



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA (INE)
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Armindo Antonio Guerra Junior

Sistema de Previsão de Performance para Alunos em Cursos Online (SPPACO)

Florianópolis
2023

Armindo Antonio Guerra Junior

Sistema de Previsão de Performance para Alunos em Cursos Online (SPPACO)

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Ciência da Computação.

Orientador: Profa. Luciana de Oliveira Rech, Dra.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Guerra Junior, Armino Antonio
Sistema de Previsão de Performance para Alunos em
Cursos Online (SPPACO) / Armino Antonio Guerra Junior ;
orientadora, Luciana de Oliveira Rech, 2023.
91 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Ciência da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciência da Computação. 2. Previsão de Desempenho de
Alunos EAD. 3. Processamento de Linguagem Natural. 4.
Redes Neurais Artificiais Profundas. I. Rech, Luciana de
Oliveira. II. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Ciência da Computação. III.
Título.

Armando Antonio Guerra Junior

Sistema de Previsão de Performance para Alunos em Cursos Online (SPPACO)

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Profa. Patricia Della Mía Plentz, Dra.
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)

Prof. Mauro Roisenberg, Dr.
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)

Prof. Ricardo Alexandre Reinaldo de Moraes, Dr..
Departamento de Ciência da Informação
Universidade Federal de Santa Catarina (UFSC)

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Ciência da Computação.

Coordenação do Programa de
Pós-Graduação

Profa. Luciana de Oliveira Rech, Dra.
Orientador

Florianópolis, 2023.

Dedico esse trabalho à minha mãe, Célia Terezinha
Móseno Guerra, cujo a sua fortaleza reside no amor
incondicional a sua família.

AGRADECIMENTOS

Agradeço à minha orientadora, professora Luciana de Oliveira Rech, não somente por ter aceitado guiar meu trabalho, mas também pela compressão e incentivo frente aos momentos delicados que enfrentei. Agradeço à minha família, pela referência de honestidade, trabalho e força. Agradeço em especial as minhas irmãs, Elizabete Olinda Guerra e Eliane Cristina Guerra, por contribuírem com meu desenvolvimento estudantil e ao meu pai, Armindo Antonio Guerra, pela referência intelectual e de quem eu, felizmente, herdei o amor pelo conhecimento e pelo futebol. Agradeço à minha esposa, Aline Back, por me ensinar cotidianamente a amar, respeitar e persistir no que realmente vale a pena. E, por fim, agradeço ao Itaú Unibanco, pelo incentivo contínuo para o aprimoramento individual dos seus colaboradores.

*“viver é como andar de bicicleta,
para se manter em equilíbrio é preciso manter-se em movimento”
(Albert Einstein)*

RESUMO

A discussão sobre os principais fatores que causam a evasão dos alunos de cursos de Ensino a Distância é bastante ampla, passando por fatores sociais, institucionais e econômicos. Dentre todos esses fatores, um específico que se destaca é o baixo desempenho acadêmico dos alunos, que é percebido nas sucessivas reprovações, culminando na desistência definitiva do curso. Alguns autores tratam as reprovações como causa explícita da desistência ou evasão, outros, por outro lado, elegem as reprovações como um variável importante do processo. Independentemente dos motivos pelos quais as taxas de evasão atingem valores tão elevados, é importante que a sociedade educacional, bem como seus gestores, pensem em estratégias para combatê-la. Com base nesse cenário, esse trabalho propõe a criação do Sistema de Previsão de Performance para Alunos em Cursos Online (SPPACO), que é um sistema de previsão de performance de alunos, baseado em Processamento de Linguagem Natural e Redes Neurais Artificiais Profundas, para ser utilizado como ferramenta de mitigação das reprovações e, portanto, contribuindo com o combate a evasão. Para criação do SPPACO foram testadas as principais arquiteturas de redes neurais artificiais profundas do estado da arte. As arquiteturas foram comparadas exaustivamente entre si e, também, em conjunto, com intuito de identificar a arquitetura mais adequada para o problema proposto. Com essa metodologia alcançou-se até 98% de acurácia nas previsões nos testes realizados. Por fim, vislumbrando uma possível integração sistêmica do SPPACO com as plataformas educacionais, foi proposta uma arquitetura conceitual de serviço das previsões realizadas.

Palavras-chave: Desempenho de Alunos *online*. Processamento de Linguagem Natural. Redes Neurais Artificiais.

ABSTRACT

The discussion about the main factors that cause students to drop out of Distance Learning courses is quite broad, including social, institutional and economic factors. Among all these factors, one specific one that stands out is the low academic performance of students, which is seen in successive failures, culminating in definitive withdrawal from the course. Some authors treat failures as an explicit cause of withdrawal or evasion, others, on the other hand, choose failures as an important variable in the process. Regardless of the reasons why dropout rates reach such high values, it is important that the educational society, as well as its managers, think of strategies to combat it. Based on this scenario, this work proposes the creation of the Performance Prediction System for Students in Online Courses (SPPACO), which is a student performance prediction system, based on Natural Language Processing and Deep Artificial Neural Networks, to be used as a tool to mitigate failures and, therefore, contributing to the fight against evasion. To create SPPACO, the main state-of-the-art deep artificial neural network architectures were tested. The architectures were exhaustively compared with each other and also together, with the aim of identifying the most suitable architecture for the proposed problem. With this methodology, up to 98% accuracy was achieved in predictions in the tests carried out. Finally, envisioning a possible systemic integration of SPPACO with educational platforms, a conceptual service architecture for the predictions made was proposed.

Keywords: Student Performance. Natural Language Processing. Artificial Neural Networks.

LISTA DE FIGURAS

Figura 1 – Número de Ingressos em Cursos de Graduação por Modalidade (INEP, 2020)	17
Figura 2 – Modelo Matemático do Perceptron	24
Figura 3 – Dados linearmente separáveis (à esquerda) e Não linearmente separáveis (à direita)	26
Figura 4 – Representação de uma Rede Perceptron de Múltiplas Camadas (SOBREIRO <i>et al.</i> , 2008)	27
Figura 5 – Representação de uma operação de convolução	29
Figura 6 – Rede Neural Artificial <i>feedforward</i> (à direita) e Rede Neural Recorrente (à esquerda)	30
Figura 7 – Matriz de Confusão	32
Figura 8 – Curva ROC	35
Figura 9 – Arquitetura utilizada pelos autores (LIU; GUO, 2019) para classificação de textos	41
Figura 10 – Tabela com a comparação dos resultados obtidos pelo trabalho (LIU; GUO, 2019)	42
Figura 11 – O Framework MVHGNN (LI, M. <i>et al.</i> , 2022)	43
Figura 12 – Rede Neural para o Sistema proposto em (KEHINDE <i>et al.</i> , 2022) .	44
Figura 13 – Diagrama de funcionamento do <i>Student-Formulator</i> (HUSSAIN; KHAN, 2023)	46
Figura 14 – Diagrama de funcionamento do <i>Hybrid Machine Learning Framework</i> (EVANGELISTA, 2021)	47
Figura 15 – Rede Neural do trabalho (YUSOF; ABD KHALID, 2021)	48
Figura 16 – <i>Deep Neural Network Based Educational Data Classification Model</i> (SURENTHIRAN; RAJALAKSHMI; SUJATHA, 2021)	49
Figura 17 – Modelos 1, 2 e 3 treinados para o CORC-Net (YADAV; SRIVASTAVA, 2020)	51
Figura 18 – Rede Neural para o Sistema CORC-Net (YADAV; SRIVASTAVA, 2020)	51
Figura 19 – Diagrama das Tabelas de Banco de Dados Utilizadas	57
Figura 20 – Textos ao Longo do Tratamento	58
Figura 21 – Separação das Palavras em Categorias	58
Figura 22 – Algoritmo de Treinamento da Rede C-BiLSTM	61
Figura 23 – Histórico da Acurácia e Erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação	62
Figura 24 – Proposta de Arquitetura para o SPPACO	63
Figura 25 – Obtenção dos vetores de palavras com <i>Word2vec</i> utilizando o algoritmo <i>Skip-Gram</i>	67

Figura 26 – Similaridade da palavra aluno com <i>Word2vec</i>	67
Figura 27 – Configuração dos pesos da Rede Neural Utilizando a <i>Embedding Matrix</i>	67
Figura 28 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no primeiro mês	69
Figura 29 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no primeiro mês	69
Figura 30 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no primeiro mês	70
Figura 31 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no primeiro mês	71
Figura 32 – Arquiteturas de Redes Neurais de acordo com Instituto Asimov (SINGH; LAARSE, 2021)	83
Figura 33 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras . .	84
Figura 34 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras . .	84
Figura 35 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras	84
Figura 36 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras	85
Figura 37 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras .	85
Figura 38 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras .	85
Figura 39 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras	86
Figura 40 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras	86
Figura 41 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras .	86
Figura 42 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras .	87
Figura 43 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras	87

Figura 44 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras	87
Figura 45 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no primeiro mês	88
Figura 46 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no primeiro mês	88
Figura 47 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no primeiro mês	88
Figura 48 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no primeiro mês	89
Figura 49 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no segundo mês	89
Figura 50 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no segundo mês	89
Figura 51 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no segundo mês	90
Figura 52 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no segundo mês	90
Figura 53 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no terceiro mês	90
Figura 54 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no terceiro mês	91
Figura 55 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no terceiro mês	91
Figura 56 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no terceiro mês	91

LISTA DE TABELAS

Tabela 1 – Tabela comparativa com os trabalhos do estado da Arte para previsão do desempenho de alunos	54
Tabela 2 – Resultados de experimentos com textos postados no primeiro mês de curso	62
Tabela 3 – Resultados experimentais com textos postados no primeiro mês de curso	71
Tabela 4 – Previsões acertados pelo modelo	72
Tabela 5 – Previsões erradas pelo modelo	73

LISTA DE ABREVIATURAS E SIGLAS

CNN	Convolutional Neural Network
EAD	Ensino a Dinstância
IES	Instituições de Ensino Superior
KNN	K-Nearest Neighbor
PLN	Processamento de Linguagem Natural
RNN	Recurent Neural Network
SVM	Support Verctor Machines

SUMÁRIO

1	INTRODUÇÃO	17
1.1	MOTIVAÇÃO	17
1.2	PROPOSTA	19
1.3	OBJETIVOS	20
1.3.1	Objetivos Específicos	20
1.4	CONTRIBUIÇÕES	20
1.5	PUBLICAÇÕES	21
1.6	ORGANIZAÇÃO DO TRABALHO	21
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	APRENDIZADO DE MÁQUINA	22
2.1.1	Tipos de Aprendizado de Máquina	22
2.1.1.1	Aprendizado Supervisionado	22
2.1.1.2	Aprendizado Não Supervisionado	23
2.1.1.3	Aprendizado por Reforço	23
2.1.2	Redes Neurais	23
2.1.2.1	Perceptron de Única Camada	24
2.1.2.2	Função de Ativação	25
2.1.2.3	Perceptron de Múltiplas Camadas	26
2.1.2.4	Aprendizado nas Redes Neurais Artificiais	27
2.1.2.5	Redes Neurais Convolucionais	28
2.1.2.6	Redes Neurais Recorrentes	29
2.1.3	Métricas de Avaliação de Modelos	30
2.1.3.1	Regressão	30
2.1.3.2	Classificação	31
2.1.3.3	Matriz de Confusão	31
2.1.3.4	Acurácia	32
2.1.3.5	Precisão	33
2.1.3.6	Revocação	33
2.1.3.7	Score F1	33
2.1.3.8	Curva ROC	33
2.2	PROCESSAMENTO DE LINGUAGEM NATURAL	35
2.2.1	Modelos de Linguagem	35
2.2.2	Modelos <i>n-gram</i>	36
2.2.3	<i>Word Embeddings</i>	37
2.3	CLASSIFICAÇÃO DE TEXTOS	38
2.4	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO	38
3	REVISÃO DOS TRABALHOS CORRELACIONADOS	40

3.1	CLASSIFICAÇÃO DE TEXTOS	40
3.2	PREVENDO DESEMPENHO DE ALUNOS	42
3.2.1	Multi-view hypergraph neural networks (MVHGNN)	42
3.2.2	Prediction of Students' performance with Artificial Neural Network using Demographic Traits	44
3.2.3	Student-Performulator	45
3.2.4	A Hybrid Machine Learning Framework	46
3.2.5	Precision Education Reviews: A case study on predicting student's performance	48
3.2.6	Deep Neural Network Based Educational Data Classification Model	48
3.2.7	Student Performance Prediction from E-mail Assessments Using Tiny Neural Networks (CorC-Net)	50
3.2.8	Deep learning for predicting students academic performance . .	52
3.2.9	An Approach to predict a students academic performance using Recurrent Neural Network (RNN)	52
3.3	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO	53
4	ARQUITETURA PROPOSTA	56
4.1	COLETA, TRATAMENTO E ROTULAGEM DOS DADOS	57
4.1.1	Bando de Dados	57
4.1.2	Limpeza, Organização e Tratamento dos Dados	58
4.2	PRÉ-TREINO E CRIAÇÃO DAS <i>WORD EMBEDDING</i>	59
4.3	AVALIAÇÃO, TREINAMENTO E ESCOLHA DA ARQUITETURA DE REDE NEURAL PROFUNDA	60
4.3.1	Treinamento do Modelo de Aprendizado de Máquina	60
4.4	PROPOSTA DE ARQUITETURA PARA O SPPACO	62
4.4.1	Camada de Pré-processamento e Treino do Modelo	63
4.4.2	Interface do SPPACO	64
4.5	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO	64
5	ANÁLISE DOS RESULTADOS OBTIDOS E DISCUSSÃO	66
5.1	BASE DE DADOS USADA PARA O TREINAMENTO DOS MODELOS	66
5.2	CRIAÇÃO DAS <i>WORD EMBEDDINGS</i>	66
5.3	CONFIGURAÇÃO DOS PARÂMETROS PARA TREINAMENTO . . .	67
5.4	RESULTADOS OBTIDOS	68
5.4.1	Arquitetura MLP	68
5.4.2	Arquitetura CNN	69
5.4.3	Arquitetura CLSTM	70
5.4.4	Arquitetura C-BiLSTM	70
5.4.5	Exemplos de previsões acertados pelo modelo	71
5.4.6	Exemplos previsões erradas pelo modelo	72

5.5	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO	73
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	74
6.1	REVISÃO DAS MOTIVAÇÕES E OBJETIVOS	74
6.2	VISÃO GERAL DO TRABALHO	74
6.3	CONTRIBUIÇÕES	75
6.4	LIMITAÇÕES	75
6.5	TRABALHOS FUTUROS	76
	REFERÊNCIAS	78
	APÊNDICE A – ARQUITETURAS DE REDES NEURAIIS	83
	APÊNDICE A – GRÁFICOS DOS EXPERIMENTOS	84
A.1	POR QUANTIDADE DE PALAVRAS	84
A.1.1	De 10 a 50 palavras	84
A.1.2	De 50 a 100 palavras	85
A.1.3	De 100 a 200 palavras	86
A.2	POR DATA DA POSTAGEM	88
A.2.1	Postados no Primeiro Mês	88
A.2.2	Postados no Segundo Mês	89
A.2.3	Postados no Terceiro Mês	90

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Os cursos de ensino a distância (EAD) no Brasil em 2020 receberam mais matrículas do que os presenciais. Os dados fazem parte do Censo da Educação Superior 2020, divulgado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep) e pelo Ministério da Educação (MEC) (INEP, 2020). Esse fenômeno havia sido constatado em 2019, apenas na rede privada. Dos mais de 3,7 milhões de ingressantes em 2020, mais de 2 milhões (53,4%) optaram por cursos a distância e 1,7 milhão (46,6%) pelos presenciais.

O ano de 2020 também foi o ano em que houve a explosão de casos de COVID-19 no Brasil e, por consequência, a interrupção do ano letivo presencial. Desse modo, o ensino a distância apresentou-se como única alternativa para que algumas instituições de ensino, mesmo para aquelas que não adotam essa modalidade dentre as suas principais metodologias, dessem continuidade as suas atuações, como aponta do relatório do Censo EAD BR (BR, 2020). Ainda de acordo com o relatório, 34% das instituições de ensino viram um aumento significativo no número de matrículas e a simples procura pela modalidade aumentou em até 50% durante a pandemia.

Embora o aumento tenha intensificado-se em 2020 por conta da COVID-19, o aumento da representatividade dos cursos EAD pode ser notada já em anos anteriores, como mostra a **Figura 1**. Ao observar o período de 2010 a 2020, por exemplo, o número de matrículas em cursos presenciais diminuiu 13,9%, enquanto nos cursos EAD aumentou 428,2%. Em 2010, a participação percentual dos novos alunos em cursos superiores *Online* era de 17,4%, em 2020, alcançou 53,4% dos estudantes. A queda significativa nas matrículas presenciais é notada com clareza a partir de 2015. Em contrapartida, o crescimento na modalidade EAD nesse mesmo ano começa a crescer mais do que os anos anteriores.

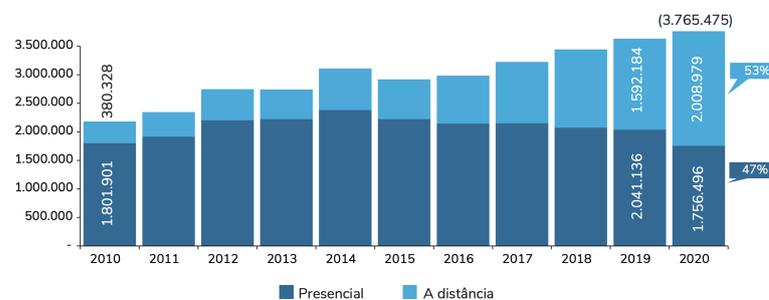


Figura 1 – Número de Ingressos em Cursos de Graduação por Modalidade (INEP, 2020)

Dada a importância que o tema adquire, é natural que os profissionais especializados na área dediquem-se em analisar os principais fatores que influenciam o

comportamento dos alunos que optam por realizar seus estudos de modo remoto. Nesse contexto, assuntos como evasão e reprovação de alunos, que frequentemente ganham destaque dentre as métricas avaliadas nos relatórios do Censo EAD BR, apresentam-se como desafios importantes e que exigem atenção não somente dos profissionais especializados, mas também de todas as instituições de ensino superior (IES) que possuem alunos em EAD.

Ao analisar o fenômeno com um pouco mais de detalhes, observam-se alguns indicadores que são, de fato, alarmantes, como as taxas de evasão de alunos no ano de 2014, que alcançaram aproximadamente 75% nos cursos técnicos e cerca de 40% no Ensino Superior (VENTURINI *et al.*, 2014). Outro estudo, realizado pelo Mapa do Ensino Superior SEMESP (2018), informa que as taxas de evasão variam de 36,6% nas IES privadas e 30,4% nas IES públicas. Embora a taxa de concluintes dos cursos EAD tenham parado de apresentar queda em 2016, a evasão ainda é um problema sério para as IES, tanto presenciais, quanto EAD.

De acordo com Oliveira e Bittencourt (2020), a permanência do aluno até a conclusão do curso traz benefícios tanto para a sociedade quanto para o mercado de trabalho, que contará com profissionais mais qualificados e com renda maior. O insucesso, por sua vez, traz prejuízos financeiros para as instituições de modo geral e, em especial quando se tratam de cursos públicos, para o governo, que não consegue implantar políticas sociais que atinjam o seus objetivos, como também argumenta Oliveira e Bittencourt (2020).

A discussão sobre os principais fatores que causam a evasão dos alunos é bastante ampla, passando por fatores sociais, institucionais e econômicos, como salienta Noronha, Carvalho e Santos (2001) ao avaliar as dificuldades acadêmicas de modo geral. Dentre todos esses fatores, um específico que se destaca é o baixo desempenho acadêmico dos alunos, que é percebido nas sucessivas reprovações, culminando na desistência definitiva do curso, como pontua Cislaghi *et al.* (2008). Alguns autores tratam as reprovações como causa explícita da desistência ou evasão, como é o caso de Cislaghi *et al.* (2008). Outros, por outro lado Silva *et al.* (2017), elegem as reprovações como um variável importante do processo e não como sua causa.

Como já mencionado, a evasão prematura de alunos matriculados em cursos oferecidos pelo governo, por exemplo, é preocupante porque causa desperdício de recursos financeiros e educacionais. Portanto, independentemente dos motivos pelos quais as taxas de evasão atingem valores tão elevados, é importante que a comunidade educacional, bem como seus gestores, pensem em estratégias para combatê-la. Uma possibilidade de combater a evasão é implementando métodos que mitiguem as suas principais causas. A reprovação dos alunos, já identificada como uma das principais causas Cislaghi *et al.* (2008), pode ser combatida adotando medidas preventivas para os alunos que demonstrem significativo risco de reprovação.

1.2 PROPOSTA

É consenso científico que a linguagem humana reflete personalidade, estilo de pensamento, ligações sociais e estados emocionais (MAJUMDER *et al.*, 2017). O texto é uma das maneiras utilizadas pelas pessoas para comunicação, de modo geral, e pode ser considerada como um tipo específico de linguagem. Os textos registrados pelos alunos são dados abundantes nas plataformas de ensino a distância, sejam eles produzidos em fóruns de discussão, na resolução de problemas propostos pelos professores, ou mesmo na interação social entre alunos e entre alunos e professores.

Como técnica computacional para trabalhar com os textos e utilizá-los de forma analítica, existe o Processamento de Linguagem Natural, ou simplesmente PLN, que é uma área multidisciplinar que engloba Matemática, Ciência da Computação, Inteligência Artificial e Linguística com o objetivo de analisar problemas relacionados à geração, interpretação e manipulação da linguagem humana. Devido à grande quantidade de textos disponíveis na internet, blogs, portais de notícias, redes sociais, etc., a PLN atualmente recebe muita atenção dos pesquisadores e também se tornou um campo orientado a dados (do inglês, *data driven*).

Ao longo do tempo, várias abordagens de Aprendizado de Máquina (do inglês, *Machine Learning*), que aproveitam a quantidade de dados para mapear padrões, têm sido amplamente utilizadas em aplicações de PLN. Dentre eles podemos destacar *Nayve Bayes*, *k-nearest Neighbors* (KNN), *Hidden Markov Models*, *Conditional Random Fields*, *Decision Tress*, *Random Forest* e *Support Verctor Machines* (SVMs). No entanto, por questões de eficiência e assertividade, essas abordagens foram recentemente substituídas, ou pelo menos aprimoradas, por Redes Neurais Artificiais Profundas (OTTER; MEDINA; KALITA, 2019).

Redes Neurais Artificiais Profundas, também conhecidas pelo termo em inglês *Deep Learning*, têm alcançado resultados notáveis em diversas áreas do conhecimento humano. Como exemplos, podemos citar Visão Computacional, Reconhecimento de Fala e Classificação de Textos (LIU; GUO, 2019). Para Classificação de Textos, que considera textos como variáveis preditoras em um modelo de Aprendizado de Máquina Supervisionado e que será nosso objeto de estudo, dois tipos são aplicados com bons resultados; Redes Neurais Convolucionais (CNNs) e Redes Neurais Recorrentes (RNNs) (LIU; GUO, 2019). As CNNs são capazes de aprender uma resposta local a partir de dados temporais ou espaciais, mas não têm a capacidade de aprender correlações sequenciais de modo satisfatório. Em contrapartida, as RNNs são especializadas em modelagem sequencial, mas não podem extrair recursos em paralelo. De acordo com Liu e Guo (2019) ambos os modelos, quando usados em conjunto e com a adição de outras técnicas como *Word Embeddings*, por exemplo, podem apresentar resultados significativos na classificação de textos.

Como destacado nos parágrafos anteriores, a evasão de alunos nos cursos de

EAD é um problema relevante no âmbito educacional. Outro ponto destacado são as reprovações dos alunos, ora tratada como causa, ora tratada como variável importante no processo. A área de pesquisa relacionada à previsão do desempenho dos alunos é multidimensional e pode ser explorada e analisada através de várias perspectivas, aplicando técnicas estatísticas para prever a evasão de forma direta, ou como também é a proposta desse trabalho, analisando e prevendo os fatores que mais influenciam a evasão, como as reprovações.

Portanto, levando em consideração o problema identificado e as ferramentas computacionais que se tem a disposição, o presente trabalho propõe o SPPACO, que é um Sistema de Previsão de Performance de Alunos para Cursos *Online* para ser utilizado como ferramenta de mitigação das reprovações e, portanto, contribuindo com o combate a evasão.

1.3 OBJETIVOS

O objetivo geral desse trabalho é prever de forma eficiente a reprovação de alunos de cursos *Online* com intuito contribuir com o combate a evasão.

1.3.1 Objetivos Específicos

Vislumbrando alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

1. Encontrar as arquiteturas de Redes Neurais Profundas mais adequadas para classificação de textos e para previsão de performance de alunos;
2. Realizar a previsão do desempenho do aluno como um projeto de classificação de textos;
3. Propor e Prototipar uma arquitetura para um sistema de Aprendizado de Máquina para prever o desempenho alunos;

1.4 CONTRIBUIÇÕES

Além da revisão sistemática da literatura relacionada e da exposição organizada dos principais conceitos técnicos envolvidos no trabalho de Classificação de Textos utilizando como ferramentas Processamento de Linguagem Natural e Redes Neurais Profundas, pode-se considerar como contribuições desse trabalho:

1. Identificação de uma arquitetura de Redes Neurais Profundas para realizar a classificação dos textos e, portanto, a previsão do desempenho dos alunos com alta precisão;
2. Elaboração da arquitetura para um Sistema de previsão baseado em Redes Neurais Profundas.

3. Avaliação experimental detalhada dos testes para escolha da Rede Neural Profunda Mais adequada para previsão do desempenho de alunos de cursos *Online*.

1.5 PUBLICAÇÕES

Os resultados preliminares da Prototipação, Testes, Modelagem e Implementações foram submetidos para a conferência *18th International Conference on Machine Learning and Data Mining (MLDM 2022)* sob o título de *Predicting Student Performance of Online Courses with Deep Learning and NLP from Texts in Portuguese (pt-br)*. A submissão foi aceita e consta nos *proceedings*, podendo ser acessada em http://ibai-publishing.org/html/proceedings_2022/pdf/proceedings_mldm_2022.pdf.

1.6 ORGANIZAÇÃO DO TRABALHO

Nesse capítulo são descritas as motivações para a realização do presente trabalho e os objetivos a serem alcançados. O restante do documento está organizado da seguinte maneira:

- **Capítulo 2 - Fundamentação Teórica:** Apresenta os conceitos de Aprendizado de Máquina e Processamento de Linguagem Natural necessários para sustentar o presente trabalho;
- **Capítulo 3 - Estado da Arte:** Apresenta os principais trabalhos correlatos ao tema de previsão de desempenho de alunos implementados com Aprendizado de Máquina;
- **Capítulo 4 - SPPACO:** Apresenta as técnicas de tratamento de dados aplicadas, testes, escolha e implementação do modelo de Aprendizado de Máquina que suporta as previsões e, por fim, uma proposta de arquitetura sistêmica;
- **Capítulo 5 - Análise dos Resultados:** Apresenta os resultados obtidos após a utilização das técnicas de tratamento de dados aplicadas, testes, escolha e implementação do modelo de Aprendizado de Máquina e a comparação com os modelos listados como estado da arte para previsão do desempenho de alunos;
- **Capítulo 6 - Considerações Finais e Trabalhos Futuros:** Avalia de modo geral os resultados obtidos com a realização desse trabalho e apresenta algumas sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão exploradas as contribuições mais relevantes, as correntes de pensamento mais influentes e as descobertas que moldam as áreas de pesquisa que embasam o presente trabalho. Buscar-se-á estabelecer uma base sólida e sintética para servir de sustentação conceitual para melhor compreensão dos assuntos abordados. O capítulo divide-se em três seções; na **Seção 2.1**, serão explorados os conceitos relacionados a Aprendizado de Máquina, na **Seção 2.2** serão explorados os conceitos relacionados a Processamento de Linguagem Natural e, por fim, na **Seção 2.3** serão explorados os conceitos relacionados Classificação de Textos, unindo os conceitos das duas seções anteriores.

2.1 APRENDIZADO DE MÁQUINA

Ao longo do tempo, várias abordagens de Aprendizado de Máquina têm sido amplamente utilizadas em aplicações de Processamento de Linguagem Natural. Dentre elas podemos destacar *Nayve Bayes*, *k-nearest Neighbors* (KNN), *Hidden Markov Models*, *Conditional Random Fields*, *Decision Tress*, *Random Forest* e *Support Vector Machines* (SVMs). No entanto, por questões de eficiência e assertividade, essas abordagens foram recentemente substituídas, ou pelo menos aprimoradas, por Redes Neurais Artificiais Profundas (OTTER; MEDINA; KALITA, 2019). Dentro desse contexto, as seções que se seguem apresentam os conceitos necessários para a compreensão das técnicas Redes Neurais utilizadas pelo presente trabalho.

2.1.1 Tipos de Aprendizado de Máquina

Redes Neurais é um tipo específico de Aprendizado de Máquina (do inglês *Machine Learning*), que por sua vez faz parte da disciplina Inteligência Artificial. De acordo com Goodfellow, Bengio e Courville (2016), Aprendizado de Máquina é essencialmente uma forma de estatística aplicada para estimar funções não triviais. Os algoritmos de Aprendizado de Máquina podem ser divididos em três categorias;

1. Aprendizado Supervisionado;
2. Aprendizado Não Supervisionada;
3. Aprendizado por Reforço.

2.1.1.1 Aprendizado Supervisionado

Quando estão disponíveis no conjunto de dados um desfecho ou rótulo, pode-se utilizar o Aprendizado Supervisionado, que é a busca iterativa por uma função que generalize o mapeamento entre os dados históricos e seus rótulos associados (GOODFELLOW; BENGIO; COURVILLE, 2016). Problemas de Aprendizado Supervisionado

são segregados em problemas de Regressão e Classificação. Em um problema de Regressão, busca-se prever os resultados em uma saída contínua, o que significa a tentativa em mapear variáveis de entrada para alguma função contínua. Em problemas de Classificação, busca-se prever os resultados em uma saída discreta, ou seja, busca-se mapear variáveis de entrada em categorias ou classes distintas.

2.1.1.2 Aprendizado Não Supervisionado

O Aprendizado Não Supervisionado, por outro lado, permite abordar problemas com pouca ou nenhuma ideia dos desfechos e/ou rótulos. Esse tipo de Aprendizado utiliza propriedades úteis da estrutura intrínseca dos dados, quando analisados em conjunto. Nesse caso, ao invés de estimar uma função que mapeie resultados de entrada e saída, é possível aplicar técnicas para agrupar os dados. Algoritmos dessa modalidade, de modo geral, utilizam cálculos matemáticos para associar os dados em grupos (*do inglês, clusters*) a partir de alguma característica que os tornem semelhantes entre si (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma implementação clássica desse tipo de abordagem é o Algoritmo *k-means*, que consiste em um método de conglomerar os dados em torno de posições cartesianas do plano ou do espaço, denominados centróides. O número de centróides num primeiro momento pode ser escolhido de forma aleatória e busca-se o número ótimo de centróides que minimalize a distância dos dados aos seus respectivos.

2.1.1.3 Aprendizado por Reforço

Aprendizado por Reforço (*do inglês, Reinforcement Learning*), de acordo Academy (2021), é uma técnica de Aprendizado de Máquina cujo o treinamento baseia-se na tomada de uma sequência de decisões em um ambiente incerto e potencialmente complexo, de modo que cada decisão pode ser punida dadas as configurações iniciais atribuídas à máquina. No Aprendizado por Reforço, o sistema de inteligência artificial está preparado para enfrentar situações imprevisíveis e utiliza tentativa e erro para encontrar uma solução para o problema proposto. Algoritmos dessa natureza possuem uma característica menos determinista no curto prazo e são menos utilizados em relação aos demais acima citados, que encontraram utilização massificada, dado a quantidade de dados e capacidade computacional que temos à disposição atualmente.

2.1.2 Redes Neurais

Redes Neurais (ou Redes Neurais Artificiais) são modelos computacionais de Aprendizado de Máquina inspirados no sistema nervoso central humano, cujo a estratégia para solucionar os problemas baseia-se em uma estrutura matemática que busca simular o trânsito de informações das sinapses nervosas. A estrutura base dessa abor-

dagem foi proposta em 1958 e recebeu do seu criador, Frank Rosenblatt, o nome de Perceptron (ROSENBLATT, 1958). O Perceptron é considerada a forma mais simples de configuração de uma rede neural artificial, uma vez que é constituída de uma única camada neural, ou de um único neurônio artificial. As arquiteturas mais sofisticadas de Redes Neurais Artificiais são, em certa medida, a união de vários neurônios conectados ou não entre si para receber informações, processá-las e devolver um resultado.

2.1.2.1 Perceptron de Única Camada

Como já mencionado, o Perceptron é um modelo matemático de um neurônio biológico (ROSENBLATT, 1958). Enquanto que nos neurônios reais os dendritos são os componentes responsáveis por receber impulsos elétricos dos axônios, ou ainda, de outros neurônios, no Perceptron, o que transita entre os seus componentes são valores numéricos. Nas sinapses nervosas humanas os sinais elétricos são modulados pela força do contato sináptico, permitindo que a intensidade do sinal elétrico transmitido seja controlado, gerando resultados diferentes de acordo com a necessidade de interpretação do sinal recebido. No Perceptron, por sua vez, esse controle é feito por um valor numérico denominado peso. Em um processo de treinamento que ocorre em uma rede neural artificial, o peso é inicializado aleatoriamente e muda de valor a cada iteração dos dados. No Perceptron, portanto, assim como nas redes neurais artificiais mais complexas, os pesos juntamente com o outro valor numérico conhecido como bias (ou, viés) são os responsáveis pelo aprendizado da rede.

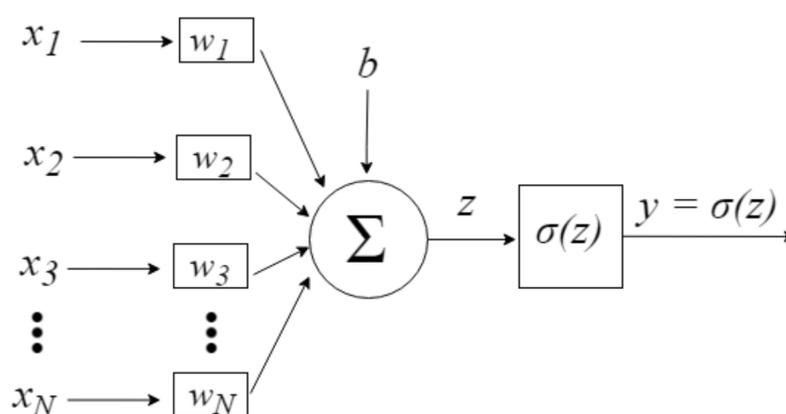


Figura 2 – Modelo Matemático do Perceptron

Dada a sua característica matemática, o Perceptron de uma única camada pode ser utilizado para dividir duas classes linearmente separáveis, que é o objetivo também de outros modelos de Aprendizado Supervisionado. Como pode ser observado na **Figura 2**, os sinais da entrada, representados pelo vetor $x = [x_1, x_2, x_3, \dots, x_n]$ ao chegarem ao neurônio artificial, são multiplicados pelos respectivos pesos, represen-

tados pelo vetor $w = [w_1, w_2, w_3, \dots, w_n]$, gerando o valor z , comumente denominado potencial de ativação, de acordo com a expressão:

$$z = \sum_{i=1}^n x_i * w_i + b \quad (1)$$

O termo adicional b provê um grau de liberdade a mais na equação, que não é afetado pelos valores de entrada. O Valor de b correspondendo tipicamente ao bias. O valor z passa, por fim, por uma função matemática (σ), com a característica de ser não linear, responsável por limitar o valor a um certo intervalo de interesse e que produzirá o valor final de saída y do neurônio artificial. A função σ é conhecida como Função de Ativação e é a grande responsável por decidir se o neurônio será ou não utilizado na rede neural.

2.1.2.2 Função de Ativação

As Funções de Ativação são elementos fundamentais na estrutura de uma rede neural. Como já mencionado na seção 2.1.2.1, elas são responsáveis por definir se o neurônio será ou não ativado, ou seja, se ele será ou não utilizado para definir a Aprendizagem da rede. De modo geral, a Função de Ativação é uma transformação não linear aplicada ao sinal de entrada. Esta saída processada é, então, enviada para a próxima camada de neurônios como nova entrada. Quando não temos a função de ativação, os pesos e bias simplesmente fazem uma transformação linear. Uma equação linear é simples de resolver, mas é limitada na sua capacidade de resolver problemas complexos. Uma rede neural sem função de ativação é essencialmente apenas um modelo de regressão linear simples.

Como exemplos de Funções de Ativação amplamente utilizadas podem-se citar:

- **Sigmoide** - $Sigmoide(x) = \frac{1}{(1+e^{-x})}$;
- **Tangente hiperbólica** - $Tanh(x) = 2 * Sigmoide(2x) - 1$;
- **Softmax** - $\sigma(x)_j = e^{-x_j} \sum_{k=1}^k e^{x_k}$ para todo j, \dots, k
- **ReLU (do acrônimo em inglês *Rectified Linear Unit*)** - $f(x) = \max(0, x)$

A escolha de qual Função de Ativação utilizar, depende das propriedades do problema que se pretende resolver. Não há uma regra universal. Entretanto, alguns fatores podem ser úteis, de acordo com a Academy (2021). Funções Sigmóide e suas combinações geralmente são utilizadas em classificadores. Funções Sigmóide e Tanh (Tangente Hiperbólica) às vezes são evitadas devido ao problema de Dissipação do Gradiente (seção 2.1.2.6). Portanto, quando há a necessidade de utilizar iteratividade, é conveniente escolher outras funções. E, por fim, a função ReLU é uma função de ativação geral e é usada na maioria dos casos atualmente. De modo geral, é a primeira a ser testada.

2.1.2.3 Perceptron de Múltiplas Camadas

Como mencionado na seção 2.1.2.1, um único perceptron pode ser utilizado para criar um classificador para separar dados linearmente separáveis. Entretanto, boa parte dos problemas encontrados no cotidiano não podem ser representados desse modo, logo são considerados Não linearmente separáveis. A **Figura 3** apresenta um exemplo de Dados linearmente separáveis (à esquerda) e Não linearmente separáveis (à direita).

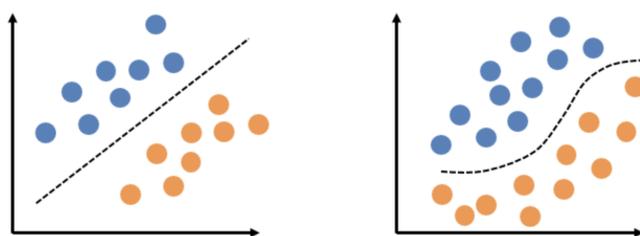


Figura 3 – Dados linearmente separáveis (à esquerda) e Não linearmente separáveis (à direita)

Para solucionar esse problema, a solução pode ser a combinação de vários Perceptrons conectados em uma organização semelhante a uma rede. Como essa solução é união de vários modelos matemáticos inspirados no neurônio humano, essa rede é conhecida como Rede Neural Artificial. A organização de vários Perceptrons totalmente conectados em uma rede é denominada Perceptron de Múltiplas Camadas, também conhecido pelo termo em inglês *Deep Feedforward Networks* ou *Feedforward Neural Networks*.

Perceptron de Múltiplas Camadas são Modelos de Aprendizado Profundo por excelência (GOODFELLOW; BENGIO; COURVILLE, 2016). O objetivo desse tipo de arquitetura é aproximar alguma função f que mapeie os dados de entrada e devolvam alguma saída. A estratégia para solucionar esse problema é semelhante a outros classificadores, ou seja, para um $y = f(x)$ mapeia-se uma entrada x para uma categoria y . O que diferencia a Rede Neural Artificial de outro classificador comum é a estratégia de mapeamento utilizada. A **Figura 4** mostra uma representação visual de uma Rede Perceptron de Múltiplas Camadas.

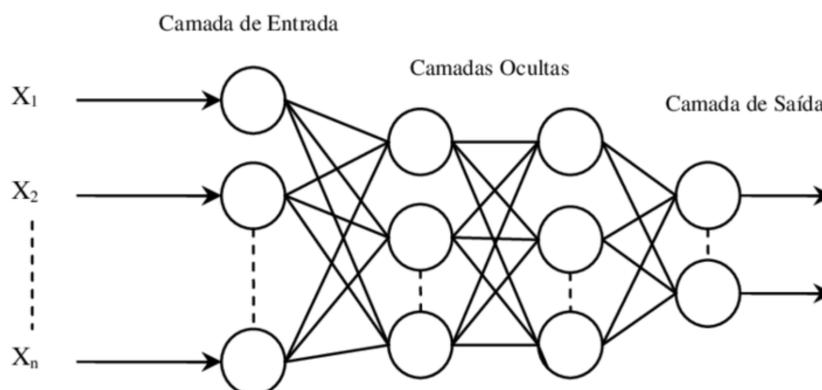


Figura 4 – Representação de uma Rede Perceptron de Múltiplas Camadas (SOBREIRO *et al.*, 2008)

A **Figura 4** é um grafo direcionado onde cada nó representa um neurônio. Basicamente, quatro componentes são necessários em uma arquitetura de Redes Neurais Artificiais. Da esquerda para a direita, o primeiro componente representa os dados de entrada do problema, exemplificados pelos valores X_1, X_2, \dots, X_n . O segundo, representa a camada de entrada da rede, ou os neurônios artificiais escolhidos para receber os valores de entrada. O terceiro componente representa as camadas ocultas. Por conta da existência dessas camadas ocultas que as Redes Neurais Artificiais são também conhecidas como Redes Neurais Profundas (do inglês, *Deep Learning*). E, por fim, a última camada é a camada representa a saída esperada pelo problema. Por exemplo, se essa Rede Perceptron de Múltiplas Camadas for utilizada para realizar uma classificação binária, de modo geral, teremos dois neurônios nessa camada.

2.1.2.4 Aprendizado nas Redes Neurais Artificiais

O processo de Aprendizado em uma Rede Neural Artificial consiste em passar os dados pelas camadas da rede de modo iterativo. Ao final de cada iteração contabiliza-se, então, a discrepância entre os valores que chegam para a camada de saída e os valores que deveriam ser, de fato, observados. Reinicia-se a iteração tantas vezes quanto necessário de modo a diminuir a discrepância entre os valores.

O processo de passar os dados pela rede é conhecido pelo seu termo em inglês *feedforward* e o processo de atualização dos pesos e bias após a comparação é conhecido como *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986). A função que é utilizada para comparação dos resultados dos dados ao passar pela rede é denominada Função de Custo e pode ser definida por:

$$c(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2 \quad (2)$$

onde, w representa todos os pesos na rede, b todos os bias, n é o número total de entradas de treinamento e a é o vetor de saídas da rede quando x é entrada. A

notação $\|v\|$ indica a função de comprimento usual para um vetor v . Na Função de Custo c observa-se que $c(w,b)$ não é negativo, pois o resultado da subtração é elevado ao quadrado. Além disso, o custo $c(w,b)$ torna-se pequeno ($w,b \approx 0$) quando $y(x)$ se aproxima da saída a para todas as entradas de treinamento x . Pode-se afirmar que o algoritmo de treinamento atinge seu objetivo com sucesso se ele encontra pesos e bias de forma que $c(w,b) \approx 0$.

O trabalho de treinamento é, em essência, mas não somente, a busca pela otimização da Função de Custo com intuito de encontrar os valores w e b tais que $c(w,b) \approx 0$. Um dos algoritmos que pode ser utilizado para essa otimização é o Algoritmo de Descida do Gradiente. A Descida do Gradiente é um recurso matemático utilizado computacionalmente para otimizar funções complexas de modo iterativo. Dada alguma função arbitrária, o objetivo, portanto, é encontrar o mínimo espacial da respectiva função.

2.1.2.5 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (ou, simplesmente CNNs, acrônimo do termo em inglês *Convolutional Neural Networks*), são tipos especializados de Rede Neurais Artificiais para processamento de dados que possuem uma topologia conhecida como grade (do inglês, *grid-like*) (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse tipo de rede é comumente aplicada a problemas que envolvem séries temporais e textos, que podem ser considerados grades unidimensionais ou vetoriais. Outra aplicação bastante comum ocorre no processamento de imagens, que são casos em que os dados são grades de 2 ou 3 dimensões, também conhecidas como grades matriciais. A aplicação de CNNs para predição e processamento de imagens também é conhecida como Visão Computacional.

A convolução é um tipo específico de operação linear. As CNNs, portanto, são simplesmente Redes Neurais Artificiais que usam a convolução dentre os cálculos no *feedforward*, ao invés da multiplicação geral de matrizes, como ocorre nas Redes Neurais Perceptron de Múltiplas Camadas. Esse processo permite com que as CNNs sejam capazes de obter eficientemente respostas locais das grades aplicando convolução, além de diminuir a dimensão de uma camada subsequente comparada a sua camada posterior.

Uma CNN é capaz de capturar com sucesso, por exemplo, as dependências espaciais em uma imagem através da aplicação de filtros (do inglês, *kernels*). Ao mesmo tempo é possível usar as CNNs para diminuição de dimensionalidade, aplicando a operação matemática adequada é possível também extrair ou evidenciar características importantes nas imagens, como bordas, contrastes de cores, etc. A operação de Convolução pode ser observada, a título de exemplo, na **Figura 5**, onde a **Matriz A** (5x5) é percorrida por um filtro (2x2) e a cada deslocamento a média

aritmética dos valores abarcados pelo filtro são calculados. Ao final do processo, ter-se-ia uma nova matriz (4x4) com os valores obtidos pelas operações convolucionais.

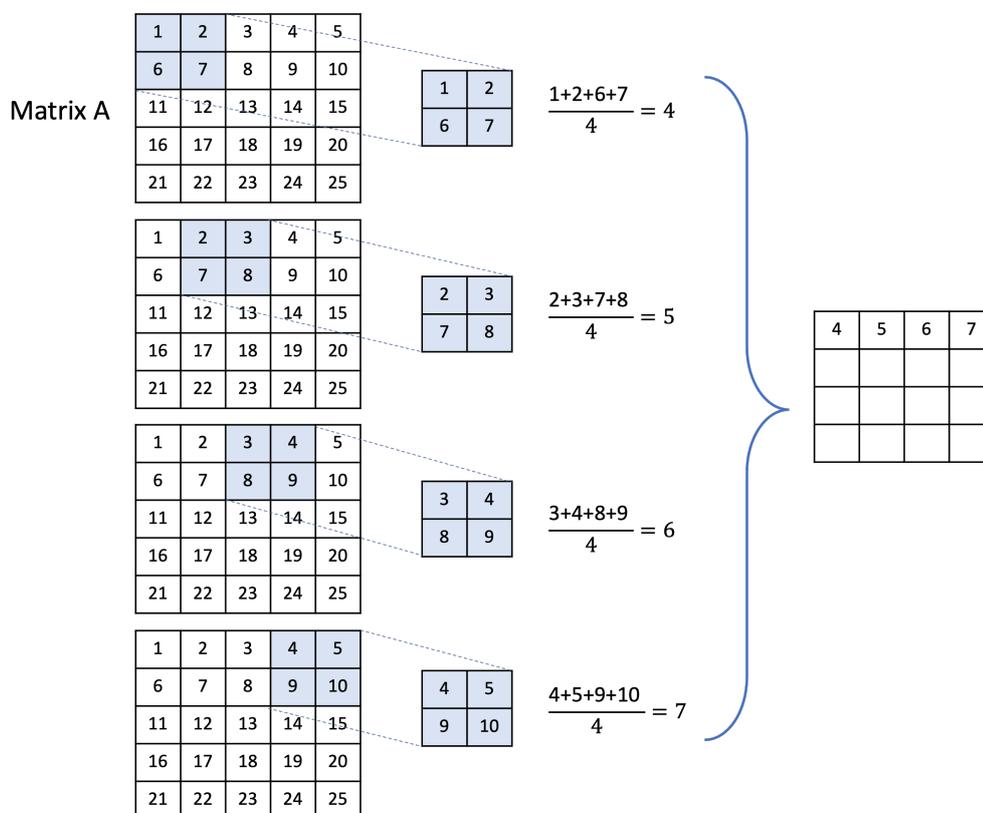


Figura 5 – Representação de uma operação de convolução

Textos podem ser considerados matrizes de uma dimensão, também conhecidas como vetores. Portanto, as operações convolucionais também podem ser aplicadas para problemas de Aprendizado de Máquina que envolvam textos.

2.1.2.6 Redes Neurais Recorrentes

Redes Neurais Recorrentes (do inglês, *Recurrent Neural Network* ou simplesmente (RNN), diferentemente das Redes Perceptrons de Múltiplas Camadas (2.1.2.1) e das Redes Neurais Convolucionais (2.1.2.5), são estruturas que permitem que seus neurônios artificiais, além da conexão tradicional *feedforward*, comuniquem-se recorrentemente consigo mesmos. A **Figura 6** exemplifica visualmente a recorrência do neurônio conectando a ele mesmo na RNN (representada à esquerda) evidenciando a principal diferença para uma Rede Perceptron de Múltiplas Camadas *feedforward* (à direita). Dada a sua capacidade de recorrência, As RNNs, são famílias de Redes Neurais Artificiais que podem ser utilizadas com eficiência para processamento de dados sequenciais como Séries Temporais e Textos.

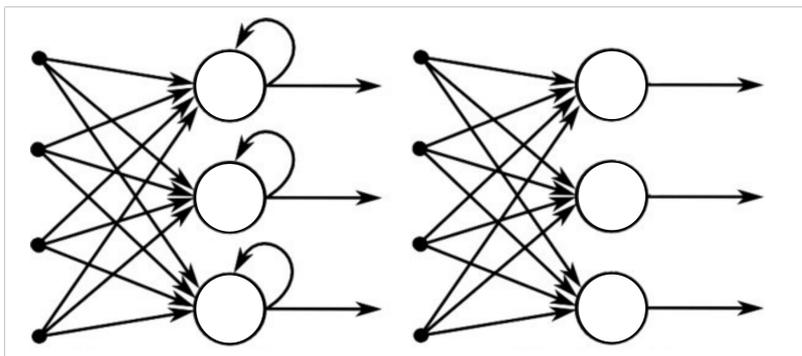


Figura 6 – Rede Neural Artificial *feedforward* (à direita) e Rede Neural Recorrente (à esquerda)

A informação sequencial que passa pelos neurônios da rede que possuem recorrência é preservada no estado oculto da rede. Essa etapa pode repetir-se várias vezes à medida em que a informação avança em cascata para afetar o processamento de cada novo exemplo. Essas correlações entre eventos são separadas em vários momentos temporais e são denominadas "dependências de longo prazo", porque um evento no tempo depende e é uma função de um ou mais eventos que vieram antes. Essa dependências de longo prazo tem um funcionamento semelhante a uma memória, o que torna as RNNs eficientes para solucionar pro com dados sequenciais.

2.1.3 Métricas de Avaliação de Modelos

Ao desenvolver projetos de Aprendizado de Máquina é fundamental a escolha da métrica com a qual o modelo de Aprendizado será avaliado. O valor obtido por intermédio dessas métricas, de modo geral, nos mostrará a qualidade como a qual o modelo faz as previsões esperadas. Existem diversas métricas que podem ser aplicadas em contextos diferentes e algumas são mais adequadas que outras dependendo da técnica de Aprendizado escolhida.

Quando o problema de Aprendizado de Máquina é Supervisionado, via de regra, as métricas são divididas em métricas que avaliam modelos de Regressão e modelos de Classificação. Nas seções seguintes são apresentadas as principais para cada tipo de modelo.

2.1.3.1 Regressão

A avaliação da performance de um algoritmo de Aprendizado de Máquina aplicado a um determinado conjunto de dados é realizada por meio da mensuração do quão bem as previsões reproduzem o valor observado para a resposta de interesse. Portanto, é preciso quantificar o quanto o valor predito (y'_i) se aproxima de seu valor observado (y_i) (JAMES *et al.*, 2013). No contexto de regressão, o erro quadrático médio

(EQM ou, MSE do inglês *mean squared error*) é uma métrica utilizada com frequência, e é dado por:

$$EQM = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (3)$$

A raiz quadrada extraída de EQM resulta em um valor na mesma unidade que os dados originais. Sua interpretação refere-se à distância média entre os valores observados e os preditos pelo modelo. O EQM será pequeno se as respostas preditas forem muito próximas das observadas e será grande se, para algumas observações, a resposta predita e a observada diferirem significativamente.

Adicionalmente, como o objetivo final da modelagem preditiva é obter previsões acuradas em novos dados, a performance preditiva real do modelo deve ser avaliada a partir da mensuração de seu EQM em dados que o algoritmo de treinamento não conhece (JAMES *et al.*, 2013). Esses dados são conhecidos como dados de teste e a técnica de separar aleatoriamente os dados em teste e treino é conhecido pelo termo em inglês *Cross Validation*.

2.1.3.2 Classificação

Um modelo de classificação tem como objetivo decidir a qual classe uma nova observação pertence dentre as classes possíveis. A avaliação de um modelo de classificação é feita a partir da comparação entre as classes preditas pelo modelo e as classes verdadeiras de cada exemplo. Todas as métricas de classificação têm como objetivo comum medir quão distante o modelo está da classificação ótima, porém existem diversas formas de realizar esse trabalho.

2.1.3.3 Matriz de Confusão

Uma forma bastante simples de visualizar a performance de um modelo de classificação é através de uma matriz de confusão, **Figura 7**. Essa matriz dispõe o cruzamento dos valores segregados pelas classes nas quais eles foram preditos e os compara com as classes nas quais esses valores foram, de fato, observados. Genericamente os valores são separados em valores positivos, ou representados pelo número 1, e negativos, representados pelo número 0. Os valores que foram observados com a classe positiva e foram preditos também com essa mesma classe recebem na matriz de confusão o nome de Verdadeiros Positivos (ou, simplesmente VP). Os valores que foram observados com a classe positiva, entretanto foram preditos como negativos, são conhecidos como Falsos Negativos (FN). Os valores que foram observados com a classe negativa e foram preditos também na classe negativa, são conhecidos como Verdadeiros Negativos (VN). Por fim, Os valores que foram observados com a classe negativa e foram preditos na positiva, são conhecidos como Falsos Positivos (FP).

A maioria das métricas utilizadas para avaliar modelos de classificação analisam os resultados relativos dentre os valores dispostos na matriz de confusão.

		Valores Observados	
		1	0
Valores Preditos	1	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	0	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Figura 7 – Matriz de Confusão

2.1.3.4 Acurácia

A Acurácia (ou *Accuracy* em inglês) apresenta quantos dos exemplos foram, de fato, classificados corretamente, independente da classe. Essa métrica é definida pela razão entre o que o modelo acertou e o número total de exemplos. A Acurácia pode ser calculada pela equação (4):

$$Acuracia = \frac{VP + VN}{VP + VN + FP + FN} \quad (4)$$

Apesar da Acurácia ser uma métrica simples, de fácil manuseio e diretamente interpretável, ela muitas vezes não é adequada para alguns tipos de problemas. Por exemplo, pode-se considerar um modelo de classificação criado para discernir entre pacientes que possuem câncer, portanto pertencem a classe negativa, e pacientes que não possuem a doença, pertencentes a classe positiva. Dentre todos os pacientes observados, totalizando 1000, 990 não possuem câncer (classe positiva). Caso o modelo seja ingênuo e classifique todos os exemplos como positivos, ele obterá uma acurácia de 99%. Esse valor num primeiro momento aparenta uma qualidade muito elevada do modelo. No entanto, é trivial perceber que a avaliação é inadequada. Pois, embora acerte na maioria absoluta dos casos, não é útil para avaliar se as predições ocorrem de modo assertivo.

O exemplo acima citado, é um típico exemplo onde as classes para o problema analisado estão desbalanceadas, ou seja, a representatividade de uma classe sobrepõe significativamente a outra. Nos casos opostos, em que se tem os dados balanceados, a Acurácia pode ser utilizada sem maiores prejuízos, desde que as classes também sejam igualitárias do ponto de vista de classificação. Para mitigar problemas

com classes desbalanceadas, pode-se utilizar métricas como Precisão, Revocação ou a combinação de ambas, que é a Score F1.

2.1.3.5 Precisão

A Precisão (ou *Precision* em inglês) é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e o total de exemplos classificados como positivos. A Precisão dá maior ênfase para os erros classificados como Falso Positivo (FP). A Precisão pode ser calculada pela equação (5):

$$Precisao = \frac{VP}{VP + FP} \quad (5)$$

2.1.3.6 Revocação

Em oposição a Precisão, a Revocação, (ou *Recall* em inglês), também conhecida como sensibilidade ou taxa de verdadeiro positivo, dá maior ênfase para os erros classificados como falsos negativos. Esta métrica é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos. A Revocação pode ser calculada pela equação (6):

$$Revocacao = \frac{VP}{VP + FN} \quad (6)$$

2.1.3.7 Score F1

A métrica Score F1, também conhecida pelo termo em inglês *F-measure*, leva em consideração tanto a Precisão quanto a Revocação. Ela é definida pela média harmônica entre as duas. A Score F1 pode ser calculada pela equação (7):

$$ScoreF1 = 2 * \frac{Precisao * Revocacao}{Precisao + Revocacao} \quad (7)$$

Uma das características da média harmônica é que se a Precisão ou a Revocação forem valores muito baixos ou próximos de zero, o F1 score também será baixo. Desta forma, para que o F1 score seja alto, tanto a Precisão como a Revocação devem ser valores elevados. Esse fato torna F1 score umas métricas mais gerais para avaliar uma modelo de classificação.

2.1.3.8 Curva ROC

Alguns modelos de classificação contabilizam simplesmente a classe na qual o novo dado foi classificado, 0 ou 1. Entretanto, existem situação em que é desejável obter a probabilidade de um novo dado pertencer a uma das classes possíveis. Nesse caso, é necessário definir um limiar de classificação. Acima deste limiar, um exemplo

é classificado como positivo, caso contrário, é classificado como negativo. O limiar de classificação influencia o valor das métricas mencionadas anteriormente (Acurácia, Precisão, etc), e sua escolha deve levar em consideração o custo de cada erro.

A Curva ROC (do inglês, *Receiver Operating Characteristic*) pode ser utilizada para avaliar a performance de um classificador para diferentes limiares de classificação. Ela é construída medindo a Taxa de Falsos Positivos (TFP) e a Taxa de Verdadeiros Positivos (TVP) para cada limiar de classificação possível, conforme as equações (8) e (9).

$$TFP = \frac{VP}{VP + FN} \quad (8)$$

$$TVP = \frac{FP}{FP + VN} \quad (9)$$

A Curva ROC é, então, visualizada por meio de um gráfico, que mostra visualmente o compromisso entre falsos positivos e verdadeiro positivos na escolha do limiar. Quanto mais alto o limiar, maior é taxa de verdadeiros positivos (VP), porém a taxa de falsos positivos (FP) também será maior. No caso extremo em que todos os exemplos são colocados na classe positiva, ambas as taxas chegam a 100%, enquanto no outro extremo, ambas ficam em 0%. Quanto mais próxima a curva estiver do canto superior esquerdo, melhor será a predição do modelo, dado que ele teria 100% de TVP e 0% de TFP. A área sob a curva ROC (do inglês, *Area Under the Curve*, ou simplesmente AUC) pode ser utilizada como métrica de qualidade de um modelo, dado que quanto mais próxima a curva estiver do canto superior esquerdo, maior será a área sob a curva e melhor será o modelo. Uma vantagem desta métrica é que ela não é sensível ao desbalanceamento de classes, como ocorre com a Acurácia. A **Figura 8** mostra um exemplo de uma Curva ROC.

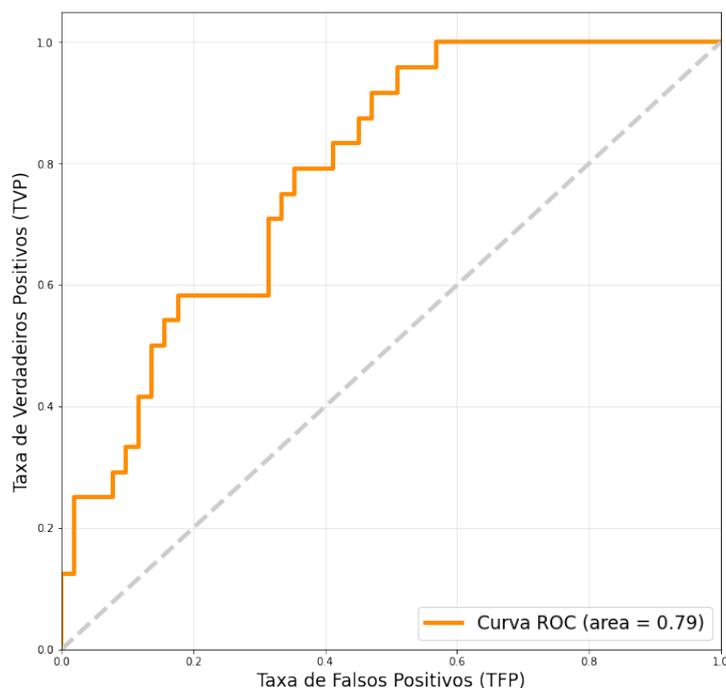


Figura 8 – Curva ROC

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural, ou simplesmente PLN, é uma área multidisciplinar que engloba Matemática, Ciência da Computação, Inteligência Artificial e Linguística com o objetivo de analisar problemas relacionados à geração, interpretação e manipulação da linguagem humana. Devido à grande quantidade de textos disponíveis na internet, blogs, portais de notícias, redes sociais, etc., a PLN atualmente recebe muita atenção dos pesquisadores e também se tornou um campo orientado a dados (do inglês, *data driven*).

Pode-se dizer que Processamento de Linguagem Natural (PLN) é uma área em franco crescimento na comunidade científica e também em aplicações de mercado. Por conta desse fato, existe um número significativamente grande de conceitos marginais à PLN. As seções seguintes são responsáveis por apresentar os principais ou, pelo menos, os necessários para embasamento teórico da proposta apresentada por esse trabalho.

2.2.1 Modelos de Linguagem

Entende-se linguagem, dentro do contexto desse trabalho, como um conjunto estruturado ou não estruturado de símbolos visando algum tipo de interação entre interlocutores, sejam eles humanos e/ou máquinas. Existem diversas representações que podem ser consideradas linguagens, duas delas recebem maior atenção quanto utiliza-se o computador como ferramenta de análise; Linguagens Formais e Linguagens

Naturais.

As linguagens de programação como Java ou Python possuem modelos precisamente definidos e podem, eventualmente, ser caracterizadas pela sequencialidade dos dados. Vejamos o exemplo do código “*print(2 + 2)*” que é um programa válido na linguagem Python, enquanto que “*2) + (2print*”, que possui os mesmos caracteres em ordem diferente, não é. Levando em consideração o exemplo acima, é fácil projetar que há infinitas possibilidades de programas em linguagens de programação, logo, enumera-las, pode não ser conveniente. Uma alternativa é estabelecer um conjunto de regras para serem seguidas por todos os usuários dessa linguagem, ou estabelecer formalmente tais regras. Linguagens com essas características são ditas Linguagens Formais.

Por outro lado, existem linguagens como o inglês, espanhol ou português, que não podem ser caracterizadas como um conjunto de sentenças definitivas, pois, além de serem intrinsecamente ambíguas, são muito grandes e estão em constante mutação ao longo do tempo. Tais linguagem são conhecidas como Naturais (RUSSELL; NORVIG, 2009).

Todos concordam, por exemplo, que “O banco da praça é branco” é uma sentença em português, mas as pessoas podem discordar sobre a gramaticalidade e o significado da sentença, uma vez que “banco” pode ser interpretado como uma agência bancária ou, simplesmente, como um assento. Portanto, é mais frutífero definir um modelo de linguagem natural como uma distribuição de probabilidade sobre sentenças em vez de um conjunto definitivo de regras. Ou seja, em vez de perguntar se uma sequência de palavras é ou não membro do conjunto que define o idioma, perguntamos por:

$$P(S = \textit{palavras}) \tag{10}$$

Ou, qual a probabilidade de que uma sentença aleatória pertencer ao grupo palavras (RUSSELL; NORVIG, 2009).

2.2.2 Modelos *n-gram*

Um texto escrito em um idioma qualquer é composto basicamente por caracteres; letras, dígitos, pontuações e espaços (e caracteres mais exóticos em alguns casos). Dessa forma, um modelo de linguagem simples para esse caso é uma distribuição de probabilidade sobre a sequências de caracteres. Escrevemos $P(c_1 : n)$ para a probabilidade de uma sequência de n caracteres, de c_1 até c_n . Uma sequência de símbolos escritos de comprimento n é chamada de *n-gram* (da raiz grega para escrita de letras). Um modelo de distribuição de probabilidade de n sequências de letras é, então, chamado de Modelo *n-gram* (RUSSELL; NORVIG, 2009). Importante

destacar que podemos ter modelos *n-grams* com sequências de palavras, sílabas ou outro conjunto de unidades, não apenas com caracteres.

Podemos definir o modelo *n-gram* como uma cadeia de Markov¹ de ordem $n - 1$. Um modelo *trigram* (apenas 3 grams) pode ser escrito da seguinte forma:

$$P(c_j|c_{1:n-1}) = P(c_j|c_{i-2:i-1}) \quad (11)$$

2.2.3 Word Embeddings

Para utilizar métodos computacionais para processamento e análise de textos é necessário convertê-los para um formato numérico adequado à respectiva técnica que será utilizada. Nos primeiros trabalhos de PLN o modelo de espaço vetorial era normalmente utilizado, no qual os valores das palavras individuais eram estimados pelo método de ponderação (baseado na frequência com que as palavras aparecem) formando o vetor de características de cada documento. Essa técnica é conhecida como saco de palavras (do inglês, *Bag of Words*, ou simplesmente, BOW) (ZHONG; LI, W., 2019).

No entanto, dada a quantidade atual de dados, o tamanho dos vocabulários utilizados e o fato de que BOW não é eficiente na captura de informações semânticas, os pesquisadores investigaram outras formas de representar documentos e palavras em um vetor denso, de baixa dimensão e semanticamente significativos (ZHONG; LI, W., 2019). Um método que recentemente atraiu atenção é a hipótese de distribuição, que assume que palavras contextualmente semelhantes têm semântica semelhante (WANG; ZHOU; JIANG, 2020). Os métodos de representação de palavras baseados na hipótese de distribuição são divididos principalmente em três tipos (WANG; ZHOU; JIANG, 2020):

1. Representação baseada em matriz, também conhecida como representação distributiva;
2. representação distribuída baseada em *cluster*;
3. representação distribuída baseada em redes neurais, também conhecidas como *Word Embeddings*.

Dentre as técnicas de Word Embeddings, uma que tem se destacado por obter resultados significativos é o *Word2vec* (MIKOLOV *et al.*, 2013), que é um algoritmo de Aprendizado de representação distribuída concebido para aprender representações vetoriais densas contínuas para palavras em espaço vetorial de baixa dimensão. O Algoritmo consiste no uso conjunto de dois modelos relacionados: o modelo contínuo de saco de palavras (do inglês, *Continuous Bag Of Words*, ou simplesmente, CBOW)

¹ Em matemática, uma cadeia de Markov é um caso particular de processo estocástico com estados discretos em que a distribuição de probabilidade do próximo estado depende apenas do estado atual e não da sequência de eventos que precederam ou que vieram depois.

e o modelo *Skip-Gram*. O CBOW prevê a palavra atual a partir de suas palavras de contexto circundantes em uma frase dentro de uma janela centrada na palavra atual. Enquanto o modelo *Skip-Gram*, prevê as palavras de contexto circundantes de uma determinada palavra em uma frase dentro de uma janela simétrica.

Uma característica importante nesse tipo de abordagem é que os vetores de palavras no espaço vetorial estão próximos uns dos outros quando as palavras correspondentes são semanticamente semelhantes entre si. Isso nos dá o benefício de poder inferir palavras semanticamente semelhantes comparando a distância entre os vetores de palavras criados por *Word2vec*. Então, usando *Word2vec* é possível criar uma *Embedding Matrix* que mapeia cada palavra para seu vetor correspondente.

2.3 CLASSIFICAÇÃO DE TEXTOS

A classificação de textos é uma técnica de Processamento de Linguagem Natural que utiliza Aprendizado de Máquina para encontrar padrões entre os textos e seus rótulos associados para discriminação e classificação futura de novos textos para seus respectivos rótulos. Dentre os problemas que classificação de textos resolve podemos destacar análise de sentimentos, categorização de documentos e detecção de spam.

De acordo com Liu e Guo (2019) Redes Neurais Artificiais Profundas têm alcançado resultados notáveis também para essa atividade. Duas arquiteturas específicas de Redes Neurais Artificiais Profundas são aplicados com bons resultados; Redes Neurais Convolucionais (CNNs) e Redes Neurais Recorrentes (RNNs). As CNNs são capazes de aprender uma resposta local a partir de dados temporais ou espaciais, mas não têm a capacidade de aprender correlações sequenciais de modo satisfatório. Em contrapartida, as RNNs são especializadas em modelagem sequencial, mas não podem extrair recursos em paralelo. De acordo com Liu e Guo (2019) ambos os modelos, quando usados em conjunto podem apresentar resultados significativos na classificação de textos.

2.4 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO

A fundamentação teórica tem como objetivo proporcionar ao leitor um contexto abrangente para o entendimento do tópico de pesquisa em questão. É por intermédio desse capítulo que há um aprofundamento das teorias, modelos e perspectivas que sustentam o respectivo estudo. O intuito é explorar as contribuições mais relevantes, as correntes de pensamento mais influentes e as descobertas que moldam a pesquisa. Dentro desse contexto, para alcançar o respectivo objetivo, esse presente capítulo dividiu-se em três seções; na **Seção 2.1**, foram explorados os conceitos relacionados a Aprendizado de Máquina, na **Seção 2.2** foram explorados os conceitos relacionados a Processamento de Linguagem Natural e, por fim, na **Seção 2.3** foram explorados os

conceitos relacionados Classificação de Textos, unindo os conceitos das duas seções anteriores.

3 REVISÃO DOS TRABALHOS CORRELACIONADOS

Esse Capítulo apresenta uma revisão dos trabalhos relacionados a aplicação de Aprendizado de Máquina, dentre outras técnicas, para a previsão do desempenho de alunos. O objetivo principal deste capítulo é explorar pesquisas anteriores que abordaram tópicos semelhantes, identificando as principais abordagens, métodos e contribuições no contexto da análise preditiva para o desempenho estudantil. Ao final deste capítulo é exposto um panorama sobre as pesquisas anteriores relacionadas ao tema, para viabilizar a identificação de lacunas e oportunidades para a contribuição deste trabalho.

O **Capítulo 3** está organizado em três seções. Na primeira, **Seção 3.1**, abordar-se-á a estratégia de Aprendizado de Máquina aplicada a Classificação de Textos, uma vez que o intuito do presente trabalho é a realização de previsões a partir dos textos postados pelos alunos. Na segunda, **Seção 3.2**, serão analisados trabalhos recentes envolvendo aplicação de Aprendizado de Máquina para a previsão do desempenho estudantil especificamente. Na terceira e última, **Seção 3.3**, serão analisados comparativamente os trabalhos listados nas seções anteriores, com intuito de identificar as lacunas que a proposta desse presente trabalho pretende preencher.

3.1 CLASSIFICAÇÃO DE TEXTOS

O problema de classificação de textos utilizando Aprendizado de Máquina é amplamente discutido na comunidade acadêmica, um bom resumo sobre os principais tópicos e técnicas utilizadas pode ser encontrado no *survey* publicado pelos autores Kowsari *et al.* (2019). Um trabalho que se destacou recentemente foi discutido no artigo Liu e Guo (2019) que propõe uma nova arquitetura de *Deep Learning* para solucionar os problemas relacionadas com a classificação de textos. Nesse trabalho os autores utilizaram em conjunto algumas técnicas bem estabelecidas por apresentarem bons resultados e nomeou a arquitetura final como *Bidirectional LSTM with attention mechanism and convolutional layer for text classification* (ou, simplesmente AC-BiLSTM). A ideia dessa nova arquitetura proposta é utilizar o melhor que cada técnica dispõe para que, ao final, obtenham-se resultados que as técnicas utilizadas individualmente não conseguiriam apresentar. A **Figura 9** apresenta um diagrama para a arquitetura proposta por (LIU; GUO, 2019).

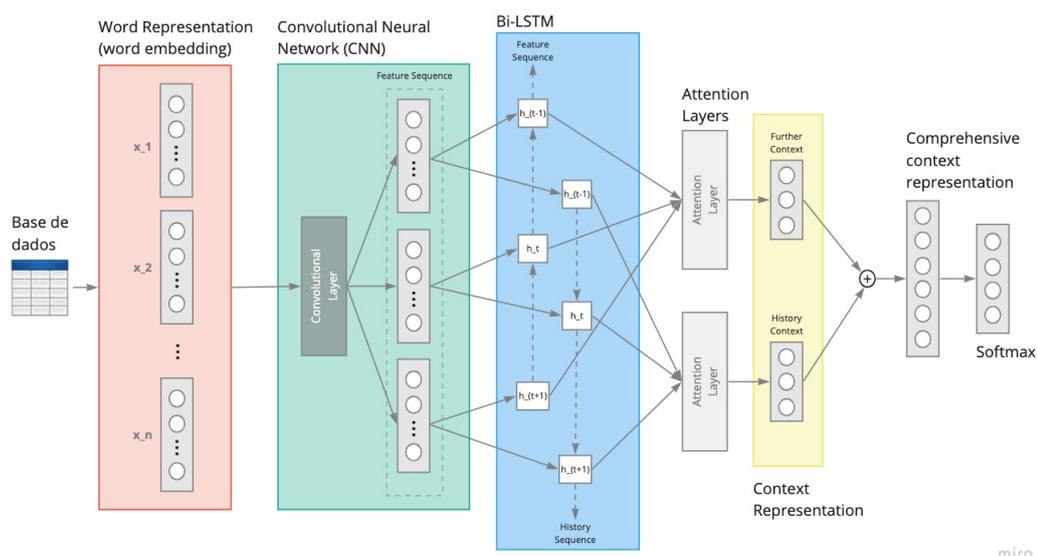


Figura 9 – Arquitetura utilizada pelos autores (LIU; GUO, 2019) para classificação de textos

A primeira camada (da esquerda para direita), área hachurada em vermelho, representa a camada de entrada para arquitetura AC-BiLSTM. Trata-se da parte da rede neural onde são tratados os textos e as *word embeddings*, que são os equivalentes numéricos e probabilísticos das palavras obtidas com algoritmo *skip-gram*, com intuito de adequá-las para alimentar os pesos das camadas de Rede Neural e fazer o treinamento do modelo de Aprendizado de Máquina.

Na área hachura em verde, estão representados os filtros convolucionais unidimensionais, que atuam na extração de características distintas das frases e reduzem as dimensões dos dados de entrada.

Na área hachurada em azul, está representada a arquitetura Bi-LSTM, que é usado nesse contexto para extrair as informações contextuais dos recursos produzidos pela camada convolucional. Existem duas representações de LSTM, uma para percorrer os vetores da esquerda para a direita, para dar ênfase aos dados históricos, a serem analisados como normalmente é realizada a leitura de uma frase no idioma em inglês, e outra para os dados percorridos na direção contrária, da direita para esquerda, para dar ênfase na análise dos vetores na direção contrária a leitura.

Na área hachurada em cinza, estão representadas as camadas de Mecanismos de Atenção (do inglês, *Attention Mechanism*) que são empregadas para dar um foco às informações extraídas da camada oculta anterior, diminuindo a possibilidade e dissipação do gradiente. Como pode ser observado, foram utilizadas duas camadas de Mecanismos de Atenção, a exemplo de como foi feito nas arquiteturas LSTM. Por fim, os recursos extraídos com as camadas de Mecanismos de Atenção são agrupados e classificados utilizando uma função *softmax*, representado pela área hachurada em amarelo.

Seis conjuntos de dados rotulados foram utilizados para avaliar a nova arquitetura. Os resultados das seguem apresentados na **Figura 10**. Percebe-se claramente que a arquitetura proposta em Liu e Guo (2019) é mais assertiva, levando-se em consideração como métrica de avaliação a acurácia, do que as utilizadas pelos autores como *baseline*, o que justifica a adoção dessa técnica para o presente trabalho.

Model	MR	IMDB	SST-1	SST-2	RT-2k	Subj
SVM	–	89.2	40.7	79.4	87.4	91.7
MNB	79.0	86.6	–	–	85.9	93.6
NBSVM	79.4	91.2	–	–	89.5	93.2
RAE	77.7	–	43.2	82.4	–	–
MV-RNN	79.0	–	44.4	82.9	–	–
RNTN	–	–	45.7	85.4	–	–
Paragraph-Vec	–	–	48.7	87.8	–	–
DCNN	–	–	48.5	86.8	–	–
CNN-static	81.0	–	45.5	86.8	–	93.0
CNN-non-static	81.5	–	48.0	87.2	–	93.4
CNN-multichannel	81.1	–	47.4	88.1	–	93.2
DRNN	–	–	49.8	86.6	–	–
Multi-task LSTM	–	–	49.6	87.9	–	–
Tree-LSTM	–	–	50.6	86.9	–	–
P-LSTM	–	91.5	–	–	89.3	93.8
C-LSTM	–	–	49.2	87.8	–	–
LSTM	80.1	87.0	48.0	86.4	86.7	91.3
BiLSTM	80.3	87.9	48.4	88.0	87.2	92.3
AC-BiLSTM	83.2	91.8	48.9	88.3	93.0	94.0

Figura 10 – Tabela com a comparação dos resultados obtidos pelo trabalho (LIU; GUO, 2019)

3.2 PREVENDO DESEMPENHO DE ALUNOS

Nessa seção serão expostos e analisados os principais trabalhos que formam o Estado da Arte especificamente para a previsão do desempenho de Alunos.

3.2.1 Multi-view hypergraph neural networks (MVHGNN)

Mengran Li *et al.* (2022) propuseram uma Rede Neural de hiper grafos de visão múltipla, denominada MVHGNN (do acrônimo em inglês *Multi-view hypergraph neural networks*), para prever o desempenho acadêmico dos alunos da Universidade Tecnológica de Beijing, China, (BJUT-STUD). O objetivo desse trabalho é utilizar hiper grafos para capturar informações semânticas que ficam ocultas no comportamento dos alunos, transformá-las em variáveis preditoras para uma Rede Neural Artificial Profunda e, então, fazer previsões do desempenho dos alunos com alta precisão.

Para treinar a rede foram utilizados dados comportamentais dos alunos, tais como entrada, saída e quantidade de idas a biblioteca, banheiro, ginásios de esportes e refeitório. O método usa um hipergrafo de múltiplas visualizações para representar a associação entre os alunos e seus respectivos comportamentos dentro da Universidade. As informações semânticas contidas nos comportamentos são identificadas e devidamente representadas e os relacionamentos são gerados pelo módulo CAT (do acrônimo em inglês *Cascaded Attention Transformation*), que atribui diferentes pesos a cada hipergrafo para obter o mapa de atenção, semelhante a técnica utilizada pela

arquitetura proposta em Vaswani *et al.* (2017). O desempenho acadêmico do aluno é previsto, então, por uma Rede Neural Profunda de hipergrafo.

Nos experimentos o MVHGNN é comparado com vários outros métodos de Aprendizado de Máquina e de Redes Neurais Artificiais Profundas. De acordo com o autores, o MVHGNN não apenas melhora o desempenho da previsão, mas também torna o modelo interpretável. Para os experimentos foram analisados 2.368.066 registros de dados comportamentais de 11.652 alunos no semestre da primavera do ano de 2019 durante um período de 145 dias. Para amostragem foram selecionados 25%, 50%, 75% e 100% dos dados para realizar diferentes experimentos. Com intuito de evitar viés, foi aplicada a validação cruzada com 5 partições nos dados (5-folds). As pontuações finais são obtidas através do cálculo da média e desvio padrão de todos os resultados da avaliação. Os resultados experimentais finais alcançaram a Precisão de 92.4% e 92.0% de $f1_score$. A **Figura 11** apresenta graficamente o fluxo de funcionamento para o MVHGNN.

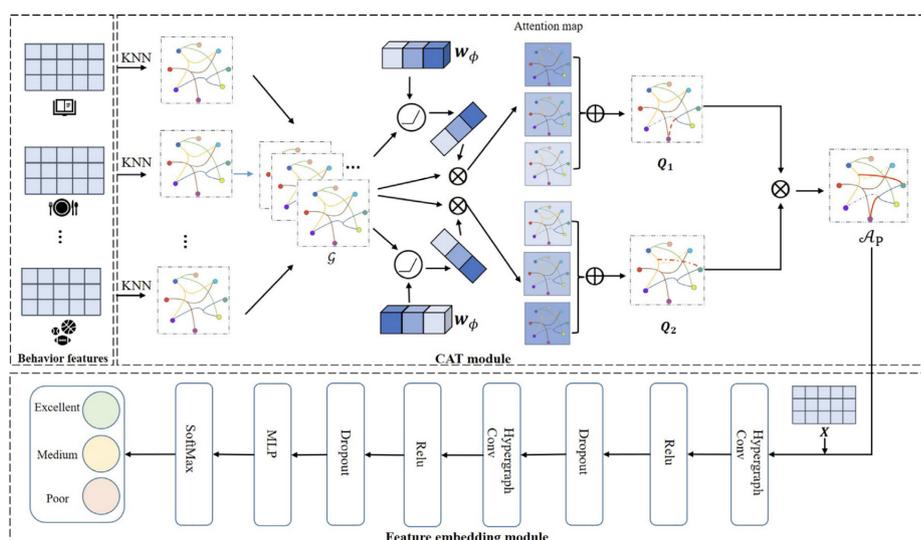


Figura 11 – O Framework MVHGNN (LI, M. *et al.*, 2022)

A estrutura geral do MVHGNN é mostrada na **Figura 11**. Como é possível observar, da esquerda para a direita, os atributos que representam os comportamentos dos alunos são agrupados com o algoritmo KNN (do acrônimo em inglês, *K-Nearest Neighbors*) e convertidos para a representação de grafos. Em seguida os grafos são combinados, por meio de operações matemáticas, para compor o hiper grafo de multi visualização e utiliza-se o módulo CAT, através do mecanismo de auto-atenção, para aprender os pesos que representam a semântica comportamental dos alunos. Por fim, o hiper grafo serve como entrada de um Rede Neural Baseada em grafos para gerar, ao fim do treinamento, o resultado com as classes da previsão (Excelente, Médio e Pobre).

3.2.2 Prediction of Students' performance with Artificial Neural Network using Demographic Traits

Kehinde *et al.* (2022) propuseram uma abordagem para previsão do desempenho dos alunos usando predominantemente características demográficas. O objetivo desse trabalho é desenvolver um sistema altamente preciso de previsão de desempenho estudantil com Redes Neurais Artificiais. O sistema servirá ao propósito de auxiliar a universidade na seleção de candidatos (alunos) para admissão universitária utilizando como atributos preditores os registros acadêmicos de alunos anteriormente admitidos e que continuaram sua trajetória acadêmica com sucesso.

As variáveis de entrada utilizadas, para serem usadas como atributos preditores, foram obtidas experimentalmente do *UCL Machine Learning Repository* e estão publicamente disponíveis no portal de competição de Aprendizado de Máquina Kaggle¹. A relação entre as características demográficas e a variável resposta foi examinada com técnicas estatística como a distorção e curtose do conjunto de dados. Os valores discrepantes da avaliação estatística de distorção e curtose foram removidos no estágio de seleção de variáveis preditoras usando a matriz de correlação, mantendo apenas dados que possuem um correlação fraca entre si e correlação forte com a variável resposta.

A arquitetura de rede apresentada na **Figura 12** mostra como o modelo está estruturado. São cinco camadas densas com 80, 120, 20, 10 e 3 neurônios respectivamente. Todos os neurônios totalmente conectados utilizando camadas de *Dropout* para redução de dimensionalidade com intuito de evitar aprendizagem dos ruídos dos dados de entrada. A rede foi treinada em lotes divididos em oito partes e executou o *feedforward* em quinhentas épocas otimizando a função de Erro Absoluto Médio (MAE) como função de perda.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 80)	2480
dropout_1 (Dropout)	(None, 80)	0
dense_2 (Dense)	(None, 120)	9720
dropout_2 (Dropout)	(None, 120)	0
dense_3 (Dense)	(None, 20)	2420
dropout_3 (Dropout)	(None, 20)	0
dense_4 (Dense)	(None, 10)	210
dropout_4 (Dropout)	(None, 10)	0
dense_5 (Dense)	(None, 3)	33

Figura 12 – Rede Neural para o Sistema proposto em (KEHINDE *et al.*, 2022)

¹ <https://www.kaggle.com/dipam7/student-grade-prediction>

O sistema diferencia-se, de acordo com os autores, por conta do alto valor de acurácia alcançado devido a eficiência do processo de seleção de variáveis preditoras. A abordagem proposta não se restringe apenas às características demográficas. Além disso, considera a adição novas variáveis preditoras de naturezas diversas como testes futuros para aprimorar o sistema. O melhor resultado obtido para previsão de desempenho dos alunos foi de 92,3% de Precisão.

3.2.3 Student-Performulator

Hussain e Khan (2023) propuseram uma abordagem baseada em Algoritmos Genéticos (do inglês *Genetic Algorithm* - GA) para automatizar a previsão do desempenho dos alunos. Essa pesquisa teve como principal objetivo obter uma melhor precisão para a classificação com um baixo valor de erro quadrático médio. O classificador de árvore de decisão baseado em GA proposto alcançou resultados significativos em relação aos demais trabalhos do estado a arte.

O conjunto de dados utilizado nos experimentos foi adquirido do *Khyber Pakhtunkhwa Board of Intermediate & Secondary Education Peshawar*. Os atributos são compostos pelos históricos acadêmicos dos alunos do nível Fundamental e Médio, sendo o exame do ensino Fundamental composto pelos 9^o e 10^o anos e o exame Médio pelos 11^o e 12^o anos. O melhor resultado obtido para previsão de desempenho dos alunos foi de 96,64% de Acurácia. A **Figura 13** apresenta graficamente o fluxo de funcionamento para o sistema *Student-Performulator*. Técnicas de pré-processamento foram utilizadas para gerenciar as instâncias com valores ausentes e extrair atributos que compõem as informações pessoais dos alunos, além da remoção de dados redundantes, que é uma técnica tradicional de tratamento para variáveis preditoras em Aprendizado de Máquina.

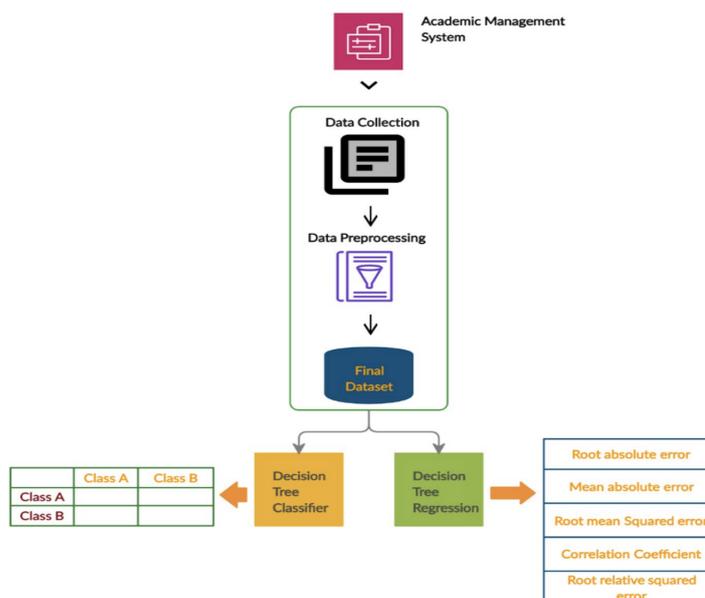


Figura 13 – Diagrama de funcionamento do *Student-Permutator* (HUSSAIN; KHAN, 2023)

Como pode ser observado na **Figura 13**, de cima para baixo, estão representados a entrada dos dados acadêmicos dos alunos, as etapas de pré-processamento para criação da base final e, por fim, o treinamento, que foi dividido em fases distintas. O Algoritmo Genético é utilizado para escolha das variáveis preditoras que alimentam ambos os modelos que serão treinados. Um modelo de Árvore de Decisão foi aplicado para Regressão (à direita), com intuito de obter o resultado numérico para as notas dos alunos e um modelo de Árvore de Decisão para Classificação (à esquerda) foi aplicado com a finalidade de obter a classificação categórica entre as classes *A* e *B* (representadas na **Figura 13** pelos termos em inglês, *Class A* e *Class B*).

3.2.4 A Hybrid Machine Learning Framework

Evangelista (2021) propôs um sistema de Aprendizado de Máquina para prever o desempenho dos alunos usando uma abordagem híbrida, denominada *A Hybrid Machine Learning Framework for Predicting Students' Performance in Virtual Learning Environment*, que consiste em uma abordagem com classificadores únicos e outra abordagem usando um método que denominou como Métodos de Conjunto (ou, do francês, *Ensemble*). Desse modo, o objetivo desse trabalho é escolher, dentre as abordagens testadas, o modelo com maior eficiência e utilizá-lo para realizar as previsões de modo automático.

Diferentemente dos classificadores únicos, tradicionalmente utilizados para Aprendizado de Máquina ao usar os dados com um único modelo de aprendizagem, o Método de Conjunto usa uma coleção de modelos e depois os combina utilizando técnicas estatísticas, tais como média, desvio padrão e sistemas de votação, como proposto

em (AMRIEH; HAMTINI; ALJARAH, 2016), para obtenção do resultado.

A estrutura deste estudo foi inspirada no *Cross Industry Standard Process for Data Mining (CRISP-DM)* (CHAPMAN *et al.*, 2000). As principais técnicas de Aprendizado de Máquina utilizadas para classificação foram *Naive Bayes*, *Decision Table*, *Random Forest*, *K-Nearest Neighbor*, *One Rule*, *J48*, *Support Vector Machine*, *Perceptron de Camadas Múltiplas* e *JRip*. De acordo com o autor, os resultados experimentais revelaram uma maior precisão preditiva ao testar os Métodos de Conjunto (*bagging*, *boosting* e *voting*) em comparação com os algoritmos de classificadores únicos. O melhor resultado obtido foi com a técnica de *boosting*, onde foi alcançada a Acurácia de 98,6%.

Ainda de acordo com o autor, essa estrutura combina a força do modelo CRISP-DM, no seu processo de mineração de dados, ao mesmo tempo que introduz uma abordagem híbrida de utilização de algoritmos de classificação e os Métodos de Conjunto para otimização dos resultados. O conjunto de dados foi obtido do segundo ano do curso de graduação em Ciências da *University of Western Ontario*. A base final representa as notas de 486 alunos. Os dados representam a participação dos alunos em diferentes atividades educacionais, tais como questionários, exames intermediários, entre outros processos utilizados pelos professores. A **Figura 14** apresenta graficamente o fluxo de funcionamento para o sistema *Hybrid Machine Learning Framework*.

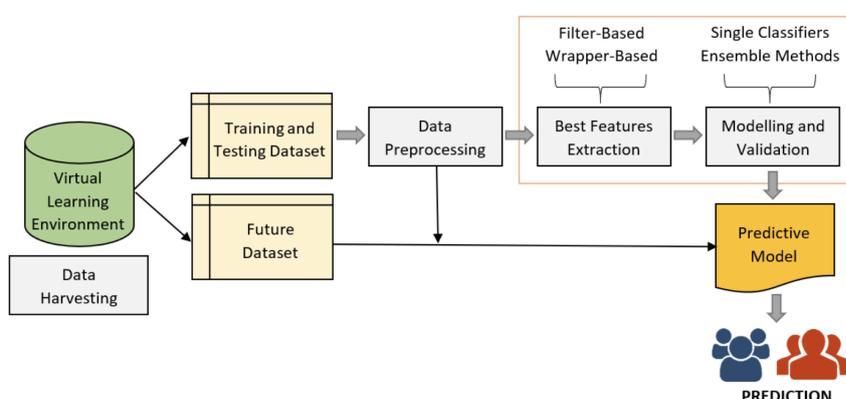


Figura 14 – Diagrama de funcionamento do *Hybrid Machine Learning Framework* (EVANGELISTA, 2021)

O componente em verde (à esquerda da **Figura 14**) representa o *software* de sistema educacional de onde os dados para os treinamentos dos modelos são obtidos automaticamente. Após a obtenção dos dados, o processo é dividido em duas etapas, representados na figura pelos nomes de *Training and Testing Dataset* e *Future Dataset*. A etapa *Training and Testing Dataset* é realizada sempre que for necessário, dado o dinamismo do comportamento dos dados. Os dados são processados, filtrados e selecionados para aplicação dos métodos propostos para a obtenção do melhor mecanismo preditivo, representado na figura pelo componente *Predictive Model*. A

etapa *Future Dataset* ocorre sempre que for necessário realizar a previsão, recebendo os dados e entregando para o *Predictive Model*.

3.2.5 Precision Education Reviews: A case study on predicting student's performance

Yusof e Abd Khalid (2021) propuseram o criação de um sistema de previsão baseado em Redes Neurais Profundas com Perceptron de Múltiplas Camadas. Para experimentação e treinamento das Redes Neurais foram utilizados dados da *Universiti Sultan Azlan Shah*, na Malásia. O estudo tem como objetivo prever o desempenho dos alunos em cursos de inglês que compõem as grades do Bacharelado em *Usuluddin* e do Bacharelado em *Syariah*. As amostras de dados foram retiradas do arquivo de desempenho dos alunos referentes ao primeiro semestre dos anos de 2019 e 2020, que consistiram em 3 turmas diferentes com aproximadamente 30 alunos cada.

De acordo com os autores, os dados de treinamento e teste foram processados no modelo Perceptron de Múltiplas Camadas, monitoradas e medidos em relação à *categorical cross entropy*. Como pode ser observado na **Figura 15**, o modelo tem duas camadas densas com cinquenta e vinte e cinco neurônios respectivamente e, por se tratar de uma classificação binária, possui uma camada final com dois neurônios. Como função de ativação para os neurônios foi utilizada *ReLU* e para obtenção da probabilidade de classificação foi utilizada a função *softmax*. A técnica *EarlyStopping* foi aplicada no conjunto de validação para evitar o aprendizado de ruídos. O melhor resultado obtido nas experimentações foi de 93% de Precisão.

```

Layer (type) Output Shape Param #
=====
dense_1 (Dense) (None, 50) 300
-----
dense_2 (Dense) (None, 25) 1275
-----
dense_3 (Dense) (None, 2) 52
=====
Total params: 1,627 Trainable params: 1,627 Non-
trainable params: 0

```

Figura 15 – Rede Neural do trabalho (YUSOF; ABD KHALID, 2021)

3.2.6 Deep Neural Network Based Educational Data Classification Model

Surenthiran, Rajalakshmi e Sujatha (2021) propuseram o *Deep Neural Network Based Educational Data Classification Model*, que consiste na utilização do Algoritmo baseado em Aprendizado Profundo (do inglês *Deep Belief Neural Network*, ou simplesmente DBNN) com Otimização de Pesquisa Atômica (do inglês, *Atom Search Optimization*, ou simplesmente ASO) para de previsão do desempenho dos alunos. O objetivo desse trabalho é utilizar o método ASO para de categorização da base histórica de desempenho dos alunos e treinar um modelo de divergência cognitiva com

múltiplas Máquinas Boltzmann Restritas (RBM) para classificar os alunos com base em seu estado comportamental.

O modelo proposto utilizou dois conjuntos de dados educacionais de código aberto (db1)² e (db2)³. O desempenho acadêmico assim como o comportamental dos alunos são insumos para criação de variáveis preditoras e para o treinamento do modelo. O db1 é formado por atributos como gênero, nacionalidade, local de nascimento, dados referentes ao estágio educacional, detalhes dos dados dos pais, entre outros. O db2 é formado por atributos como notas dos alunos, características demográficas, características sociais e características relacionadas com a escola. Todos os atributos de dados são posteriormente convertidos para o formato adequado para servir como entrada para o modelo de classificação DBNN.

Em linhas gerais o pré-processamento de dados consiste na remoção de caracteres especiais, como vírgula, ponto e vírgula e quaisquer outros caracteres especiais não necessários para o processamento do seu desempenho de dados educacionais, na inclusão de dados faltantes, de acordo com a técnica necessária para cada atributo, e normalização dos dados numéricos. A variável resposta é categorizada e classificada em Alto Nível, Médio Nível e Baixo Nível.

O desempenho do modelo proposto supera o nível de Precisão de 90% e o valor do erro ficou abaixo de 20%. De acordo com os autores, tais resultados habilitam o método proposto para ser utilizado como um mecanismo de previsão do desempenho dos alunos na área acadêmica. A **Figura 16** representa visualmente o fluxo do *Deep Neural Network Based Educational Data Classification Model*.

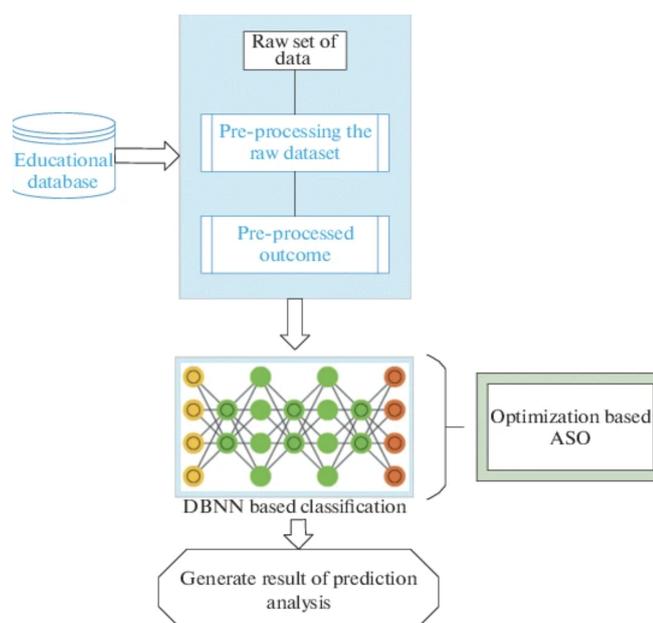


Figura 16 – *Deep Neural Network Based Educational Data Classification Model* (SURENTHIRAN; RAJALAKSHMI; SUJATHA, 2021)

² <https://www.kaggle.com/aljarah/xAPI-Edu-Data>

³ <https://archive.ics.uci.edu/ml/datasets/student+performance>

Como é possível observar na **Figura 16** a base de dados origem para o sistema é representada pelo componente *Educational database*. A primeira fase da etapa de construção do modelo preditor, representado pela área hachurada em azul, é considerada uma etapa de pré-processamento, que visa tratar os dados educacionais. Nessa etapa, então, o conjunto de dados é pré-processado para identificar valores nulos, inconsistências e valores discrepantes dos dados. Após os respectivos valores discrepantes identificados, eles são eliminados para que se tenha uma Base de de Dados Analítica para a realização do treinamento do modelo, visando a criação do modelo para realizar as previsões. Após a previsão, realizada com ajuda do DBNN, os alunos são classificados de acordo com seu nível (Baixo, Médio ou Alto). Com a classificação já realizada, têm-se acesso ao erro obtido durante o processo de retro-propagação. Por fim, a técnica de otimização ASO é empregada para otimizar o erro e melhorar o desempenho do modelo recém criado.

3.2.7 Student Performance Prediction from E-mail Assessments Using Tiny Neural Networks (CorC-Net)

Yadav e Srivastava (2020) propuseram o CorC-Net, que consiste em uma pequena Rede Neural Artificial que opera com dados em pequeno número, compostos basicamente por avaliações de alunos com base na redação de e-mails. As Redes Neurais Artificiais são normalmente construídas usando conjuntos de dados em grande volumetria. No entanto, de acordo com os autores, pequenas redes neurais podem superar essa dificuldade utilizando lotes menores, tornando-os mais eficientes durante o processo de treinamento.

O COrc-Net usa e-mails para classificação do desempenho dos alunos em relação aos quesitos Conteúdo, Clareza e Organização dos textos. Para cada um dos 3 atributos é atribuído uma pontuação e a nota final dos alunos é a soma das pontuações. Os dados que representam o desempenho (variável resposta) são classificados em *Excelente*, *Muito Bom*, *Mediano*, *Ruim*, *Inaceitável* e tratados como uma variável categórica. A codificação *One hot* é usada para converter a variável resposta em um valor de 5 bits (0s e 1s).

Para fins de construção do modelo de Aprendizado de Máquina, foram utilizadas avaliações de um curso de comunicação empresarial ministrado em um instituto de administração. 118 alunos estiveram envolvidos na fase de coleta de dados. Os indivíduos foram avaliados por professores, sob a perspectiva de que os alunos devem escrever com clareza e organização. Para avaliar a melhoria (se houver), foram atribuídas duas Tarefas de Avaliação (denominadas AT1 e AT2) para cenários distintos.

Para selecionar a configuração mais eficiente para o COrc-Net, foram utilizadas todas as 235 amostras do conjunto de dados. Primeiro foi aplicado o método *hold out*, com 50% dos dados para teste e 50% para treinamento (inicialmente sem aplicação

de *cross validation* com *K-fold*). Foram criados dois modelos e os resultados das experimentações, alternando os números de camadas ocultas e nós, são mostrados na **Figura 17**. Como também evidenciado na **Figura 17**, a arquitetura de três camadas apresentou menor perda nos dados de validação e alcançou a maior precisão. A arquitetura final do COrc-Net na **Figura 18** foi escolhida com base neste resultado.

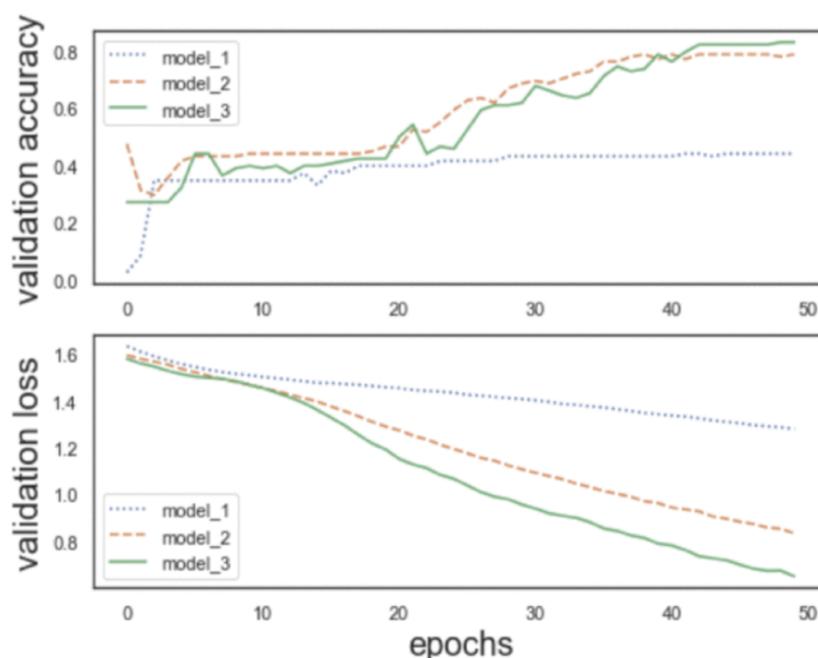


Figura 17 – Modelos 1, 2 e 3 treinados para o COrc-Net (YADAV; SRIVASTAVA, 2020)

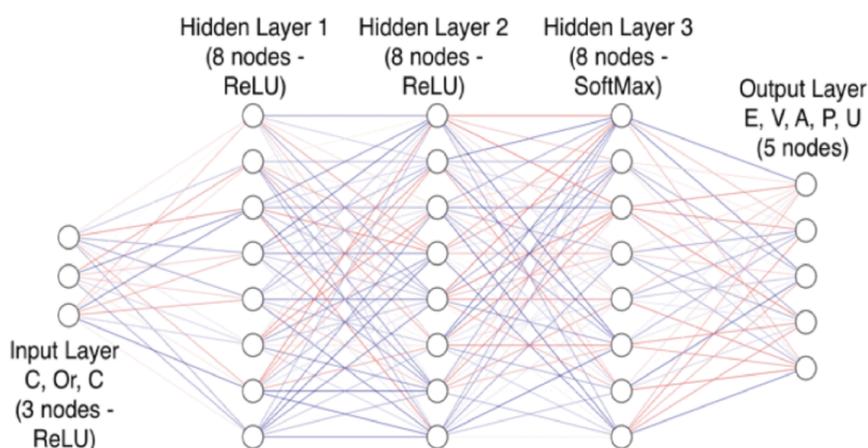


Figura 18 – Rede Neural para o Sistema COrc-Net (YADAV; SRIVASTAVA, 2020)

Para avaliar a precisão do COrc-Net e evitar quaisquer discrepâncias devido a diferenças no balanceamento das classes (mesmo sendo marginais) foi usada Curva ROC. Obtendo os valores para a taxa de verdadeiros positivos (VP) contra a taxa de falsos positivos (FP), a precisão calculada foi de 94,8%. Resultado bastante superior quando comparado com os modelos utilizados como *baseline*.

3.2.8 Deep learning for predicting students academic performance

Yunita, Santoso e Hasibuan (2019) propuseram o uso do Redes Neurais Artificiais Profundas para prever o Desempenho Acadêmico dos Alunos. O objetivo desse trabalho é treinar um modelo que possa realizar as previsões com antecedência para que seja possível tomar alguma medida e, então, evitar as respectivas reprovações.

Os dados brutos para as experimentações e treinamento do modelo são retirados de um curso de Análise de Requisitos e Projeto de Sistemas de Informação do programa de pós-graduação do Departamento de Tecnologia da Informação da Universidade da Indonésia. A coleta foi realizada da base de dados do Sistema de Gestão de Aprendizagem utilizada pela Universidade. Os registros de *Log* foram usados para examinar o comportamento dos alunos durante as atividades, enquanto as pontuações acadêmicas foram usadas como variável resposta para previsão. Como atributos foram selecionados dados como gênero, endereço, informações financeiras, idade, dados acadêmicos e comportamentais.

O tipo de Rede Neural escolhida foi a *Feedforward* com todos os neurônios na camada de entrada, nas unidades ocultas, e nas saídas totalmente conectados, configurando um dos tipos mais simples Redes Neurais Profundas. As arquiteturas Redes Neurais foram construídas usando o *Keras* sobre *TensorFlow*. As funções de ativação utilizadas foram *ReLU* e *Softmax*. A *ReLU* usada nas camadas ocultas e *Softmax* como camada de saída. Além disso, o algoritmo Adam (KINGMA; BA, 2014) foi usado para otimização. O melhor resultado alcançado foi de 85% de Precisão, usando dezoito nós de entrada com duas camadas ocultas, cada uma com doze neurônios cada, e mais dois neurônios para a camada saída.

3.2.9 An Approach to predict a students academic performance using Recurrent Neural Network (RNN)

Mondal e Mukherjee (2018) propuseram utilizar Redes Neurais Recorrentes (RNN) para prever o desempenho acadêmico dos alunos. O objetivo desse trabalho é criar um mecanismo preditivo para habilitar a previsão para os alunos do último semestre a partir da utilização de dados dos alunos do primeiro semestre, juntamente com outros quinze atributos preditores. De acordo com os autores, esse mecanismo preditivo auxilia os professores a identificar os alunos que estão em risco de reprovação, para oferecer-lhes o apoio adequado para evitar o baixo desempenho.

O conjunto de dados educacionais coletados para os experimentos foi extraído do sistema de gerenciamento de aprendizagem (LMS, do acrônimo em inglês *Learning Management System*) chamado Kalboard 360. O Kalboard 360 é um LMS multiagente, que foi projetado para facilitar o aprendizado por meio do uso de alta tecnologia. Os dados estão disponíveis publicamente por meio do portal de competição de modelos

de Aprendizado de Máquina Kaggle⁴. Dentre os atributos preditores desse conjunto podemos destacar Gênero, Local de Nascimento, Nacionalidade, Número de Faltas, Dados dos pais, entre outras.

Para fins experimentais assume-se que o desempenho final do aluno é um resultado sequencial do desempenho dos períodos anteriores, o que caracteriza um comportamento de Séries Temporais e habilita a utilização de Redes Neurais Recorrentes para solução do problema.

A arquitetura de Rede Neural proposta contém três camadas. A primeira, conhecida como camada de entrada, é a responsável por receber os atributos preditores e a última, conhecida como camada de saída, é responsável pela classificação e produção do resultado final com o desempenho dos alunos. A camada oculta está localizada entre a camada de entrada e a camada de saída e conta com quarenta neurônios. Há um processo recursivo entre os neurônios da camada oculta para conferir recorrência para a Rede Neural. Como ocorre tradicionalmente em Redes Neurais os pesos w e o valores de polarização b são inicializados aleatoriamente. Essa inicialização é proposta especificamente para nós ocultos e a função de ativação *ReLU* é definida como a parte positiva do seu argumento.

O modelo proposto alcançou uma precisão de 85,4%, superando o outros algoritmos de Aprendizado de Máquina do estado da arte que usam RNN. Embora a precisão alcançada tenha sido satisfatória, os autores destacam como trabalhos futuros submeter o modelo para ser testado com um conjunto maior e mais diverso de dados.

3.3 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO

Esse capítulo destinou-se a fazer uma revisão dos trabalhos relacionados a aplicação de Redes Neurais Artificiais, dentre outras técnicas, para a previsão do desempenho de alunos. O objetivo principal desta seção foi explorar as pesquisas publicadas por trabalhos correlatos que identificam as principais abordagens, métodos e contribuições no contexto da análise preditiva de desempenho estudantil. O intuito foi apresentar um panorama amplo sobre as pesquisas anteriores relacionadas ao tema desta dissertação, identificando lacunas e oportunidades para a contribuição deste estudo. Um resumo com as características de cada trabalho é apresentado na **Tabela 1**.

⁴ <https://www.kaggle.com/aljarah/xAPI-Edu-Data>

Tabela 1 – Tabela comparativa com os trabalhos do estado da Arte para previsão do desempenho de alunos

	Tipo de Aluno	Tipo de Curso	Tipo de Dado
(LI, M. <i>et al.</i> , 2022)	Universitário	Presencial	Comportamentais
(KEHINDE <i>et al.</i> , 2022)	Universitário	Não mencionado	Demográficas
(HUSSAIN; KHAN, 2023)	Ensino Médio	Presencial	Comportamentais
(EVANGELISTA, 2021)	Universitário	<i>Online</i>	Comportamentais
(YUSOF; ABD KHALID, 2021)	Universitário	Não mencionado	Comportamentais
(SURENTHIRAN; RAJALAKSHMI; SUJATHA, 2021)	Ensino Médio	<i>Online</i>	Comportamentais
(YADAV; SRIVASTAVA, 2020)	Não mencionado	<i>Online</i>	Texto
(YUNITA; SANTOSO; HASIBUAN, 2019)	Universitário	Não mencionado	Demográficas e comportamentais
(MONDAL; MUKHERJEE, 2018)	Ensino Médio	<i>Online</i>	Comportamentais
SPPACO	Universitário	<i>Online</i>	Texto

Na tabela **Tabela 1** entendem-se por Dados Comportamentais aqueles referentes a um comportamento ou ação tomada pelo aluno. Dados Demográficos, por sua vez, são os dados que descrevem uma população com base em uma série de variáveis-chave, tais como idade, gênero, renda, raça, etc.

Observa-se que todos os sistemas listados na **Tabela 1** dedicam-se a realizar a previsão do desempenho de alunos usando técnicas de Aprendizado de Máquina. Os trabalhos (LI, M. *et al.*, 2022), (EVANGELISTA, 2021), (YUSOF; ABD KHALID, 2021) e (YUNITA; SANTOSO; HASIBUAN, 2019) usam dados comportamentais de alunos universitários. No entanto, não utilizam textos como variáveis preditoras para o treinamento dos seus modelos e, dentre eles, apenas o trabalho (EVANGELISTA, 2021) usa dados coletados de cursos *Online*.

O trabalho (YADAV; SRIVASTAVA, 2020), embora utilize os textos de *e-mails* como variáveis preditoras, o tipo de aluno não é mencionado explicitamente. Fica subentendido que se tratam de alunos de um curso profissionalizante de cunho técnico administrativo. Além disso, as variáveis respostas são baseadas em um escopo de nota dado a qualidade de redação do e-mail nos quesitos Conteúdo, Clareza e Organização dos textos, e não propriamente relacionando o texto com o desempenho estudantil dos alunos.

Portanto, dadas as considerações anteriores, percebe-se que o SPPACO, que será apresentado no **Capítulo 4**, é o único que destina-se a fazer a previsão do desempenho de alunos de cursos universitários *Online* a partir dos textos postados

pelos alunos ao longo da sua trajetória acadêmica, além de ser mais assertivo em relação aos demais elencados como Estado da Arte.

4 ARQUITETURA PROPOSTA

É consenso científico que a linguagem humana reflete personalidade, estilo de pensamento, ligações sociais e estados emocionais (MAJUMDER *et al.*, 2017). O texto é uma das maneiras utilizadas pelas pessoas para comunicação, de modo geral, e pode ser considerada como um tipo específico de linguagem. Os textos registrados pelos alunos são abundantes nas plataformas de ensino a distância, sejam eles produzidos em fóruns de discussão, na resolução de problemas propostos pelos professores, ou mesmo na interação social entre alunos e entre alunos e professores.

Redes Neurais Artificiais Profundas, também conhecidas pelo termo em inglês *Deep Learning*, têm alcançado resultados notáveis em diversas áreas do conhecimento humano. Como exemplos, podemos citar Visão Computacional, Reconhecimento de Fala e Classificação de Textos (LIU; GUO, 2019). Para Classificação de Textos, que considera textos como variáveis preditoras em um modelo de Aprendizado de Máquina Supervisionado e que será nosso objeto de estudo, dois tipos são aplicados com bons resultados; Redes Neurais Convolucionais (CNNs) e Redes Neurais Recorrentes (RNNs) (LIU; GUO, 2019). As CNNs são capazes de aprender uma resposta local a partir de dados temporais ou espaciais, mas não têm a capacidade de aprender correlações sequenciais de modo satisfatório. Em contrapartida, as RNNs são especializadas em modelagem sequencial, mas não podem extrair recursos em paralelo. De acordo com (LIU; GUO, 2019) ambos os modelos, quando usados em conjunto e com a adição de outras técnicas como *Word Embeddings*, por exemplo, podem apresentar resultados significativos na classificação de textos.

Como destacado no **Capítulo 1**, a evasão de alunos, bem como suas reprovações nos cursos de EAD, são problemas sérios no âmbito educacional que precisam ser devidamente tratados. Portanto, dentro desse cenário, o presente trabalho propõe utilizar o Estado da Arte de Classificação de Textos com Aprendizado de Máquina para criar um sistema para realizar a previsão da performance dos alunos (Aprovado ou Reprovado) de cursos online com intuito orientar os Agentes Educacionais¹ para adotar alguma ação para prevenir reprovações. A solução será denominada Sistema de Previsão de Performance para Alunos em Cursos Online (ou, simplesmente SPPACO). A seguintes etapas serão percorridas para criação do SPPACO:

1. Coleta, Tratamento e Rotulagem dos Dados
2. Pré-treino e criação das *Word Embedding*
3. Avaliação, treinamento e escolha da arquitetura de Rede Neural Profunda
4. Proposta de Arquitetura para o SPPACO

¹ Profissionais da área da Educação de modo geral, cujo trabalho seja impactado pelo desempenho dos alunos.

4.1 COLETA, TRATAMENTO E ROTULAGEM DOS DADOS

Como fonte de dados para este trabalho foram utilizadas tabelas de banco de dados de alunos dos cursos de Ensino à Distância (EAD) da Universidade Federal de Santa Catarina (UFSC). Procurou-se trabalhar com uma base final, onde os alunos não podem ser identificados diretamente. Mesmo com tais cuidados, um projeto foi submetido ao Comitê de Ética em Pesquisa com Seres Humanos da UFSC (CEPSH), com o objetivo de formalizar a pesquisa e reforçar a responsabilidade relacionada ao tratamento dos dados. Após o processamento e avaliação do projeto pelo CEPSH, a autorização para acesso aos dados foi recebida.

4.1.1 Bando de Dados

A base de dados completa possui 419 tabelas, das quais 7 foram utilizadas. Após todo o processamento a base final ficou com 50.356 linhas, representando 259 cursos e 3.492 alunos. Cada registro corresponde a um texto deixado pelo aluno nos fóruns da plataforma e foi vinculado ao respectivo resultado do aluno no curso ao qual o fórum está vinculado. Um aluno foi considerado **Aprovado** quando a nota alcançada foi igual ou superior a 6. Nesse caso o aluno recebe o rótulo 1. Os alunos com nota inferior a 6 foram considerados **Reprovados** e receberam o rótulo 0. Um diagrama das tabelas usadas é mostrado na **Figura 19**.

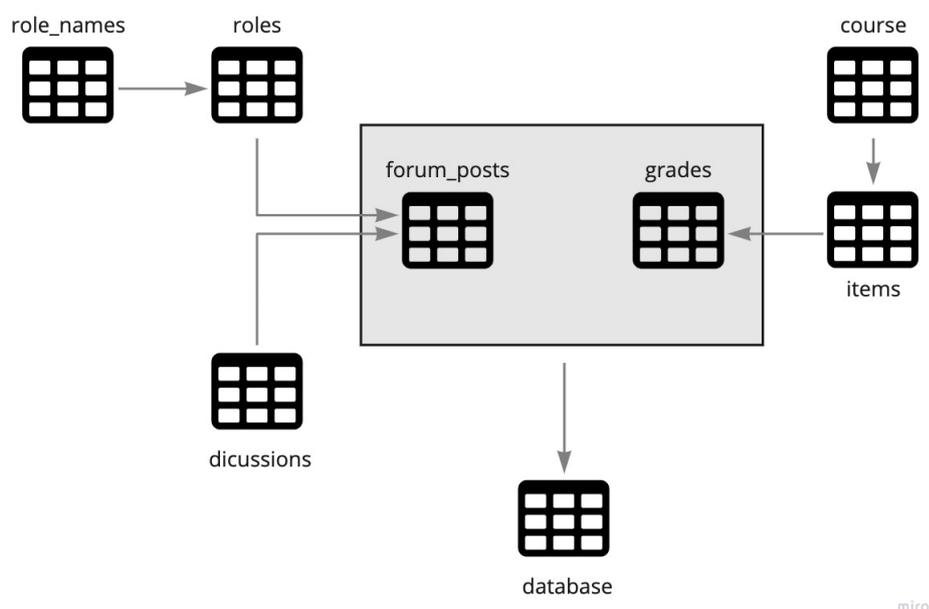


Figura 19 – Diagrama das Tabelas de Banco de Dados Utilizadas

4.1.2 Limpeza, Organização e Tratamento dos Dados

Para preparar os textos dos alunos, a fim de utilizá-los com algoritmos de *Deep Learning*, foram aplicadas algumas técnicas de limpeza e tratamento de dados. Pontuações, caracteres especiais, *tags html*, *e-mails*, urls e números de telefone foram removidos. As palavras em português que quando retiradas da frase não afetam o entendimento geral (*stopwords*) também foram removidas. Posteriormente, cada palavra de cada texto foi separada (*tokens*), a fim de transformá-las em valores numéricos únicos (ID's). E, finalmente, as linhas com a coluna de texto em branco foram removidas. A **Figura 20** mostra um exemplo de texto digitado por alunos sem tratamento, limpo e separado e em formato numérico, respectivamente.

```
1 base_curso.msg[1]
'Toda essa desconfiança com um diploma EaD gerou-me uma dúvida: o que diz o diploma? Estará nele descrito claramente c
urso a distância? Não que eu me importe, mas o mercado sim! Será que a mudança, o caminho para a aceitação deste tipo
de ensino não deveria partir daí? Sem distinção, sem preconceitos! '
```

```
1 sentences[1]
'toda desconfiança diploma ead geroume dúvida diz diploma estará nele descrito claramente curso distância importe merc
ado sim mudança caminho aceitação deste tipo ensino deveria partir daí distinção preconceitos'
```

```
1 print(sequences[1])
[231, 5015, 3881, 185, 21781, 85, 221, 3881, 1503, 3609, 2280, 3224, 36, 214, 17746, 80, 37, 437, 520, 2281, 357, 184,
70, 302, 238, 2030, 5619, 6126]
```

Figura 20 – Textos ao Longo do Tratamento

Os textos digitados pelos alunos não obedecem a uma regra quanto ao número de palavras, por isso foram criadas algumas categorias para analisar a distribuição dos números das palavras. A **Figura 21** representa as distribuições.

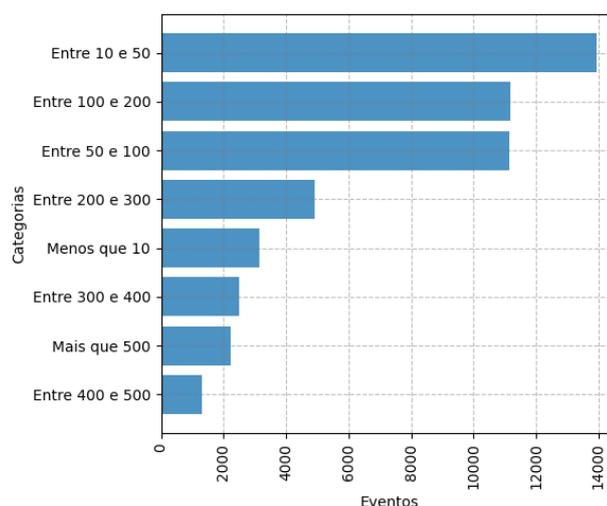


Figura 21 – Separação das Palavras em Categorias

Palavras que aparecem pouco, como nomes próprios, por exemplo, recebem uma marca *out of vocabulary* (ou, simplesmente, OOV) para ser desconsiderada ao

treinar a *Embedding Matrix*. Textos com menos palavras foram completados com zeros até ficarem do mesmo tamanho das sentenças maiores (*padding*) para ter no processo final todas as frases com a mesma dimensão.

4.2 PRÉ-TREINO E CRIAÇÃO DAS *WORD EMBEDDING*

Como visto na seção 2.2.3, *Word2Vec* é um dos métodos mais influentes e amplamente utilizados para a representação de palavras em aplicações de Processamento de Linguagem Natural (PLN). O *Word2Vec* é baseado na hipótese distribucional, que postula que palavras que aparecem em contextos semelhantes tendem a ter significados semelhantes. O método explora coocorrências de palavras em um grande corpus de texto para aprender representações vetoriais que codificam informações sobre o contexto em que as palavras ocorrem.

O treinamento do *Word2Vec* é realizado usando algoritmos de otimização, como o Gradiente Descendente. Os vetores de palavras são inicializados aleatoriamente e, durante o treinamento, são ajustados iterativamente para minimizar a função de perda, que mede a diferença entre as previsões do modelo e os valores reais das palavras de contexto. O vetor otimizado para representar a palavra é conhecido como *Word Embedding*. As duas abordagens principais que são usadas para implementar o *Word2Vec* são as arquiteturas *Skip-gram* e CBOW (Continuous Bag of Words) proposta por (MIKOLOV *et al.*, 2013). Nesse trabalho optou-se pela utilização do *Skip-gram*.

É comum utilizar a *Embedding Matrix*, que é a matriz com todas as *Word Embeddings* obtidas de um corpus, pré-treinadas por outros grupos de pesquisa. No Brasil, por exemplo, temos o NILC-Embeddings (HARTMANN *et al.*, 2017), que é um repositório para armazenar e compartilhar *Word Embeddings* geradas para a língua portuguesa. Embora a pesquisa e implementação realizada em (HARTMANN *et al.*, 2017) seja muito bem feita, esse trabalho opta por treinar a própria *Embedding Matrix*, pois acredita-se que o corpus extraído dos fóruns dos cursos a distância utilizados possuem características intrínsecas que seriam perdidas ao adotar outra abordagem mais ampla.

Detalhes sobre a implementação da *SPPACO Embedding Matrix* serão apresentadas no **Capítulo 5**. Após obter a *Embedding Matrix*, é possível usá-la como os pesos iniciais na primeira camada das redes neurais. Nesse contexto a criação da *SPPACO Embedding Matrix* pode ser considerado um pré-treino da arquitetura final de Redes Neurais Profundas que suportará a previsão de desempenho de alunos.

4.3 AVALIAÇÃO, TREINAMENTO E ESCOLHA DA ARQUITETURA DE REDE NEURAL PROFUNDA

A classificação de textos utilizando Redes Neurais Artificiais Profundas pode ser considerada uma tarefa de modelagem sequencial, semelhante, nesse sentido, as Séries Temporais. Por esse motivo, dadas as suas características de recorrência e a capacidade de armazenamento de memória de curto e de longo prazo, as Redes Neurais Recorrentes (RNNs) (FUNAHASHI; NAKAMURA, 1993) são frequentemente usadas. Como exposto na seção 2.1.2.6, uma RNN em sua implementação inicial, ao ser utilizada para análise de sequências de dados longas, sofrem de dois problemas inerentes ao seu funcionamento, explosão e a dissipação de gradiente. Nesse contexto, surgiram as Redes Neurais com Memória de Curto e Longo Prazo (ou, simplesmente LSTM, do seu acrônimo em inglês *Long Short Term Memory*) (HOCHREITER; SCHMIDHUBER, 1997). As LSTMs são RNNs dotadas de unidades de memória de longo prazo também, e, portanto, auxiliam a resolver satisfatoriamente os problemas relacionados ao gradiente.

Além da questão sequencial intrínseca aos vetores que representam as palavras, a questão temporal também tem influência direta. Dependendo da ordem com a qual se analisa um vetor de palavras, elas podem adquirir um contexto semântico diferente. As Redes Neurais com Memória de Curto e Longo Prazo Bidirecionais (ou, simplesmente *Bi-LSTM*) (GRAVES; SCHMIDHUBER, 2005) são evoluções naturais das redes LSTM, pois são capazes de percorrer as sequências em ambas as direções temporais. Por esta razão, o BiLSTM é geralmente superior a LSTM para trabalhar com textos.

A representação vetorial de textos geralmente tem uma dimensão elevada. O vetor de alta dimensão como entrada para um LSTM causará um aumento acentuado nos parâmetros da rede e dificultará sua otimização. A arquitetura de Redes Neurais Convolucionais (CNNs) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) pode ser útil para extrair recursos dos dados enquanto reduz a sua dimensionalidade. Por esse motivo, a operação de convolução podem ser usada neste contexto. Portanto, as arquiteturas LSTM, CNNs e BiLSTM são geralmente testadas para classificar os textos utilizados em conjunto. No presente trabalho, o melhor resultado, do ponto de vista de Acurácia, foi a implementação da arquitetura híbrida C-BiLSTM inspirada no trabalho Liu e Guo (2019).

4.3.1 Treinamento do Modelo de Aprendizado de Máquina

O algoritmo de treinamento da arquitetura híbrida *Convolutional Bidirectional Long Short-term Memory* (ou, simplesmente CBiLSTM) é representado na **Figura 22**. A arquitetura possui, após a camada de inserção dos dados e *Word Embeddings*, uma camada convolucional de uma dimensão, seguida pelo uma camada de *Dropout*. Em

seguida observa-se a camada *Bidirectional LSTM*, também seguida de uma camada de *Dropout*. As camadas de *Dropout* são utilizadas com intuito de realizar uma seleção de neurônio em tempo de treinamento, o que geralmente evita sobre-aprendizagem. As camadas seguintes servem ao propósito de consolidação dos vetores e aplicação da classificação propriamente dita.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 100, 100)	5277600
conv1d (Conv1D)	(None, 100, 50)	50050
dropout (Dropout)	(None, 100, 50)	0
bidirectional (Bidirectional)	(None, 256)	183296
dropout_1 (Dropout)	(None, 256)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
activation (Activation)	(None, 1)	0
Total params: 5,543,971		
Trainable params: 266,371		
Non-trainable params: 5,277,600		

Figura 22 – Algoritmo de Treinamento da Rede C-BiLSTM

A **Figura 23** apresenta a curva de Acurácia e de Erro no decorrer do treinamento da rede. Os resultados para todos os testes realizados serão observados no **Capítulo 5** e no **Apêndice A.2**. O melhor resultado ao utilizar a arquitetura híbrida C-BiLSTM foi de 98% Acurácia, demonstrando superioridade em relação as arquiteturas usadas como base de comparação, tais como MLP, CNN e C-LSTM, que alcançaram respectivamente 95%, 96% e 95%, como mostra a tabela **Tabela 2**. Esse resultado demonstra que essa arquitetura deve ser escolhida para suportar o funcionamento do sistema SPPACO.

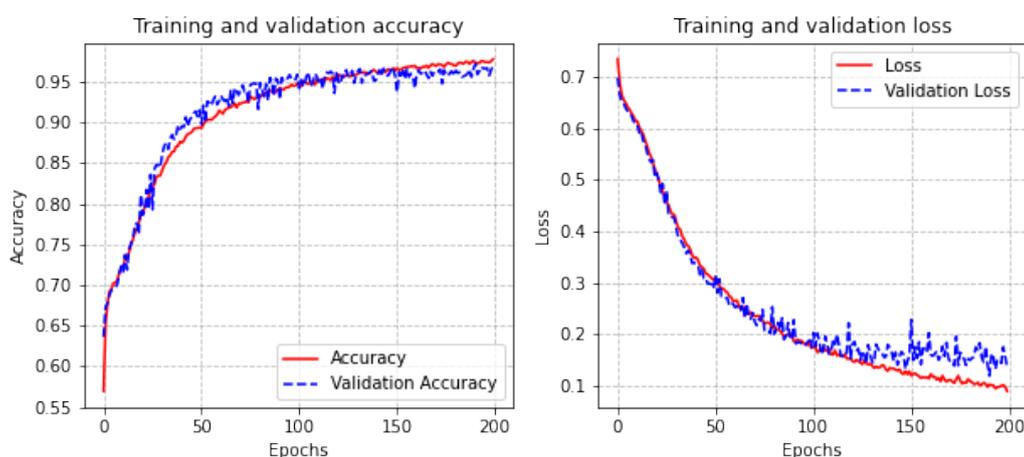


Figura 23 – Histórico da Acurácia e Erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação

Tabela 2 – Resultados de experimentos com textos postados no primeiro mês de curso

Arquitetura	Acurácia	Erro	AUC	<i>f1_score</i>
MLP	0.95	0.16	0.95	0.94
CNN	0.96	0.10	0.96	0.96
C-LSTM	0.95	0.19	0.95	0.95
C-BiLSTM	0.98	0.12	0.98	0.97

4.4 PROPOSTA DE ARQUITETURA PARA O SPPACO

Uma vez que se têm um modelo de Aprendizado de Máquina treinando para fazer a previsão do desempenho dos alunos (Representado na **Figura 24** pela letra ①), é possível transformá-lo em um modelo publicável para compor um sistema de previsão. Uma das formas comuns de fazer isso é a serialização. Essa técnica é frequentemente usada para salvar modelos de Aprendizado de Máquina, pois permite que seja feita a conversão do modelo em um arquivo para carregá-lo novamente sob demanda para fazer previsões sem precisar treiná-lo outra vez.

A serialização também favorece a reprodutibilidade e escalabilidade dos modelos treinados, pois trata-se da persistência do modelo em um arquivo físico, geralmente agnóstico ao sistema operacional. Os tipos de arquivos mais comuns nesse cenário são o *Pickle*, *HDF5 (Hierarchical Data Format)*, *ONNX (Open Neural Network Exchange)*, ou mesmo, em *JSON (JavaScript Object Notation)*.

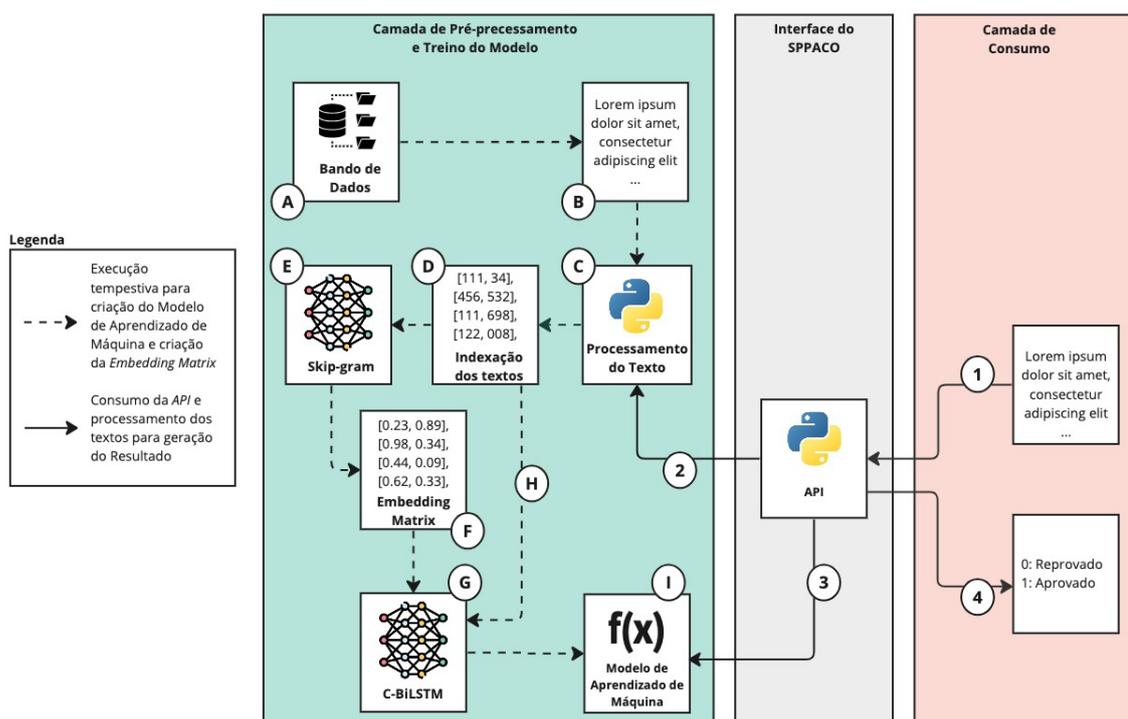


Figura 24 – Proposta de Arquitetura para o SPPACO

A **Figura 24** representa a arquitetura do sistema SPPACO, que pode ser dividida em dois principais componentes e uma camada de consumo. O primeiro componente, Camada de Pré-processamento e Treino do Modelo (área hachurada em verde), representa todo ciclo de treinamento para criação do modelo de Aprendizado de Máquina, que pode ser agendado para ser executado com a tempestividade necessária para que o modelo mantenha sua capacidade de previsão atualizada em função dos novos textos que são gerados pelos novos alunos.

O segundo componente, Interface do SPPACO (a área hachurada em cinza), representa a camada responsável pela interface do Sistema. Nessa camada, a interface ocorre por meio de uma *API* (do acrônimo em inglês *Application Programming Interface*). Por intermédio dela, os textos dos alunos, os quais as previsões precisam ser feitas, são enviados para processamento e previsão de desempenho, ação representada pela seta ①. Por fim, há a representação dos textos que precisam ser preditos e da resposta gerada pelo sistema, denominada Camada de Consumo (área hachurada em laranja).

4.4.1 Camada de Pré-processamento e Treino do Modelo

Para os novos textos que serão recebidos para previsão também é necessário fazer o mesmo pré-processamento realizado ao treinar o modelo (representado pela seta ② na **Figura 24**), como exposto na seção 4.1.2. Tal ação justifica-se para que o texto a ser predito seja convertido para o formato cujo o modelo espera receber. Uma

vez que as ações foram realizadas, aciona-se o modelo treinado para inferência do resultado, ato representado na seta (3). Por fim, com os textos pré-processados e devidamente mapeados, é gerada uma resposta, com a previsão propriamente dita, para ser retornada pela chamada da *API*, etapa representada pela seta (4).

Na Camada de Pré-processamento e Treino do Modelo o item (A) representa o banco de dados de onde podem ser extraídos os textos inseridos pelos alunos como dados históricos para o treinamento do modelo. O item (B) representa os textos selecionados, extraídos e rotulados do banco de dados. O item (C) representa o processamento de dados que é realizado nos textos. O item (D) representa os textos pré-processados e indexados para criação e pré-treino das *Word Embeddings*. O item (E) representa o algoritmo de *Word2Vec* escolhido. O item (F) representa a *Embedding Matrix* e, por fim, o item (G) representa a arquitetura de Rede Neural Profunda que será utilizada para o treinamento, no caso do presente trabalho a arquitetura C-BiLSTM.

Após utilização da *Embedding Matrix* como entrada da rede para configuração inicial dos pesos com os vetores probabilístico das palavras ((F)), as sentenças textuais indexadas dos alunos ((D)), juntamente com os rótulos (1:Aprovado, 0:Reprovado), são entregues para o treinamento do modelo. Após a execução tempestiva de todo o ciclo representado por essa camada, a *Embedding Matrix*((F)) e o Modelos de Aprendizado de Máquina ((I)) são criados e/ou atualizados.

4.4.2 Interface do SPACO

A Interface do SPACO, como acima mencionado, representa a camada responsável pela interface do Sistema. Nessa camada, o mecanismo de interface e conexão como os elementos da Camada de Pré-processamento e Treino do Modelo, é representado por uma *API*. Nesse momento os textos dos alunos, os quais as previsões precisam ser feitas, são enviados para processamento e previsão de desempenho, representado pela seta (1) na **Figura 24**. Em um ambiente produtivo outros elementos precisariam estar presentes nessa camada, como por exemplo, autenticação dos usuários da *API* e o Monitoramento e Observabilidade das Respostas.

4.5 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO

Esse capítulo destinou-se a apresentar o SPACO, que se trata de um Sistema de Previsão de Performance para Alunos em Cursos Online, como mecanismo preditivo para possibilitar antecipação de ações para alunos com risco de reprovação e, então, mitigar o baixo desempenho de alunos e a evasão. A seção 4.1 apresentou a etapa de Coleta, Tratamento e Rotulagem dos Dados, necessária para adequação dos dados para o propósito de realizar as experimentações e treinamento do modelo de Apreindizado de Máquina que será responsável pelas previsões. A seção 4.2

apresentou a etapa de Pré-treino e criação das *Word Embedding*, responsável por criar os valores probabilísticos e contextuais das palavras do corpus utilizado por esse trabalho, com intuito de alimentar, inicialmente, os pesos da Rede Neural utilizada. A seção 5.4 apresenta a etapa de Avaliação, treinamento e escolha da arquitetura de Rede Neural Profunda mais adequadas para fazer as previsões do desempenho do aluno. E, por fim, a seção 4.4 apresenta uma proposta para arquitetura para o sistema SPPACO.

5 ANÁLISE DOS RESULTADOS OBTIDOS E DISCUSSÃO

Esse capítulo tem por objetivo explorar os resultados obtidos a partir das investigações experimentais conduzidas acerca da aplicação de Linguagem de Processamento Natural e Redes Neurais Artificiais Profundas para treinamento de um modelo de previsão de desempenho de alunos de Cursos EAD para suportar o SPPACO.

5.1 BASE DE DADOS USADA PARA O TREINAMENTO DOS MODELOS

Como fonte de dados para este trabalho foram utilizadas tabelas de banco de dados da plataforma *Moodle*, utilizada pelos professores e alunos dos cursos de Ensino à Distância (EAD) da Universidade Federal de Santa Catarina (UFSC). A base de dados completa possui 419 tabelas, das quais 7 foram utilizadas. Após todo o processamento a base final ficou com 50.356 registros, representando 259 cursos que correspondem aos registros de 3.492 alunos. Cada registro corresponde a um texto deixado pelo aluno nos fóruns da plataforma e foi vinculado ao respectivo resultado do aluno no curso ao qual o fórum está vinculado. Um aluno foi considerado **Aprovado** quando a nota alcançada foi igual ou superior a 6. Nesse caso o aluno recebe o rótulo 1. Alunos com nota inferior a 6 foram considerados **Reprovados**, recebendo o rótulo 0. Portanto, a base de dados é adequada para classificação binária de textos.

A Representatividade de alunos reprovados é inferior aos alunos que foram aprovados. Portanto, para evitar viés na aprendizagem do modelo, a técnica de balanceamento de carga (SMOTE), que consiste em equilibrar o número de registros de ambas as classes (aprovados e reprovados), foi utilizada.

5.2 CRIAÇÃO DAS *WORD EMBEDDINGS*

Para criar a *SPPACO Embedding Matrix* foi usada uma implementação em Python da ferramenta **Gensim** na versão 3.8.3, que é uma biblioteca de código aberto para modelagem de tópicos não supervisionados e Processamento de Linguagem Natural. A **Figura 25** mostra o método em Python utilizado para obtenção dos vetores *Word2vec* e a **Figura 26** fornece um exemplo da aplicação da *Embedding Matrix* para obter palavras semelhantes à palavra "aluno" no corpus utilizado neste trabalho.

```
# training w2c with skip-gram algorithm
w2v = Word2Vec(corpus,
               size=EMB_DIM,
               window=5,
               min_count=5,
               negative=15,
               sg = 1, #skip-gram
               workers=multiprocessing.cpu_count())

word_vectors = w2v.wv
```

Figura 25 – Obtenção dos vetores de palavras com *Word2vec* utilizando o algoritmo *Skip-Gram*

```
In [47]: 1 word_vectors.similar_by_word("aluno")
Out[47]: [('estudante', 0.8087526559829712),
          ('alunos', 0.7965363264083862),
          ('educador', 0.7912662029266357),
          ('educando', 0.761611819267273),
          ('educandos', 0.7417762875556946),
          ('professor', 0.715225875377655),
          ('plateia', 0.6950972080230713),
          ('ensinoaprendizagem', 0.6872051358222961),
          ('autodidata', 0.6790434122085571),
          ('discente', 0.6764541268348694)]
```

Figura 26 – Similaridade da palavra **aluno** com *Word2vec*

Como já mencionado no **Capítulo 4**, depois de obter a *Embedding Matrix*, é possível usá-la como os pesos iniciais na primeira camada das redes neurais, como exemplifica a **Figura 27**. Nesse contexto a criação da *SPPACO Embedding Matrix* pode ser considerado um pré-treino da arquitetura final de Redes Neurais Profundas que suportará a previsão de desempenho de alunos.

```
model_v1 = Sequential()
model_v1.add(Embedding(input_dim = VOCAB_SIZE,
                      output_dim = EMB_DIM,
                      weights = [embeddings_matrix],
                      input_length = MAX_LENGTH,
                      trainable = False))
```

Figura 27 – Configuração dos pesos da Rede Neural Utilizando a *Embedding Matrix*

5.3 CONFIGURAÇÃO DOS PARÂMETROS PARA TREINAMENTO

Os dados foram separados em 80% para treinamento, 10% para validação e 10% para teste. Os dados de treinamento são entregues em lotes e a cada época de passagem dos dados pela rede neural os valores previstos são comparados com os valores reais nos dados de validação, com fim de otimização da função de custo no *backpropagation*. A acurácia do modelo é medida com a porção de dados de teste, para que não haja viés na avaliação final do modelo.

Durante o treinamento na camada *Embedding*, o vetor de entrada é definida como a n-ésima palavra no vocabulário. A *Embedding Matrix* foi treinada a partir do

corpus disponível com dimensão 100. A dimensão da memória das redes LSTM e BiLSTM foi configurada para 128 e as camadas convolucionais foram configuradas com 50 filtros de tamanho 10. O tamanho do lote de treinamento (*epoch*) para todos os conjuntos de dados foi de 256. A taxa de abandono (*dropout*) na maioria dos testes foi de 0,6. Um algoritmo de retro propagação com o método de otimização estocástica (Adam) foi usado para treinar a rede ao longo do tempo com uma taxa de aprendizado entre 0,001 e 0,0001. A função de perda utilizada foi a *Binary Cross Entropy*.

5.4 RESULTADOS OBTIDOS

Como já mencionado na **Seção 4.1**, para homogeneizar os vetores das sentenças dos alunos em relação a dimensão, foi aplicada a técnica de *padding*, que consiste, após a *tokenização*, em completar as sentenças menores com zeros até ficarem do mesmo tamanho das sentenças maiores. Com intuito de identificar se haveria viés em relação ao número de palavras ao treinar as Redes Neurais Artificiais e para validar a hipótese de que seria possível encontrar relação entre os textos e os rótulos de desempenho dos alunos, os primeiros testes consideraram o treinamento de modelos com populações segregadas por números de palavras.

Os resultados desses testes foram publicados no trabalho Guerra Junior A. A. (2022) e podem ser acompanhados em detalhes no **Apêndice A.1**, onde constam todos gráficos de acompanhando da Acurácia e Erro nos diversos testes realizados.

Uma vez confirmada a hipótese, para os testes subsequentes e para criação do modelo preditivo que suporta o SPPACO, houve a segregação das populações de dados pela data com a qual os textos foram publicados pelos alunos no *Moodle*. Considerou-se três amostras; textos publicados até o fim do primeiro mês de curso, até o fim do segundo e, por fim, do terceiro mês. Os resultados dos treinamentos realizados para o primeiro mês de postagem serão apresentados nas seções seguintes. Todos os demais gráficos podem ser visualizados no **Apêndice A.2**.

5.4.1 Arquitetura MLP

Percebe-se ao observar a **Figura 53** que o treinamento realizado com a arquitetura Perceptron de Múltiplas Camadas (MLP, do acrônimo em inglês *Multi Layer Perceptron*) já apresenta bons resultados, mesmo se compararmos com o trabalho do estado da arte Yadav e Srivastava (2020) que também usa texto como variáveis predictoras. O trabalho Yadav e Srivastava (2020) obteve 94,8% de Acurácia, enquanto que a arquitetura MLP desse presente trabalho obteve resultado ligeiramente superior, com 95% de Acurácia.

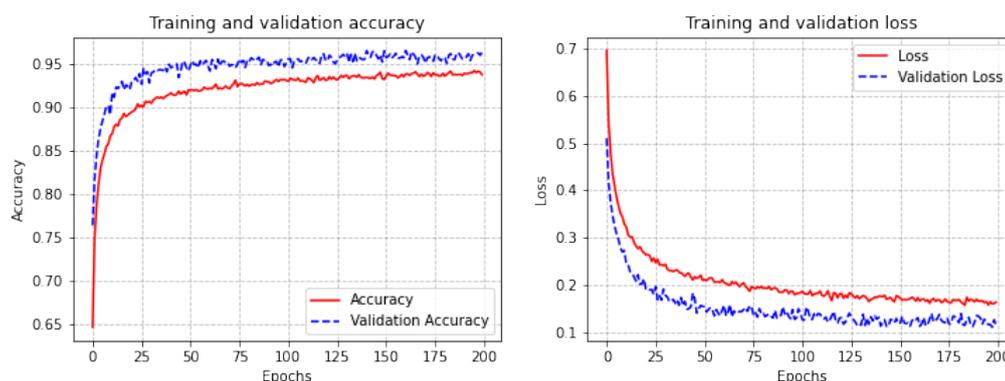


Figura 28 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no primeiro mês

A diferença encontrada entre as curvas de Acurácia para os dados de treino (em vermelho) e de validação (em azul) pode apontar para a sobre-aprendizagem, que ocorre quando a Rede Neural aprende, além dos padrões, os ruídos dos dados, deixando o modelo com uma capacidade limitada de generalização. Esse fato pode ocorrer nas MLPs por conta da sua capacidade limitada de identificação de características intrínsecas a esse tipo de arquitetura, assim como também por conta da dificuldade de lidar com vetores unidimensionais sequenciais de grande dimensão, como são os textos.

5.4.2 Arquitetura CNN

Percebe-se ao observar a **Figura 54** que o treinamento realizado com a arquitetura de Redes Neurais Convolucionais (CNN, do acrônimo em inglês *Convolution Neural Network*) apresenta bons resultados, semelhante a arquitetura MLP, ficando ligeiramente superior com 96% de Acurácia.

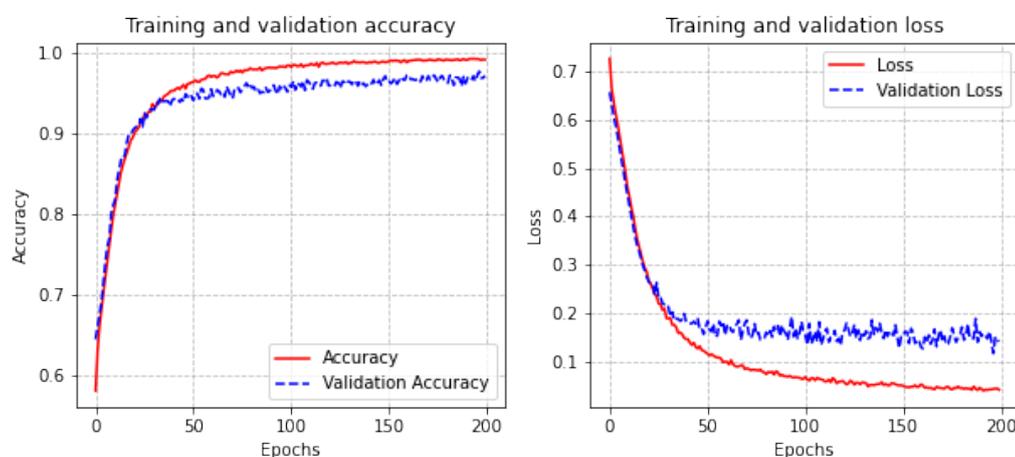


Figura 29 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no primeiro mês

As CNNs são excelentes arquiteturas para extração de características dos dados e para redução de dimensionalidade. Entretanto não são boas opções quando há a necessidade de analisar vetores unidimensionais de grande dimensão com características sequenciais, como são os textos. Esse fato pode justificar a diferença encontrada entre as curvas de Acurácia para os dados de treino e de validação, a exemplo de como ocorre com as MLPs.

5.4.3 Arquitetura CLSTM

Percebe-se ao observar a **Figura 55** que o treinamento realizado com a arquitetura CLSTM (do acrônimo em inglês *Convolutional Long Short-Term Memory*) apresenta bons resultados, semelhante as arquiteturas anteriores, ficando 95% de Acurácia.

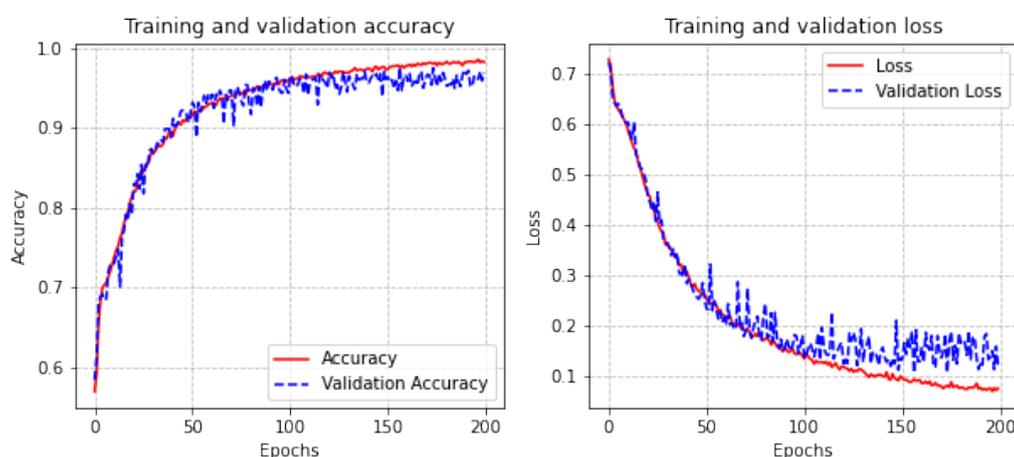


Figura 30 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no primeiro mês

A arquitetura híbrida CLSTM (CNN + LSTM) aproveita-se da capacidade de extração de características dos dados e de redução de dimensionalidade das CNNs, e aproveita-se também da capacidade de manter memória de curto e de longo prazo das LSTMs. Tais fatos conduzem esse tipo de arquitetura a apresentar bons resultados com textos e Séries Temporais. Embora o resultado de Acurácia tenha sido semelhante ao teste com as MLPs (95%), nota-se em menor intensidade a distância entre as curvas de Acurácia de treino e validação, permitindo a essa arquitetura oferecer resultados superiores em relação generalização do modelo.

5.4.4 Arquitetura C-BiLSTM

Como pode ser observado na **Figura 56** o treinamento realizado com a arquitetura C-BiLSTM (do acrônimo em inglês *Convolutional Bidirectional Long Short-Term Memory*) apresentou o melhor resultado de todos os testes realizados (98% de Acu-

rácia), superando, portanto, as demais arquiteturas anteriormente mencionadas e os demais trabalhos do Estado da Arte citados no **Capítulo 3**.

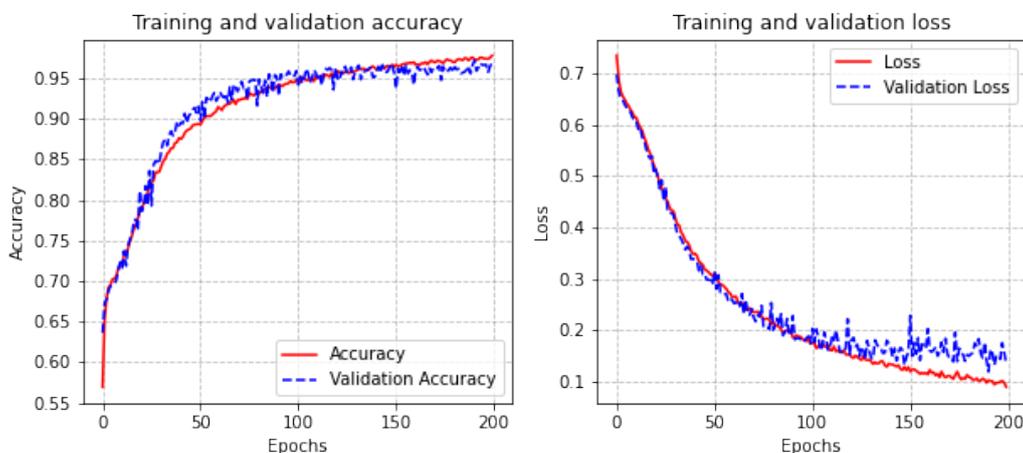


Figura 31 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no primeiro mês

A arquitetura híbrida C-BiLSTM aproveita-se de todas as vantagens citadas para arquitetura CLSTM com a adição da possibilidade de percorrer o vetor em ambas os sentidos temporais, da esquerda para a direita, como se lê um texto no idioma em Português, e da direita para esquerda, no sentido contrario. Esses fatores justificamos os resultados obtidos, tornando essa a melhor arquitetura para suportar as previsões de desempenho dos alunos para o sistema SPPACO. A tabela **Tabela 3** apresenta um resumo com as métricas extraídas dos testes realizados com as quatro arquiteturas.

Tabela 3 – Resultados experimentais com textos postados no primeiro mês de curso

Arquitetura	Acurácia	Erro	AUC	<i>f1_score</i>
MLP	0.95	0.16	0.95	0.94
CNN	0.96	0.10	0.96	0.96
C-LSTM	0.95	0.19	0.95	0.95
C-BiLSTM	0.98	0.12	0.98	0.97

5.4.5 Exemplos de previsões acertados pelo modelo

Na **Tabela 4** podem ser observados exemplos de textos postados pelo alunos e os resultados de previsão obtidos pelo modelo C-BiLSTM para os respectivos textos. Na **Tabela 4** estão representados apenas as previsões acertados pelo modelo.

Tabela 4 – Previsões acertados pelo modelo

Texto Postado	Resultado Real	Resultado Predito
<i>A socialização afeta a produção de identidades pelos agentes sociais à medida que o indivíduo interage com os elementos do contexto social, ativamente e não de forma passiva como ocorre na doutrinação, internalizando a cultura e conhecimentos pertinentes ao grupo e contribuindo com o mesmo através da busca pela satisfação de suas necessidades. A partir dessa interação, é atribuída a cada indivíduo uma ou mais identidades sociais (como é visto perante a sociedade) dependendo de suas características e ligações, e também estruturada a sua auto-identidade (que diz respeito à personalidade e características pessoais). De</i>	Aprovado	Aprovado
<i>Acredito que nas compras públicas não pode haver parceria com fornecedores para que exista transparência nos processos, e não existir preferência ou vantagens para apenas um prestador de serviço. Por mais burocrático que sejam os processos de compras em nosso país, ele tende a ser o mais correto possível para que exista ética nas transações. A verba utilizada é pública, portanto as ações não podem ser prioritárias para poucos. As auditorias (feitas pelo Tribunal de Contas da União, Controladoria Geral da União e outros) estão cada vez mais eficazes e exigentes, para que o dinheiro da população seja utilizado de forma certa e justa para com todos.</i>	Reprovado	Reprovado

5.4.6 Exemplos previsões erradas pelo modelo

Na **Tabela 5** podem ser observados exemplos de textos postados pelo alunos e os resultados de previsão obtidos pelo modelo C-BiLSTM para os respectivos textos. Na **Tabela 5** estão representados apenas as previsões erradas pelo modelo.

Tabela 5 – Previsões erradas pelo modelo

Texto Postado	Resultado Real	Resultado Predito
<i>A socialização e marcada pela estrutura social de uma determinada época, seja ela hoje chamada de geração X, Y, ou outra letra que convier. a distinção entre socialização e doutrinação ou lavagem cerebral esta na capacidade de aguçar o desejo de questionar e aceitar mudanças, conviver com o novo e aceitar quem possa implementar uma nova forma de conviver distinta da minha, isto e relevante pois em vários países Europeus - aceitos como culturalmente mais avançados) – situações envolvendo comportamentos de xenofobia (aversão aos que é diferente) e tem produzido conflitos e movimentos que radicalizam e exercem a doutrinação, motivados pela necessidade de consumir bens e serviços , na qual o "estrangeiro" e visto como empecilho. A identidade e prejudicada pela falta de liberdade e capacidade de mediação de circunstancias, sobre tudo pela falta de dialogo e acesso a grupos radicais.</i>	Reprovado	Aprovado
<i>Responsabilidade social e ética são importantes na administração de uma organização porque além de contribuir com a sociedade e o meio em que atua, irá se auto beneficiar. Hoje em dia é muito comum que investidores pesquisem as empresas antes de investir nelas, e os mesmos, muitas vezes deixam de investir em organizações por não terem um trabalho ético e social.</i>	Aprovado	Reprovado

5.5 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO

Esse capítulo destinou-se a apresentar os testes experimentais realizados para escolha da melhor arquitetura de Redes Neurais Profundas para realizar a previsão do desempenho dos alunos de Cursos EAD da UFSC para suportar o sistema SPPACO. Nesse contexto, a **Seção 5.1** apresentou a natureza e a origem dos dados que foram utilizados para os treinamentos dos modelos. A **Seção 5.2** apresentou a criação da *Embedding Matrix* para configuração inicial dos pesos das redes. A **Seção 5.3** apresentou a parametrização das Redes Neurais. E, por fim, a **Seção 5.4** apresentou os resultados dos treinamentos das arquiteturas de Redes Neurais Profundas testadas.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

6.1 REVISÃO DAS MOTIVAÇÕES E OBJETIVOS

Os cursos de ensino a distância no Brasil foram destaque nos últimos anos no que diz respeito a procura por essa modalidade de ensino. Nesse contexto, assuntos como evasão e reprovação apresentaram-se como desafios importantes e que exigem atenção de toda comunidade educacional. Naturalmente, tais assuntos também atraem a atenção dos pesquisados na busca por técnicas que mitiguem as causas da evasão, assim como também das reprovações.

O levantamento do Estado da Arte realizado por esse trabalho demonstrou a preocupação que a academia tem em propor técnicas para mitigar as principais causas do baixo desempenho dos alunos. Todos os modelos apresentados utilizam Aprendizado de Máquina para criar mecanismos preditivos com intuito de antecipar quais alunos seriam reprovados para, então, adotar alguma medida para evitar o baixo desempenho.

Entretanto, também observou-se que há lacunas para abranger todas as possibilidades de modalidades de ensino, tipos de alunos e diversidade na utilização de tipos de dados. Notou-se, especificamente, que não há trabalhos que destinam-se a propor métodos para fazer a previsão do desempenho de alunos de cursos universitários *Online* a partir dos textos postados pelos alunos ao longo da sua trajetória acadêmica. Portanto, o objetivo geral desse trabalho foi a criação de um sistema baseada em Redes Neurais Profundas e Processamento de Linguagem Natural para um sistema de previsão de performance de alunos de cursos *Online* que fosse eficiente na previsão de alunos que seriam reprovados para contribuir com o combate a evasão.

Visando atender esse objetivo, os seguintes objetivos específicos foram perseguidos:

1. Encontrar as arquiteturas de Redes Neurais Profundas mais adequadas para classificação de textos e para previsão de performance de alunos;
2. Realizar a previsão do desempenho do aluno como um projeto de classificação de textos;
3. Propor e Prototipar uma arquitetura para um sistema de Aprendizado de Máquina para prever o desempenho alunos;

6.2 VISÃO GERAL DO TRABALHO

Com intuito de alcançar os objetivos específicos anteriormente listados, o presente trabalho organizou-se do seguinte modo. No **Capítulo 2** foram apresentados os conceitos de Aprendizado de Máquina e Processamento de Linguagem Natural necessários para suportar o entendimento dos assuntos aqui abordados. Já no **Capítulo 3** foram apresentados os principais trabalhos correlatos ao tema de previsão de desem-

penho de alunos implementados com Aprendizado de Máquina. O **Capítulo 4** trouxe a descrição do SPPACO, como sistema de previsão de desempenho de alunos para cursos *Online*. Por fim, no **Capítulo 5**, foram apresentados os resultados obtidos após a utilização das técnicas de tratamento de dados, testes, escolha e implementação de arquiteturas de Redes Neurais Profundas e a comparação com os modelos listados como estado da arte para previsão do desempenho de alunos.

6.3 CONTRIBUIÇÕES

De acordo com os objetivos definidos para este trabalho, pode-se listar as seguintes contribuições:

1. Identificação de uma arquitetura de Redes Neurais Profundas para realizar a classificação dos textos e, portanto, a previsão do desempenho dos alunos com alta precisão;
2. Elaboração da arquitetura para um Sistema de previsão baseado em Redes Neurais Profundas.
3. Avaliação experimental detalhada dos testes para escolha da Rede Neural Profunda Mais adequada para previsão do desempenho de alunos de cursos *Online*.

Com o objetivo de divulgar os resultados obtidos com o trabalho e, principalmente, submeter seus resultados para uma avaliação da comunidade científica que trata do tema desta dissertação, foi produzido um artigo *Predicting Student Performance of Online Courses with Deep Learning and NLP from Texts in Portuguese (pt-br)* submetido e aceito pela conferência *18th International Conference on Machine Learning and Data Mining (MLDM 2022)*. A classificação Qualis-CC da respectiva conferência é B3.

6.4 LIMITAÇÕES

Embora o trabalho tenha atingido os objetivos desejados, algumas decisões tomadas no seu desenvolvimento trazem limitações para sua reprodutibilidade e utilização com a mesma assertividade encontrada nos testes aqui realizados. Estas limitações são discutidas a seguir.

Ao optar por treinar a própria *Embedding Matrix*, utilizando o corpus obtido dos textos dos alunos oriundos da Plataforma *Moodle*, automaticamente dificultou-se a reprodutibilidade dos resultados, uma vez que para ter acesso a esses dados é necessário solicitar formalmente para comitê de Ética em Pesquisa com Seres Humanos da UFSC (CEPSH).

Outra limitação que pode ser destacada nesse trabalho é a dificuldade de interpretação dos resultados produzidos pela Rede Neural, uma vez que a arquitetura adotada para suportar o sistema de previsão é híbrida e complexa. Os dados são reduzidos por Redes Convolucionais e Analisados em ambas os sentidos temporais dos vetores com Bi-LSTMs e reduzidos ainda mais com uma camada final totalmente conectada. Tais ações, embora profícuas para produção da classificação mais assertiva, dificultam o entendimento dos meandros que acometem com os dados ao passar pela rede.

Um ponto também importante a ser observado é que os registros dos alunos que não deixaram comentários durante a sua participação no respectivo curso, foram excluídos da base e, portanto, não serão utilizadas para treinar o modelo que fará as previsões. Logo, o modelo não será capaz de realizar previsões para esses alunos.

6.5 TRABALHOS FUTUROS

Como trabalhos futuros podemos destacar alguns pontos que possivelmente mitiguem as limitações previamente elencadas. Para contornar a questão do treinamento customizado da *Embedding Matrix*, sugere-se testar os mesmos mecanismos de previsão utilizados por esse trabalho com outras *Word Embeddings* disponíveis publicamente, como por exemplo a utilização do repositório NILC-Embeddings (HARTMANN *et al.*, 2017).

Outro ponto que fica como sugestão de trabalho futuro é a busca por mecanismos de interpretabilidade de estruturas de Redes Neurais Profundas com arquiteturas complexas, com intuito de ofertar aos pretensos usuários dos resultados produzidos por esse trabalho, alguma inferência causal do baixo desempenho dos alunos analisados.

Pode-se destacar também que os testes realizados com as populações segregadas por quantidade de palavras apresentaram aumento dos desempenhos das arquiteturas de rede aplicadas quando testadas com maior números de palavras. Tal fato, pode, empiricamente, sugerir que é mais preciso prever a Aprovação e Reprovação de alunos a partir dos textos com maior número de palavras. Sugere-se, portanto, explorar o assunto com maior detalhe em trabalhos futuros.

Os dados utilizados para o treinamento dos modelos abarca alunos que estiveram matriculados nos cursos até o mês de maio de 2020. Portanto, não foram utilizados exemplos de textos dos alunos durante a pandemia da COVID-19. Sugere-se, dessa forma, reproduzir o estudo com dados mais atuais, inclusive coletados durante a pandemia, para avaliar o comportamento das previsões com os possíveis impactos causados pelas consequências da COVID-19 nos alunos em questão.

Sugere-se também, caso haja dados que suportem, a aplicação das arquiteturas aqui testadas para a previsão direta de evasão. E, por fim, sugere-se testar alguns

mecanismos preditivos baseados na arquitetura Transformers (VASWANI *et al.*, 2017), que tem ganhado significativa relevância com o lançamento de alguns produtos baseados na sua estrutura, como por exemplo o ChatGPT, da empresa Norte Americana OpenAI.

REFERÊNCIAS

ACADEMY, Data Science. **Deep Learning Book**. 2021. Disponível em:

<https://www.deeplearningbook.com.br/>.

AMRIEH, Elaf Abu; HAMTINI, Thair; ALJARAH, Ibrahim. Mining educational data to predict student's academic performance using ensemble methods. **International journal of database theory and application**, NADIA, v. 9, n. 8, p. 119–136, 2016.

BR, EAD. Censo EAD. BR: Relatório analítico da aprendizagem a distância no Brasil 2020. **São Paulo**, 2020.

CHAPMAN, Pete; CLINTON, Julian; KERBER, Randy; KHABAZA, Thomas; REINARTZ, Thomas; SHEARER, Colin; WIRTH, Rüdiger *et al.* CRISP-DM 1.0: Step-by-step data mining guide. **SPSS inc**, v. 9, n. 13, p. 1–73, 2000.

CISLAGHI, Renato *et al.* Um modelo de sistema de gestão do conhecimento em um framework para a promoção da permanência discente no ensino de graduação. Florianópolis, SC, 2008.

EVANGELISTA, Edmund. A hybrid machine learning framework for predicting students' performance in virtual learning environment. **International Journal of Emerging Technologies in Learning (iJET)**, International Journal of Emerging Technology in Learning, v. 16, n. 24, p. 255–272, 2021.

FUNAHASHI, Ken-ichi; NAKAMURA, Yuichi. Approximation of dynamical systems by continuous time recurrent neural networks. **Neural networks**, Elsevier, v. 6, n. 6, p. 801–806, 1993.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GRAVES, Alex; SCHMIDHUBER, Jürgen. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. **Neural networks**, Elsevier, v. 18, n. 5-6, p. 602–610, 2005.

GUERRA JUNIOR A. A., de Oliveira Rech L. Predicting Student Performance of Online Courses with Deep Learning and NLP from Texts in Portuguese (pt-br). *In*: IBAI

PUBLISHING. MACHINE Learning and Data Mining in Pattern Recognition MLDM 2022. [S.l.: s.n.], 2022. P. 33–45.

HARTMANN, Nathan; FONSECA, Erick; SHULBY, Christopher; TREVISO, Marcos; RODRIGUES, Jessica; ALUISIO, Sandra. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. **arXiv preprint arXiv:1708.06025**, 2017.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HUSSAIN, Shah; KHAN, Muhammad Qasim. Student-performulator: Predicting students' academic performance at secondary and intermediate level using machine learning. **Annals of data science**, Springer, v. 10, n. 3, p. 637–655, 2023.

INEP. **Censo da Educação Superior**. [S.l.]: Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep), 2020.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor; TIBSHIRANI, Robert. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

KEHINDE, Adeniyi Jide; ADENIYI, Abidemi Emmanuel; OGUNDOKUN, Roseline Oluwaseun; GUPTA, Himanshu; MISRA, Sanjay. Prediction of students' performance with artificial neural network using demographic traits. *In*: RECENT Innovations in Computing: Proceedings of ICRC 2021, Volume 2. [S.l.]: Springer, 2022. P. 613–624.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KOWSARI, Kamran; JAFARI MEIMANDI, Kiana; HEIDARYSAFA, Mojtaba; MENDU, Sanjana; BARNES, Laura; BROWN, Donald. Text classification algorithms: A survey. **Information**, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 150, 2019.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LI, Mengran; ZHANG, Yong; LI, Xiaoyong; CAI, Lijia; YIN, Baocai. Multi-view hypergraph neural networks for student academic performance prediction. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 114, p. 105174, 2022.

LIU, Gang; GUO, Jiabao. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. **Neurocomputing**, v. 337, p. 325–338, 2019. ISSN 0925-2312.

MAJUMDER, Navonil; PORIA, Soujanya; GELBUKH, Alexander; CAMBRIA, Erik. Deep learning-based document modeling for personality detection from text. **IEEE Intelligent Systems**, IEEE, v. 32, n. 2, p. 74–79, 2017.

MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MONDAL, Arindam; MUKHERJEE, Joydeep. An Approach to predict a student's academic performance using Recurrent Neural Network (RNN). **Int. J. Comput. Appl**, v. 181, n. 6, p. 1–5, 2018.

NORONHA, BN; CARVALHO, Beatriz Montiani; SANTOS, Fabício F Foganhole. Perfil dos alunos evadidos da Faculdade de Economia, Administração e Contabilidade campus Ribeirão Preto e avaliação do tempo de titulação dos alunos atualmente matriculados. **Documento de Trabalho. NUPES–Núcleo de Pesquisa sobre Ensino Superior, Universidade de São Paulo, São Paulo, SP, Brasil**, 2001.

OLIVEIRA, WP de; BITTENCOURT, Wanderley José Mantovani. A evasão na EaD: uma análise sobre os dados e relatórios, ano base 2017, apresentados pelo Inep, UAB e Abed. **Educação Pública**, v. 20, n. 3, 2020.

OTTER, Daniel W.; MEDINA, Julian R.; KALITA, Jugal K. **A Survey of the Usages of Deep Learning in Natural Language Processing**. [S.l.: s.n.], 2019. arXiv: 1807.10854 [cs.CL].

ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

- RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.
- RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3rd. USA: Prentice Hall Press, 2009. ISBN 0136042597.
- SEMESP. **Mapa do Ensino Superior no Brasil-2018**. [S.l.]: Sindicato das Matenedoras do Ensino Superio São Paulo, 2018.
- SILVA, Fernanda Cristina da *et al.* Gestão da evasão na EAD: modelo estatístico preditivo para os cursos de graduação a distância da Universidade Federal de Santa Catarina, 2017.
- SINGH, A; LAARSE, JPR van der. The Asimov Institute, 2021.
- SOBREIRO, Vinicius Amorim; ARAÚJO, PHSL; MENDONÇA, Michelle Magalhães; NAGANO, Marcelo Seido. Uma estimação do valor da commodity de açúcar utilizando redes neurais artificiais. **Pesquisa & Desenvolvimento em Engenharia de Produção**, v. 7, p. 36–52, 2008.
- SURENTHIRAN, S; RAJALAKSHMI, R; SUJATHA, SS. Student performance prediction using atom search optimization based deep belief neural network. **Optical Memory and Neural Networks**, Springer, v. 30, p. 157–171, 2021.
- VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.
- VENTURINI, Jamila *et al.* Censo EAD. BR: Relatório analítico da aprendizagem a distância no Brasil 2014, 2014.
- WANG, Shirui; ZHOU, Wenan; JIANG, Chao. A survey of word embeddings based on deep learning. **Computing**, Springer, v. 102, n. 3, p. 717–740, 2020.
- YADAV, Nikhil; SRIVASTAVA, Kajal. Student performance prediction from e-mail assessments using tiny neural networks. *In: IEEE. 2020 IEEE Integrated STEM Education Conference (ISEC)*. [S.l.: s.n.], 2020. P. 1–6.

YUNITA, Ariana; SANTOSO, Harry Budi; HASIBUAN, Zainal A. Deep Learning for Predicting Students' Academic Performance. *In: IEEE. 2019 Fourth International Conference on Informatics and Computing (ICIC)*. [S.l.: s.n.], 2019. P. 1–6.

YUSOF, Mohammad Hafiz Mohd; ABD KHALID, Ilyani. Precision education reviews: a case study on predicting student's performance using feed forward neural network. *In: IEEE. 2021 International Conference of Technology, Science and Administration (ICTSA)*. [S.l.: s.n.], 2021. P. 1–4.

ZHONG, Junmei; LI, William. Detecting Customer Churn Signals for Telecommunication Industry Through Analyzing Phone Call Transcripts with Recurrent Neural Networks. *In: MLDM (1)*. [S.l.: s.n.], 2019. P. 93–103.

APÊNDICE A – ARQUITETURAS DE REDES NEURAIS

A **Figura 32** é uma coletânea feita pelo Instituto Asimov (SINGH; LAARSE, 2021) das principais arquiteturas de Redes Neurais utilizadas atualmente.

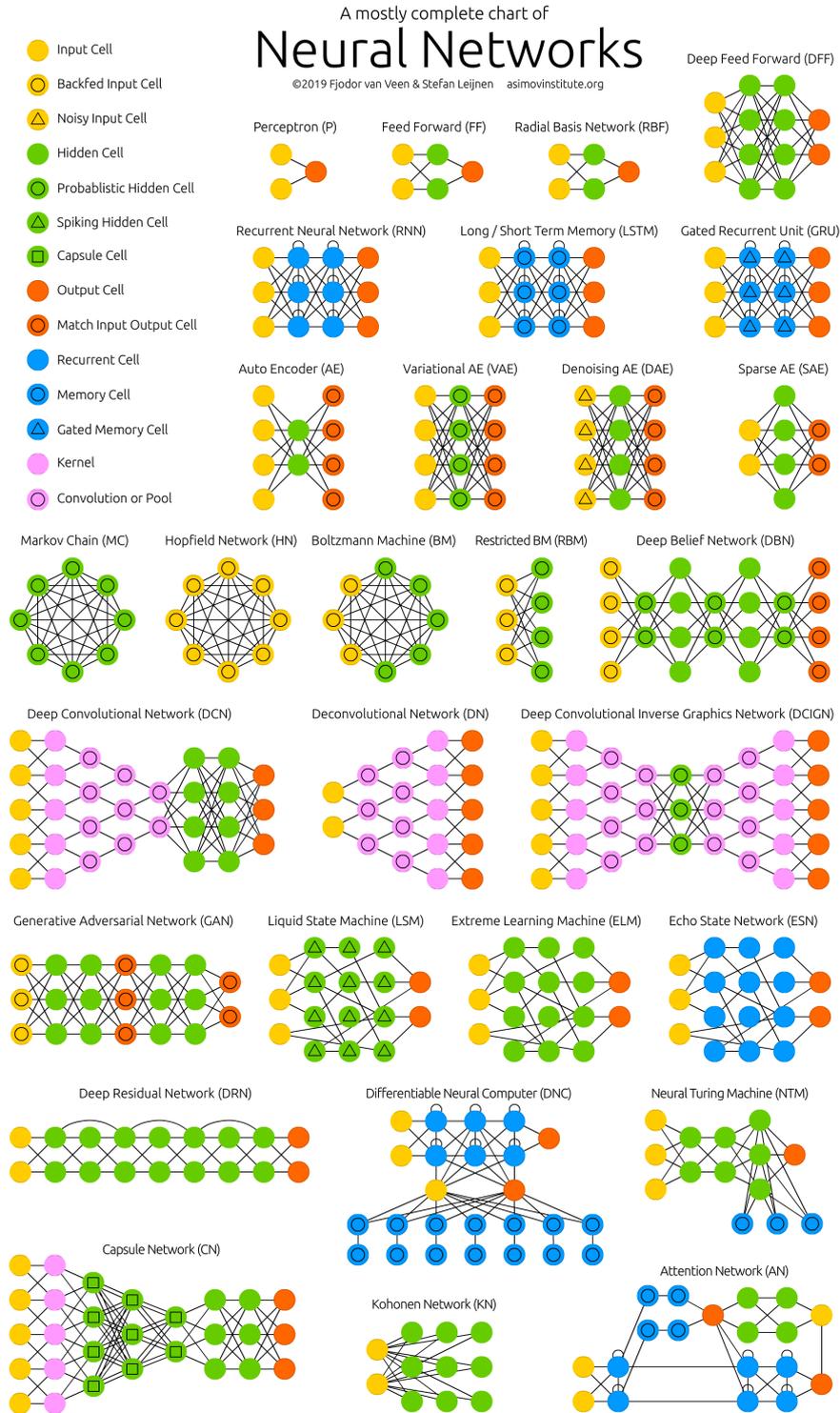


Figura 32 – Arquiteturas de Redes Neurais de acordo com Instituto Asimov (SINGH; LAARSE, 2021)

APÊNDICE A – GRÁFICOS DOS EXPERIMENTOS

A.1 POR QUANTIDADE DE PALAVRAS

A.1.1 De 10 a 50 palavras

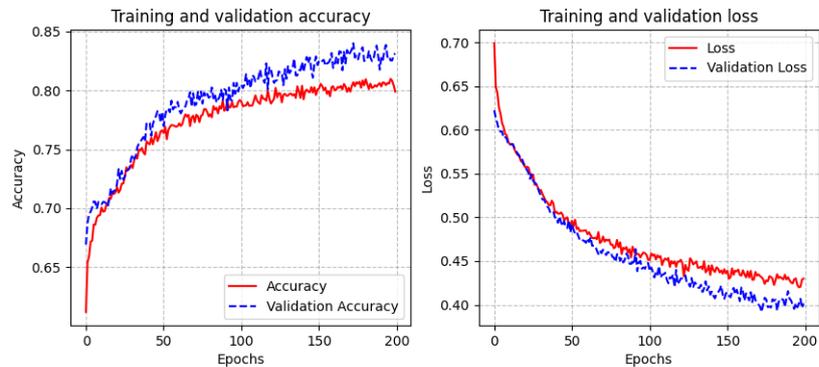


Figura 33 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras

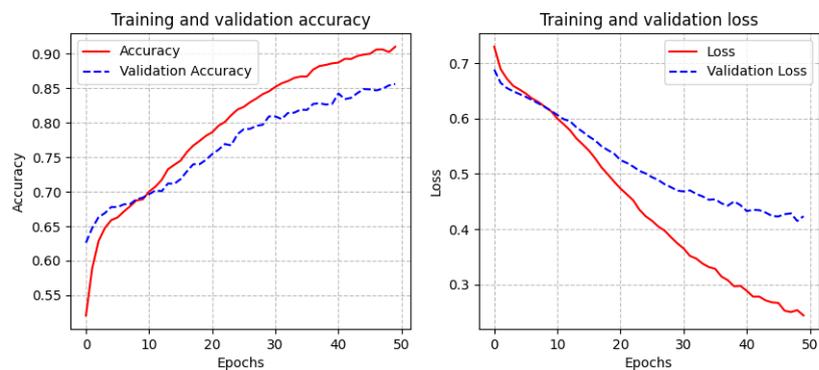


Figura 34 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras

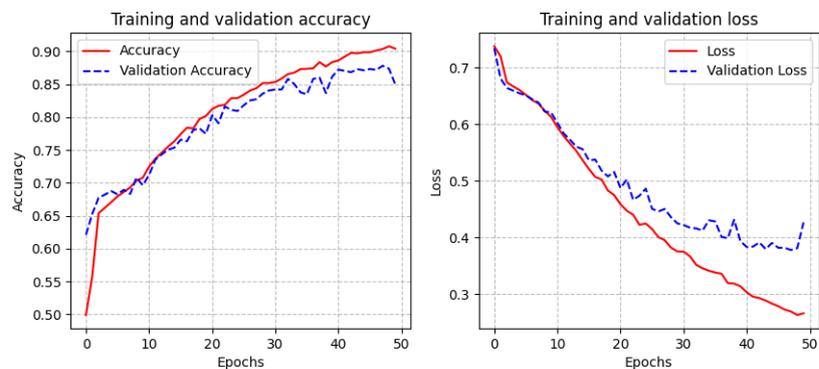


Figura 35 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras

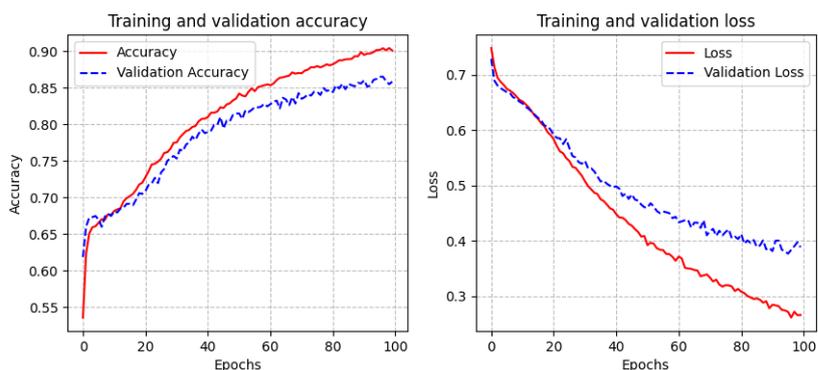


Figura 36 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 10 a 50 palavras

A.1.2 De 50 a 100 palavras

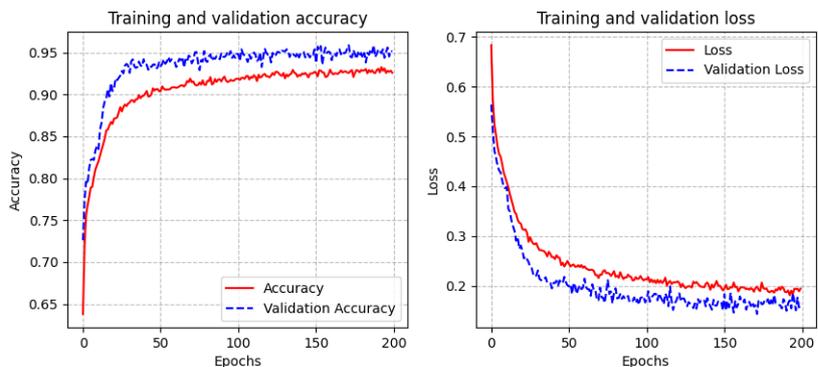


Figura 37 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras

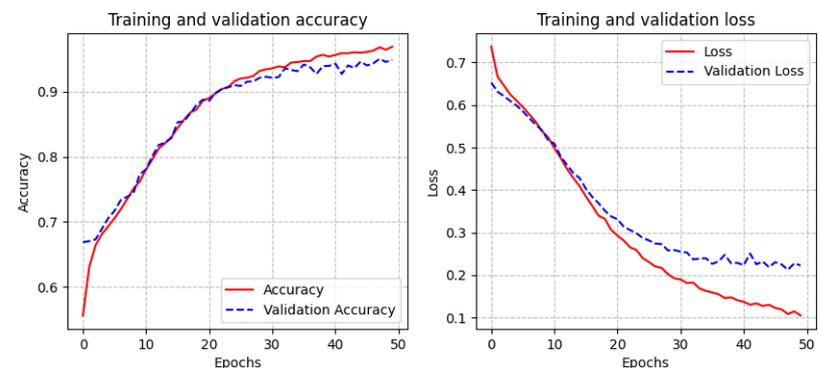


Figura 38 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras

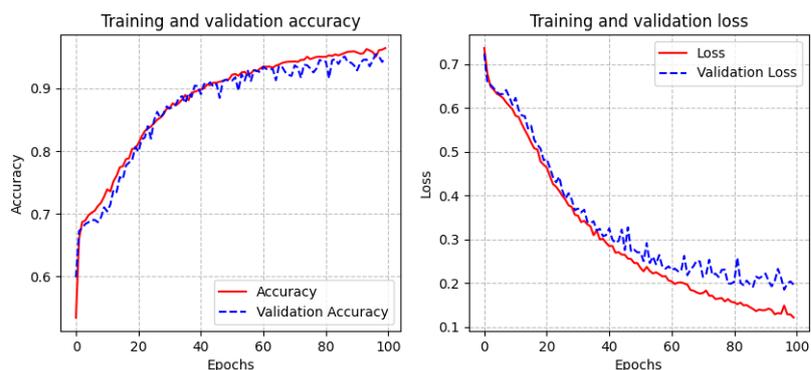


Figura 39 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras

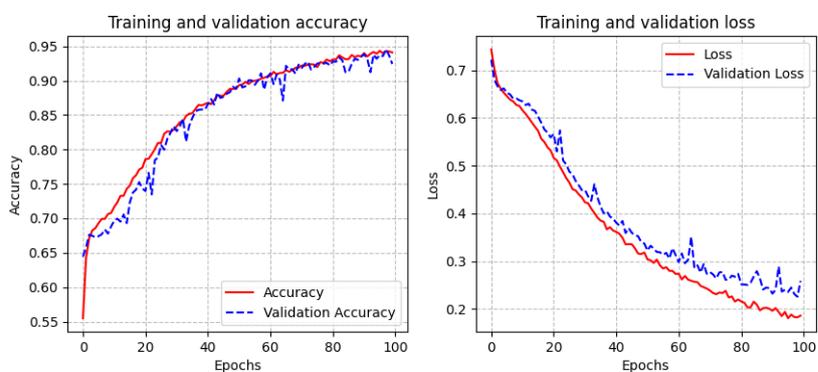


Figura 40 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 50 a 100 palavras

A.1.3 De 100 a 200 palavras

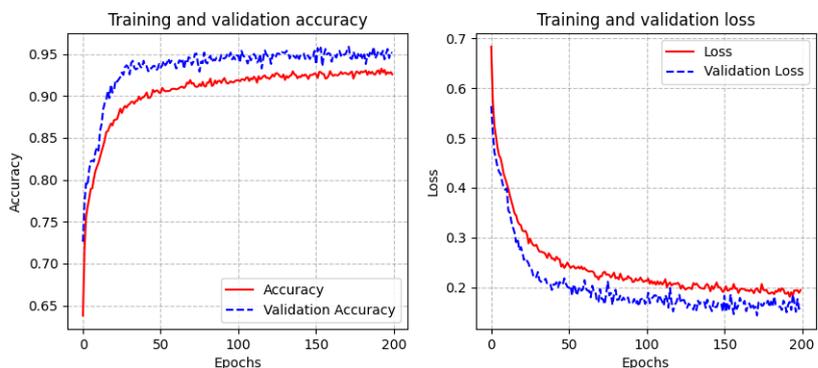


Figura 41 – Histórico da precisão e erros da arquitetura MLP aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras

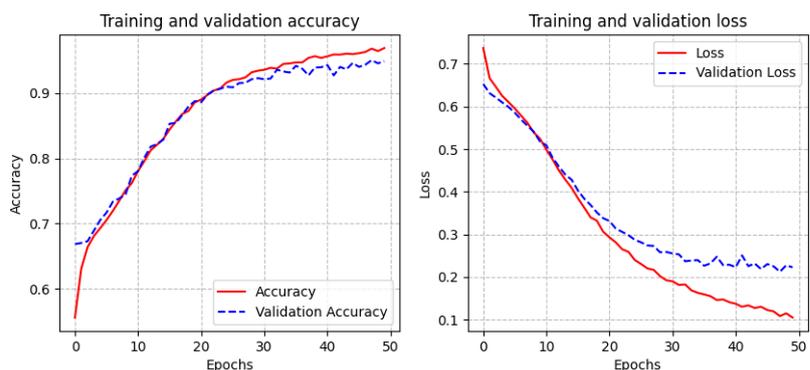


Figura 42 – Histórico da precisão e erros da arquitetura CNN aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras

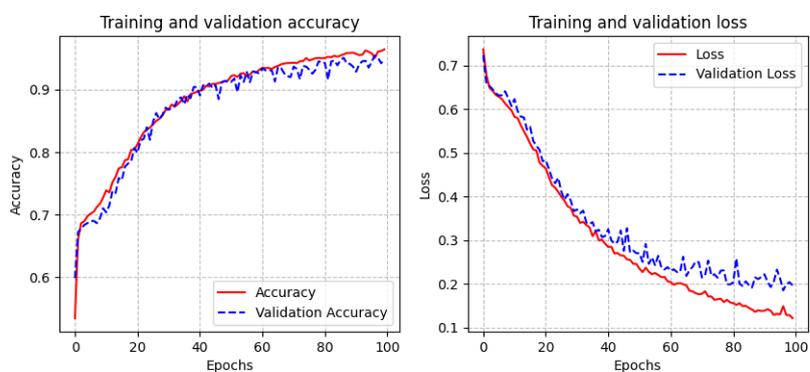


Figura 43 – Histórico da precisão e erros da arquitetura C-LSTM aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras

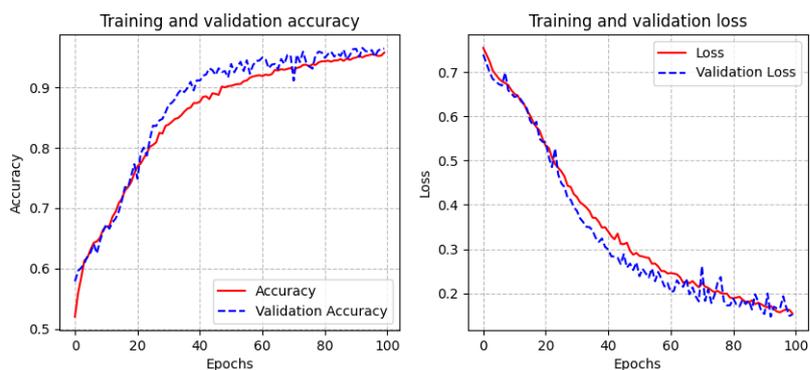


Figura 44 – Histórico da precisão e erros da arquitetura C-BiLSTM aplicada aos dados de treinamento e validação para a categoria de 100 a 200 palavras

A.2 POR DATA DA POSTAGEM

A.2.1 Postados no Primeiro Mês

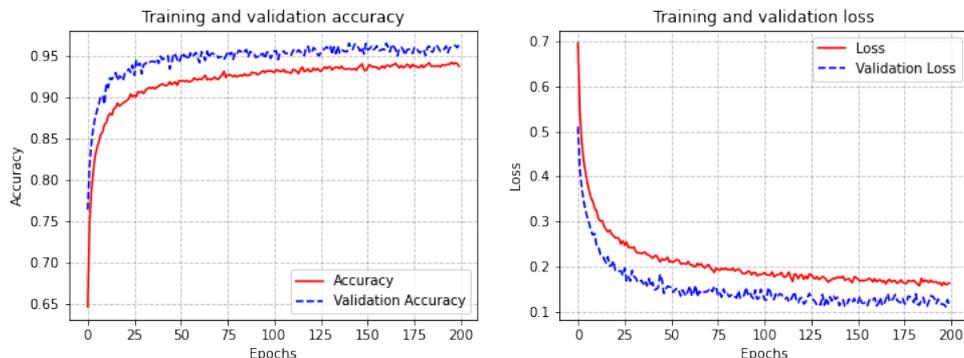


Figura 45 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no primeiro mês

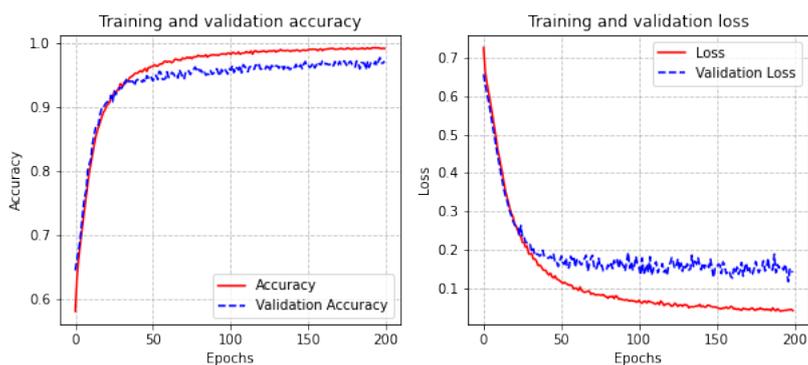


Figura 46 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no primeiro mês

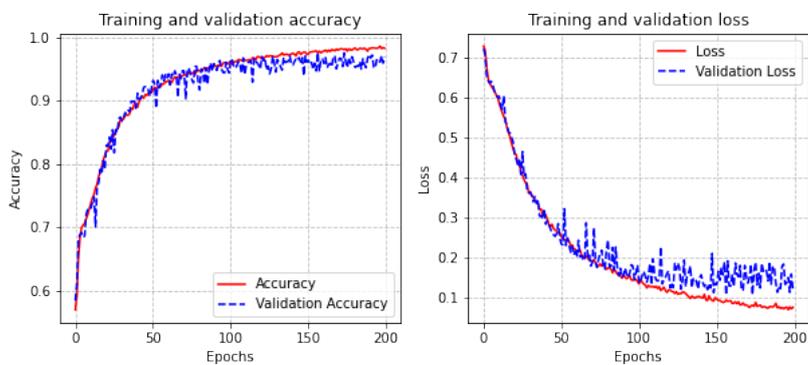


Figura 47 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no primeiro mês

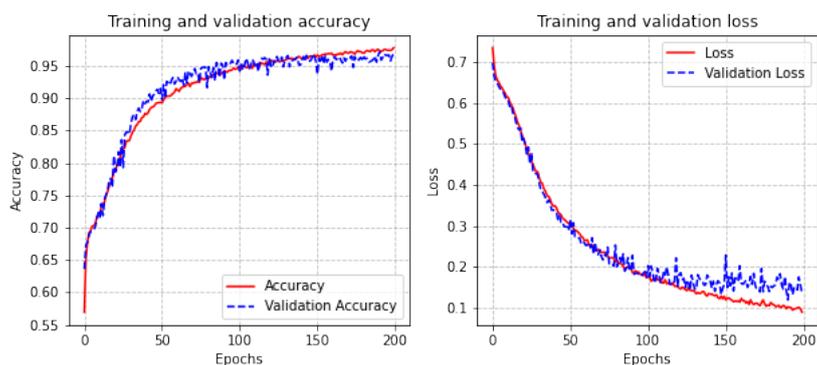


Figura 48 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no primeiro mês

A.2.2 Postados no Segundo Mês

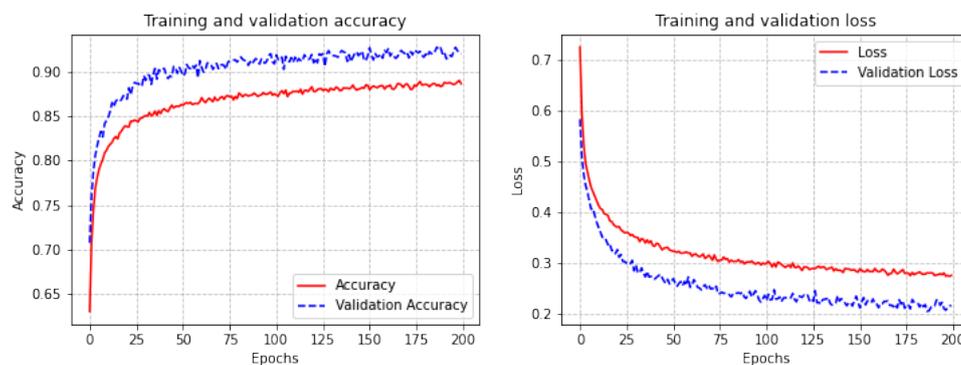


Figura 49 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no segundo mês

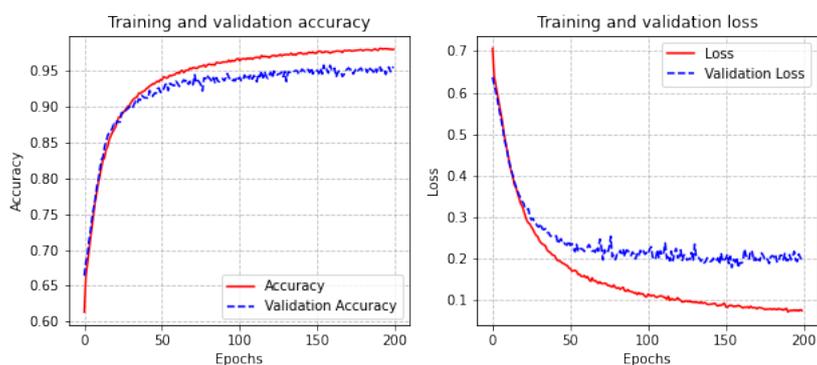


Figura 50 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no segundo mês

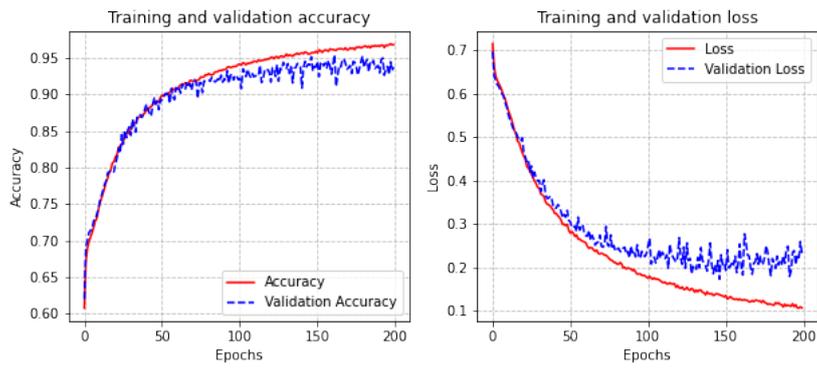


Figura 51 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no segundo mês

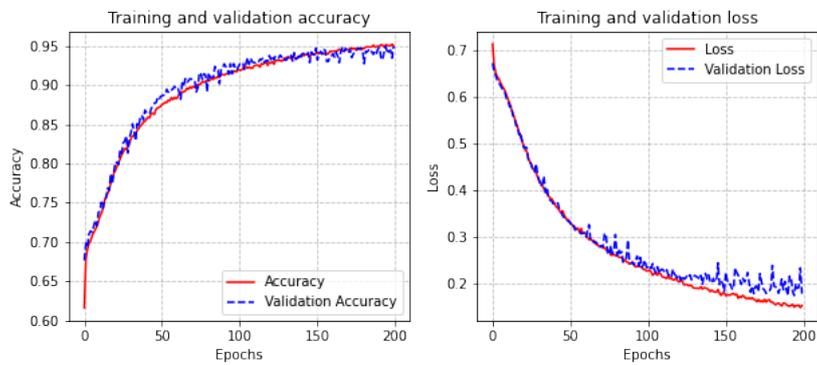


Figura 52 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no segundo mês

A.2.3 Postados no Terceiro Mês

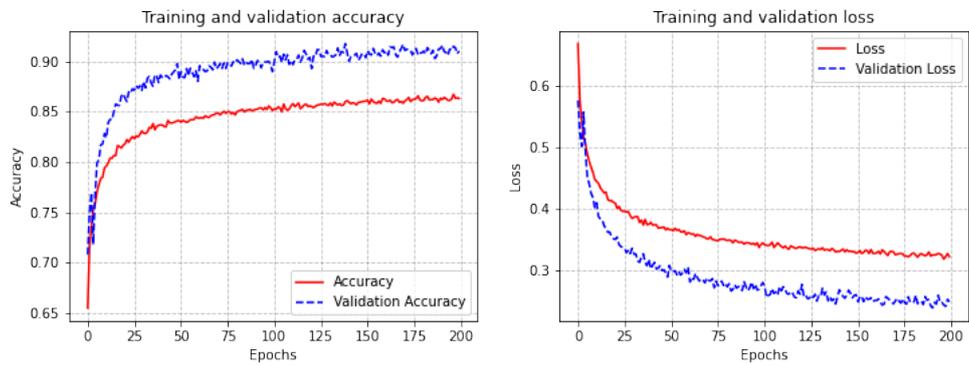


Figura 53 – Histórico da Acurácia e Erro da arquitetura MLP aplicada aos textos postados no terceiro mês

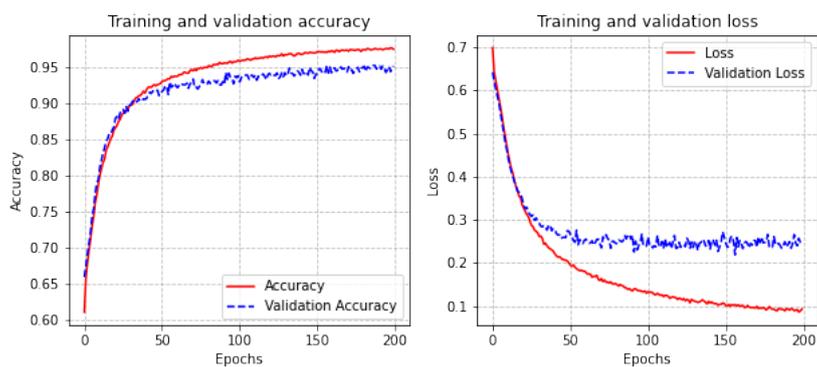


Figura 54 – Histórico da Acurácia e Erro da arquitetura CNN aplicada aos textos postados no terceiro mês

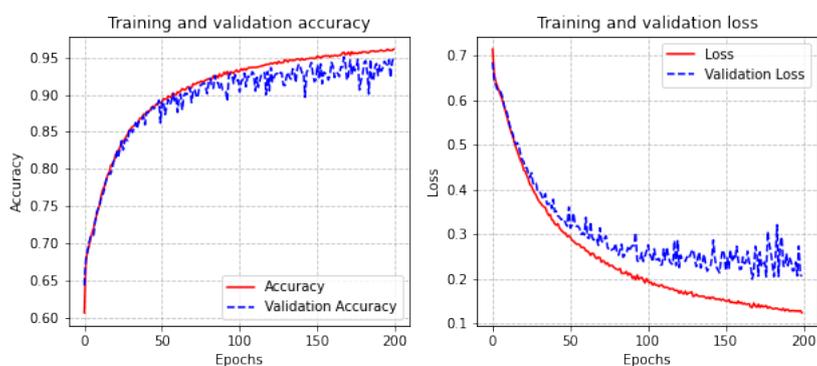


Figura 55 – Histórico da Acurácia e Erro da arquitetura CLSTM aplicada aos textos postados no terceiro mês

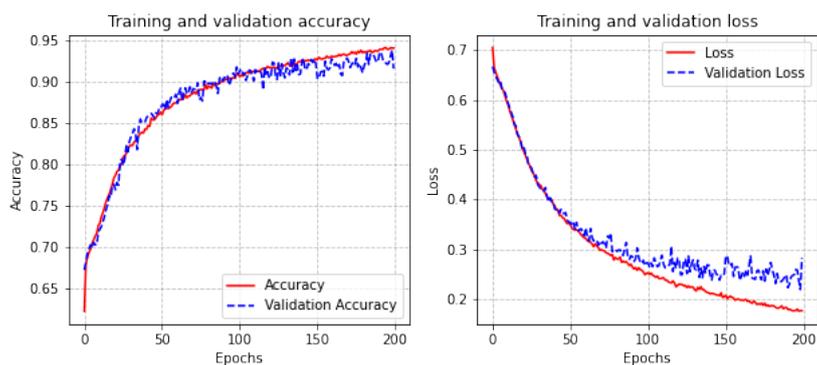


Figura 56 – Histórico da Acurácia e Erro da arquitetura C-BiLSTM aplicada aos textos postados no terceiro mês