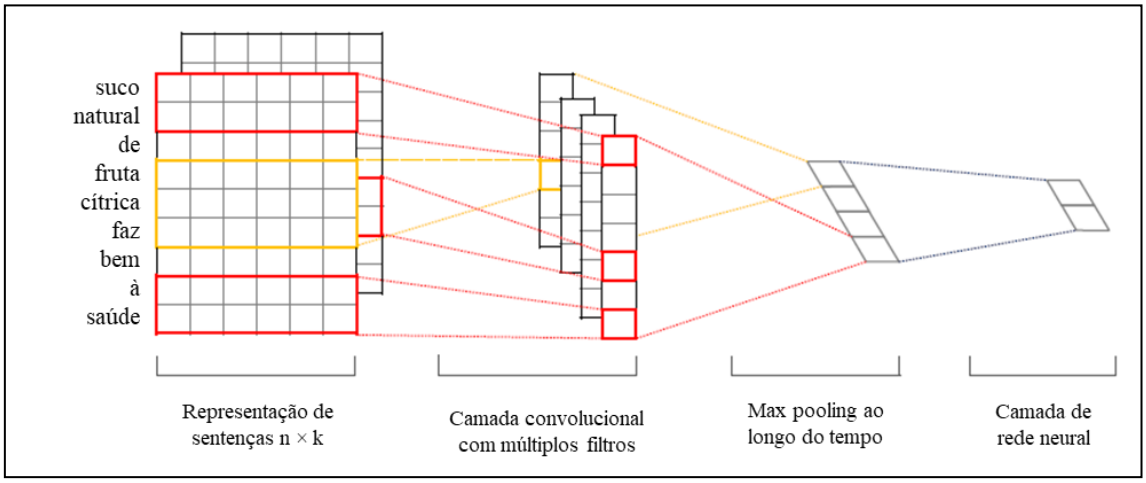

Figura 13 - Exemplo de uma Rede Convolucional aplicada a textos (Adaptado de [22]).

5. Teoria da Informação e a Combinação de Modelos

A teoria da informação, proposta por SHANNON em 1948 [64], visava a encontrar os limites para o processamento de sinais e operações de comunicação, que envolvem a compressão, o armazenamento e a transmissão de dados. Shannon demonstrou que a medida de compressão de dados equivale à taxa de símbolos requerida para representar uma mensagem e permitir sua reconstrução no destino, e que a menor taxa na qual uma sequência gerada por uma fonte estocástica pode ser transmitida e reconstruída perfeitamente está relacionada com um parâmetro básico da fonte estocástica chamado entropia [65].

O conceito de entropia, oriundo da termodinâmica, reflete o grau de irreversibilidade de um sistema, sendo, por vezes, associado ao seu grau de desordem. Na teoria da informação, a entropia está diretamente ligada à quantidade de informação presente em uma mensagem, isto é, quanto maior a informação presente na mensagem maior sua entropia e maior a incerteza para o recebedor da mensagem.

Considerando-se eventos aleatórios, a entropia é a medida do grau de incerteza médio que descreve a variável aleatória. Sendo, então, *X* uma variável aleatória com função de probabilidade *p(x)*, sua entropia é definida por [66]:

$$H(X) = -\sum_{x=1} p(x) \log_2 p(x), \tag{5.1}$$

onde se considera que $0 \times \log_2 0 = \lim_{\varepsilon \to 0} (\varepsilon \times \log_2 \varepsilon) = 0$.

Em um problema de classificação de *n* classes, as saídas de uma rede neural com função de ativação *softmax* apresentam seus resultados em termos probabilísticos e que, posteriormente, irão definir a qual classe pertence o registro de entrada. Ou seja,

aplicando-se o conceito da entropia de Shannon (Equação 5.1) à saída da rede, pode-se obter um grau de incerteza inerente ao resultado.

Dessa forma, observando-se as situações limítrofes de um problema de classificação com *n* classes, em uma saída da rede do tipo (1, 0, 0, ..., 0), onde a rede aponta 100% de certeza na sua saída, a entropia é igual a 0. Por outro lado, uma saída do tipo $(1/n, 1/n, 1/n, ..., 1/n)$, onde a rede aponta probabilidades iguais para cada classe, leva a uma entropia igual a $\log_2 n$. Ou seja, a entropia calculada para cada saída da rede neural está compreendida no intervalo $0 \leq H \leq \log_2 n$ [65].

No presente trabalho, esses conceitos foram aplicados à saída da rede neural profunda, que fornece 3 classes de saída. Em seguida, fez-se a normalização da entropia, dividindo-se o resultado por $\log_2 3$, obtendo-se uma entropia no intervalo [*0,1*]. Posteriormente, alguns cenários foram analisados por meio de filtros, onde apenas as saídas com entropia abaixo de um determinado limiar são consideradas, conforme detalhado na seção 6.5.

5.1. Combinação de Modelos

A combinação de modelos, conhecida como *Ensemble Learning*, é muito usada em algoritmos de aprendizado de máquina para se obter um resultado superior ao apresentado pelos modelos individualmente. Estes métodos são tidos como capazes de combinar resultados de modelos fracos, pouco melhores que preditores aleatórios, em modelos fortes, com elevada acurácia [67].

Diversas são as formas de se combinar os modelos, como por exemplo: médias, muito usado para regressores, onde se usa a média das previsões individuais pra obter a previsão combinada; votação, muito usado para classificadores, onde a saída é o resultado de um sistema de votação, com ou sem pesos, dos classificadores individuais; o *boosting,* que envolve o treinamento de múltiplos modelos em sequência, onde a função erro de um modelo depende da performance do anterior; e a árvore de decisão,

onde o processo de combinação segue uma seleção baseada em escolhas binárias, que resultam em uma estrutura em forma de árvore [67] [68].

Este trabalho usa como forma de combinação a Votação Plural [67], na qual cada modelo vota em um resultado de saída (uma classe $Y(C_i)$), o modelo combinado faz a soma dos votos em cada classe e, por fim, escolhe a classe que recebeu o maior número de votos. Este sistema de votação, quando normalizado de [0,1], permite uma interpretação dos votos como uma distribuição de probabilidades ($P(C_i)$), tornando-se conveniente a aplicação da entropia de Shannon como meio de buscar maior assertividade.

$$P(C_i) = \frac{1}{N} \times \sum_{x=1}^{N} Y(C_i) \tag{5.2}$$

A Tabela 2 apresenta o exemplo de um sistema de votação plural com 20 modelos e 3 classes. Após cada modelo apresentar sua saída para um determinado registro, é feita a contagem de votos e a normalização, sendo os valores apresentados em probabilidades. Em seguida, usa-se novamente o conceito da entropia de Shannon para calcular a entropia do resultado. Esses conceitos são uteis para a aplicação dos filtros definidos na seção 6.5.

Tabela 2 – Exemplo de sistema de votação plural

Classe	Mod. 1	Mod. 2	Mod. 3	Mod. 4	Mod. 5	...	Mod. 20	Soma	P(Ci)	Entropia
A	1	0	1	0	1	...	1	11	0,55	0,843
B	0	1	0	0	0	...	0	7	0,35	
C	0	0	0	1	0	...	0	2	0,10	

6. Construção do Modelo e dos Cenários

Conforme apresentado na seção 1.2, o presente trabalho faz uso de diversas técnicas de *machine learning* para avaliar seu desempenho no auxílio à tomada de decisão no mercado de ações. Para isso, o processo de modelagem seguiu os passos indicados na Figura 14. Em linhas gerais, essa modelagem segue três principais fases:

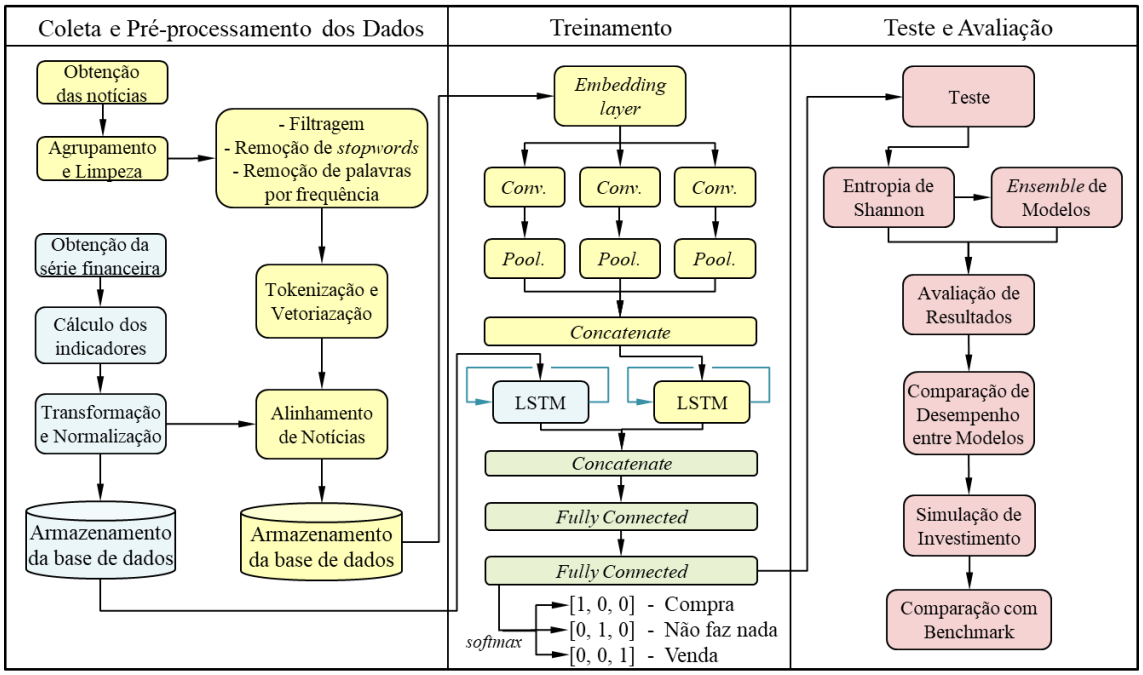

Figura 14 – Diagrama esquemático do modelo de previsão para auxílio à tomada de decisão. (Elaborado pelo autor)

- Coleta e Pré-processamento dos dados: Nesta fase foi realizada toda a coleta das séries de preço da ação BBAS3 (Banco do Brasil ON), cálculo dos indicadores, transformação e normalização dos dados, obtenção do *dataset* de notícias, disponibilizado por FIGUEIREDO [69], limpeza, filtragem, tokenização, vetorização, alinhamento das notícias com os dias de operação e separação dos diferentes *datasets* a serem testados. Estes pontos serão detalhados na seção 6.2;
- Treinamento: Implementação do modelo SI-RCNN, realização do treinamento estático e também por janelas deslizantes, conforme itens 6.4 e 6.5;

- Teste e Avaliação: Após o treinamento dos modelos, realiza-se o teste, aplica-se um filtro nas operações baseado na Entropia de Shannon e realiza-se o *ensemble* de modelos para verificar os ganhos obtidos com essas técnicas (seção 6.5). Depois, os resultados são comparados e uma simulação de investimento é feita e comparada aos *benchmarks* do mercado (seção 6.6).

6.1. Tecnologia Adotada

Os problemas que envolvem aprendizado profundo requerem o uso de ferramentas computacionais que, em conjunto com técnicas estatísticas e algoritmos de aprendizagem, permitam a identificação de padrões nos dados e auxiliem na geração de conhecimento. São muitas as ferramentas disponíveis hoje em dia, desde softwares específicos para *machine learning* como KNIME, Keel e Weka, passando por softwares de cálculo de alta performance, como MATLAB®, Scilab e GNU Octave, e, até mesmo, linguagens de programação como Python e R, que dispõem de vasto conteúdo de *toolbox* gerado pelas comunidades de *data mining* e *machine learning*.

6.1.1 Arquitetura em Python

O trabalho em questão trata de séries temporais, onde cada valor depende não apenas da observação imediatamente anterior, mas também de um longo período de registros de preços e volumes movimentados nas ações. Para resolver este tipo de problema, são usados algoritmos de Redes Neurais Profundas disponíveis para Python como ferramenta, devido sua alta capacidade de análise de dados e vasto conteúdo disponibilizado em suas *toolboxes*.

A arquitetura usada consiste do ecossistema Anaconda suportando o uso do Jupyter notebook, a partir de onde são acessados o Python, as bibliotecas Numpy, gensim e NLTK. Acima do Numpy, operam outras bibliotecas que dão suporte ao trabalho, que são o matplotlib, pandas e TensorFlow. A Figura 15 ilustra essa arquitetura.

Os modelos foram treinados e testados usando-se o Google® Cloud, onde foi possível configurar uma máquina virtual com 4 vCPUs, 16GB de memória RAM e uma GPU NVIDIA® Tesla K80, com sistema operacional Debian com TensorFlow 1.13.1 m21 (com Intel® MKL-DNN/MKL e CUDA 10.0), que permitiu a execução dos modelos com relativa agilidade.

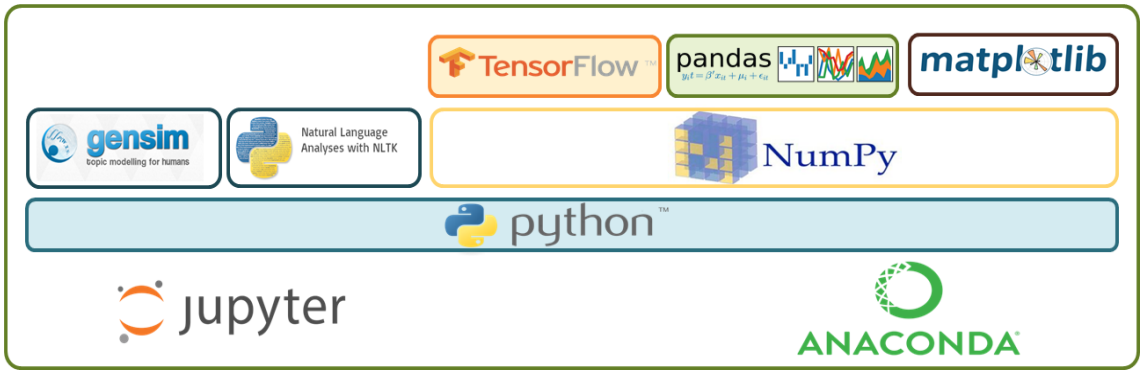

Figura 15 - Ecossistema Anaconda e bibliotecas Python usadas. (Elaborado pelo autor)

6.2. Obtenção e Pré-processamento dos Dados

Os dados são a base de todo trabalho de *machine learning*, pois é a partir deles que os modelos são treinados para reconhecer padrões e realizar suas tarefas preditivas ou descritivas. Por isso, é importante que a obtenção de dados seja originária de uma fonte confiável e que tenha qualidade nos registros. Dados sem confiabilidade podem invalidar todo um trabalho de pesquisa e a má qualidade dos registros geram esforços adicionais de limpeza e pré-processamento, podendo inviabilizar a execução do treinamento de um modelo.

6.2.1 Obtenção da Série Financeira

No presente trabalho, optou-se pelo uso de um ativo de elevada liquidez e que tivesse seu desempenho atrelado diretamente à economia nacional, uma vez que o *data set* de notícias a ser usado é uma base em português. Descartou-se, inicialmente, as ações ligadas a commodities (petróleo, mineração e siderurgia), pelo fato de seus preços

serem influenciados diretamente pelos mercados mundiais, tensões entre países e oscilação do preço das commodities no mercado internacional.

Nesse cenário, entre os ativos de maior liquidez que compõem o Ibovespa, os bancos encontravam-se em posição privilegiada. Entre os 3 maiores bancos listados no índice, Itaú e Bradesco possuem ações ordinárias (ON) e preferenciais (PN), enquanto Banco do Brasil possui apenas ações ordinárias. Uma vez que pequenos detalhes de governança das empresas também podem influenciar os preços das ações ON e PN de forma diferente [4], optou-se pelo uso das ações do Banco do Brasil ON (BBAS3) como base para o estudo, para minimizar os possíveis ruídos advindos da diferença ON/PN.

Escolhido o ativo, os dados foram obtidos diretamente do site da B3[1] por meio do *download* dos arquivos em formato CSV, disponibilizados diretamente ao público. Foram obtidos inicialmente 11 anos de dados, iniciando em 02/01/2007 até 29/12/2017, para realização do cálculo dos indicadores e, após o alinhamento com as notícias, foi usado apenas o período de 28/09/2007 a 17/01/2016, totalizando 3034 dias corridos e 2050 dias de efetiva negociação das ações BBAS3.

6.2.1.1 Detalhamento dos Dados

Após a obtenção dos dados, uma das primeiras tarefas a se desempenhar é a identificação do *dataset*, suas variáveis de entrada e seu comportamento estatístico. A partir dessa análise preliminar, é possível determinar quais caminhos tomar nas escolhas dos modelos.

[1] Disponível em http://www.b3.com.br/en_us/market-data-and-indices/data-services/market-data/historical-data/equities/historical-data

As operações nos mercados de capitais fornecem uma base de dados de séries temporais com indicativo de cotação de abertura, fechamento, máxima e mínima de preços, além da quantidade de negócios realizados, volume de papéis negociado e volume financeiro movimentado. Para cada ativo, a partir destes dados, pode-se gerar inúmeros indicadores de momento, de tendência e mistos e, com isso, obter ferramentas que indiquem a oscilação provável da cotação do papel.

Para este trabalho, foram usados dados históricos de 2050 dias dos preços de fechamento ajustados para a ação BBAS3 e, em seguida, foram aplicados os indicadores descritos na seção 2.4.1. O resultado pode ser observado na Tabela 3, onde é apresentado o dicionário de dados.

Tabela 3 - Dicionário de Dados do Data Set

Coluna	Variável	Descrição	Tipo de Variável	Restrições
1	data_pregao	Data do Pregão	Data	Formato AAAAMMDD
2	price	Preço de Fechamento	Real	Valor Positivo
3	stoc_k	Indicador estocástico rápido	Real	Valor entre [0,100]
4	stoc_d	Indicador estocástico lento	Real	Valor entre [0,100]
5	momentum	Indicador *Momentum*	Real	-
6	roc	Indicador *Rate of Change*	Real	Valor Positivo
7	will_r	Indicador William %R	Real	Valor entre [0,100]
8	ad_osc	Indicador Acumulação/Distribuição	Real	-
9	disparity	Indicador *Disparity* 5	Real	Valor Positivo

6.2.1.2 Análise exploratória dos dados

Analisando-se os dados, não foram observados dados faltantes ou *outliers*, o que é resultado de um sistema totalmente informatizado e com elevada confiabilidade, como o da bolsa de valores. Por se tratar de uma série temporal, em que há forte correlação de valores dentro de cada uma das variáveis, a análise exploratória dos dados deve ser realizada de maneira diferente de quando se trabalha com dados atemporais. Ou seja,

não se deve observar dados estatísticos como média, mediana e desvio padrão dos dados brutos, bem como histogramas e matriz de correlação com base na correlação de Pearson [57].

As séries temporais financeiras são séries não-estacionárias, uma vez que suas características estatísticas de média e variância não são constantes ao longo do tempo e sua covariância não é função de *t* e *t-k*. Essas observações são facilmente identificáveis diante de um gráfico da série de preços e de suas funções de autocorrelação e autocorrelação parcial (Figura 16). A função de autocorrelação mede o grau de correlação de uma variável no instante *t* consigo mesma em um instante *t+k* e a função de autocorrelação parcial mede o grau de correlação de uma variável no instante *t* consigo mesma em um instante *t+k*, porém excluindo-se as influências dos instantes *t+1* a *t+k-1* [70].

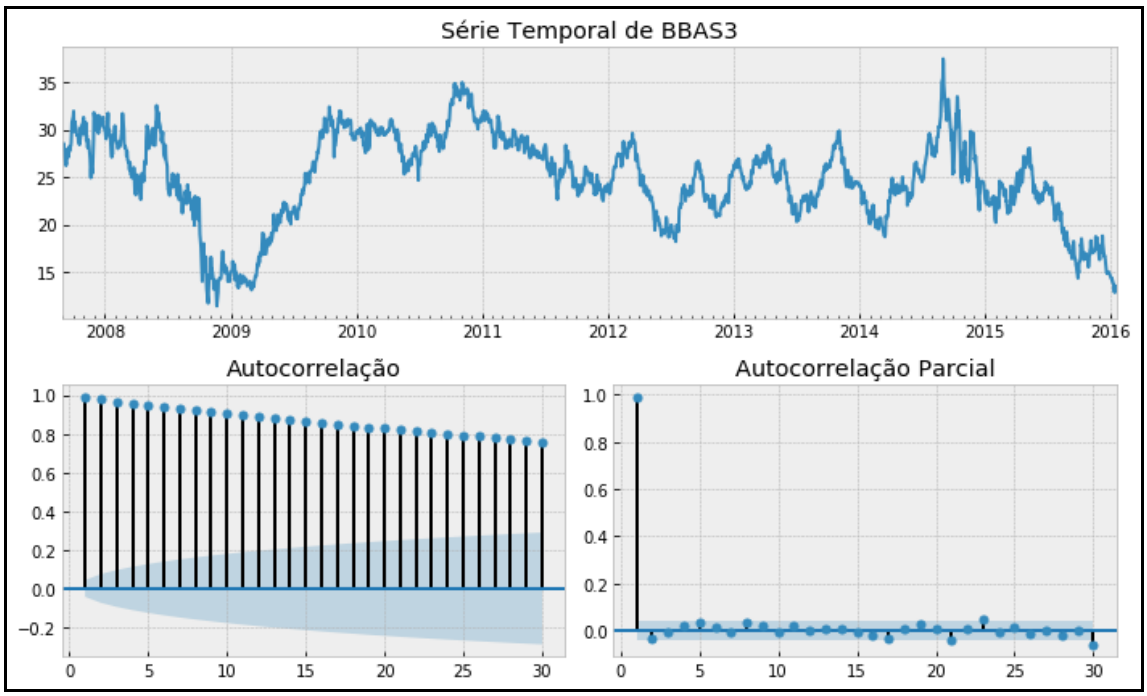

Figura 16 - Gráfico do preço de fechamento, Autocorrelação e Autocorrelação Parcial de BBAS3. Pode-se identificar que os preços não se desenvolvem em torno de uma média μ constante e que sua variância também varia não uniformemente ao longo do tempo. O lento decaimento da autocorrelação entre os preços também é um indicativo de não-estacionariedade. (Elaborado pelo autor)

Em econometria, a análise de séries financeiras é preferencialmente feita a partir das séries de log-retornos dos preços de fechamento ($r_t = ln(P_t) - ln(P_{t-1})$), uma vez que investidores estão mais interessados nos retornos e a série de retornos possui propriedades estatísticas mais interessantes do que a série de preços [70].

Dessa forma, foram gerados os gráficos da série de log-retornos, autocorrelação e autocorrelação parcial, conforme Figura 17, onde se observa a eliminação da tendência e o desenvolvimento da série em torno de zero, indicando que a média possui pequena variação. Porém, fica claro também que a série passa por fortes períodos de volatilidade nos meses do final de 2008, final de 2015 e final de 2016, indicando uma grande flutuação da variância ao longo do tempo. Observando-se as funções de autocorrelação e autocorrelação parcial para um período de 30 dias, pode-se verificar baixíssima ou nenhuma correlação serial, uma vez que praticamente todos os valores estão dentro do intervalo de confiança de 95%, indicando que não são significantes.

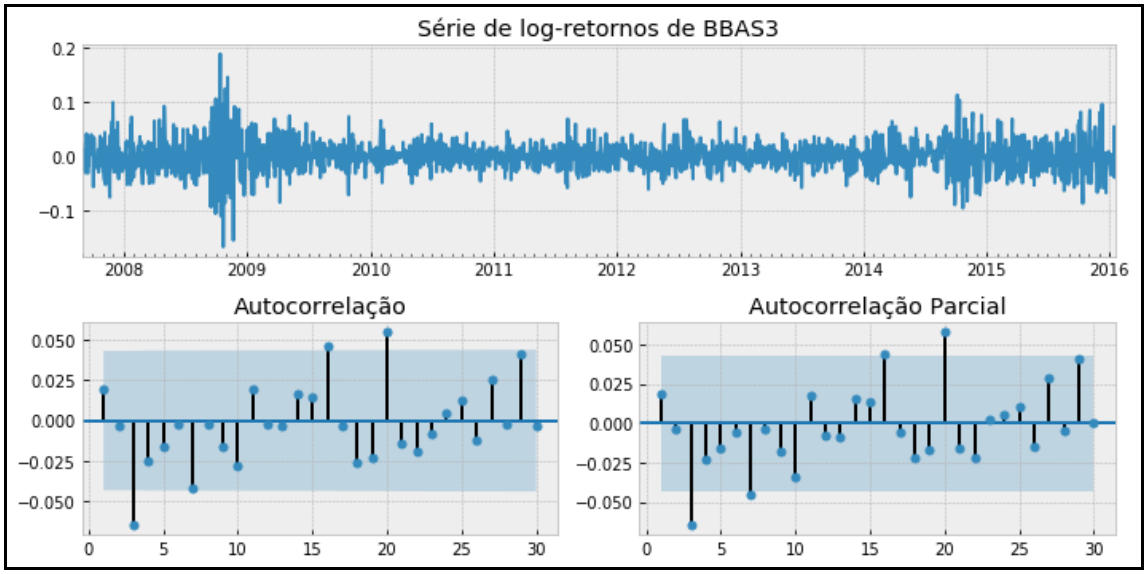

Figura 17 - Gráfico do log-retorno do preço de fechamento, função de autocorrelação e função de autocorrelação parcial. Observa-se que não há correlação serial significativa. (Elaborado pelo autor)

Diante do apresentado, tanto a série de preços quanto a de log-retornos apresentam comportamento não-estacionário. Além disso, testes de não linearidade em séries de

retornos sugerem que estas séries são não-lineares [70]. Ou seja, para um bom modelo de previsão, deve-se fazer uso de modelos que consigam capturar as não-linearidades presentes na série, o que confirma a adequação do uso de um modelo de Redes Neurais Profundas.

No entanto, de acordo com o informado na seção 1.1, o modelo SI-RCNN não usa a série de preços e de retornos como dado de entrada para previsão. Os dados de entrada da rede são apenas os indicadores das linhas 3 a 9 da Tabela 3, tendo sido todos eles calculados usando-se uma janela de 5 dias.

A normalização dos dados foi feita aplicando-se o *z-score* para se obter uma distribuição da variável com média nula e desvio padrão um. Esta normalização evita que o treinamento da rede seja enviesado por valores muito elevados de alguns indicadores.

6.2.2 Obtenção e Pré-processamento da Base de Notícias

A base de notícias usada neste trabalho foi cedida por FIGUEIREDO [69], que fez a aquisição por meio de um rastreador de rede (*web crawler*) desenvolvido, em sua tese de doutorado, para rastreamento do site da agência Thomson Reuters. Ao todo, foram fornecidos 176.144 arquivos individuais em formato HTML que possuíam, entre outros dados, o título da notícia, o corpo do texto, a data e a hora de publicação.

Inicialmente, foi feita a importação de cada arquivo, a extração dos dados relevantes (data, dia da semana, hora, título e corpo do texto) e a consolidação de todos os registros em um único arquivo CSV para facilitar seu pré-processamento.

6.2.2.1 Detalhamento e Limpeza dos Documentos

Com a base de notícias devidamente consolidada, foi feita a remoção de notícias duplicadas. Posteriormente, os corpos de texto foram descartados, devido sua elevada quantidade de palavras, o que elevaria sobremaneira o custo computacional do trabalho,

sem necessariamente trazer resultados. Além disso, como o trabalho proposto não faz análise e operações no *intraday*, o horário de publicação das notícias não tem relevância para o modelo, sendo este dado também descartado. Assim, o conjunto de dados ficou apenas com os registros de título da notícia, data da publicação e dia da semana.

Conforme apresentado na Tabela 4, apesar de o *data set* possuir 176.144 registros individuais de notícias, há de fato 90.552 registros únicos de notícias, sendo a notícia que se repete mais vezes ("Yellen diz que mercado de trabalho dos EUA ain..."), foi repetida 12 vezes. Após a remoção das duplicatas, pode-se observar, na Tabela 5, que a contagem de registros totais é a mesma dos registros únicos e que a frequência máxima de uma mesma notícia é 1. Também é interessante observar que a frequência máxima de notícias em um só dia é de 94 notícias no dia 15/10/2014.

Tabela 4 – Descrição básica do *dataset* indicando quantidade de registros, quantidade de registros únicos de cada tipo, registro que mais se repete e frequência com que se repete.

	date	dia_sem	titulo
count	176144	176144	176144
unique	2975	7	90552
top	2014-10-15	quinta-feira	Yellen diz que mercado de trabalho dos EUA ain...
freq	254	35519	12

Tabela 5 – Descrição do *dataset* após remoção de duplicatas indicando novos valores para a quantidade de registros, quantidade de registros únicos de cada tipo, registro que mais se repete e frequência com que se repete.

	date	dia_sem	titulo
count	90552	90552	90552
unique	2975	7	90552
top	2014-10-15	quinta-feira	Tractebel não concorda com divisão de custo té...
freq	94	18331	1

Após esse passo, foi feito o alinhamento da série de notícias com a séria financeira, de modo que todos as notícias ficassem associadas a um dia de operação. Ou seja, foi

considerado que todas as notícias que foram publicadas durante um dia impactam apenas no próximo dia de operação do mercado. Com isso, por exemplo, notícias publicadas na sexta-feira, no sábado e no domingo são agrupadas e alinhadas aos dados de sexta-feira, para entrar na rede alinhadas com esse dia e auxiliar na previsão das ações na segunda-feira. Esse procedimento foi feito de maneira geral em todo o conjunto de dados de notícias, incluindo todos os dias sem operação.

Cabe ressaltar que uma limitação nesse procedimento é que as notícias que são publicadas durante o pregão têm impacto imediato nas ações, com seus efeitos durando menos de 5 minutos e não se perpetuando para períodos de tempo maiores [27]. Porém, essa é uma restrição inerente a qualquer modelo que faça uso de dados diários, de modo que isso não recebeu nenhum tipo de tratamento neste trabalho.

A Tabela 6 apresenta as estatísticas do *dataset* de notícias após este alinhamento. Observa-se que há 2030 dias com notícias no período de 28/09/2007 a 17/01/2016, enquanto há 2050 dias de negociações. Isso se deve ao fato de haver dias em que nenhuma notícia foi capturada pela base de dados. Outro ponto importante a se notar é a quantidade máxima de notícias por dia, um total de 117. Por ser um valor muito elevado e exigir grande custo computacional para seu processamento no modelo de Redes Neurais Profundas, uma redução na dimensionalidade é aplicada neste *dataset*.

Tabela 6 – Estatísticas do *dataset* de notícias após o alinhamento com a base de dados de indicadores.

	título
contagem	2030
média	44,61
desvio padrão	16,85
mínimo	9
25%	30
50%	44
75%	57
máximo	117

6.2.2.2 Filtragem de Notícias

Uma vez que a base possui toda sorte de notícias sobre empresas e economia, ela pode conter informações irrelevantes para a ação em estudo, servindo apenas como ruído em meio a notícias realmente importantes.

Com base nisso, foi feita uma filtragem do conjunto de dados para manter apenas as notícias de maior relevância para o setor financeiro usando-se as palavras-chave: bancos, economia, PIB, Selic, desemprego, inadimplência, inflação, IPCA, IGP-M, IGP-DI, IBC-Br, dívida interna, dívida pública, previdência, déficit, balança comercial, dólar, euro, iuan, câmbio, moedas, Banco do Brasil, Itaú, Bradesco, Santander, Bank of America, Merril Lynch, taxas de juro, taxas DI, empréstimo, crédito, financiamento, BACEN, Banco Central, BNDES, Bank of England (BoE), Bank of Japan (BoJ), Banco Central Europeu BCE e Federal Reserve (FED).

Isso resultou em uma base de dados mais enxuta, com um total de 24.349 notícias, também distribuído em 2030 dias, com um máximo de 40 notícias por dia, média de 12 notícias por dia e 15 notícias por dia no percentil 75. Essa base de notícias foi denominada *banknews* e faz parte dos testes da mesma forma que a base completa, com o objetivo de se fazer a comparação entre ambas as estratégias.

Além disso, ambos os conjuntos de dados (completo e *banknews*) receberam uma remoção de sinais de pontuação e padronização em letras minúsculas. A partir deste ponto, houve nova divisão dos *datasets*, mantendo uma versão com *stopwords* e outra onde as *stopwords* foram removidas. A remoção de *stopwords* ajuda a reduzir o tamanho do *dataset* retirando palavras de elevada frequência e que não trazem consigo relevante significado para o texto. Ainda assim, optou-se também por testar as notícias com as *stopwords*, para efeito de comparação.

Na seção 6.2.2.3, a redução de dimensionalidade é novamente tratada e as mudanças na dimensionalidade dos datasets são resumidas na Tabela 7 e na Tabela 8.

6.2.2.3 Redução de Dimensionalidade e Representação Vetorial dos Documentos

Após limpeza e filtragem dos documentos, verificou-se que os dois conjuntos (completo e *banknews*) ainda possuíam elevada dimensionalidade e, consequentemente, elevado custo computacional para processamento. Considerando o *dataset* completo, há 90.552 notícias em 2050 dias, com até 117 notícias em um único dia e até 20 palavras em uma única notícia, totalizando 935.922 palavras. Sabe-se que, em uma representação matricial, as dimensões são fixas para todos os registros. Ou seja, uma vez estabelecido que serão usadas todas as notícias e todas as palavras, é necessário usar um tensor 2050 x 117 x 20 (dias x notícias x palavras), totalizando 4.797.000 posições, mesmo que a maior parte (82,4%) dos elementos desse tensor sejam iguais a zero. Diante disso, três estratégias foram adotadas para redução de dimensionalidade:

- Limitação do número máximo de notícias por dia: A Figura 18 apresenta o histograma da quantidade de notícias por dia. Pode-se verificar uma elevada frequência de dias que contam com 20 a 70 notícias. Esta distribuição bimodal é resultado do agrupamento de notícias dos fins de semana e feriados no dia útil anterior. Sendo assim, a quantidade máxima de notícias foi definida pelo percentil 75, que limita esse valor a 57 notícias por dia;
- Limitação do número máximo de palavras por notícia: uma vez que a remoção de muitas palavras pode causar uma perda de semântica em uma frase, esta limitação foi menos restritiva. Optou-se por usar o percentil 99, gerando uma redução de 20 para 18 palavras por notícia;
- Remoção de palavras com baixa frequência: A frequência de aparição de palavras em um *dataset* muitas vezes pode determinar sua importância. Em conjuntos muito grandes, uma palavra com frequência muito baixa pode ser irrelevante para o problema analisado. Diante disso, foram testados filtros com frequência mínima de palavra variando de 2 a 10 e também 15, 20, 25, 30 e 50.

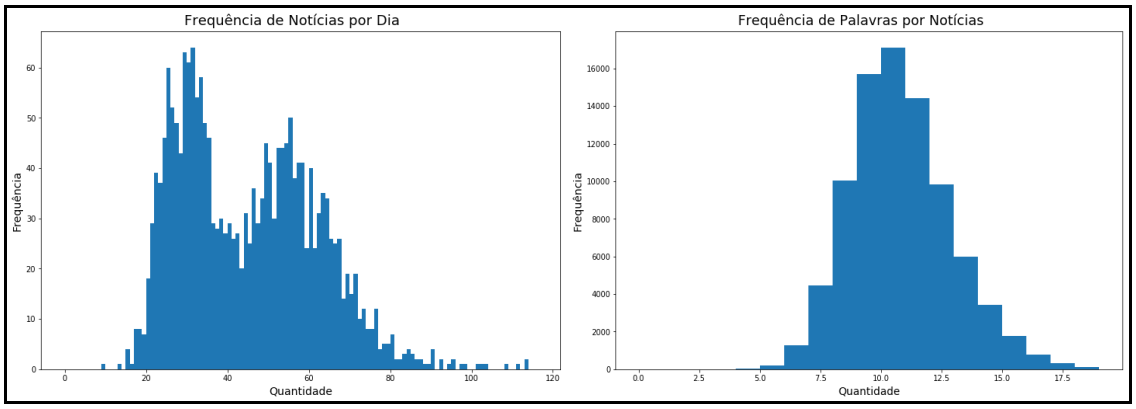

Figura 18 – Histogramas de frequência de notícias por dia e de palavras por notícia.

As estratégias de limitação de notícias e palavras levaram a uma redução de 11% na quantidade total de palavras, passando a 832.604. No entanto, quando se considera a dimensionalidade do tensor que armazena essas palavras, tem-se uma redução de 56%, totalizando 2.103.300 posições, aumentando assim a densidade de elementos diferentes de zero, ou seja, a densidade de informação passa de 17,6% para 39,5%.

Tabela 7 – Variação do tamanho do vocabulário, máximo de notícias por dia, máximo de palavras por notícia e palavras sem representação no dicionário para o *dataset* completo de acordo com os filtros de frequência aplicados.

COMPLETO								
Dataset	COM *stopwords*				SEM *stopwords*			
Frequência mínima	Tamanho do Vocabulário	Máx. Notícias por dia	Máx. palavras por notícia	Palavras sem representação	Tamanho do Vocabulário	Máx. Notícias por dia	Máx. palavras por notícia	Palavras sem representação
1	26480	57	18	6605	26367	57	12	6605
2	13728	57	18	1183	13636	57	12	1183
3	10385	57	17	533	10291	57	12	533
4	8657	57	17	310	8567	57	12	310
5	7559	57	17	220	7473	57	12	220
6	6818	57	17	174	6734	57	12	174
7	6203	57	17	145	6121	57	12	145
8	5749	57	17	121	5667	57	12	121
9	5380	57	17	109	5298	57	12	109
10	5066	57	17	96	4988	57	12	96
15	4003	57	17	50	3930	57	12	50
20	3379	57	17	35	3308	57	12	35
25	2912	57	17	29	2845	57	12	29
30	2586	57	17	20	2520	57	12	20
50	1786	57	17	10	1728	57	12	10

Tabela 8 – Variação do tamanho do vocabulário, máximo de notícias por dia, máximo de palavras por notícia e palavras sem representação no dicionário para o *dataset banknews* de acordo com os filtros de frequência aplicados.

BANKNEWS								
Dataset	COM *stopwords*				SEM *stopwords*			
Frequência mínima	Tamanho do Vocabulário	Máx. Notícias por dia	Máx. palavras por notícia	Palavras sem representação	Tamanho do Vocabulário	Máx. Notícias por dia	Máx. palavras por notícia	Palavras sem representação
1	9676	15	17	1251	9579	15	12	1251
2	5158	15	17	189	5078	15	12	189
3	3793	15	17	87	3717	15	12	87
4	3104	15	17	57	3030	15	12	57
5	2667	15	17	42	2595	15	12	42
6	2351	15	17	37	2282	15	12	37
7	2146	15	17	31	2079	15	12	31
8	1960	15	17	28	1895	15	12	28
9	1819	15	17	26	1754	15	12	26
10	1690	15	17	23	1628	15	11	23
15	1290	15	17	19	1234	15	11	19
20	1029	15	17	7	979	15	11	7
25	873	15	16	6	825	15	11	6
30	773	15	16	6	726	15	11	6
50	537	15	16	6	495	15	11	6

Já a estratégia de remoção de palavras com baixa frequência, resultou na criação de 15 cenários distintos, que combinados aos dois datasets (completo e *banknews*) e ao uso ou não de *stopwords* resulta em 60 cenários, conforme pode ser visto na Tabela 7 e na Tabela 8. Estas tabelas também apresentam algumas informações relevantes, como o tamanho do vocabulário, que influencia no tamanho da matriz *embedding*, responsável pela vetorização das palavras na entrada da rede e os limites de notícia por dia e de palavras por notícia, baseado nas regras de percentil adotadas anteriormente. A coluna "Palavras sem representação" indica a quantidade de palavras no *dataset* que não possuem correspondência na matriz *embedding* disponibilizada pelo NILC da USP. Nestes casos, adotou-se uma inicialização aleatória, com distribuição normal de media 0 (zero) e desvio padrão 1 (um), e deixou-se que a rede treinasse também a camada de *embedding*, para que estas palavras também se ajustassem ao problema.

Cabe ressaltar que estes 60 cenários foram analisados tanto com o uso de *word2vec skip-gram* quanto com o uso do GloVe, ou seja, o total de cenários analisados inicialmente foi de 120.

6.3. Arquitetura da rede

A arquitetura da rede neural profunda não sofreu grandes alterações com relação à usada em VARGAS *et al*. [16]. A Figura 19 apresenta cada uma das camadas da rede, sendo detalhado a seguir, de acordo com as etapas numeradas, o processamento de um único registro (notícias de um dia e indicadores de 5 dias):

(1) Camada *embedding*, que recebe o *dataset* de notícias e faz a transcrição de cada palavra para um vetor de 300 dimensões. Considerando-se *n* palavras em todas as notícias de um único dia, esta matriz de entrada seria $1 \times n \times 300$.

(2) A mesma matriz de notícias vetorizada segue para 3 ramos distintos de redes convolucionais, cada uma com um tamanho de filtro diferente. A primeira tenta capturar características (*features)* em grupos de 3 palavras, a segunda em grupos de 4 palavras e a terceira em grupos de 5 palavras, de modo que cada uma possa capturar um detalhe diferente, e adota-se um preenchimento com zeros (*padding*) na dimensão das palavras da matriz original. Os filtros também buscam uma redução de dimensionalidade, de modo que são usados filtros de dimensões 3x64, 4x64 e 5x64, respectivamente, e depois uma função de ativação ReLU, resultando em matrizes de saída com dimensões $1 \times n \times 64$ em cada ramo.

(3) Esses filtros são seguidos por um *max pooling* de dimensão 2 e deslocamento (*stride*) de 2, cujo objetivo é extrair as características (*features*) mais relevantes de cada grupo de palavras. Neste ponto, a dimensão da matriz se reduz para $1 \times n/2 \times 64$.

(4) Extraídas as principais *features* de cada convolução, os resultados são concatenados em uma única matriz de dimensão $1 \times n/2 \times 192$.

(5) A rede LSTM de 128 passos recebe o resultado do processamento das camadas convolucionais e processa, passo a passo, as *features*, buscando obter relações semânticas e sintáticas, resultando num vetor de saída de 1×128 dimensões.

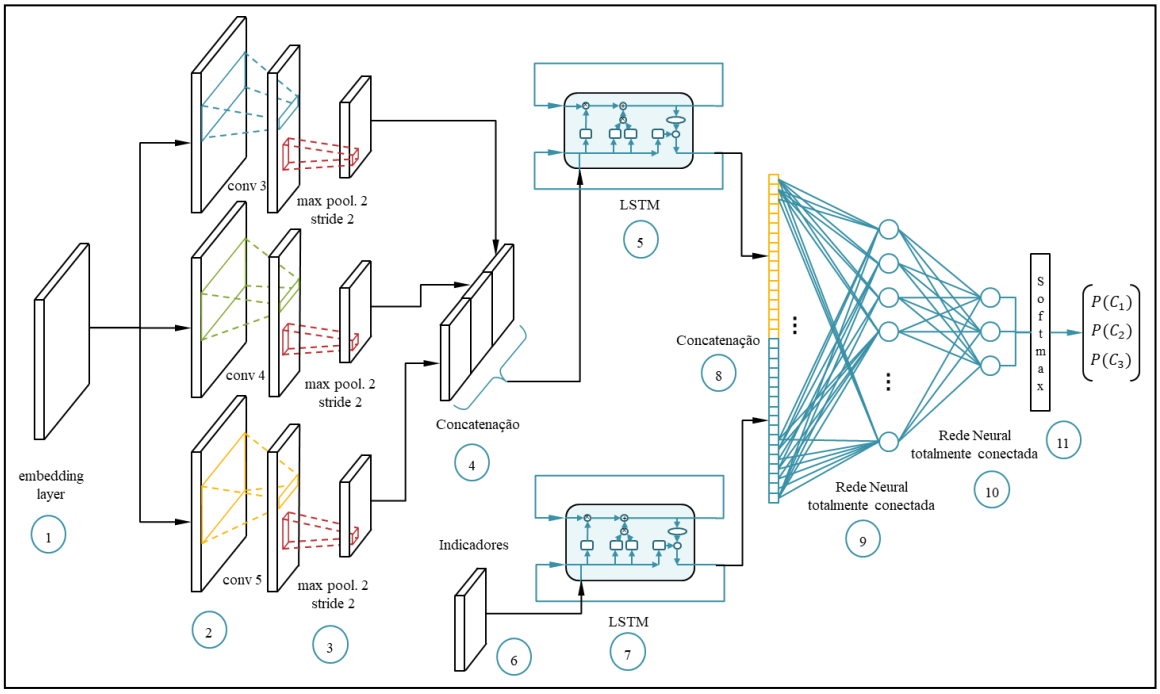

Figura 19 – Arquitetura da Rede Neural Profunda usada no trabalho (Elaborado pelo autor)

(6) Paralelamente a isso, a rede recebe a série de indicadores de dimensão *1×5×7* (5 dias e 7 indicadores).

(7) Estes indicadores são enviados para uma outra LSTM de 128 passos, onde seus valores são processados em busca de captar correlações temporais entre os dados. Essa rede resulta numa saída de *1×128.*

(8) Uma nova concatenação é feita, dessa vez entre o vetor de saída da LSTM que processou as notícias e a saída da LSTM que processou os indicadores, gerando uma saída de dimensão de *1×256*. Na saída desta concatenação, é aplicado um *dropout* [71] de 60% para evitar o *overfitting* da rede.

(9) O vetor resultante da concatenação é enviado para uma rede neural totalmente conectada de 30 neurônios com função de ativação logística para redução de dimensionalidade e para captação de não-linearidades. Após esta camada, também é aplicado um *dropout* de 60%.

(10) No final da rede encontra-se a principal alteração na arquitetura com relação ao trabalho anterior. Foi acrescentado um neurônio na camada de saída, resultando em uma saída ternária, em vez de uma saída binária.

(11) Após essa saída, é aplicada uma função de ativação *softmax*, com o intuito de apresentar a saída em termos de probabilidades, permitindo, assim, uma análise de entropia conforme indicado no capítulo 5.

Para efeito de treinamento da rede, as probabilidades de saída são comparadas aos resultados verdadeiros, a entropia cruzada é usada como função de erro e o Adam como método estocástico de otimização do erro [72].

Quando da realização dos testes, as saídas dos 3 neurônios finais da rede são tratadas com os conceitos da entropia de Shannon. Em cada cenário da aplicação de um limiar de entropia, os resultados com valores superiores ao limite estabelecido não geram decisão de investimento, conforme exemplificado na Figura 20. Espera-se, com isso, que a eliminação das saídas com baixo grau de certeza reduza o erro de previsão e o risco de prejuízos nas operações. Os resultados que passarem por este filtro apresentam saídas [1,0,0] indicando compra, [0,1,0] para não executar nenhuma operação ou [0,0,1] para venda.

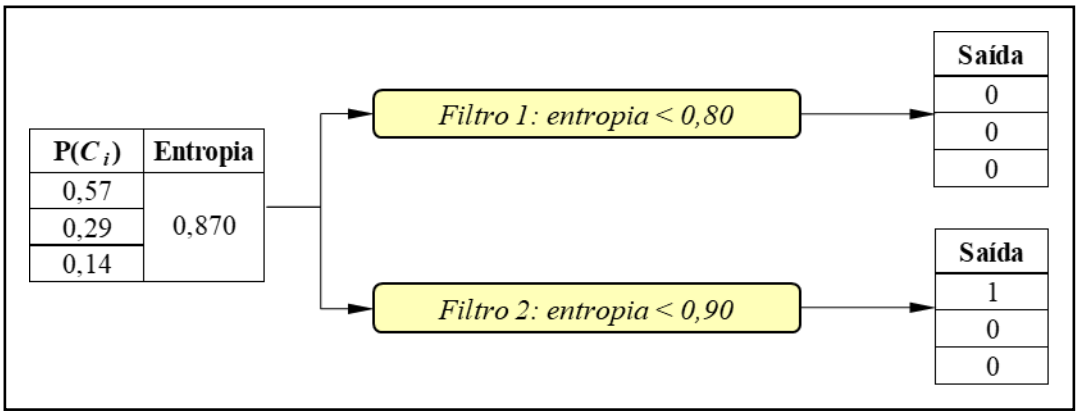

Figura 20 – Exemplos de dois filtros de entropia para uma saída da rede. Um filtro que exija entropia menor do que 0,80, não indicará nenhuma operação. Já um filtro menos restritivo, como 0,90, indicará uma operação de compra. (Elaborado pelo autor)

Esta arquitetura de rede é usada com os diversos *datasets* de entrada, variando suas características de pré-processamento, conforme seção 6.5. Posteriormente, os cenários são agrupados por tipo de treinamento e realiza-se uma combinação dos resultados, usando o sistema de votação indicado na seção 5.1, com o objetivo de determinar se essa técnica leva a maior assertividade.

6.4. Divisão do Dataset e Formas de Treinamento

Apesar da grande diversidade de conjuntos de notícias para entrada no modelo, todos compartilham de uma característica comum: 2050 dias de operação. Sendo assim, a divisão dos *datasets* seguiu o mesmo conceito de usar os primeiros 90% dos registros para treinamento e os 10% finais para teste.

Com relação às formas de treinamento dos modelos, esta pesquisa faz uso de duas formas básicas:

- Estática: similar aos treinamentos realizados em problemas que não possuem a temporalidade como fator determinante. Neste caso, faz-se uma subdivisão do dataset de treinamento, mantendo-se 80% do total efetivamente para treinamento o modelo e os 10% seguintes como validação. Após atingir o critério de parada, a rede efetua o teste nos 10% finais (Figura 21).

  Como critérios de parada, foram estabelecidos o limite máximo de 400 épocas de treinamento ou um erro de generalização maior do que 15%. Seja, $E_{va}(t)$ o erro de validação na época *t* e $E_{opt}(t)$ o menor erro de validação até a época *t*, o erro de generalização é definido por [73]:

$$GL(t) = 100 \times \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right) \tag{6.1}$$

- Janela Deslizante (*Sliding Window*): este tipo de treinamento é considerado o mais adequado em problemas de séries temporais, pois permite que o modelo se ajuste aos eventuais ciclos e tendências da série de dados [74]. Ele é realizado

definindo-se uma janela de treinamento *j* e um passo *s*. Treina-se o modelo com os dados de 1 a *j*, em seguida, avança-se uma quantidade *s* de registros, treina-se com dados de *s* a *j+s* e segue-se avançando até terminar o *dataset* de treinamento.

Neste trabalho, o conceito da validação foi incorporado ao da janela deslizante, resultando no treinamento ilustrado na Figura 21. Definida uma janela de treinamento *j*, adotou-se também uma janela de validação, de tamanho $v = s$. A primeira janela de treinamento *j* com validação *v* foi treinada por até 200 épocas ou quando o erro de generalização ultrapassasse 15%, conforme equação 6.1. Em seguida, fez-se o avanço do passo *s* e, daí por diante, realizou-se o treinamento por até 50 épocas ou quando o erro de generalização ultrapassasse 15%.

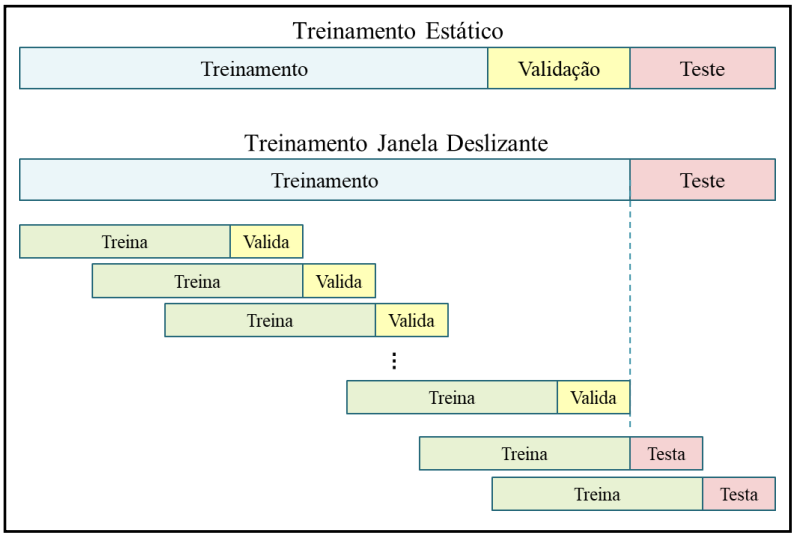

Figura 21 – Ilustração das formas de treinamento adotadas. Na parte de cima o treinamento estático usando 80% dos dados para treinamento, 10% para validação e 10% para teste. Na parte de baixo, o treinamento por janela deslizante, usando 90% dos dados para treinamento divididos em *n* janelas de treinamento tamanho *j* e validação de tamanho *v*. (Elaborado pelo autor)

Essa diferença, deve-se ao fato de haver necessidade de um treinamento maior no início do modelo, até que os seus pesos completamente aleatórios sejam ajustados para o problema, e as janelas seguintes servem apenas como um ajuste

fino para incorporar mudanças na série. Após chegar ao fim do *dataset* de treinamento, o mesmo procedimento de retreinar por janelas foi repetido após cada grupo de *s* dias de testes.

O presente estudo avaliou 5 tamanhos de janelas de treinamento diferentes: 250, 500, 750, 1000 e 1250 dias, que equivalem a 1, 2, 3, 4 e 5 anos de dados; e janelas de validação de 25 dias, equivalentes a 5 semanas de validação, aproximadamente 1 mês.

6.5. Resumo dos Cenários Analisados

Este trabalho se desenvolveu em torno de diversas variantes do *dataset* de notícias, que geraram cenários que foram simulados e analisados a partir de seus resultados, sendo alguns deles descartados ao longo da pesquisa. A primeira rodada de avaliação visou a responder os questionamentos presentes nos objetivos intermediários, com base nos 120 cenários descritos na seção 6.2.2.3.

As simulações realizadas do modelo SI-RCNN para uma classificação binária são apresentadas no Anexo A. A partir destes resultados, observa-se que o modelo pode ser aplicado à ação BBAS3 com resultados similares aos obtidos em VARGAS *et al.* [16] para ações da Chevron, mostrando, também, que há potencial no seu uso em outros ativos do mercado brasileiro. Além disso, para o problema em questão, melhores resultados são obtidos com o uso das *stopwords*, indicando que pode haver perda de semântica ao se retirar todas essas palavras do contexto.

Com relação ao filtro de frequência de palavras, *word2vec* versus *GloVe* e *dataset* completo versus *banknews*, não houve superioridade de nenhum cenário especificamente. Porém, considerando-se os resultados agregados, pode-se verificar resultados superiores no uso das frequências mínimas de 2, 10, 15, 25. Uma maior discussão desses resultados é apresentada na seção 7.1.

Após isso, os 20 cenários que apresentaram melhores resultados foram usados na fase seguinte do trabalho, na busca de respostas aos objetivos principais. A principal

alteração foi a adoção de uma saída ternária, onde, posteriormente, se aplicaram outras variações na janela de treinamento, entropia de Shannon e *Ensemble* de Modelos. Essas variações resultaram em novos 120 cenários, sendo 20 de treinamento estático e 100 usando janela deslizante (20 com cada tamanho de janela). A Figura 22 apresenta a árvore de cenários que foi aplicada na segunda etapa da pesquisa.

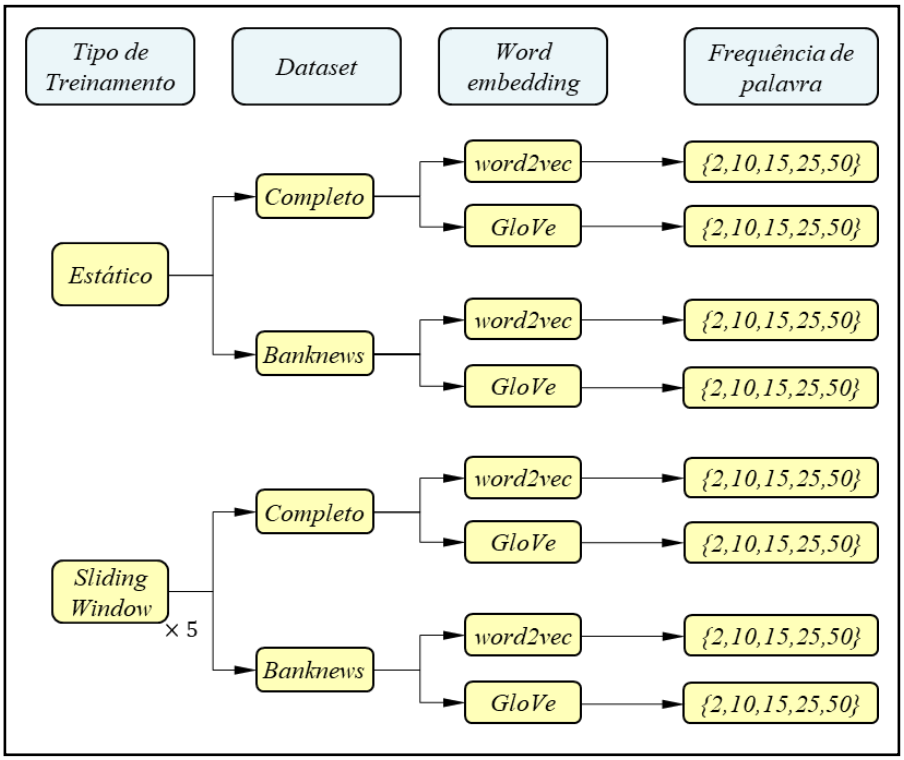

Figura 22 – Árvore de cenários com os 120 cenários analisados na segunda rodada de avaliação. (Elaborado pelo autor)

Cabe ressaltar que todos os diferentes cenários de texto foram testados usando-se o mesmo *dataset* de indicadores apresentados na seção 6.2.1. Diante da extensão de cenários derivados das variações do *dataset* de notícias, ficou inviável a aplicação de variações adicionais no conjunto de indicadores, como inclusão de preços do ativo, novos indicadores, alteração nas janelas de cálculo dos indicadores e utilização de mais de um ativo do mesmo setor, sendo estas variantes deixadas como propostas de trabalho futuro. No entanto, também foram simulados cenários sem os dados de texto, contando apenas com os dados dos indicadores aplicados a cada tipo de treinamento.

Após o treinamento de cada um desses cenários, os testes foram executados e as saídas do modelo foram enviadas para um filtro baseado na entropia de Shannon. Ou seja, calculada a entropia para cada registro, o resultado indica o grau de certeza da previsão – indicação de compra, venda ou não fazer nada. Se este valor for superior ao limiar estabelecido, executa-se a operação indicada, caso contrário, o modelo não opera. Foram aplicados limiares em 0,70, 0,80, 0,85, 0,90, 0,95 e 0,99 e os resultados foram comparados, com o intuito de verificar a efetividade da entropia de Shannon neste tipo de problema.

Após a aplicação da entropia em cada cenário, foi realizado o *ensemble* de modelos por tipo de treinamento, resultando em 6 resultados dessa combinação: *ensemble* estático, *ensemble sliding window* 250, *ensemble sliding window* 500, *ensemble sliding window* 750, *ensemble sliding window* 1000 e *ensemble sliding window* 1250. A construção do *ensemble* obedeceu aos seguintes critérios:

- Para cada dia de teste verificou-se a quantidade de indicações de compra, não fazer nada e venda, após a aplicação da entropia de Shannon.
- Esses valores são então normalizados e escritos em termos de probabilidades e, em seguida, é novamente aplicada a Entropia de Shannon, com o mesmo limiar usado na saída dos cenários individuais
- Se o resultado for superior ao limiar estabelecido, executa-se a operação indicada, caso contrário o modelo não opera.

Cada um desses resultados combinados foi comparado com os resultados individuais, avaliando-se os ganhos obtidos com o uso dessa tecnologia. Em resumo, o trabalho conta com 120 cenários apresentados na Figura 22, um cenário sem texto e um cenário ensemble para cada forma de treinamento, totalizando 132 cenários. Aplicando-se os 8 limiares da entropia a cada um desses cenários, tem-se um total de 1056 resultados, que são apresentados e analisados no capítulo 7.

6.6. Medidas de Avaliação dos Resultados

Primeiramente, definiu-se as classes do modelo segundo um critério que resultasse no melhor balanceamento possível das classes. Com base nisso, foi estabelecido um limite de retorno de *±0,85%* no log-retorno, a partir do qual seria interessante realizar operação. Ou seja, retornos abaixo de *-0,85%* correspondem à classe venda (*S - Sell*), retornos entre *-0,85%* e *0,85%*, inclusive, correspondem à classe não fazer nada (*H – Hold*) e acima de *0,85%* correspondem à classe compra (*B – Buy*). O resultado dessa divisão pode ser visto na Tabela 9.

Tabela 9 – Divisão das classes reais e sua prevalência no *dataset*

Classe	**N° Registros**	**Percentual**
B	675	32,9%
H	641	31,3%
S	734	35,8%
Total	2050	100%

Para avaliação dos resultados obtidos pelo modelo foram usados dois critérios, sendo o primeiro baseado na avaliação de modelos de *machine learning*, uma vez que se deseja comparar a efetividade das técnicas aplicadas. A segunda avaliação é baseada em conceitos do mercado financeiro, uma vez que um modelo não só deve ser capaz de realizar boas previsões, mas também trazer retornos superiores aos *benchmarks*, mesmo diante das restrições operacionais a que precisa se submeter.

6.6.1 AUC (*Area Under the ROC Curve*)

O espaço ROC (*Receiver Operating Characteristic*) é uma técnica gráfica amplamente usada na teoria de detecção de sinais para visualizar o desempenho de classificadores, baseada nas suas taxas de acerto e de alarmes falsos. Essa técnica provê uma medida de desempenho de classificadores superior a medidas como acurácia, erro, precisão e recuperação, uma vez que não favorece a classe de maior frequência em problemas desbalanceados [75][76].

Considerando-se as classes verdadeiras como *B*, *H* e *S*, e as classes preditas como $\hat{B}$, $\hat{H}$ e $\hat{S}$, respectivamente, pode-se construir a matriz de confusão do problema como na Tabela 10, onde são apresentados os valores de predição Verdadeira e Falsa para cada classe, bem como os valores totais, onde $\hat{N}_i$ corresponde ao total de valores preditos para a classe $i = \{B, H, S\}$, N_i corresponde ao total de valores reais para a classe $i = \{B, H, S\}$ e N corresponde ao total de registros de teste.

Tabela 10 – Matriz de confusão para o modelo em estudo

	$\hat{B}$	$\hat{H}$	$\hat{S}$	Total
B	V_B	F_H	F_S	N_B
H	F_B	V_H	F_S	N_H
S	F_B	F_H	V_S	N_S
Total	$\hat{N}_B$	$\hat{N}_H$	$\hat{N}_S$	N

Uma vez que antes de montar a matriz de confusão há a aplicação do filtro baseado na entropia de Shannon, os resultados que não atingem o limiar imposto não são classificados como Verdadeiro ou Falso de nenhuma das classes, sendo então estes resultados excluídos das estatísticas do modelo. Ou seja, a AUC é calculada apenas com base nas predições que possuem um mínimo de grau de certeza.

O Espaço ROC é construído com base na Taxa de Verdadeiro Positivo, também conhecida como recuperação (REC) - que corresponde à assertividade do modelo com relação à classe correspondente - e na Taxa de Falso Positivo, que é o complemento da especificidade (SPE) - que corresponde à quantidade de alarmes falsos dentro do universo falsos negativos, como pode ser visto na Figura 23. Essas taxas são calculadas como:

$$TVP_i = \frac{V_i}{N_i}, \tag{6.2}$$

$$TFP_i = 1 - SPE_i = \frac{F_i}{N - N_i}, \tag{6.3}$$

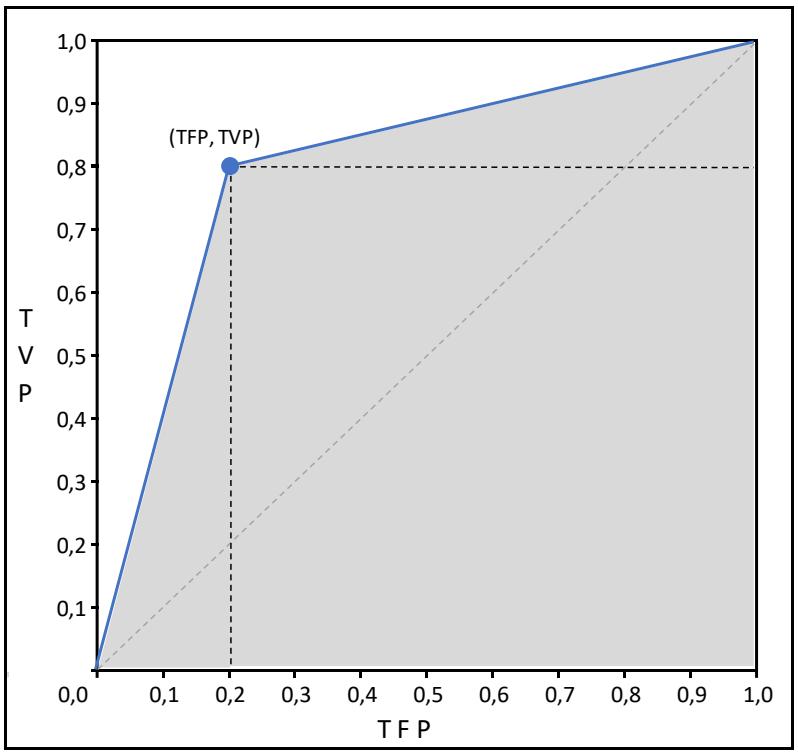

Figura 23 – Exemplo de gráfico do espaço ROC, apresentando como as medidas de Recuperação e Especificidade se relacionam para a construção da AUC. (Adaptado de EVSUKOFF [75])

A AUC refere-se à área sob a curva ROC, região hachurada da Figura 23, limitada pelo quadrilátero formado pelos vértices (0,0), (1,0), (1,1) e (TFP_i, TVP_i). Em problemas de classificação de múltiplas classes, a AUC do classificador deve ser calculado com base na AUC_i de cada classe, onde a classe *i* é considerada a classe positiva e as demais classes como classe negativa. Este cálculo pode ser realizado de forma simplificada pela soma das áreas de dois triângulos e um quadrado:

$$AUC_i = \frac{TFP_i \times TVP_i}{2} + (1 - TFP_i) \times TVP_i + \frac{TFP_i \times (1 - TVP_i)}{2}, \tag{6.4}$$

Posteriormente, calcula-se a AUC como uma média das *AUCi* ponderados pelas respectivas probabilidades a priori (P_i) de cada classe [75], apresentados na Tabela 11:

$$AUC = \sum_{i=B,H,S} AUC_i \times P_i, \tag{6.5}$$

Tabela 11 – Divisão das classes reais no *dataset* de teste e suas probabilidades a priori

Classe	N° Registros	Percentual
B	92	44,9%
H	49	23,9%
S	64	31,2%
Total	205	100%

Essa análise ajuda a compreender como essa medida não favorece uma classe de maior prevalência. Considerando-se o exemplo clássico de problema de classificação desbalanceado, a detecção de fraude em cartões de crédito possui, em média, a classe negativa para detecção em 99,9% do *dataset* e a classe positiva para detecção em 0,1% apenas. Qualquer classificador que apresente como saída 100% dos resultados na classe negativa resultará em uma acurácia de 99,9%, porém não faria nenhuma detecção de fraude e seria sem utilidade. Ou seja, apesar de sua *TVP* ser igual a 1, sua *TFP* também é igual a 1, resultando em uma *AUC = 0,5*, resultado equivalente ao puramente aleatório.

Sua interpretação leva à conclusão de que quanto mais próxima de 1 é a área, melhor o classificador. Ou seja, quando TVP_i tende a 1 e TFP_i tende a zero, o classificador é capaz de reconhecer bem a classe positiva e marcar muitos verdadeiros, cometendo poucos erros de falso positivo. Casos em que a *AUC < 0,5* indicam que o classificador apresenta resultado inferior a um classificador aleatório.

6.6.2 *Benchmarks* de Mercado

Além de alcançar bons resultados de AUC como classificador, um modelo de previsão de séries temporais financeiras precisa ser capaz de obter lucro frente às restrições impostas pelo mercado e superar os *benchmarks*. A principal restrição que o problema enfrenta é com relação à execução das operações, conforme observado na seção 2.1.

Como o modelo analisa dados diários históricos após o fechamento do mercado, a execução de uma operação indicada pelo modelo só pode ser realizada no dia útil seguinte à análise. Uma vez que o modelo tem por premissa efetuar a operação de compra ou de venda, quando houver indicação, a única maneira de garantir que uma ordem seja executada a um preço conhecido, sem enviesar o resultado do modelo, é com a realização das operações a mercado exclusivamente nos leilões de abertura e de fechamento do mercado. Ou seja, nas indicações de compra, o modelo efetua a compra no leilão de abertura do dia seguinte e a venda no leilão de fechamento, e nas indicações de venda, vende na abertura e compra no fechamento.

Com isso, há risco de o modelo acertar a direção do mercado e ainda assim obter retorno financeiro negativo, pois o modelo pode indicar compra, o fechamento de *t+1* pode ser superior ao fechamento de *t*, mas o preço de abertura de *t+1* pode ser superior ao fechamento de *t+1*.

Outras restrições à operacionalização do modelo são: a possibilidade de acertar movimentos cujos retornos financeiros são ligeiramente superiores a 0 e errar movimentos com perdas elevadas; as taxas de corretagem, emolumentos e impostos sobre serviço e o imposto de renda de 20% sobre o lucro nas operações *daytrade*, cobrados conforme Tabela 12. Atualmente, há corretoras no mercado que não cobram taxas de corretagem nas operações, sendo assim, este trabalho considera corretagem zero e ISS sobre corretagem também igual a zero.

Tabela 12 – Tarifas e impostos incidentes nas operações

Taxa / Imposto	**Percentual**	**Incidência**
Negociação	0,004032%	Sobre o valor da operação
Liquidação	0,020000%	Sobre o valor da operação
Corretagem	0%	Não incidente
ISS	5%	Sobre a taxa de corretagem
IRPF	20%	Sobre o lucro da operação

Ao final dos testes, calculados os retornos diários e cumulativo, os cenários são comparados com benchmarks do mercado. Neste trabalho, adotou-se como referências o

CDI, o Ibovespa, a estratégia ingênua (Naïve), que consiste em repetir a operação direcional do dia anterior, e o *buy-and-hold*, que reflete fielmente a evolução do preço da ação, cujos retornos são apresentados na Figura 24.

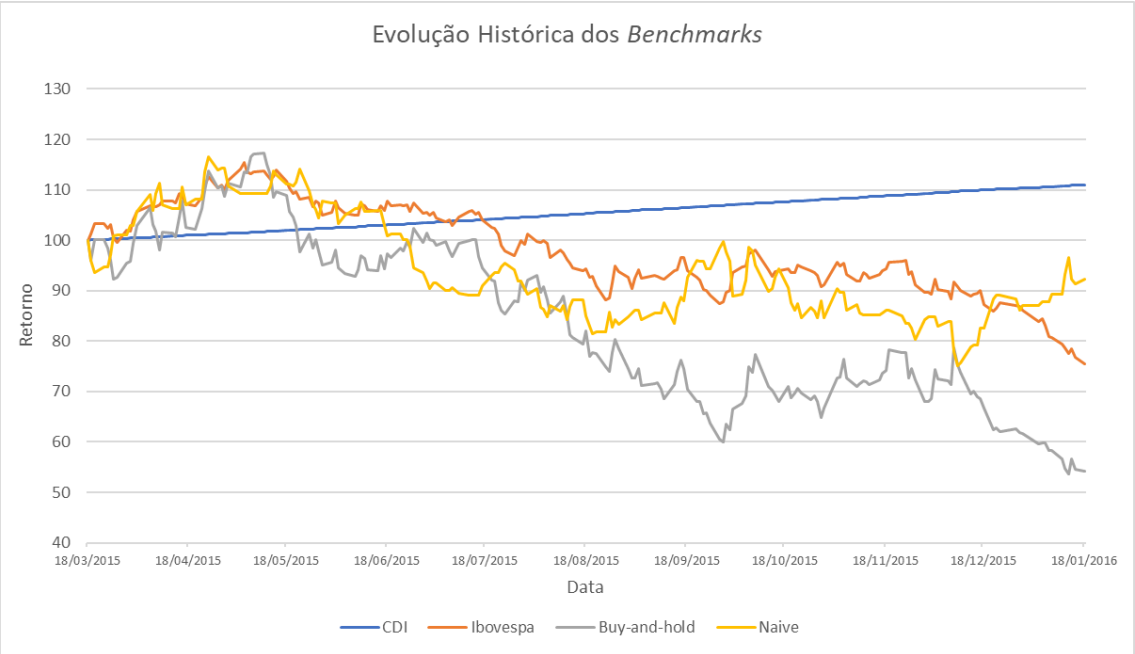

Figura 24 – Gráfico da evolução dos *benchmarks* no período de teste do modelo (19/03/2015 a 18/01/2016) em termos de base 100. (Fonte: B3. Elaborado pelo autor)

7. Resultados

Os resultados obtidos pelo modelo são apresentados em tabelas de forma agrupada por tipo de treinamento. Nelas é possível ver os resultados de AUC para cada cenário e as variações de acordo com o aumento do nível de certeza requerido pela entropia de Shannon. Cabe ressaltar que, ao aumentar a restrição da entropia, as operações somente são permitidas às saídas que se enquadrem nestes critérios e, em situações limites, onde o modelo não apresenta nenhuma previsão com baixa entropia, sua AUC final é "N/A", uma vez que o sistema não foi capaz de operar.

Os cenários são identificados seguindo-se a nomenclatura "*dataset_stopwords*_filtrodefrequência_vetorização", ou seja, o cenário chamado "banknews_csw_k25_glove" corresponde ao uso do *dataset* de notícias *banknews*, com stopwords, frequência mínima da palavra de 25 vezes e representação GloVe.

Os resultados dos cenários por tipo de treinamento são apresentados nas tabelas ímpares, da Tabela 13 à Tabela 23. Nelas podem ser vistos os resultados de AUC para os 20 diferentes cenários de texto, o resultado da combinação de modelos, identificado na linha "*ensemble*", o resultado do cenário sem texto e uma média das AUCs dos cenários com texto, separados de acordo com o limiar de entropia aplicado. Esta estrutura de apresentação dos resultados permite que seja feita a avaliação da eficácia da entropia e da combinação de modelos dentro de cada tipo de treinamento.

As tabelas pares, desde a Tabela 14 até a Tabela 24, apresentam quadros resumo, por tipo de treinamento, das quantidades de cenários com AUC maior do que *0,5*, menor do que *0,5*, iguais a *0,5* e sem operação. A forma de apresentação permite verificar como o filtro de entropia influencia na assertividade dos modelos, em especial quando se observa a evolução da relação entre as quantidades de cenários com AUC maior que 0,5 e as com AUC menor que 0,5 (última linha das tabelas). Resultados crescentes nesses valores, quando se aumenta a restrição de entropia, indicam um aumento na assertividade dos cenários em geral, reduzindo em maior escala as taxas de falso

positivo por classe, ainda que as taxas de verdadeiro positivo por classe possam estar reduzindo também.

Posteriormente, faz-se um comparativo entre os resultados da combinação de modelos do treinamento estático e dos treinamentos dinâmicos, mostrado na Figura 25 e na Figura 26. Esta forma de apresentação dos resultados permite uma verificação das vantagens do treinamento de janela deslizante sobre o treinamento estático, bem como permite verificar qual tamanho de janela apresenta melhor resultado para o problema estudado.

Na Tabela 13, observam-se valores de AUC abaixo de 0,5 na maior parte dos cenários na coluna "sem entropia". Isso reflete a dificuldade do modelo em capturar as relações entre as notícias, indicadores e as mudanças do mercado ao longo do tempo. Da mesma forma, o cenário *ensemble* é incapaz da apresentar um bom resultado, tanto em termos absolutos (*AUC<0,5*), quanto em comparação com os cenários individuais. Nota-se também a baixa certeza nas previsões, uma vez que um limiar de entropia de 0,95 é suficiente para que todos os cenários, com exceção do cenário sem texto, não executem nenhuma operação.

Na Tabela 14, que mostra a evolução das quantidades de cenários de acordo com o resultado de AUC, percebe-se que quanto mais restritivo o limiar de entropia, menor a quantidade de cenários com *AUC < 0,5*. Isso é um indicativo de que o filtro de entropia impede que as saídas das redes com baixo grau de certeza resultem em operações erradas.

A Tabela 15 apresenta os resultados dos cenários com o treinamento *sliding window* com janela de 250 dias. Observam-se valores de AUC iguais a 0,5 em muitos cenários, devido à rede indicar uma mesma resposta para todos os dias de teste, resultando em 100% de verdadeiro positivo e também de falso positivo em uma das classes. A possível causa disso é o pequeno tamanho da janela de treinamento, que pode enviesar o modelo, por uma tendência de curto prazo, a indicar sempre um determinado tipo de operação.

Tabela 13 – AUC para os 22 cenários do treinamento estático e as variações com a mudança no limiar da entropia

Treinamento Estático Cenario	Sem Entropia	Entropia < 0,99	Entropia < 0,98	Entropia < 0,95	Entropia < 0,9	Entropia < 0,85	Entropia < 0,8	Entropia < 0,7
banknews_csw_k02_glove	0,489	0,489	0,512	N/A	N/A	N/A	N/A	N/A
banknews_csw_k02_w2v-skip	0,493	N/A	N/A	N/A	N/A	N/A	N/A	N/A
banknews_csw_k10_glove	0,521	**0,529**	**0,513**	N/A	N/A	N/A	N/A	N/A
banknews_csw_k10_w2v-skip	0,469	N/A	N/A	N/A	N/A	N/A	N/A	N/A
banknews_csw_k15_glove	0,500	N/A	N/A	N/A	N/A	N/A	N/A	N/A
banknews_csw_k15_w2v-skip	0,484	0,474	0,503	N/A	N/A	N/A	N/A	N/A
banknews_csw_k25_glove	0,518	N/A	N/A	N/A	N/A	N/A	N/A	N/A
banknews_csw_k25_w2v-skip	0,483	0,462	0,493	N/A	N/A	N/A	N/A	N/A
banknews_csw_k50_glove	0,490	0,503	N/A	N/A	N/A	N/A	N/A	N/A
banknews_csw_k50_w2v-skip	0,506	0,502	0,489	N/A	N/A	N/A	N/A	N/A
complete_csw_k02_glove	0,471	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k02_w2v-skip	0,479	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k10_glove	0,486	0,481	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k10_w2v-skip	0,490	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k15_glove	0,502	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k15_w2v-skip	0,489	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k25_glove	0,472	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k25_w2v-skip	0,498	N/A	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k50_glove	0,505	0,469	N/A	N/A	N/A	N/A	N/A	N/A
complete_csw_k50_w2v-skip	0,499	0,491	0,498	N/A	N/A	N/A	N/A	N/A
ensemble	0,485	0,494	0,492	N/A	N/A	N/A	N/A	N/A
sem_texto	**0,529**	0,501	0,509	0,498	N/A	N/A	N/A	N/A
AUC média dos cenários de texto	0,492	0,489	0,501	N/A	N/A	N/A	N/A	N/A

Tabela 14 – Quadro resumo de AUC e entropia nos cenários de treinamento estático

Treinamento Estático	Sem Entropia	Entropia < 0,99	Entropia < 0,98	Entropia < 0,95	Entropia < 0,9	Entropia < 0,85	Entropia < 0,8	Entropia < 0,7
Quant, Cenários com AUC > 0,5	6	4	4	0	0	0	0	0
Quant, Cenários com AUC < 0,5	15	7	4	1	0	0	0	0
Quant, Cenários com AUC = 0,5	1	0	0	0	0	0	0	0
Quantidade de Cenários Sem Operação	0	11	14	21	22	22	22	22
Relação entre Qnt AUC>0,5 e AUC<0,5	0,40	0,57	1,00	0,00	N/A	N/A	N/A	N/A

Da mesma forma, o cenário *ensemble* também recebe esse viés, não conseguindo apresentar um bom resultado, tanto em termos absolutos quanto em comparação com os cenários individuais. A exceção fica por conta de quando se limita a entropia em 0,8, onde o cenário combinado apresenta resultado superior aos individuais. Nota-se também a baixa certeza nas previsões, uma vez que um limiar de entropia de 0,9 é o suficiente para que a maioria dos cenários deixe de fazer previsões, e um limiar de 0,7 torne todos os cenários sem utilidade.

Observando-se a Tabela 16, pode-se verificar que o número de cenários com AUC > 0,5 e o número com AUC < 0,5 aumentam até o limiar de 0,98 e depois diminuem. Esse

comportamento difere do esperado, uma vez que se espera que os modelos incorram em menos erros e que a relação entre quantidade de cenários com AUC > 0,5 e AUC < 0,5 aumente progressivamente. Apesar disso, conforme o filtro da entropia fica mais restritivo, verifica-se novamente o comportamento adequado dos resultados, com a diminuição da quantidade de cenários com AUC baixo.

Tabela 15 – AUC para os 22 cenários do treinamento *sliding window* 250 e as variações com a mudança no limiar da entropia

Treinamento SW 250 / **Cenario**	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
banknews_csw_k02_glove	0,495	0,494	0,504	0,503	0,498	N/A	N/A	N/A
banknews_csw_k02_w2v-skip	0,478	0,478	0,493	0,491	0,503	0,501	N/A	N/A
banknews_csw_k10_glove	0,478	0,485	0,482	0,493	0,494	0,502	N/A	N/A
banknews_csw_k10_w2v-skip	0,500	0,500	0,507	N/A	N/A	N/A	N/A	N/A
banknews_csw_k15_glove	0,500	0,502	0,504	0,502	0,504	0,500	0,516	N/A
banknews_csw_k15_w2v-skip	0,500	0,508	0,503	0,500	0,501	0,494	0,514	N/A
banknews_csw_k25_glove	**0,543**	0,512	0,508	0,508	**0,510**	**0,510**	0,502	N/A
banknews_csw_k25_w2v-skip	0,528	**0,535**	0,514	0,505	N/A	N/A	N/A	N/A
banknews_csw_k50_glove	0,500	0,502	0,500	0,495	0,485	0,468	0,501	N/A
banknews_csw_k50_w2v-skip	0,500	0,527	0,503	N/A	N/A	N/A	N/A	N/A
complete_csw_k02_glove	0,500	0,500	0,474	N/A	N/A	N/A	N/A	N/A
complete_csw_k02_w2v-skip	0,500	0,500	0,474	N/A	N/A	N/A	N/A	N/A
complete_csw_k10_glove	0,500	0,500	0,474	**0,509**	N/A	N/A	N/A	N/A
complete_csw_k10_w2v-skip	0,500	0,500	0,474	N/A	N/A	N/A	N/A	N/A
complete_csw_k15_glove	0,500	0,500	0,474	0,491	N/A	N/A	N/A	N/A
complete_csw_k15_w2v-skip	0,511	0,501	**0,523**	0,506	N/A	N/A	N/A	N/A
complete_csw_k25_glove	0,500	0,500	0,474	N/A	N/A	N/A	N/A	N/A
complete_csw_k25_w2v-skip	0,500	0,500	0,474	0,491	N/A	N/A	N/A	N/A
complete_csw_k50_glove	0,506	0,493	0,490	0,498	N/A	N/A	N/A	N/A
complete_csw_k50_w2v-skip	0,500	0,500	0,474	N/A	N/A	N/A	N/A	N/A
ensemble	0,500	0,500	0,502	0,500	0,498	0,495	**0,521**	N/A
sem_texto	0,500	0,500	0,483	N/A	N/A	N/A	N/A	N/A
AUC média dos cenários de texto	0,502	0,502	0,491	0,500	0,499	0,496	0,508	N/A

Tabela 16 – Quadro resumo de AUC e entropia nos cenários de treinamento *sliding window* 250

Treinamento SW 250	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
Quant, Cenários com AUC > 0,5	5	7	9	7	4	4	5	0
Quant, Cenários com AUC < 0,5	3	5	13	6	4	3	0	0
Quant, Cenários com AUC = 0,5	14	10	0	1	0	0	0	0
Quantidade de Cenários Sem Operação	0	0	0	8	14	15	17	22
Relação entre Qnt AUC>0,5 e AUC<0,5	1,67	1,40	0,69	1,17	1,00	1,33	N/A	N/A

Os resultados dos cenários com o treinamento *sliding window* com janela de 500 dias são apresentados na Tabela 17. Ao se usar uma janela de treinamento maior que os 250 dias, encontram-se resultados de AUC superiores em quase todos os cenários e,

consequentemente, uma melhora na média dos resultados individuais e também do *ensemble*. Neste caso, observa-se que a aplicação conjunta do limiar da entropia e do *ensemble* resultam em resultados frequentemente superiores à média, culminando com um AUC superior a todos os cenários individuais, quando se solicita ao modelo uma entropia menor do que 0,7 para operar.

Tabela 17 - AUC para os 22 cenários do treinamento *sliding window* 500 e as variações com a mudança no limiar da entropia

Treinamento SW 500	**Sem**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**
Cenario	**Entropia**	**< 0,99**	**< 0,98**	**< 0,95**	**< 0,9**	**< 0,85**	**< 0,8**	**< 0,7**
banknews_csw_k02_glove	0,499	0,514	0,505	0,502	N/A	N/A	N/A	N/A
banknews_csw_k02_w2v-skip	0,479	0,488	0,500	0,500	N/A	N/A	N/A	N/A
banknews_csw_k10_glove	0,497	0,511	0,515	0,523	0,514	0,509	0,500	0,512
banknews_csw_k10_w2v-skip	0,533	0,522	0,513	0,504	N/A	N/A	N/A	N/A
banknews_csw_k15_glove	0,477	0,490	0,498	0,497	0,503	0,501	0,499	0,501
banknews_csw_k15_w2v-skip	0,496	0,497	0,501	0,510	N/A	N/A	N/A	N/A
banknews_csw_k25_glove	**0,537**	**0,532**	**0,533**	**0,531**	**0,526**	0,524	**0,525**	0,517
banknews_csw_k25_w2v-skip	0,486	0,497	0,493	0,487	0,497	0,501	N/A	N/A
banknews_csw_k50_glove	0,500	0,506	0,505	0,502	N/A	N/A	N/A	N/A
banknews_csw_k50_w2v-skip	0,504	0,518	0,513	0,494	0,496	0,495	0,498	0,505
complete_csw_k02_glove	0,516	0,509	0,507	0,502	0,504	0,506	0,489	0,494
complete_csw_k02_w2v-skip	0,477	0,482	0,491	0,487	0,488	0,486	0,498	N/A
complete_csw_k10_glove	0,522	0,494	0,496	0,501	0,499	N/A	N/A	N/A
complete_csw_k10_w2v-skip	0,512	0,519	0,522	0,521	0,521	**0,528**	0,506	0,508
complete_csw_k15_glove	0,530	0,521	0,520	0,521	0,516	0,503	0,503	0,504
complete_csw_k15_w2v-skip	0,495	0,499	0,492	0,499	N/A	N/A	N/A	N/A
complete_csw_k25_glove	0,524	0,517	0,517	0,517	0,512	0,518	0,502	0,500
complete_csw_k25_w2v-skip	0,474	0,503	0,498	0,492	0,501	0,496	0,501	0,507
complete_csw_k50_glove	0,486	0,504	0,500	0,494	0,496	0,499	N/A	N/A
complete_csw_k50_w2v-skip	0,479	0,504	0,502	0,503	N/A	N/A	N/A	N/A
ensemble	0,495	0,531	0,528	0,521	0,518	0,525	0,521	**0,533**
sem_texto	0,466	0,482	0,483	0,488	N/A	N/A	N/A	N/A
AUC média dos cenários de texto	0,501	0,506	0,506	0,505	0,506	0,505	0,502	0,505

Tabela 18 – Quadro resumo de AUC e entropia nos cenários de treinamento *sliding window* 500

Treinamento SW 500	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
Quant, Cenários com AUC > 0,5	8	14	13	14	9	9	7	9
Quant, Cenários com AUC < 0,5	13	8	9	8	5	4	4	1
Quant, Cenários com AUC = 0,5	1	0	0	0	0	0	0	0
Quantidade de Cenários Sem Operação	0	0	0	0	8	9	11	12
Relação entre Qnt AUC>0,5 e AUC<0,5	0,62	1,75	1,44	1,75	1,80	2,25	1,75	9,00

A Tabela 18 mostra, para o treinamento sw500, que os resultados dos modelos apresentam o comportamento esperado com a aplicação mais restritiva da entropia de Shannon, ou seja, enquanto aumenta a quantidade de cenários com AUC > 0,5,

diminuem os cenários com AUC < 0,5. Isso fica evidente na última linha do quadro, onde a relação de AUCs aumenta da esquerda para a direita.

Prosseguindo no aumento do tamanho da janela de treinamento, a Tabela 19 apresenta os resultados dos cenários com o treinamento *sliding window* com janela de 750 dias. Nota-se, pela comparação entre as médias das AUCs, uma melhora nos resultados em relação aos cenários com janela de 500 dias, indicando que esse tamanho de janela melhora a capacidade dos modelos de capturar as relações entre notícias e indicadores, bem como prever corretamente o direcional do mercado. Isso também pode ser observado pelos valores de AUC terem ultrapassado pela primeira vez 0,55. Esses resultados superiores refletem diretamente nos resultados do modelo *ensemble*, que alcança AUCs acima da média dos cenários individuais.

Tabela 19 - AUC para os 22 cenários do treinamento *sliding window* 750 e as variações com a mudança no limiar da entropia

Treinamento SW 750	**Sem**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**
Cenario	**Entropia**	**< 0,99**	**< 0,98**	**< 0,95**	**< 0,9**	**< 0,85**	**< 0,8**	**< 0,7**
banknews_csw_k02_glove	**0,538**	0,538	0,539	0,541	**0,546**	**0,545**	**0,560**	0,519
banknews_csw_k02_w2v-skip	0,488	0,501	0,502	0,491	0,488	0,492	0,498	0,506
banknews_csw_k10_glove	0,479	0,480	0,484	0,491	0,493	0,492	0,487	0,491
banknews_csw_k10_w2v-skip	0,483	0,490	0,488	0,486	0,489	0,491	0,489	0,494
banknews_csw_k15_glove	0,511	0,531	0,532	0,531	0,532	0,531	0,527	0,529
banknews_csw_k15_w2v-skip	0,479	0,480	0,484	0,486	0,496	0,495	0,496	0,511
banknews_csw_k25_glove	0,497	0,521	0,521	0,522	0,523	0,526	0,524	0,523
banknews_csw_k25_w2v-skip	0,503	0,480	0,490	0,511	0,503	0,497	0,498	0,499
banknews_csw_k50_glove	0,486	0,504	0,505	0,507	0,504	0,502	0,508	0,514
banknews_csw_k50_w2v-skip	0,496	0,507	0,509	0,510	0,508	0,515	0,518	0,517
complete_csw_k02_glove	0,507	0,529	0,530	0,531	0,529	0,528	0,530	0,514
complete_csw_k02_w2v-skip	0,484	0,491	0,491	0,490	0,490	0,486	0,493	0,491
complete_csw_k10_glove	0,506	0,515	0,513	0,514	0,512	0,517	0,515	0,516
complete_csw_k10_w2v-skip	0,494	0,506	0,504	0,501	0,503	0,502	0,507	0,504
complete_csw_k15_glove	0,490	0,493	0,492	0,492	0,501	0,497	0,503	0,495
complete_csw_k15_w2v-skip	0,503	0,503	0,504	0,500	0,500	0,513	0,514	0,510
complete_csw_k25_glove	0,513	0,523	0,521	0,519	0,516	0,522	0,516	0,512
complete_csw_k25_w2v-skip	0,470	0,491	0,502	0,508	0,505	0,507	0,512	0,509
complete_csw_k50_glove	0,534	**0,553**	**0,549**	**0,543**	0,543	0,542	0,541	0,532
complete_csw_k50_w2v-skip	0,497	0,502	0,504	0,503	0,501	0,506	0,514	**0,534**
ensemble	0,523	0,517	0,512	0,520	0,529	0,517	0,533	0,529
sem_texto	0,482	0,504	0,507	0,497	0,496	0,502	N/A	N/A
AUC média dos cenários de texto	0,498	0,507	0,508	0,509	0,509	0,510	0,513	0,511

Sob a ótica das quantidades de cenários por grupo de AUC, o aumento da restrição da entropia resulta em mais modelos com AUC elevado e menos com AUC < 0,5,

apresentando uma evolução gradativa da relação de AUC > 0,5 e AUC < 0,5, conforme pode ser visto na Tabela 20.

Tabela 20 – Quadro resumo de AUC e entropia nos cenários de treinamento *sliding window* 750

Treinamento SW 750	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
Quant, Cenários com AUC > 0,5	9	15	16	14	16	15	15	16
Quant, Cenários com AUC < 0,5	13	7	6	8	6	7	6	5
Quant, Cenários com AUC = 0,5	0	0	0	0	0	0	0	0
Quantidade de Cenários Sem Operação	0	0	0	0	0	0	1	1
Relação entre Qnt AUC>0,5 e AUC<0,5	0,69	2,14	2,67	1,75	2,67	2,14	2,50	3,20

Aumentando-se o tamanho de janela de treinamento *sliding window* para 1000 dias (Tabela 21), observa-se resultados similares ao observado com 750 dias, apesar de não haver nenhum cenário de destaque. Além disso, esta janela de treinamento também traz melhora nos resultados do *ensemble*, oscilando por valores de AUC superiores a 0,5 em todos os cenários com restrição de entropia. Ou seja, a restrição da entropia melhora os resultados dos cenários individuais e essa melhora também se reflete no *ensemble*.

Tabela 21 - AUC para os 22 cenários do treinamento *sliding window* 1000 e as variações com a mudança no limiar da entropia

Treinamento SW 1000	**Sem**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**
Cenario	**Entropia**	**< 0,99**	**< 0,98**	**< 0,95**	**< 0,9**	**< 0,85**	**< 0,8**	**< 0,7**
banknews_csw_k02_glove	0,491	0,495	0,490	0,502	0,499	0,492	0,494	0,498
banknews_csw_k02_w2v-skip	0,492	0,484	0,492	0,503	0,513	0,519	0,505	N/A
banknews_csw_k10_glove	0,525	0,523	0,514	0,509	0,508	0,505	0,498	0,501
banknews_csw_k10_w2v-skip	0,492	0,493	0,512	0,512	0,500	0,494	0,497	N/A
banknews_csw_k15_glove	0,484	0,487	0,504	0,508	0,498	0,500	0,500	N/A
banknews_csw_k15_w2v-skip	0,513	0,499	0,505	0,516	N/A	N/A	N/A	N/A
banknews_csw_k25_glove	0,530	**0,530**	**0,525**	0,519	0,519	**0,525**	0,518	0,517
banknews_csw_k25_w2v-skip	0,472	0,479	0,472	0,496	0,505	0,496	0,497	0,499
banknews_csw_k50_glove	0,485	0,502	0,507	0,499	0,500	0,506	0,519	0,514
banknews_csw_k50_w2v-skip	0,490	0,500	0,505	0,501	0,505	0,509	0,508	0,514
complete_csw_k02_glove	0,488	0,489	0,491	0,500	0,508	0,516	**0,528**	0,502
complete_csw_k02_w2v-skip	0,498	0,498	0,503	0,504	0,500	0,498	0,499	0,510
complete_csw_k10_glove	0,491	0,512	0,515	0,518	0,501	0,494	0,499	N/A
complete_csw_k10_w2v-skip	0,502	0,494	0,492	0,493	0,503	0,499	N/A	N/A
complete_csw_k15_glove	**0,534**	0,519	0,516	0,514	**0,523**	0,517	0,510	0,505
complete_csw_k15_w2v-skip	0,513	0,493	0,494	0,494	0,487	0,490	0,498	0,508
complete_csw_k25_glove	0,505	0,500	0,496	0,505	0,506	0,506	0,504	0,499
complete_csw_k25_w2v-skip	0,512	0,513	0,514	0,518	0,514	0,514	0,511	0,511
complete_csw_k50_glove	0,478	0,487	0,494	0,491	0,483	0,488	0,492	0,493
complete_csw_k50_w2v-skip	0,456	0,458	0,450	0,477	0,489	0,496	N/A	N/A
ensemble	0,513	0,514	0,517	**0,525**	0,512	0,519	0,526	**0,526**
sem_texto	0,528	0,502	0,502	0,501	0,502	0,502	0,498	N/A
AUC média dos cenários de texto	0,497	0,498	0,500	0,504	0,503	0,503	0,504	0,505

Em contrapartida, o quadro apresentado na Tabela 22 apresenta resultados inferiores ao do treinamento com 750 dias, com menor quantidade de cenários com AUC > 0,5 e ligeiro aumento dos cenários sem operação ao longo do aumento da restrição da entropia, porém, ainda conduzindo a uma evolução gradativa da relação de AUC > 0,5 e AUC < 0,5.

Tabela 22 – Quadro resumo de AUC e entropia nos cenários de treinamento *sliding window* 1000

Treinamento SW 1000	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
Quant, Cenários com AUC > 0,5	10	10	13	16	15	12	9	10
Quant, Cenários com AUC < 0,5	12	12	9	6	6	9	10	4
Quant, Cenários com AUC = 0,5	0	0	0	0	0	0	0	0
Quantidade de Cenários Sem Operação	0	0	0	0	1	1	3	8
Relação entre Qnt AUC>0,5 e AUC<0,5	0,83	0,83	1,44	2,67	2,50	1,33	0,90	2,50

Tabela 23 - AUC para os 22 cenários do treinamento *sliding window* 1250 e as variações com a mudança no limiar da entropia

Treinamento SW 1250	**Sem**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**	**Entropia**
Cenario	**Entropia**	**< 0,99**	**< 0,98**	**< 0,95**	**< 0,9**	**< 0,85**	**< 0,8**	**< 0,7**
banknews_csw_k02_glove	0,508	0,508	0,506	0,503	0,504	0,504	0,512	0,516
banknews_csw_k02_w2v-skip	**0,542**	**0,547**	**0,548**	**0,546**	0,536	0,537	0,538	0,530
banknews_csw_k10_glove	0,509	0,507	0,507	0,510	0,508	0,506	0,512	0,523
banknews_csw_k10_w2v-skip	0,530	0,520	0,521	0,525	0,530	0,533	0,532	0,521
banknews_csw_k15_glove	0,531	0,533	0,535	0,543	**0,550**	**0,558**	**0,561**	**0,566**
banknews_csw_k15_w2v-skip	0,498	0,498	0,498	0,498	0,497	0,499	0,495	0,505
banknews_csw_k25_glove	0,533	0,536	0,535	0,528	0,536	0,540	0,538	0,537
banknews_csw_k25_w2v-skip	0,507	0,512	0,515	0,514	0,516	0,521	0,528	0,516
banknews_csw_k50_glove	0,533	0,543	0,545	0,542	0,527	0,527	0,528	0,504
banknews_csw_k50_w2v-skip	0,492	0,502	0,506	0,507	0,505	0,504	0,502	0,520
complete_csw_k02_glove	0,458	0,476	0,481	0,492	0,493	0,508	0,507	0,505
complete_csw_k02_w2v-skip	0,483	0,483	0,481	0,485	0,482	0,496	0,492	0,520
complete_csw_k10_glove	0,496	0,486	0,485	0,487	0,478	0,477	0,487	0,488
complete_csw_k10_w2v-skip	0,487	0,479	0,479	0,488	0,498	0,488	0,494	0,488
complete_csw_k15_glove	0,478	0,479	0,490	0,483	0,500	0,503	0,508	0,506
complete_csw_k15_w2v-skip	0,478	0,465	0,472	0,475	0,500	0,502	0,503	0,499
complete_csw_k25_glove	0,493	0,495	0,492	0,494	0,493	0,495	0,500	0,496
complete_csw_k25_w2v-skip	0,509	0,514	0,517	0,511	0,500	0,500	0,497	0,498
complete_csw_k50_glove	0,539	0,530	0,526	0,512	0,516	0,505	0,507	0,499
complete_csw_k50_w2v-skip	0,513	0,507	0,504	0,505	0,510	0,506	0,504	0,508
ensemble	0,513	0,490	0,488	0,495	0,500	0,499	0,506	0,532
sem_texto	0,489	0,483	0,488	0,489	N/A	N/A	N/A	N/A
AUC média dos cenários de texto	0,506	0,506	0,507	0,507	0,509	0,511	0,512	0,512

Por fim, a Tabela 23 apresenta os resultados para o maior tamanho de janela de treinamento para o *sliding window*: 1250 dias. Apesar da significativa melhora nos resultados individuais com o *dataset banknews*, essa melhora não foi capitalizada pelo

ensemble dos modelos, tendo oscilado em torno da AUC = 0,5 e só obtendo significativa melhora no cenário com entropia mais restritiva (entropia < 0,7).

Da mesma forma que os cenários com janela de 1000 dias, houve aumento na quantidade de cenários com AUC > 0,5 e melhora ao longo do aumento da restrição da entropia, conforme pode ser visto na Tabela 24, levando a uma evolução gradativa da relação de AUC > 0,5 e AUC < 0,5. Cabe ressaltar que este tamanho de janela permitiu a mais nítida melhoria nesta relação de AUC, assim como também trouxe maior quantidade de cenários com AUC superior a 0,5.

Tabela 24 – Quadro resumo de AUC e entropia nos cenários de treinamento *sliding window* 1250

Treinamento SW 1250	**Sem Entropia**	**Entropia < 0,99**	**Entropia < 0,98**	**Entropia < 0,95**	**Entropia < 0,9**	**Entropia < 0,85**	**Entropia < 0,8**	**Entropia < 0,7**
Quant, Cenários com AUC > 0,5	12	12	12	12	14	15	16	15
Quant, Cenários com AUC < 0,5	10	10	10	10	7	6	5	6
Quant, Cenários com AUC = 0,5	0	0	0	0	0	0	0	0
Quantidade de Cenários Sem Operação	0	0	0	0	1	1	1	1
Relação entre Qnt AUC>0,5 e AUC<0,5	1,20	1,20	1,20	1,20	2,00	2,50	3,20	2,50

Compilando-se os resultados dos quadros resumo apresentados, observa-se na Figura 25 um gráfico com a evolução da relação entre a quantidade de cenários com AUC > 0,5 e AUC < 0,5 com o aumento da restrição do limiar de entropia. Esse gráfico indica que conforme aumenta-se o nível de certeza requerido dos modelos, maior a quantidade de cenários que respondem com AUCs maiores que 0,5, ao menos tempo que diminui o número de cenários com AUC baixo. Ou seja, há um indicativo de melhoria nos resultados finais de previsão quando aplicados limiares mais restritivos na entropia.

Além disso, agrupando-se os resultados dos modelos *ensemble*, é possível avaliar a evolução de seus desempenhos ao longo de situações mais restritivas de entropia, como apresentado na Figura 26. Pode-se perceber uma inclinação positiva nas curvas de evolução da AUC por tipo de treinamento conforme se solicita maiores níveis de certeza das respostas. Ou seja, em geral, as respostas de baixa certeza (ou elevada entropia) descartadas são predições ruins dos modelos, e que resultariam em falso positivo para a classe. Então, conforme são descartadas, os resultados de melhor certeza sobressaem e resultam em melhoria na AUC do modelo combinado.

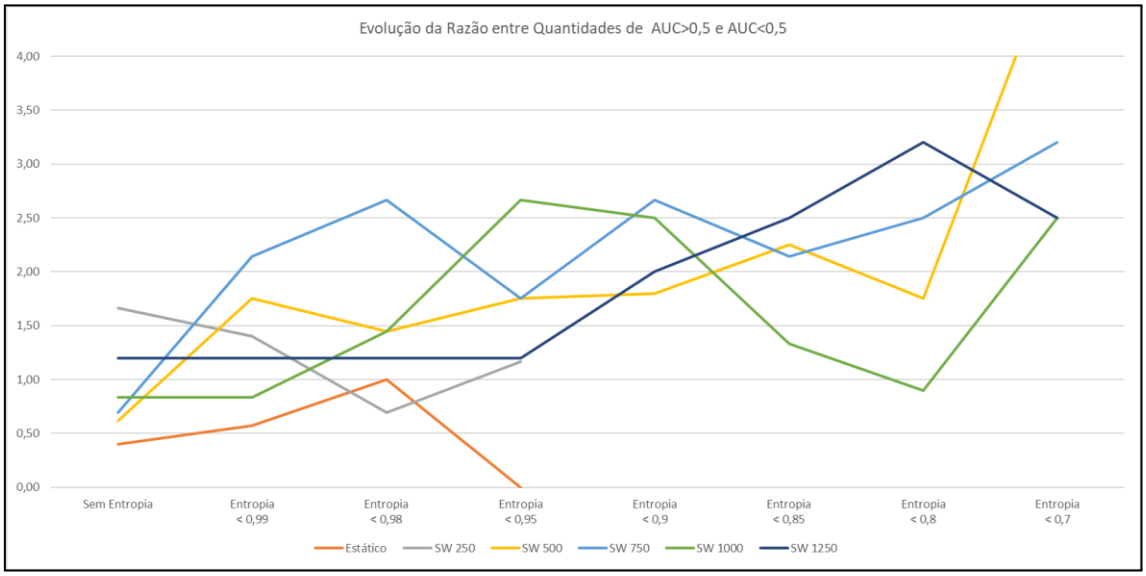

Figura 25 – Evolução por tipo de treinamento da relação entre a quantidade de cenários com AUC > 0,5 e AUC < 0,5 com o aumento da restrição do limiar de entropia.

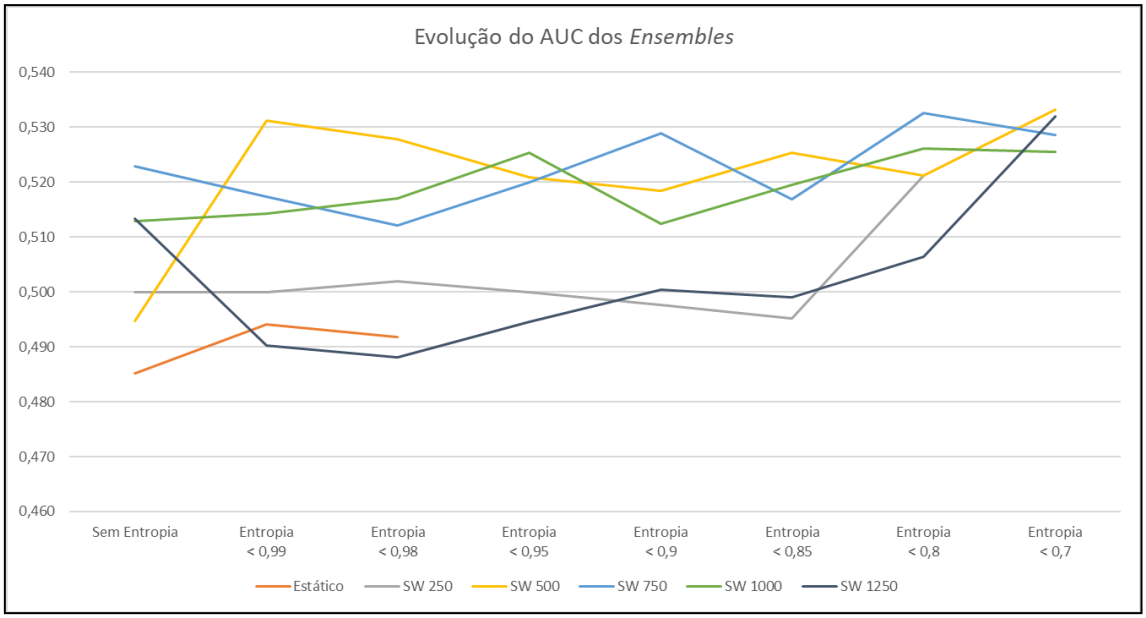

Figura 26 – Evolução do resultado da AUC dos *ensembles* conforme aumenta a restrição do limiar de entropia.

7.1. Discussão

A compilação e a análise dos resultados apresentados permitem uma avaliação abrangente dos cenários testados ao longo deste estudo, endereçando os questionamentos abertos na seção 1.2, onde se enumeraram os objetivos deste trabalho.

- Aplicabilidade do modelo SI-RCNN a uma ação do mercado brasileiro:

Os resultados dos testes para validação do modelo SI-RCNN à ação do Banco do Brasil (BBAS3) são apresentados no Anexo A. Essa divisão nos testes foi necessária, uma vez que o modelo original apresenta uma saída binária, indicando apenas compra ou venda, enquanto a presente pesquisa requer uma saída ternária, para permitir a adequada comparação entre cenários com e sem entropia, de modelos individuais e combinados.

Para efeito desta análise, considerou-se um comparativo de acurácia (ACC) dos modelos, uma vez que o resultado apresentado por VARGAS *et al.* [16] fez uso dessa medida de avaliação. Ao todo, foram rodados 120 cenários em treinamento estático e os resultados de validação para cada cenário são apresentados na Tabela 27, enquanto os de teste são apresentado na Tabela 28. A Tabela 25 apresenta um comparativo das estatísticas apresentadas por estes resultados e o resultado apresentado pelo modelo SI-RCNN original (VARGAS *et al.* [16]).

Tabela 25 – Resultado do modelo SI-RCNN em VARGAS *et al.* [16] e valores de média, máxima, mínima e mediana dos cenários com a ação BBAS3.

Modelo	**Validação ACC (%)**	**Teste ACC (%)**
SI-RCNN [16]	60,45	56,84
Média	54,93	53,39
Máxima	61,46	56,59
Mínima	51,22	49,76
Mediana	55,12	53,66

Observa-se que os melhores resultados deste trabalho são similares aos apresentados no artigo de VARGAS *et al.* [16], indicando que o modelo se aplica a uma ação do

mercado brasileiro com notícias em português. No entanto, em termos de média e mediana, as pequenas diferenças nas acurácias podem ser explicadas por dois diferentes fatores:

- A qualidade das representações vetoriais de palavras, uma vez que o *word2vec* em inglês é treinado com um corpus de mais de 100 bilhões de palavras gerando mais de 3 milhões de vetores de palavras[2], enquanto em português usa apenas 1,3 bilhões gerando 935 mil vetores palavras [40];

- A diferente estrutura de linguagem entre o português e o inglês, que pode requerer pequenos ajustes de parâmetros nas redes convolucionais para captar melhor a semântica em português.

- Vantagens no uso de notícias:

Observando-se de maneira geral todos os resultados apresentados nas tabelas ímpares, da Tabela 13 à Tabela 23, é notório que os cenários que fizeram uso de notícias textuais obtiveram resultados superiores aos que fizeram uso apenas dos indicadores. Além dos resultados de AUC em geral serem superiores, a exigência de maior grau de certeza para operar, ou seja, uma entropia menor nas saídas da rede, rapidamente retira os modelos sem texto de operação, mostrando que mesmo com alguns resultados positivos, há uma incerteza muito grande nas previsões.

Apesar de BECKMANN [27] afirmar, em seu estudo, que o impacto das notícias no mercado de ações dura menos de 5 minutos, as notícias que são publicadas com os mercados fechados não têm meios de refletir nos preços das ações imediatamente. Os resultados apresentados na presente pesquisa têm a contribuir para o rol de trabalhos que indicam que as notícias podem auxiliar na previsão dos movimentos diários das

[2] https://code.google.com/archive/p/word2vec/

ações, da mesma forma que ZHAI, *et al.* [10], LUSS e D'ASPREMONT [11], DING *et al*. [12], AKITA *et al*. [13] e VARGAS, *et al*. [14] [16].

Além disso, estes resultados se distanciam da Hipótese dos Mercados Eficientes, pois, ainda que as previsões sejam pouco superiores ao resultado puramente aleatório, eles indicam que é possível realizar previsões assertivas com base em notícias, uma vez que as ações ainda não incorporaram essas informações aos preços.

- Diferenças no treinamento estático e janela deslizante:

Com relação ao uso da janela deslizante, esta pesquisa testou 5 variantes no tamanho da janela e cada uma apresentou particularidades nos resultados. Enquanto uma janela pequena de 250 dias não é capaz de treinar o modelo adequadamente, como pode ser visto pela elevada quantidade resultados de AUC $\leq$ 0,5 da Tabela 15, uma janela de 500 dias consegue apresentar resultados melhores (Tabela 17). Porém, este tamanho de janela ainda não traz certeza nas previsões, sendo incapaz de gerar resultados em alguns cenários de entropia mais restritiva.

O uso de janelas maiores obteve resultados mais satisfatórios, como, por exemplo, nas janelas de 750, 1000 e 1250 dias, Tabela 19, Tabela 21 e Tabela 23, respectivamente. Observa-se melhora nas previsões e constante evolução da AUC nos cenários *ensemble* em ambientes de entropia restritiva, com destaque para os resultados com janela de 750 dias. No entanto, é importante ressaltar que mais estudos devem ser realizados para se encontrar tamanho de janela e de passo ideais, que permitam um treinamento que maximize a AUC dos modelos.

Apesar dos resultados insatisfatórios com janela de 250 dias, todos os cenários com janela deslizante apresentaram resultados superiores ao treinamento estático, tanto individualmente quanto no *ensemble*, indicando uma vantagem para o uso de janela deslizante em problemas que envolvam séries temporais.

- Resultados da aplicação da entropia de Shannon:

Um dos principais pontos desta pesquisa, a aplicação da Entropia de Shannon aos resultados dos modelos trouxe uma sensível melhora nos resultados. Com o aumento da restrição no limiar de entropia, cenários que apresentam saídas de elevada incerteza chegavam ao ponto de não realizar nenhuma operação, tornando-se virtualmente descartáveis. Por outro lado, os cenários que apresentaram maior grau de certeza nos resultados, notadamente os treinados com janela deslizante de 750, 1000 e 1250 dias, deixam de apresentar AUCs abaixo de 0,5 e passam a apresentar resultados de AUC superiores a 0,5.

Os gráficos da Figura 25 e da Figura 26 mostram claramente a melhoria nos resultados, conforme se aumenta a restrição de entropia imposta aos cenários. Em outras palavras, a entropia de Shannon aplicada a modelos de previsão de séries temporais financeiras protege o investidor de duas formas: descartando respostas de baixa certeza, que elevam a taxa de falso positivo, e eliminando modelos de baixa confiabilidade.

Cabe ressaltar que a aplicação de um filtro baseado na entropia de Shannon é muito útil em um problema no qual o usuário pode se abster de tomar uma decisão, como neste caso que envolve decisão de investimento. No entanto, os resultados também demonstram que um filtro de entropia muito restritivo pode inutilizar um modelo.

Apesar deste trabalho ter apresentado os melhores resultados com filtros em torno de 0,8 e 0,7, para uma aplicação real mais testes devem ser realizados, com o objetivo de encontrar a linha de corte ideal, que maximize as taxas de acerto e minimize as taxas de erro, promovendo maiores retornos financeiros com o mínimo de risco possível – o que resulta em um outro problema de otimização com restrições.

- Vantagens no uso de *ensemble* de modelos:

Outro ponto de elevada relevância nesta pesquisa, o *ensemble* de modelos não apresenta um resultado melhor do que cada modelo individualmente, principalmente em cenários sem restrição de entropia, apesar de seu AUC ser superiores à média dos AUCs dos

cenários individuais. No entanto, como apresentado na Figura 26, essa técnica apresenta gradativo aumento na AUC quando combinado à entropia de Shannon, carregando consigo mais confiabilidade nos resultados.

Conforme esperado, a combinação de modelos mostra que é muito dependente dos resultados de cada um dos modelos que o compõem. Ou seja, ele é capaz de combinar as respostas de cada modelo e retornar um melhor resultado, mas somente se os cenários individuais apresentarem um mínimo de assertividade para transmitir ao *ensemble*.

- Objetivos intermediários:

Com relação aos objetivos intermediários, os resultados apresentados no Anexo A novamente dão o direcionamento. O uso de dois algoritmos de vetorização de palavras não evidenciou superioridade de nenhum dos dois. Tanto *word2vec skip-gram* quanto o *GloVe* apresentaram resultados similares, alternando-se entre os melhores resultados de acurácia.

No entanto, quando comparados os resultados obtidos com e sem o uso das *stopwords*, há um indicativo de que há perda de semântica ao se retirar todas estas palavras do contexto, uma vez que os resultados com *stopwords* apresentam melhores acurácias.

Analisando-se nos resultados por filtro de frequência de palavras, pode-se verificar melhores acurácias no uso das frequências mínimas de 2, 10, 15, 25 e 50 sobre os demais cenários. Apesar do destaque destes valores, não é possível afirmar que esses tamanhos de filtro de frequência influenciam sobremaneira nos resultados. Convém, porém, salientar que os cenários com filtro 50 apresenta um menor *dataset* de entrada no modelo e, consequentemente, consome menos recurso computacional.

Novamente é importante observar que, devido às limitações desta pesquisa, estes resultados não são exaustivos e maiores avaliações devem ser realizadas para encontrar o tamanho ideal de janelas de treinamento, otimizar a linha de corte para o filtro de entropia, combinar modelos com diferentes topologias e cenários com outros tipos de indicadores, buscando maior assertividade.

7.2. Simulações de Investimento

Após avaliar os resultados dos cenários, faz-se necessário executar simulações de investimentos com o intuito de verificar se estes são capazes de gerar retornos financeiros em ambientes tão restritivos e de elevada competitividade, como detalhado na seção 6.6.2. Este trabalho trata da aplicação de redes neurais profundas para auxílio à tomada de decisão, mantendo-se a discricionariedade do operador de executar ou não a operação. No entanto, nesta simulação, optou-se por acatar todos os indicativos de compra e venda apresentados pelos modelos.

Além disso, cabe aqui reafirmar que a assertividade de um modelo não é garantia de retorno financeiro pelos seguintes motivos principais:

- O direcional dos modelos está ajustado para prever alta ou queda na relação entre fechamento em *t+1* e fechamento em *t*, enquanto só é possível operar na abertura de *t+1* e fechamento de *t+1*, relação que não necessariamente possui o mesmo direcional;
- O modelo pode acertar muitos movimentos de baixa amplitude e errar os de grande amplitude, resultando em retorno negativo;
- Não foi implementada nenhuma estratégia de *stop loss*, que ajuda a limitar perdas;
- Há custos operacionais, tarifas e impostos que incidem nas operações de compra e de venda, independentemente de sua lucratividade, além da incidência do imposto de renda de 20% sobre os lucros.

Considerando-se o direcionamento dado por esta pesquisa, optou-se por avaliar dois modelos que apresentam as qualidades indicadas na seção 7.1:

- Tipo de Treinamento: *Sliding Window* com janela de 750 dias, melhor resposta de treinamento para os modelos individuais e também para o ensemble;
- Modelos ensemble, pois apresentam resultados superiores aos modelos individuais;

- Limiar de entropia em 0,8 e 0,7, exigindo bom nível de certeza nas respostas.

Os resultados da simulação de investimento podem ser visualizados na Tabela 26 e na Figura 27. Observa-se que os modelos considerados conseguiram superar os benchmarks do mercado, uma vez que o período de teste em questão apresenta um forte comportamento de queda no mercado de ações brasileiro. Ou seja, estratégias de comprar e segurar não são indicadas nesse período, e por isso seus resultados são bastante negativos.

Tabela 26 – Retorno dos investimentos e dos *benchmarks* no final do período de teste

Investimento	CDI	Ibovespa	Buy-and-hold	Naive	Ensemble (entr. < 0,8)	Ensemble (entr. < 0,7)
Retorno (%)	11,0%	-24,6%	-45,8%	-7,7%	38,7%	27,4%

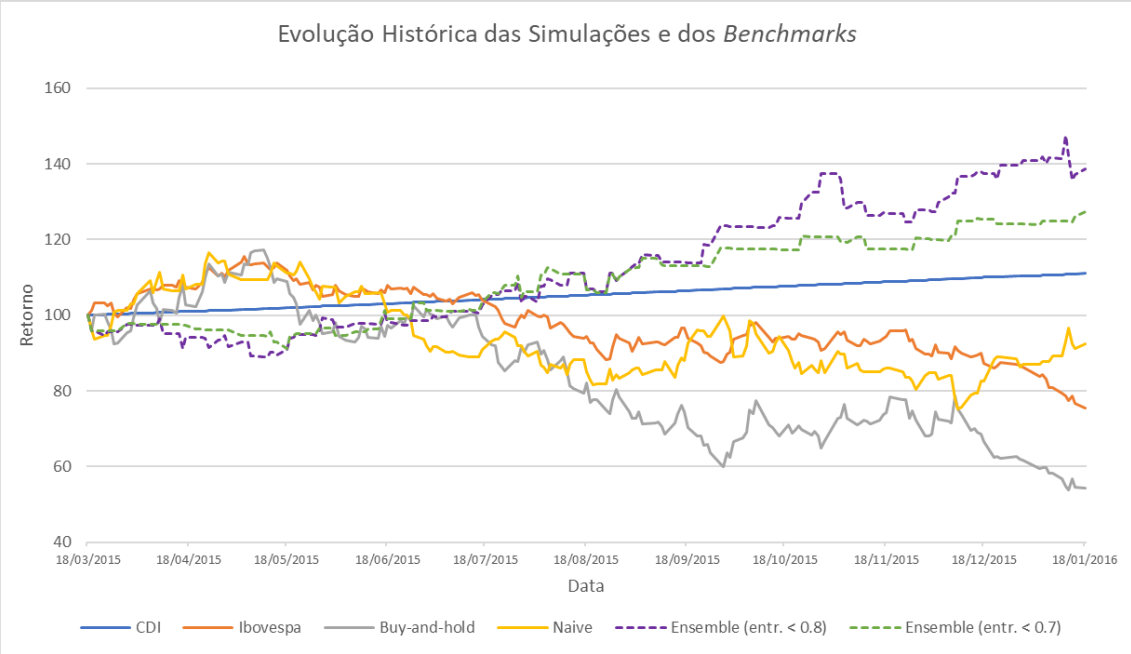

Figura 27 – Evolução do retorno histórico dos modelos *ensemble* e dos *benchmarks* no período de simulação do *dataset* de teste (19/03/2015 a 18/01/2016) na base 100.

8. Conclusões

Este trabalho teve por finalidade identificar e testar técnicas de *machine learning* capazes de compor um bom sistema de suporte à decisão de investimento em ações usando notícias e análise técnica. Inicialmente, usou-se a arquitetura de redes neurais convolucionais e recorrentes aplicada por VARGAS *et al.* [16] para validação no cenário brasileiro e, em seguida, aplicaram-se as modificações nas saídas e os testes nos demais cenários.

Primeiramente, é importante destacar que a citada arquitetura apresentou bons resultados para o cenário de uma ação brasileira usando notícias em língua portuguesa. Este foi o ponto de partida do estudo, pois, somente após isso, foi possível fazer a modificação das saídas do modelo de uma estrutura binária para uma saída ternária. Essa mudança mostrou-se mais adequada para efeitos comparativos entre os modelos *ensemble* e os cenários individuais, com e sem a aplicação da entropia de Shannon.

Após estes ajustes na saída do modelo e a definição dos cenários de teste, detalhados na seção 6.5, foi possível verificar a superioridade dos resultados com uso de notícias sobre os cenários que usam apenas indicadores, notadamente quando da aplicação do treinamento por janela deslizante, uma vez que carrega mais informações, melhorando sua capacidade de generalização.

Os diferentes tipos de treinamento aplicados mostraram que o uso da janela deslizante para problemas envolvendo séries temporais é mais adequado que o uso de um treinamento estático, como tradicionalmente é usado nos problemas de *machine learning* atemporais. Ainda que os melhores resultados tenham sido obtidos com janelas maiores, mesmo o cenário com a menor janela testada ainda apresentou maior assertividade nos resultados que o modelo estático.

É importante observar também que o trabalho não foi exaustivo na busca pela janela ideal de treinamento. No entanto, a Figura 26 mostra que os cenários com janela de 500, 750 e 1000 dias apresentaram os melhores resultados de *ensemble*, indicando que pode

haver uma janela de treinamento que se ajuste melhor ao modelo e resulte numa maior assertividade.

A aplicação de um filtro baseado na entropia de Shannon às saídas dos modelos trouxe uma maior certeza às indicações de compra e venda, ao eliminar predições de elevada entropia e manter apenas as melhores previsões. Ao se construir um modelo de auxílio à tomada de decisão no mercado financeiro, é fundamental a aplicação de tal técnica com o intuito de aumentar a confiabilidade dos resultados, mitigar os riscos e proteger o investidor. Entretanto, a presente pesquisa não foi capaz de indicar o limiar ideal para a entropia, mas mostrou que uma maior busca de certeza nas saídas da rede conduz a uma melhor assertividade em uma quantidade menor de operações.

Apesar de ser uma técnica amplamente usada em *machine learning*, a combinação de modelos mostrou vantagem com relação aos modelos individuais. No entanto, essa técnica depende muito de boas previsões dos modelos individuais, para que sua combinação traga resultados positivos, sendo um claro exemplo do clássico "*Garbage in, garbage out"*. Diante disso, seu uso associado à entropia de Shannon, que funciona como um controle de qualidade das saídas, mostra elevado potencial na melhora dos resultados, tanto de assertividade quanto de retorno financeiro.

Por fim, cabe ressaltar que os resultados apresentados neste trabalho indicam que algoritmos de redes neurais profundas processando notícias são capazes de realizar previsão dos movimentos diários das ações, auxiliando a tomada de decisão, e permitindo a obtenção de resultados superiores aos benchmarks, mesmo diante das restrições existentes num ambiente de simulação e sem aplicação de regras de gestão de risco. Esta conclusão se distancia da Hipótese dos Mercados Eficientes, uma vez que mostra que os preços ainda não incorporaram as informações publicadas com os mercados fechados, permitindo ao operador obter vantagens com o uso dessa característica.

8.1. Trabalhos Futuros

A busca por um modelo capaz de realizar previsões de elevada confiabilidade para o mercado financeiro requer extensa pesquisa que permeia os diversos campos da economia, teoria da informação e *machine learning*. Este último em especial se desenvolve mais a cada dia com o surgimento de novas técnicas e adaptação de outras já existentes.

Apesar do amplo esforço da pesquisa, a elevada complexidade do problema não permite a exploração exaustiva de todas as técnicas e parâmetros ajustáveis. Sendo assim, as propostas de trabalhos futuros se dividem em três principais vertentes:

- Financeira: alteração dos períodos usados para calcular os indicadores, usar os preços do ativo ou série de log-retornos e combinar outros indicadores. É interessante também implementar regras de gestão de risco, de modo a reduzir as perdas nas previsões erradas.
- Teoria da Informação: buscar a linha de corte ideal do filtro de entropia, que solicite elevado grau de certeza nas previsões, mas que não seja tão restritivo a ponto de não propor operações, resultando em elevada assertividade;
- *Machine Learning*: aprofundar as pesquisas em busca da janela ideal de treinamento por janela deslizante, aplicar outras arquiteturas de rede convolucional para melhor captura da semântica, como as *Dynamic CNN* [77], e implementação de técnicas de aprendizado por reforço, que têm contribuído fortemente em pesquisas recentes [78][79].

Referências Bibliográficas

[1]. ROSS, S., WESTERFIELD, R., JAFFE, J.; *Corporate finance*, Editora McGraw-Hill/Irwin, 2005.

[2]. ELDER, A.; *Come into my trading room*, John Wiley & Sons, 1ª Edition, 2005.

[3]. NOFSINGER, J.; The impact of public information on investors, *Journal of Banking & Finance 25*, 1339-1366, 2001.

[4]. DAMODARAN, A.*; Investment Philosophies: Successful Strategies and the Investors Who Made Them WorkInvestment Philosophies*, Wiley Finance, 2nd Edition, 2012.

[5]. FAMA, E.; The Behavior of stock prices, *The Journal of Business*, Vol. 38, No. 1, pp. 34-105, Jan. 1965.

[6]. FAMA, E.; Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, Vol. 25, No. 2, 1970.

[7]. MALKIEL, B.; *A Random Walk Down Wall Street: The Time-tested Strategy for Successful Investing.* W.W. Norton, New York, 1973.

[8]. MIZUNO, H., KOSAKA, M., YAJIMA, H., KOMODA, N.; Application of neural network to technical analysis of stock market prediction, *Studies in Informatic and control*, vol. 7, no. 3, pp. 111-120, 1998

[9]. LEIGH, W., PURVIS, R., RAGUSA, J.; Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: A Case Studying Decision Support, *Decision Support Systems*, no. 32, pp. 361–377, 2002.

[10]. ZHAI, Y., HSU, A., HALGAMUGE, S.; Combining news and technical indicators in Daily stock price trends prediction, *Advances in Neural Networks – ISNN 2007*, 1087-1096, 2007.

[11]. LUSS, R., D'ASPREMONT, A.; *Predicting Abnormal Returns from News Using Text Classification*, arXiv:0809.2792, 2009.

[12]. DING, X., ZHANG, Y., LIU, T., DUAN, J.*;* Deep Learning for Event-Driven Stock Prediction. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2015.

[13]. AKITA, R., YOSHIHARA, A., MATSUBARA, T., UEHARA, K.; Deep learning for stock prediction using numerical and textual information, *15th International Conference on Computer and Information Science (ICIS)*, 2016.

[14]. VARGAS, M. R., DE LIMA, B. S., EVSUKOFF, A. G.; Deep learning for stock market prediction from financial news articles. *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 60-65, 2017.

[15]. FISCHER, T., KRAUSS, C.*;* Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research Volume 270*, Issue 2, Pages 654-669, 2018.

[16]. VARGAS, M. R., DOS ANJOS, C. E. M., BICHARA, G. L. G., EVSUKOFF, A. G.; Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles, 2018 *International Joint Conference on Neural Networks* (IJCNN), IEEE, 2018.

[17]. KARPATHY, A.; *The Unreasonable Effectiveness of Recurrent Neural Networks*. Disponível em http://karpathy.github.io/2015/05/21/rnn-effectiveness/. Consultado em agosto/2018.

[18]. HOCHREITER, S., SCHMIDHUBER, J.; Long short-term memory. *Neural Computation.* Vol. 9, N° 8, p.1735–1780, 1997.

[19]. LECUN, Y., BENGIO, Y.; Convolutional networks for images, speech, and time-series. *The handbook of brain theory and neural networks*, Pages 255-258 MIT Press Cambridge, MA, USA, 1995.

[20]. COLLOBERT, R., WESTON, J.; A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[21]. COLLOBERT, R., WESTON, J., BOTTOU, L., *et al*.; Natural Language Processing (Almost) from Scratch, *Journal of Machine Learning Research 12*, 2493-2537, 2011.

[22]. KIM, Y.; Convolutional neural networks for sentence classification, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[23]. GIDOFALVI, G., ELKAN, C.; *Using news articles to predict stock price movements*. Department of Computer Science and Engineering, University of California, San Diego, 2001

[24]. MALAGRINO, L., ROMAN, N., MONTEIRO, A.; Forecasting stock market index daily direction A Bayesian Network approach, *Expert Systems with Applications 105*, 11-22, 2018

[25]. SCHUMAKER, R., CHEN, H.; A quantitative stock prediction system based on financial news, *Information Processing & Management Volume 45*, Issue 5, Pages 571-583, 2009.

[26]. WANG, B., HUANG, H., WANG, X.; A novel text mining approach to financial time series forecasting, *Neurocomputing, vol. 83*, pp. 136- 145, 2012

[27]. BECKMANN, M.; *Stock Price Change Prediction Using News Text Mining*, Tese de Doutorado, COPPE/UFRJ, 2017

[28]. KIA, A., HARATIZADEH, S., SHOURAKI, S.; A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices, *Expert Systems with Applications 105*, 159-173, 2018.

[29]. TAKEUCHI, L., LEE, Y., *Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks*, Stanford University, 2013

[30]. RELAÇÃO COM INVESTIDORES B3. Disponível em http://ri.bmfbovespa.com.br/, consultado em fevereiro/2019;

[31]. SISTEMA FINANCEIRO NACIONAL, *Banco Central do Brasil*, Disponível em https://www.bcb.gov.br/estabilidadefinanceira/sfn, consultado em fevereiro/2019;

[32]. ASSAF NETO, A.; *Mercado Financeiro*. 5a. ed., São Paulo: Editora Atlas, 2003.

[33]. ESTRUTURA NORMATIVA, *Brasil, Bolsa, Balcão (B3)*, disponível em http://www.b3.com.br/pt_br/regulacao/estrutura-normativa/regulamentos-e-manuais/negociacao.htm, consultado em 19 de fevereiro/2019.

[34]. KIRKPATRICK, C. D., DAHLQUIST, J.R.; *Technical Analysis: The Complete Resource for Financial Market Technicians*, FT Press, New Jersey, 2011.

[35]. FELDMAN, R., SANGER, J.; *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007

[36]. BENEVENUTO, F., RIBEIRO, F., ARAÚJO, M.; Métodos para Análise de Sentimentos em mídias sociais, *Minicurso em Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, 2015.

[37]. WEISS, S.M., INDURKHYA, N., ZHANG, T., DAMERAU, F.; *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, 2005

[38]. BENGIO, Y., DUCHARME, R., VINCENT, P., JAUVIN, C.; A Neural Probabilistic Language Model, *Journal of Machine Learning*, Research 3. 1137–1155, 2003.

[39]. DICIONARIOEGRAMATICA.COM, Quantas palavras têm os dicionários? - https://dicionarioegramatica.com.br/publicacoes-fixas/quantas-palavras-tem-os-dicionarios/, consultado em 10/03/2019.

[40]. HARTMANN, N., FONSECA, E., SHULBY, C., *et al*.; Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. *Symposium in Information and Human Language Technology*, 2017.

[41]. MIKOLOV, T., YIH, W., ZWEIG, G.; Linguistic Regularities in Continuous Space Word Representations, *Proceedings of NAACL-HLT*, pages 746–751, Atlanta, 2013.

[42]. MAATEN, L., HINTON, G.; Visualizing Data using t-SNE, *Journal of Machine Learning Reserch 9, 2579-2605, 2008*

[43]. MIKOLOV, T., CHEN, K., CORRADO, G., DEAN, J.; *Efficient Estimation of Word Representations in Vector Space*, arXiv:1301.3781v3, 2013.

[44]. PENNINGTON, J., SOCHER, R., MANNING, C.; GloVe: Global Vectors forWord Representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[45]. HAYKIN, S.; *Redes Neurais: Princípios e Prática*, 2ª ed. Porto Alegre, Bookman, 2001.

[46]. MCCULLOCH, W.S., PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943

[47]. HEBB, D.O.; *The Organization of Behavior*, McGill University, John Wiley & sons, New York, 1949.

[48]. ROSENBLATT, F.; The perceptron: a perceiving and recognizing automaton, *Project PARA*. Cornell Aeronautical Laboratory, 1957.

[49]. SILVA, I., SPATTI, D., FLAUZINO, R.; R*edes Neurais Artificiais Para Engenharia E Ciências Aplicadas. Curso Prático*, Editora Artliber, 2010.

[50]. WIDROW, B., HOFF, M.; Adaptative switching circuits, 1960 IRE *WESCON Convention Record*, pp. 96-104, 1960.

[51]. MINSKY, M., PAPERT, S.; *Perceptrons: An Introduction to Computational Geometry*, MIT Press, 1969.

[52]. RUMELHART, D., MCCLELLAND, J., HINTON, G.; *Parallel Distributed Processing*, MIT Press, 1986.

[53]. FARIA, E.; *Redes Neurais Convolucionais e Máquinas de Aprendizado Extremo Aplicadas ao Mercado Financeiro Brasileiro*. Tese de Doutorado. COPPE/UFRJ, 2018

[54]. HINTON, G. E., OSINDERO, S., THE, Y. W, A fast learning algorithm for deep belief nets, *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[55]. HINTON, G. E., SALAKHUTDINOV, R.R.; Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol. 313, 504–507, 2006.

[56]. BENGIO, Y., LECUN, Y.; Scaling Learning Algorithms towards AI, *Large-Scale Kernel Machines*, MIT Press, 2007.

[57]. CALÔBA, L. P., Redes Neurais em Modelagem de Sistemas in Aguirre, L.A. (Org.). *Enciclopédia de Automática*, 1ª ed., São Paulo, Edgar Blucher, v. 3, p. 325-344, 2007.

[58]. BENGIO, Y.; SIMARD, P.; FRASCONI, P.; Learning long-term dependencies with gradient descent is difficult, *Neural Networks IEEE Transactions* on, vol. 5, no. 2, pp. 157-166, 1994.

[59]. OLAH, C.; *Understanding LSTM Networks*, http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2005. Consultado em agosto/2017.

[60]. MILLION, E.; *The Hadamard Product*, Introduction and Basic Results, 2007.

[61]. MIT, 6.S191: *Introduction to Deep Learning*, (2017) *A 1-week extensive survey of deep learning methods and applications*. http://introtodeeplearning.com/Sequence%20Modeling.pdf. Consultado em agosto/2017

[62]. GOODFELLOW, I., BENGIO, Y., COURVILLE, A.; *Deep Learning*, MIT Press, http://www.deeplearningbook.org, 2016

[63]. LECUN, Y., BENGIO, Y.; Convolutional Networks for Images, Speech, and Time-Series, Arbib, M. A. (Eds), *The Handbook of Brain Theory and Neural Networks*, MIT Press, 1995.

[64]. SHANNON, C.; A Mathematical Theory of Communication, *The Bell System Technical Journal*, Vol. XXVII, Nº 3, 1948.

[65]. VITERBI, A., OMURA, J.; *Principles of Digital Communication and Coding*, 2nd edition, McGraw-Hill, 1979.

[66]. COVER, T., THOMAS, J; *Elements of Information Theory*, 2nd edition, John Wiley & Sons, 1991.

[67]. ZHOU, Z.; *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012;

[68]. BISHOP, C.; *Pattern Recognition and Machine Learning*, Springer, 2006.

[69]. FIGUEIREDO, M.; *Método para Representação de Conceitos por Meio de Técnicas de Análise de Textos em Sequencia Temporal*, Tese de Doutorado, COPPE/UFRJ, 2017.

[70]. TSAY, R.S.; *Analysis of Financial Time Series*, 3rd Edition, John Wiley, 2010

[71]. SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, *et al*.; Dropout: A Simple Way to Prevent Neural Networks from Overtting, *Journal of Machine Learning Research* 15, *2014.*

[72]. KINGMA, D., BA, J.L.; Adam: A Method for Stochastic Optimization, *International Conference on Learning Representations*, 2015.

[73]. PRECHELT, L.; Early Stopping – but when?, *Neural Networks: Tricks of the Trade*, Pages 55-69, 1998.

[74]. DIETTERICH, T.; Machine Learning for Sequential Data: A Review. *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 15-30, 2002.

[75]. EVSUKOFF, A.; Ensinando Máquinas, UFRJ, Rio de Janeiro, 2015;

[76]. FAWCETT, T.; An Introduction to ROC analysis. *Pattern Recognition Letters 27*, 861-874, 2006

[77]. KALCHBRENNER, N., GREFENSTETTE, E., BLUNSOM, P.; *A Convolutional Neural Network for Modelling Sentences*, arXiv:1404.2188, 2014.

[78]. MNIH, V., KAVUKCUOGLU, K., SILVER, *et al*.; Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[79]. MNIH, V., KAVUKCUOGLU, K., SILVER, D., *et al*.; *Playing Atari with Deep Reinforcement Learning*. arXiv:1312.5602, 2013

[80]. RONG, X.; *word2vec Parameter Learning Explained*. arXiv:1411.2738v4, 2016.

[81]. KURITA, K.; Paper Dissected: "Glove: Global Vectors for Word Representation" Explained. *Machine Learning Explained.* http://mlexplained.com/2018/04/29/paper-dissected-glove-global-vectors-for-word-representation-explained/, 2018. Consultado em 25/02/2019.

Anexo A – Validação do Modelo SI-RCNN

Para a validação do modelo SI-RCNN, usado por VARGAS *et al*. [16], em uma ação do mercado nacional, foi criada uma árvore de cenários que engloba uma série de parâmetros de pré-processamento que são relevantes aos modelos de Redes Neurais Profundas. Estes parâmetros são o tipo de treinamento usado, qual *dataset*, qual método de vetorização de palavras, usar ou não stopwords e qual filtro de frequência mínima de aparição no texto usar e podem ser visualizados na Figura 28.

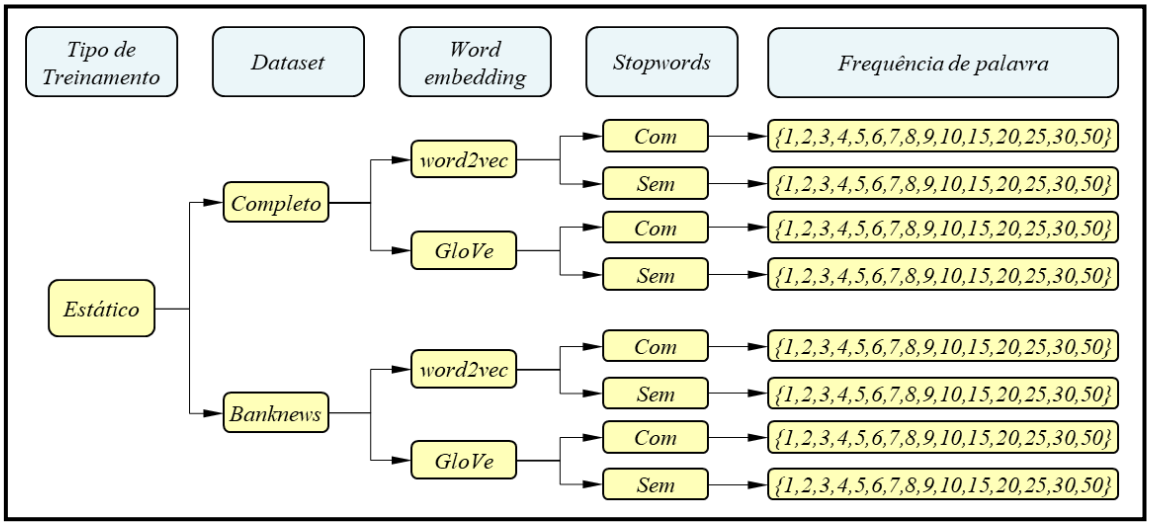

Figura 28 – Árvore de cenários com os 120 cenários analisados na primeira rodada de avaliação.

O modelo possui uma saída binária ([0,1] pra venda e [1,0] para compra) e sua arquitetura é a mesma apresentada na seção 6.3. Os resultados de AUC para os *datasets* de Validação são apresentados na Tabela 27, juntamente com os valores de Verdadeiro Positivo, Verdadeiro Negativo, Falso Positivo, Falso Negativo e acurácia. Na Tabela 28 são apresentados os resultados de AUC para os *datasets* de Teste.

Os resultados apresentados indicam que há uma ligeira superioridade nos cenários com stopwords, da mesma forma que os filtros com mínimo de palavras em 2, 10, 15, 25 e 50. Não há evidente superioridade no comparativo *dataset* completo versus *banknews* e word2vec versus GloVe.

Tabela 27 – Resultado de Validação dos 120 cenários.

Posição	Cenario	VP	VN	FP	FN	ACC	AUC
1	banknews_csw_k15_glove_300	57	69	31	48	0,615	0,616
2	banknews_csw_k50_w2v-skip_300	52	71	29	53	0,600	0,603
3	banknews_csw_k25_glove_300	61	61	39	44	0,595	0,595
4	complete_csw_k2_w2v-skip_300	54	67	33	51	0,590	0,592
5	complete_csw_k15_w2v-skip_300	47	73	27	58	0,585	0,589
6	complete_csw_k25_glove_300	50	70	30	55	0,585	0,588
7	banknews_csw_k2_w2v-skip_300	64	56	44	41	0,585	0,585
8	banknews_csw_k10_glove_300	80	40	60	25	0,585	0,581
9	banknews_ssw_k6_glove_300	79	40	60	26	0,580	0,576
10	complete_csw_k25_w2v-skip_300	42	76	24	63	0,576	0,580
11	banknews_csw_k30_glove_300	49	69	31	56	0,576	0,578
12	banknews_ssw_k2_w2v-skip_300	54	64	36	51	0,576	0,577
13	complete_csw_k10_glove_300	57	61	39	48	0,576	0,576
14	banknews_csw_k20_glove_300	84	34	66	21	0,576	0,570
15	complete_csw_k2_glove_300	53	64	36	52	0,571	0,572
16	banknews_ssw_k15_w2v-skip_300	69	48	52	36	0,571	0,569
17	banknews_csw_k5_w2v-skip_300	73	44	56	32	0,571	0,568
18	banknews_csw_k25_w2v-skip_300	73	44	56	32	0,571	0,568
19	banknews_ssw_k15_glove_300	80	37	63	25	0,571	0,566
20	banknews_csw_k1_glove_300	42	74	26	63	0,566	0,570
21	complete_ssw_k10_glove_300	45	71	29	60	0,566	0,569
22	banknews_csw_k15_w2v-skip_300	59	57	43	46	0,566	0,566
23	banknews_ssw_k7_glove_300	71	45	55	34	0,566	0,563
24	banknews_ssw_k10_w2v-skip_300	82	34	66	23	0,566	0,560
25	complete_ssw_k25_glove_300	88	28	72	17	0,566	0,559
26	complete_ssw_k6_w2v-skip_300	90	26	74	15	0,566	0,559
27	complete_csw_k7_w2v-skip_300	46	69	31	59	0,561	0,564
28	banknews_csw_k6_glove_300	47	68	32	58	0,561	0,564
29	complete_ssw_k50_w2v-skip_300	48	67	33	57	0,561	0,564
30	banknews_ssw_k8_glove_300	56	59	41	49	0,561	0,562
31	banknews_ssw_k10_glove_300	68	47	53	37	0,561	0,559
32	complete_ssw_k3_glove_300	70	45	55	35	0,561	0,558
33	complete_csw_k30_w2v-skip_300	72	43	57	33	0,561	0,558
34	banknews_ssw_k1_w2v-skip_300	75	40	60	30	0,561	0,557
35	banknews_ssw_k50_glove_300	87	28	72	18	0,561	0,554
36	banknews_ssw_k4_w2v-skip_300	92	23	77	13	0,561	0,553
37	banknews_csw_k3_w2v-skip_300	38	76	24	67	0,556	0,561
38	complete_ssw_k5_glove_300	46	68	32	59	0,556	0,559
39	banknews_ssw_k25_glove_300	48	66	34	57	0,556	0,559
40	complete_ssw_k10_w2v-skip_300	52	62	38	53	0,556	0,558
41	banknews_csw_k4_w2v-skip_300	63	51	49	42	0,556	0,555
42	banknews_csw_k7_w2v-skip_300	66	48	52	39	0,556	0,554
43	banknews_ssw_k30_glove_300	70	44	56	35	0,556	0,553
44	complete_ssw_k5_w2v-skip_300	73	41	59	32	0,556	0,553
45	banknews_ssw_k50_w2v-skip_300	73	41	59	32	0,556	0,553
46	banknews_csw_k8_glove_300	75	39	61	30	0,556	0,552
47	banknews_csw_k10_w2v-skip_300	83	31	69	22	0,556	0,550
48	complete_csw_k8_w2v-skip_300	84	30	70	21	0,556	0,550
49	banknews_csw_k4_glove_300	86	28	72	19	0,556	0,550
50	complete_ssw_k4_w2v-skip_300	90	24	76	15	0,556	0,549
51	banknews_csw_k9_w2v-skip_300	90	24	76	15	0,556	0,549
52	banknews_ssw_k8_w2v-skip_300	90	24	76	15	0,556	0,549
53	banknews_ssw_k3_glove_300	36	77	23	69	0,551	0,556
54	banknews_csw_k6_w2v-skip_300	39	74	26	66	0,551	0,556
55	banknews_csw_k50_glove_300	48	65	35	57	0,551	0,554
56	complete_ssw_k9_glove_300	53	60	40	52	0,551	0,552
57	complete_csw_k9_glove_300	57	56	44	48	0,551	0,551
58	banknews_ssw_k30_w2v-skip_300	62	51	49	43	0,551	0,550
59	complete_csw_k50_w2v-skip_300	63	50	50	42	0,551	0,550
60	banknews_csw_k8_w2v-skip_300	67	46	54	38	0,551	0,549

Posição	Cenario	VP	VN	FP	FN	ACC	AUC
61	banknews_csw_k30_w2v-skip_300	72	41	59	33	0,551	0,548
62	complete_ssw_k25_w2v-skip_300	73	40	60	32	0,551	0,548
63	banknews_ssw_k9_glove_300	73	40	60	32	0,551	0,548
64	banknews_ssw_k7_w2v-skip_300	83	30	70	22	0,551	0,545
65	banknews_csw_k20_w2v-skip_300	87	26	74	18	0,551	0,544
66	complete_ssw_k2_w2v-skip_300	31	81	19	74	0,546	0,553
67	complete_ssw_k3_w2v-skip_300	32	80	20	73	0,546	0,552
68	complete_ssw_k15_w2v-skip_300	51	61	39	54	0,546	0,548
69	complete_ssw_k50_glove_300	59	53	47	46	0,546	0,546
70	banknews_ssw_k4_glove_300	75	37	63	30	0,546	0,542
71	complete_ssw_k20_w2v-skip_300	79	33	67	26	0,546	0,541
72	complete_csw_k10_w2v-skip_300	81	31	69	24	0,546	0,541
73	complete_ssw_k1_w2v-skip_300	82	30	70	23	0,546	0,540
74	banknews_ssw_k6_w2v-skip_300	82	30	70	23	0,546	0,540
75	banknews_ssw_k25_w2v-skip_300	93	19	81	12	0,546	0,538
76	banknews_ssw_k3_w2v-skip_300	34	77	23	71	0,541	0,547
77	banknews_ssw_k1_glove_300	35	76	24	70	0,541	0,547
78	banknews_ssw_k9_w2v-skip_300	62	49	51	43	0,541	0,540
79	banknews_csw_k9_glove_300	67	44	56	38	0,541	0,539
80	complete_csw_k9_w2v-skip_300	72	39	61	33	0,541	0,538
81	complete_csw_k15_glove_300	75	36	64	30	0,541	0,537
82	complete_csw_k50_glove_300	76	35	65	29	0,541	0,537
83	complete_csw_k3_w2v-skip_300	82	29	71	23	0,541	0,535
84	complete_csw_k30_glove_300	104	7	93	1	0,541	0,530
85	banknews_csw_k2_glove_300	38	72	28	67	0,537	0,541
86	banknews_csw_k5_glove_300	42	68	32	63	0,537	0,540
87	complete_ssw_k30_w2v-skip_300	55	55	45	50	0,537	0,537
88	complete_ssw_k2_glove_300	59	51	49	46	0,537	0,536
89	banknews_ssw_k20_w2v-skip_300	74	36	64	31	0,537	0,532
90	complete_ssw_k8_glove_300	95	15	85	10	0,537	0,527
91	banknews_csw_k3_glove_300	98	12	88	7	0,537	0,527
92	complete_ssw_k15_glove_300	11	98	2	94	0,532	0,542
93	banknews_ssw_k5_glove_300	90	19	81	15	0,532	0,524
94	complete_csw_k4_w2v-skip_300	95	14	86	10	0,532	0,522
95	complete_ssw_k7_w2v-skip_300	69	39	61	36	0,527	0,524
96	banknews_ssw_k2_glove_300	74	34	66	31	0,527	0,522
97	complete_ssw_k8_w2v-skip_300	76	32	68	29	0,527	0,522
98	complete_csw_k6_glove_300	86	22	78	19	0,527	0,520
99	complete_csw_k5_w2v-skip_300	87	21	79	18	0,527	0,519
100	complete_ssw_k6_glove_300	100	8	92	5	0,527	0,516
101	complete_ssw_k30_glove_300	100	8	92	5	0,527	0,516
102	banknews_ssw_k20_glove_300	102	6	94	3	0,527	0,516
103	complete_csw_k6_w2v-skip_300	105	3	97	0	0,527	0,515
104	complete_csw_k7_glove_300	105	3	97	0	0,527	0,515
105	complete_ssw_k20_glove_300	20	87	13	85	0,522	0,530
106	banknews_csw_k1_w2v-skip_300	73	34	66	32	0,522	0,518
107	complete_ssw_k1_glove_300	73	34	66	32	0,522	0,518
108	complete_csw_k20_w2v-skip_300	99	8	92	6	0,522	0,511
109	complete_csw_k8_glove_300	101	6	94	4	0,522	0,511
110	complete_ssw_k7_glove_300	96	10	90	9	0,517	0,507
111	complete_csw_k5_glove_300	100	6	94	5	0,517	0,506
112	banknews_csw_k7_glove_300	101	5	95	4	0,517	0,506
113	complete_csw_k20_glove_300	105	1	99	0	0,517	0,505
114	complete_csw_k3_glove_300	105	1	99	0	0,517	0,505
115	complete_csw_k1_w2v-skip_300	105	0	100	0	0,512	0,500
116	banknews_ssw_k5_w2v-skip_300	105	0	100	0	0,512	0,500
117	complete_csw_k1_glove_300	105	0	100	0	0,512	0,500
118	complete_csw_k4_glove_300	105	0	100	0	0,512	0,500
119	complete_ssw_k9_w2v-skip_300	105	0	100	0	0,512	0,500
120	complete_ssw_k4_glove_300	105	0	100	0	0,512	0,500

Tabela 28 – Resultado de Teste dos 120 melhores cenários

Posição	Cenario	VP	VN	FP	FN	ACC	AUC
1	complete_csw_k2_w2v-skip_300	30	86	29	60	0,566	0,541
2	complete_csw_k10_glove_300	30	85	30	60	0,561	0,536
3	complete_csw_k7_w2v-skip_300	9	105	10	81	0,556	0,507
4	banknews_csw_k2_glove_300	33	80	35	57	0,551	0,531
5	complete_csw_k30_w2v-skip_300	32	81	34	58	0,551	0,530
6	complete_csw_k25_w2v-skip_300	29	84	31	61	0,551	0,526
7	complete_csw_k15_glove_300	26	87	28	64	0,551	0,523
8	banknews_csw_k25_glove_300	26	87	28	64	0,551	0,523
9	banknews_csw_k2_w2v-skip_300	26	87	28	64	0,551	0,523
10	banknews_ssw_k15_w2v-skip_300	22	91	24	68	0,551	0,518
11	complete_csw_k10_w2v-skip_300	21	92	23	69	0,551	0,517
12	banknews_csw_k15_glove_300	36	76	39	54	0,546	0,530
13	banknews_csw_k25_w2v-skip_300	33	79	36	57	0,546	0,527
14	banknews_csw_k20_w2v-skip_300	33	79	36	57	0,546	0,527
15	banknews_csw_k15_w2v-skip_300	30	82	33	60	0,546	0,523
16	complete_csw_k50_w2v-skip_300	27	85	30	63	0,546	0,520
17	complete_csw_k50_glove_300	26	86	29	64	0,546	0,518
18	banknews_ssw_k50_w2v-skip_300	22	90	25	68	0,546	0,514
19	complete_csw_k9_w2v-skip_300	21	91	24	69	0,546	0,512
20	banknews_ssw_k30_glove_300	19	93	22	71	0,546	0,510
21	banknews_ssw_k20_glove_300	6	106	9	84	0,546	0,494
22	banknews_csw_k50_w2v-skip_300	38	73	42	52	0,541	0,529
23	complete_csw_k25_glove_300	38	73	42	52	0,541	0,529
24	complete_csw_k2_glove_300	38	73	42	52	0,541	0,529
25	complete_csw_k15_w2v-skip_300	34	77	38	56	0,541	0,524
26	complete_csw_k4_glove_300	33	78	37	57	0,541	0,522
27	complete_csw_k30_glove_300	33	78	37	57	0,541	0,522
28	banknews_ssw_k10_glove_300	33	78	37	57	0,541	0,522
29	banknews_ssw_k8_w2v-skip_300	33	78	37	57	0,541	0,522
30	banknews_ssw_k20_w2v-skip_300	31	80	35	59	0,541	0,520
31	complete_csw_k5_glove_300	30	81	34	60	0,541	0,519
32	complete_csw_k4_w2v-skip_300	30	81	34	60	0,541	0,519
33	complete_ssw_k20_w2v-skip_300	30	81	34	60	0,541	0,519
34	banknews_ssw_k15_glove_300	30	81	34	60	0,541	0,519
35	complete_ssw_k6_w2v-skip_300	29	82	33	61	0,541	0,518
36	banknews_ssw_k3_w2v-skip_300	29	82	33	61	0,541	0,518
37	complete_ssw_k9_w2v-skip_300	28	83	32	62	0,541	0,516
38	banknews_csw_k6_glove_300	28	83	32	62	0,541	0,516
39	banknews_csw_k10_glove_300	28	83	32	62	0,541	0,516
40	banknews_csw_k7_glove_300	27	84	31	63	0,541	0,515
41	banknews_ssw_k2_w2v-skip_300	27	84	31	63	0,541	0,515
42	complete_ssw_k7_w2v-skip_300	25	86	29	65	0,541	0,513
43	banknews_ssw_k5_w2v-skip_300	25	86	29	65	0,541	0,513
44	complete_ssw_k4_w2v-skip_300	24	87	28	66	0,541	0,512
45	complete_ssw_k5_w2v-skip_300	24	87	28	66	0,541	0,512
46	complete_ssw_k10_w2v-skip_300	24	87	28	66	0,541	0,512
47	complete_ssw_k2_w2v-skip_300	22	89	26	68	0,541	0,509
48	banknews_csw_k20_glove_300	21	90	25	69	0,541	0,508
49	banknews_ssw_k30_w2v-skip_300	21	90	25	69	0,541	0,508
50	banknews_ssw_k9_glove_300	18	93	22	72	0,541	0,504
51	banknews_csw_k50_glove_300	34	76	39	56	0,537	0,519
52	banknews_ssw_k7_glove_300	33	77	38	57	0,537	0,518
53	complete_ssw_k1_glove_300	30	80	35	60	0,537	0,514
54	complete_ssw_k6_glove_300	30	80	35	60	0,537	0,514
55	banknews_ssw_k8_glove_300	30	80	35	60	0,537	0,514
56	banknews_ssw_k6_w2v-skip_300	30	80	35	60	0,537	0,514
57	banknews_csw_k9_w2v-skip_300	29	81	34	61	0,537	0,513
58	banknews_csw_k6_w2v-skip_300	28	82	33	62	0,537	0,512
59	complete_ssw_k25_w2v-skip_300	26	84	31	64	0,537	0,510
60	banknews_ssw_k1_w2v-skip_300	26	84	31	64	0,537	0,510

Posição	Cenario	VP	VN	FP	FN	ACC	AUC
61	banknews_csw_k4_glove_300	25	85	30	65	0,537	0,508
62	complete_ssw_k8_w2v-skip_300	23	87	28	67	0,537	0,506
63	banknews_ssw_k3_glove_300	23	87	28	67	0,537	0,506
64	complete_ssw_k8_glove_300	20	90	25	70	0,537	0,502
65	banknews_ssw_k50_glove_300	20	90	25	70	0,537	0,502
66	complete_ssw_k30_w2v-skip_300	6	104	11	84	0,537	0,486
67	complete_csw_k3_glove_300	36	73	42	54	0,532	0,517
68	complete_csw_k7_glove_300	34	75	40	56	0,532	0,515
69	complete_ssw_k10_glove_300	34	75	40	56	0,532	0,515
70	banknews_csw_k5_glove_300	29	80	35	61	0,532	0,509
71	complete_ssw_k7_glove_300	28	81	34	62	0,532	0,508
72	complete_ssw_k50_glove_300	23	86	29	67	0,532	0,502
73	banknews_ssw_k9_w2v-skip_300	23	86	29	67	0,532	0,502
74	complete_ssw_k1_w2v-skip_300	22	87	28	68	0,532	0,500
75	complete_ssw_k3_w2v-skip_300	22	87	28	68	0,532	0,500
76	banknews_ssw_k25_glove_300	22	87	28	68	0,532	0,500
77	complete_csw_k1_glove_300	22	87	28	68	0,532	0,500
78	complete_ssw_k30_glove_300	17	92	23	73	0,532	0,494
79	banknews_csw_k7_w2v-skip_300	14	95	20	76	0,532	0,491
80	complete_ssw_k9_glove_300	12	97	18	78	0,532	0,488
81	banknews_ssw_k6_glove_300	12	97	18	78	0,532	0,488
82	complete_csw_k9_glove_300	38	70	45	52	0,527	0,515
83	complete_ssw_k15_glove_300	34	74	41	56	0,527	0,511
84	complete_csw_k6_w2v-skip_300	34	74	41	56	0,527	0,511
85	banknews_ssw_k4_w2v-skip_300	34	74	41	56	0,527	0,511
86	complete_ssw_k3_glove_300	30	78	37	60	0,527	0,506
87	banknews_csw_k1_glove_300	30	78	37	60	0,527	0,506
88	banknews_ssw_k10_w2v-skip_300	27	81	34	63	0,527	0,502
89	banknews_ssw_k7_w2v-skip_300	25	83	32	65	0,527	0,500
90	complete_csw_k5_w2v-skip_300	24	84	31	66	0,527	0,499
91	complete_csw_k8_glove_300	22	86	29	68	0,527	0,496
92	complete_ssw_k50_w2v-skip_300	20	88	27	70	0,527	0,494
93	complete_ssw_k25_glove_300	14	94	21	76	0,527	0,486
94	complete_csw_k6_glove_300	38	69	46	52	0,522	0,511
95	complete_csw_k20_glove_300	33	74	41	57	0,522	0,505
96	complete_ssw_k5_glove_300	26	81	34	64	0,522	0,497
97	banknews_csw_k3_glove_300	26	81	34	64	0,522	0,497
98	banknews_csw_k8_w2v-skip_300	26	81	34	64	0,522	0,497
99	banknews_csw_k30_w2v-skip_300	26	81	34	64	0,522	0,497
100	banknews_ssw_k2_glove_300	25	82	33	65	0,522	0,495
101	banknews_ssw_k4_glove_300	25	82	33	65	0,522	0,495
102	complete_ssw_k20_glove_300	24	83	32	66	0,522	0,494
103	complete_ssw_k2_glove_300	18	89	26	72	0,522	0,487
104	complete_ssw_k4_glove_300	38	68	47	52	0,517	0,507
105	complete_csw_k1_w2v-skip_300	38	68	47	52	0,517	0,507
106	banknews_ssw_k25_w2v-skip_300	38	68	47	52	0,517	0,507
107	banknews_ssw_k1_glove_300	30	76	39	60	0,517	0,497
108	banknews_csw_k30_glove_300	19	87	28	71	0,517	0,484
109	banknews_csw_k4_w2v-skip_300	19	87	28	71	0,517	0,484
110	banknews_csw_k9_glove_300	15	91	24	75	0,517	0,479
111	complete_csw_k3_w2v-skip_300	40	65	50	50	0,512	0,505
112	banknews_csw_k10_w2v-skip_300	40	65	50	50	0,512	0,505
113	complete_csw_k8_w2v-skip_300	33	72	43	57	0,512	0,496
114	complete_ssw_k15_w2v-skip_300	30	75	40	60	0,512	0,493
115	banknews_csw_k8_glove_300	30	75	40	60	0,512	0,493
116	banknews_csw_k5_w2v-skip_300	30	75	40	60	0,512	0,493
117	banknews_csw_k1_w2v-skip_300	10	95	20	80	0,512	0,469
118	banknews_ssw_k5_glove_300	35	69	46	55	0,507	0,494
119	complete_csw_k20_w2v-skip_300	36	66	49	54	0,498	0,487
120	banknews_csw_k3_w2v-skip_300	21	81	34	69	0,498	0,469

Anexo B – *Word2vec e GloVe*

B.1. Word2vec

A ideia por trás do *word2vec* é desenvolver um modelo de menor complexidade e capaz de treinar vetores de palavras com alta qualidade, usando uma quantidade massiva de dados, que chega a centenas de bilhões de palavras, e gerando um vocabulário de milhões de palavras. Até então, os modelos possuíam elevada complexidade e treinavam com um corpus bem menor, com centenas de milhões de palavras [43].

Dessa forma, a arquitetura proposta busca, por meio de uma rede neural de arquitetura rasa, realizar a predição de palavras através do seu contexto e, durante esse processo, realizar o treinamento da matriz *embedding*, que é a matriz de representação distribuída das palavras. Esse treinamento pode ser realizado de duas formas, descritas em maiores detalhes a seguir[80].

B.1.1. *Continuous Bag of Words* (CBOW)

No treinamento *word2vec* usando CBOW parte-se de um contexto de palavras para encontrar a palavra alvo que se enquadra melhor naquele contexto. Intuitivamente funciona como um jogo de preencher lacunas. Por exemplo, “Suco de _____ faz bem à saúde”. Nesse contexto, palavras como “laranja”, “graviola”, “abacaxi”, “fruta” fazem sentido na frase e teriam elevada probabilidade de serem preditas, ao passo que a palavra “carro” seria altamente improvável de ser predita.

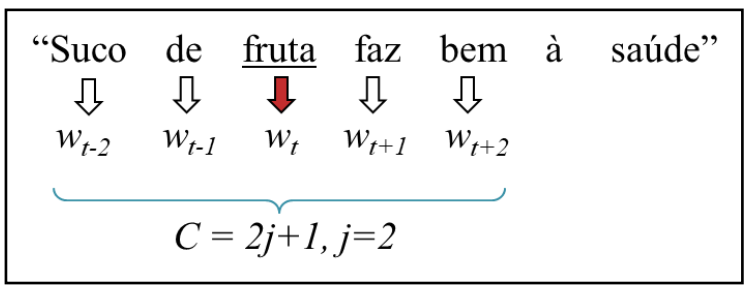

Figura 29 – Ilustração da separação de uma oração em sua palavra central e suas palavras adjacentes, definidas para uma janela igual a 2 (dois). (Elaborado pelo autor)

Na oração “Suco de fruta faz bem à saúde”, a palavra central “fruta”, representada por w_t, é a palavra alvo que deve ser predita pela rede. As palavras adjacentes formam o contexto *C*, com a janela *j* estabelecendo quantas palavras farão parte desse contexto, conforme Figura 29.

A partir disso, usa-se o vetor *one-hot* ($x_{w_{t+i}}, onde - j < i < j, i \neq 0$) de cada palavra do contexto como entrada para a rede neural e faz-se o produto interno com a matriz *W* (matriz *embedding*) - esta operação faz a transformação do vetor *one-hot* para sua representação distribuída. Com o vetor distribuído de cada palavra em mãos, calcula-se uma média desses vetores para encontrar o vetor h_t, que representa o contexto das palavras.

$$h_t = \frac{1}{2 \times j} W^T (x_{w_{t-2}} + x_{w_{t-1}} + x_{w_{t+1}} + x_{w_{t+2}}) \tag{B.1}$$

Em seguida, aplica-se o vetor h_t na matriz de contexto *W'* e aplica-se a função *softmax* para encontrar o vetor de saída u_t, que indicará a probabilidade de encontrar a palavra alvo.

$$p(w_t \mid w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}) = \frac{\exp(u_t)}{\sum_{k=1}^{V} \exp(u_k)}, \tag{B.2}$$

$$\text{onde} \quad u_t = W'^T h_t \tag{B.3}$$

O objetivo do treinamento é maximizar a probabilidade condicional de encontrar a palavra alvo, dado o contexto, ou seja, minimizar a função erro definida por:

$$E = -\log p(w_t \mid w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}) \tag{B.4}$$

O treinamento é realizado por *backpropagation* e seu funcionamento é ilustrado em linhas gerais na Figura 30, com uma simplificação para *j=1*.

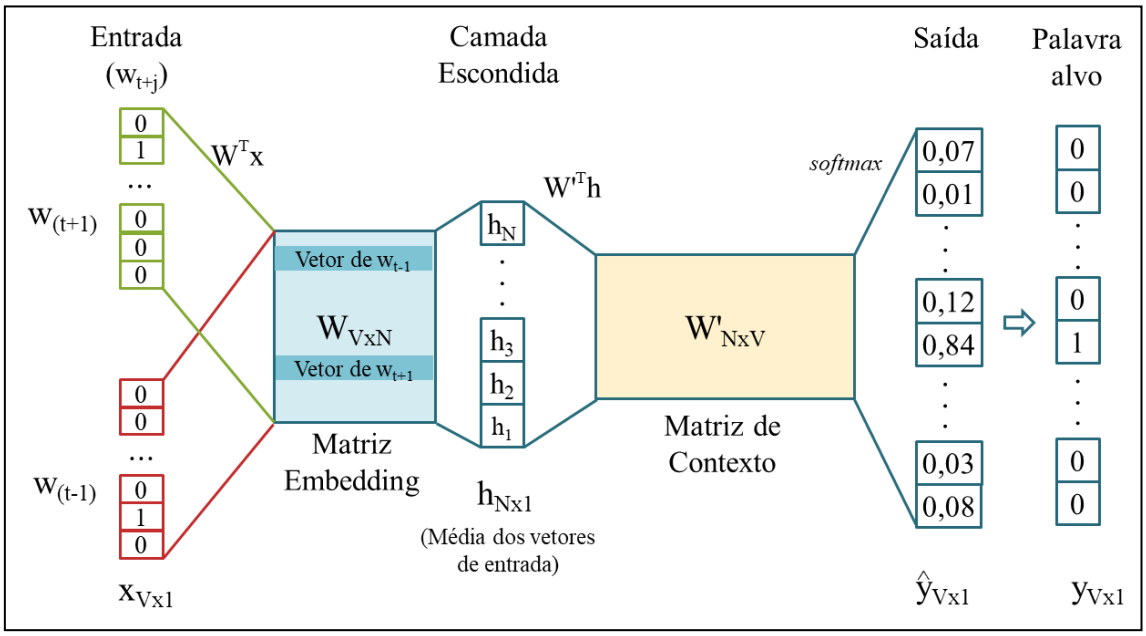

Figura 30 – Ilustração do modelo de treinamento *word2vec* CBOW para *j=1* conforme descrito por MIKOLOV *et al*. [43]. (Elaborado pelo autor)

B.1.2. *Skip-gram*

O treinamento do *word2vec Skip-gram* funciona de forma oposta ao CBOW. A partir de uma palavra alvo, deseja-se predizer as palavras que fazem sentido em um contexto. Ou seja, a palavra "fruta" pode predizer contextos como "Suco de fruta faz bem à saúde" ou "Abacaxi é uma fruta cítrica.", mas seria muito improvável predizer algo como "O carro bateu em uma fruta e capotou na avenida".

A Figura 29 ilustra a separação da palavra alvo e das palavras do contexto, que é feita da mesma forma que no CBOW. A diferença começa logo no início do treinamento, onde se usa o vetor *one-hot* (x_{w_t}) da palavra central como entrada e faz-se o produto interno com a matriz *W* (matriz *embedding*) para encontrar o vetor h_t.

$$h_t = W^T x_{w_t} \tag{B.5}$$

Em seguida, aplica-se h_t na matriz de contexto *W'* e depois a função *softmax* para encontrar o vetor de saída, que indicará a probabilidade de encontrar as palavras do contexto.

$$p(w_{t+i} = w_{T,i} \mid w_t) = \frac{\exp(u_{t,i})}{\sum_{k=1}^{V} \exp(u_k)} \ , \ -j < i < j, i \neq 0 , \tag{B.6}$$

$$\text{com} \quad u_t = W'^T h_t , \tag{B.7}$$

onde $w_{T,i}$ corresponde à saída real na posição *i*, e $u_{t,i}$ é a saída do vetor u_t na posição *i*.

O objetivo do treinamento é maximizar a probabilidade condicional de encontrar as palavras do contexto, dada a palavra alvo, ou seja, minimizar a função erro definida por:

$$E = -\log p(w_{t+i} = w_{T,i} \mid w_t), -j < i < j, i \neq 0 \tag{B.8}$$

O treinamento é realizado por *backpropagation* e seu funcionamento é ilustrado em linhas gerais na Figura 31, com uma simplificação para *j=1*.

Em sua pesquisa, MIKOLOV *et al*. [43] constataram que além de o *word2vec* ser superior aos demais algoritmos existente à época, o modelo *skip-gram* apresenta resultados superiores ao CBOW para semântica e similares para análise sintática. Esses resultados também foram apresentados por HARTMANN *et al*. [40] em sua pesquisa de *word embeddings* em português, o que motivou a escolha do *word2vec skip-gram* para compor este trabalho. Além disso, esta mesma pesquisa mostrou que os resultados com o algoritmo GloVe obteve resultados superiores ao *word2vec,* tanto em análise sintática quanto semântica, motivando também a escolha deste algoritmo para compor o presente trabalho juntamente com o *word2vec skip-gram*.

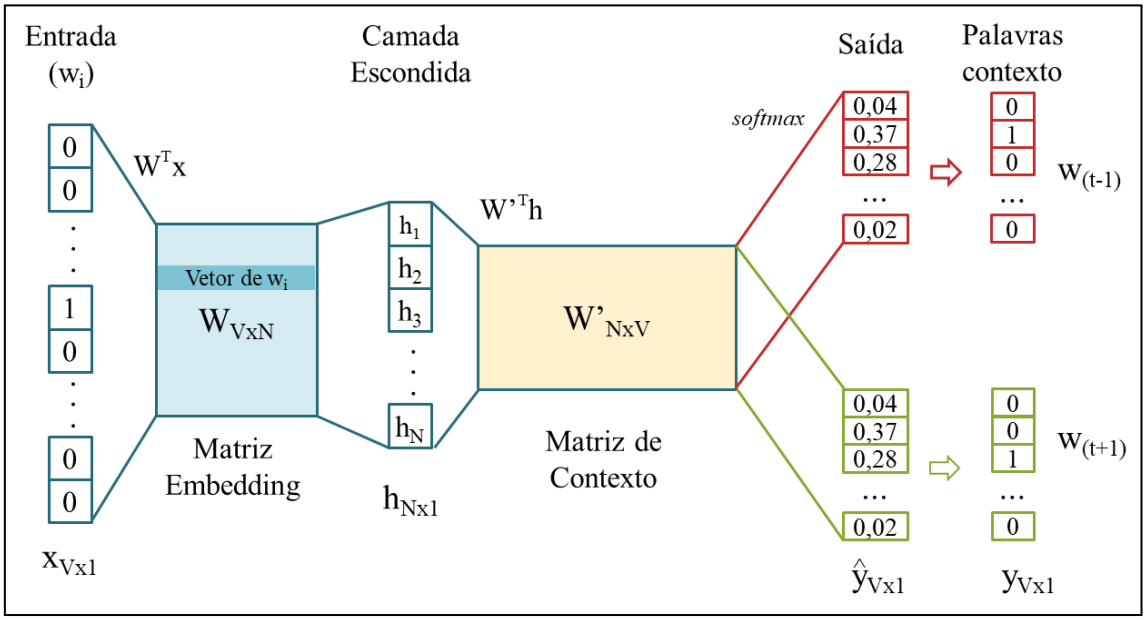

Figura 31 – Ilustração do modelo de treinamento *word2vec skip-gram* para *j=1* conforme descrito por MIKOLOV *et al*. [43]. (Elaborado pelo autor)

B.2. GloVe

O GloVe (*Global Vectors for Word Representation*) é um modelo global de regressão log-bilinear que combina a fatoração de matriz com o método de janela de contexto, ou seja, consiste em construir uma matriz de co-ocorrência global de palavras baseada na estatística de todo o corpus, não apenas no seu contexto local [44]. Por isso, ele apresenta um treinamento mais rápido, escalável para corpus muito grandes e com boa performance mesmo em corpus pequenos e vetores de representação pequenos.

O treinamento do modelo começa com a criação de uma matriz de co-ocorrência global de palavras baseada em uma janela de contexto [81]. Usando-se como exemplo as frases “Suco de fruta faz bem à saúde” e “Não há suco sem fruta” e uma janela de tamanho 2, tem-se a matriz de co-ocorrência apresentada na *Tabela 29*. Nota-se que é uma matriz simétrica, uma vez que se “suco” aparece no contexto de “fruta”, “fruta” também aparece no contexto de “suco”.

Tabela 29 – Exemplo de Matriz de co-ocorrência (Elaborado pelo autor)

	à	bem	de	faz	fruta	há	não	saúde	sem	suco
à	1	1	0	1	0	0	0	0	0	0
bem	1	1	0	1	1	0	0	1	0	0
de	0	0	1	1	1	0	0	0	0	1
faz	1	1	1	1	1	0	0	0	0	0
fruta	0	1	1	1	2	0	0	0	1	2
há	0	0	0	0	0	1	1	0	1	1
não	0	0	0	0	0	1	1	0	0	1
saúde	0	1	0	0	0	0	0	1	0	0
sem	0	0	0	0	1	1	0	0	1	1
suco	0	0	1	0	2	1	1	0	1	2

Definindo-se esta matriz de co-ocorrência por X, o elemento X_{ij} representa o número de vezes que a palavra j ocorreu no contexto da palavra i. Sendo X_i o número de vezes que qualquer palavra apareceu no contexto da palavra i, tem-se que a probabilidade da palavra j aparecer no contexto i é denotada por:

$$P_{ij} = P(j \mid i) = \frac{X_{ij}}{X_i}, \tag{B.9}$$

Partindo da premissa de que as taxas de co-ocorrência entre duas palavras em um contexto estão fortemente conectadas ao significado das palavras [81], a razão entre as probabilidades de ocorrência de palavras distintas dado um contexto (P_{ik}/P_{jk}) é função dos vetores de palavras w_i, w_j e de contexto w_k.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}, \tag{B.10}$$

Para se chegar a uma função adequada, PENNINGTON *et al.* [44] definem uma série de premissas matemáticas para se chegar a uma forma simplificada dessa relação, obtendo-se a seguinte equação:

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}), \tag{B.11}$$

Uma desvantagem dessa abordagem é dar o mesmo peso a todas as co-ocorrências, uma vez que co-ocorrências raras incorporam ruído e carregam pouca informação. Dessa forma, introduz-se uma função de peso $f(X_{ij})$ na função custo para ajustar essas situações.

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2, \tag{B.12}$$

Essa função de peso *f* possui valores pequenos, próximos de 0, para co-ocorrências raras e elevados para as muito frequente, porém, limitando esse peso a 1, para não gerar distorções, sendo definida conforme a equação 3.13 e apresentada graficamente na Figura 32.

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & se\ x < x_{\max} \\ 1 & se\ x \geq x_{\max} \end{cases}, \tag{B.13}$$

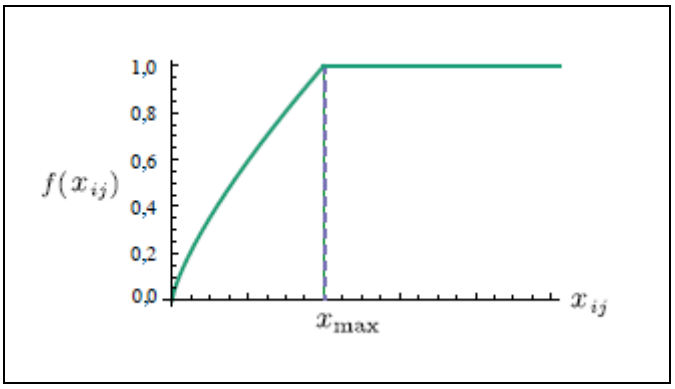

Figura 32 – Função de peso *f* com $\alpha = 3/4$ [44].