UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

EMANOEL AURELIO VIANNA FABIANO

Explorando redes neurais de grafos para predição de interações miRNA–alvo associadas a câncer em grafos heterogêneos

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof[a]. Dr[a]. Mariana Recamonde Mendoza

Porto Alegre
2023

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Fabiano, Emanoel Aurelio Vianna

Explorando redes neurais de grafos para predição de interações miRNA–alvo associadas a câncer em grafos heterogêneos / Emanoel Aurelio Vianna Fabiano. – Porto Alegre: PPGC da UFRGS, 2023.

100 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2023. Orientador: Mariana Recamonde Mendoza.

1. Predição de alvos de miRNAs. 2. Aprendizado de máquina. 3. GraphSAGE. I. Recamonde Mendoza, Mariana. II. Título.

“Eu acredito que às vezes são as pessoas que ninguém espera nada que fazem as coisas que ninguém consegue imaginar.”

— Alan Turing

AGRADECIMENTOS

Primeiramente, à professora Dr.ª Mariana Recamonde Mendoza, pelo privilégio de suas aulas, orientações, conversas, discussões e convivência durante todo o período do mestrado.

Também gostaria de agradecer a todos aqueles que contribuíram de algum modo para este trabalho. Agradeço pelo suporte intelectual e emocional que eu tive em um dos momentos de maior instabilidade e crescimento, pessoal e profissional, da minha vida.

Não poderia deixar de agradecer à minha mãe, Cecilia, que é meu grande exemplo, sempre apoiou os caminhos que escolhi e me incentivou. Por fim, a todos os meus familiares e amigos pelo apoio durante essa jornada.

RESUMO

MicroRNAs (miRNAs) são pequenos RNAs não codificantes que desempenham um papel fundamental na regulação da expressão gênica através da ligação com RNAs mensageiros (mRNAs) alvos. Estudos recentes mostram que os miRNAs estão envolvidos na regulação de mecanismos fisiológicos e de processos patológicos associados a doenças como câncer, sendo portanto importante identificar interações miRNA–alvo. Devido ao grande número de mRNAs alvos que podem existir para um único miRNA, as análises experimentais são bastante demoradas e dispendiosas. Assim, a predição computacional de alvos de miRNAs usando métodos de aprendizado de máquina (AM) tornou-se uma alternativa bastante interessante. Contudo, esta abordagem ainda apresenta limitações, como a complexidade em desenvolver os modelos com dados desbalanceados em razão do pouco número de exemplos negativos disponíveis, assim como o grande número de resultados falsos positivos. Em face destes desafios, este trabalho tem como objetivo desenvolver um modelo baseado em redes neurais de grafos para inferir padrões de interações miRNAs–alvos a partir de grafos heterogêneos, integrando dados de expressão gênica em câncer para introduzir o contexto de alterações moleculares presentes na doença. Exploramos o algoritmo HinSAGE, o qual é uma adaptação do algoritmo GraphSAGE para grafos heterogêneos disponibilizado pela biblioteca StellarGraph, com um grafo contendo interações miRNA–mRNA e mRNA–mRNA obtidas da base de dados RNAInter, e dados de expressão diferencial para 15 tipos de câncer obtidos do projeto *The Cancer Genome Atlas* (TCGA). Nossos resultados indicam que o algoritmo HinSAGE foi capaz de aprender padrões de interações miRNA–alvo a partir da própria estrutura do grafo e dos atributos dos nós, apresentando precisão média de 77%, sensibilidade de 80%, F1-score de 78% e ROC AUC de 86% nos dados de teste. Nosso modelo também mostrou-se competitivo em relação a abordagens relacionadas, destacando-se com acurácia e F1-score sempre próximos aos 90% para interações de teste comuns. Por fim, o aprendizado baseado em grafo apresentou resultados superiores a um modelo treinado com uma rede neural tradicional utilizando somente os padrões de expressão gênica. Assim, acredita-se que o uso de redes neurais de grafos se estabelece como um novo horizonte de estudo na descoberta de interações miRNAs–alvos, possibilitando poder preditivo alto e balanceado, e a amostragem de interações negativas a partir do grafo base.

Palavras-chave: Predição de alvos de miRNAs. aprendizado de máquina. GraphSAGE.

Exploring graph neural networks to predict cancer-associated miRNA-target interactions in heterogeneous graphs

ABSTRACT

MicroRNAs (miRNAs) are small non-coding RNAs that are crucial in regulating gene expression by binding to messenger RNAs (mRNAs) targets. Recent studies show that miRNAs regulate physiological mechanisms and pathological processes associated with diseases such as cancer. Therefore, it is essential to identify miRNA-target interactions. Due to the large number of target mRNAs that can exist for a single miRNA, experimental analyzes are pretty time-consuming and expensive. Thus, computational prediction of miRNA targets using machine learning (ML) methods has become an exciting alternative. However, this approach still has limitations, such as the complexity of developing models with imbalanced data due to the small number of negative examples available and many false positive results. Motivated by these challenges, this work aims to develop a model based on graph neural networks (GNNs) to infer patterns of miRNA-target interactions from heterogeneous graphs, integrating cancer gene expression data to introduce the context of molecular alterations present in the disease We explored the HinSAGE algorithm, which is an adaptation of the GraphSAGE algorithm for heterogeneous graphs provided by the StellarGraph library, with a graph containing miRNA–mRNA and mRNA–mRNA interactions obtained from the RNAInter database, and differential expression data for 15 cancer types from the The Cancer Genome Atlas (TCGA) project. Our results indicate that the HinSAGE algorithm was able to learn patterns of miRNA-target interactions from the graph structure itself and the node attributes, with an average precision of 77%, sensitivity of 80%, F1-score of 78 %, and ROC AUC of 86% on test data. Our model also proved competitive concerning related approaches, standing out with accuracy and F1-score close to 90% for common test interactions. Finally, graph-based learning presented better results than a model trained with a traditional neural network using only gene expression patterns. Thus, we believe that the use of GNNs establishes itself as a new horizon of study in the discovery of miRNA-target interactions, allowing high and balanced predictive power, and the sampling of interactions negatives from the base graph.

Keywords: miRNA target prediction. machine learning. GraphSAGE..

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
ANN	*Artificial Neural Networks*
DNA	Ácido Desoxirribonucleico
DNN	*Deep Neural Network*
DFF	*Deep Feedforward*
FN	Falsos Negativos
FP	Falsos Positivos
GNN	*Graph Neural Network*
IA	Inteligência Artificial
miRNA	MicroRNA
mRNA	RNA mensageiro
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
RNA	Ácido Ribonucleico
ROC	*Receiver Operating Characteristic*
ROC AUC	Área sob a curva ROC
SVM	*Support Vector Machine*
TCGA	*The Cancer Genome Atlas*
TFP	Taxa de Falsos Positivos
TVP	Taxa de Verdadeiros Positivos
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos

LISTA DE FIGURAS

Figura 2.1 Biogênese canônica de microRNAs e sua ação sobre o mRNA alvo. 19
Figura 2.2 Representação de um neurônio artificial utilizado como unidade de processamento em redes neurais artificiais. 24
Figura 2.3 Arquitetura de uma rede MLP *feedforward*. Para simplificar a ilustração da rede a função de ativação presente na saída do neurônio foi omitida. 25
Figura 2.4 Exemplo de grafo com 7 vértices e 8 arestas. 28
Figura 2.5 Exemplos de grafos contendo diferentes características em sua estrutura. ...29
Figura 2.6 Ilustração do funcionamento do GraphSAGE. 30
Figura 2.7 Matriz de confusão. 33
Figura 2.8 Exemplo de curva ROC para avaliação de modelos preditivos. 34
Figura 2.9 Ilustração dos diferentes tipos de correlações que podem existir entre duas variáveis contínuas. 35

Figura 4.1 Exemplo da estrutura do grafo proposto, com os nodos em vermelho representando miRNAs e os nodos em cor rosa claro denotando os mRNAs. Cada nodo possui um vetor de atributos composto pelos valores de expressão diferencial nos vários tipos de câncer analisados. As arestas são a representação das interações miRNA–mRNA ou mRNA–mRNA. 46
Figura 4.2 Histograma da distribuição original das interações considerando o *score* de confiança para registros descritos como *Strong* e *Weak*. 47
Figura 4.3 Histograma da distribuição das interações classificadas como *Strong* e *Weak* após a aplicação de filtros sobre o *score* de confiança. 48
Figura 4.4 Quantidade de interações descritas como forte e fraca sobre os diferentes conjuntos de dados criados sobre o *score* de confiança. 49
Figura 4.5 Ilustração do comportamento de indução que ocorre no algoritmo HinSAGE para a predição de links considerando o grafo heterogêneo e a tarefa de predição de alvos de miRNAs. 50
Figura 4.6 Processo de criação de conjuntos de teste, validação e treinamento a partir do grafo original. As arestas tracejadas representam arestas positivas amostradas em cada etapa, as quais são removidas do grafo para as etapas subsequentes. 53

Figura 5.1 Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 1. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 60
Figura 5.2 Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 1. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 61
Figura 5.3 Análise de desempenho nos dados de teste para os experimentos do Grupo 1. 62
Figura 5.4 Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 2. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 64
Figura 5.5 Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 2. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 64
Figura 5.6 Análise de desempenho nos dados de teste para os experimentos do Grupo 2. 65

Figura 5.7 Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 3. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 66
Figura 5.8 Análise de desempenho nos dados de teste para os experimentos do Grupo 3. 66
Figura 5.9 Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 4. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 67
Figura 5.10 Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 4. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 68
Figura 5.11 Análise de desempenho nos dados de teste para os experimentos do Grupo 4. 69
Figura 5.12 Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 5. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 70
Figura 5.13 Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 5. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 71
Figura 5.14 Análise de desempenho nos dados de teste para os experimentos do Grupo 5. 71
Figura 5.15 Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 6. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 72
Figura 5.16 Análise de desempenho nos dados de teste para os experimentos do Grupo 6. 73
Figura 5.17 Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 7. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 75
Figura 5.18 Análise de desempenho nos dados de teste para os experimentos do Grupo 7. 76
Figura 5.19 Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 8. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções. 77
Figura 5.20 Análise de desempenho nos dados de teste para os experimentos do Grupo 8. 78
Figura 5.21 Análise das curvas ROC para as 10 execuções do cenário do Experimento 14. 79
Figura 5.22 Métricas alcançadas para a execução de 10 experimentos sobre o modelo de aprendizado de máquina MLP. 87

Figura B.1 Correlação do *score* predito e do valor esperado para cada execução realizada do modelo MLP. 100

LISTA DE TABELAS

Tabela 4.1 Quantidade de interações do tipo (a) miRNA–mRNA e (b) mRNA–mRNA após o pré-processamento dos dados obtidos do RNAInter v4.0. 43
Tabela 4.2 Quantidade de amostras por tipo de câncer, grupo amostral e tipo de dado após a realização do pré-processamento dos dados de expressão gênica. 45
Tabela 4.3 Apresentação da construção do conjunto de treinamento, validação e teste .52
Tabela 4.4 Lista de hiperparâmetros e respectivos valores usados inicialmente nos experimentos. 54

Tabela 5.1 Definição dos cenários experimentais explorados no conjunto de experimentos **C1**, para explorar o potencial do algoritmo HinSAGE na predição de alvos de miRNAs. 56
Tabela 5.2 Tabela dos resultados obtidos para todas as execuções do cenário de experimento 14. Nos resultados apresentados destaca-se a execução cinco que julgamos com um bom resultados sobre todos as métricas apresentadas. 79
Tabela 5.3 Comparação de desempenho entre o modelo baseado em HinSAGE proposto neste trabalho e outras abordagens da literatura. 83
Tabela 5.4 Resumo da divisão dos dados em conjuntos de treinamento, validação e teste para a comparação com o algoritmo MLP. 85
Tabela 5.5 Resultados do coeficiente de correlação de Pearson e do erro quadrático médio para as predições do modelo MLP nos dados de teste. 86

Tabela A.1 Protocolo definido para a busca de trabalhos relacionados. 98
Tabela A.2 Revisão da Literatura 99

SUMÁRIO

1 INTRODUÇÃO 13
1.1 Justificativa 15
1.2 Descrição da estrutura do trabalho 17
2 FUNDAMENTAÇÃO TEÓRICA 18
2.1 MicroRNAs: da biogênese à expressão diferencial 18
2.1.1 Biogênese dos microRNAs 18
2.1.2 Relação de microRNAs com doenças humanas 20
2.1.3 Análise de expressão gênica em larga escala 21
2.2 Aprendizado de máquina tradicional 22
2.2.1 Aprendizado supervisionado e não-supervisionado 23
2.2.2 Redes Neurais Artificiais 24
2.3 Aprendizado profundo em grafos 27
2.3.1 Breve introdução à teoria dos grafos 28
2.3.2 Redes neurais de grafos e o algoritmo GraphSAGE 29
2.4 Avaliação de modelos preditivos 31
2.4.1 Divisão de dados com o método Holdout 32
2.4.2 Métricas de desempenho 32
3 TRABALHOS RELACIONADOS 36
3.1 Discussão 40
4 MATERIAIS E MÉTODOS 41
4.1 Coleta e pré-processamento de dados 41
4.1.1 Dados de interações miRNA–mRNA e mRNA–mRNA 41
4.1.2 Dados de expressão gênica em câncer 43
4.2 Integração de dados para geração do grafo 46
4.3 Análise exploratória do conjunto de dados e critérios de filtragem 47
4.4 Desenvolvimento do modelo preditivo 49
4.4.1 Geração dos dados de treinamento, validação e teste 51
4.4.2 Treinamento e avaliação do modelo baseado em grafos 53
5 EXPERIMENTOS E RESULTADOS 55
5.1 Definição dos experimentos 55
5.1.1 C1: Análise experimental do desempenho preditivo do algoritmo HinSAGE 55
5.1.2 C2: Comparação do modelo baseado no HinSAGE com outras abordagens 57
5.2 Resultados 58
5.2.1 Análise experimental do desempenho preditivo do algoritmo HinSAGE 59
5.2.1.1 G1: Impacto de variações no conjunto de interações 59
5.2.1.2 G2: Impacto da remoção de interações mRNA–mRNA em grafos filtrados 63
5.2.1.3 G3: Impacto da redução do número de épocas de treinamento 65
5.2.1.4 G4: Impacto da variação no tamanho de *batch* usado no treinamento 67
5.2.1.5 G5: Impacto da variação no número de nós vizinhos amostrados 69
5.2.1.6 G6: Impacto da variação do tamanho das camadas ocultas 72
5.2.1.7 G7: Impacto da variação na taxa de aprendizado 74
5.2.1.8 G8: Impacto do aumento no número de épocas de treinamento com base no melhor modelo 77
5.2.1.9 Sumário do desempenho do melhor modelo HinSAGE 78
5.2.2 Comparação do modelo baseado no HinSAGE com outras abordagens 80
5.2.2.1 Comparação com diferentes trabalhos relacionados 80
5.2.2.2 Comparação com uma rede neural artificial 84

6 CONSIDERAÇÕES FINAIS 88
6.1 Dificuldades Encontradas 89
6.2 Trabalhos Futuros 90
REFERÊNCIAS 91
APÊNDICE A — INFORMAÇÕES ADICIONAIS DA REVISÃO DA LITERATURA 98
APÊNDICE B — ANÁLISE DE CORRELAÇÃO PARA PREDIÇÕES REALIZADAS PELO MODELO MLP 100

1 INTRODUÇÃO

Os microRNAs (miRNAs) são pequenas moléculas de ácido ribonucleico (RNA) não codificantes, descritos pela primeira vez em 1993 no nematoide *Caenorhabditis elegans* (LEE; FEINBAUM; AMBROS, 1993). Inicialmente, os miRNAs foram considerados transcritos não funcionais devido à sua incapacidade de codificar proteínas. Eles estavam contidos na enorme parcela do DNA (ácido desoxirribonucleico) considerada como "DNA lixo" por não ter um papel caracterizado de gene codificante de proteína (BUDAK; ZHANG, 2017). No entanto, atualmente, sabe-se que os miRNAs desempenham um papel importante como reguladores da expressão gênica[1] em etapas pós-transcricionais, principalmente inibindo a expressão de genes específicos em diferentes espécies de plantas e animais através do silenciamento do RNA mensageiro (mRNA, de *messenger RNA*) (O'BRIEN et al., 2018).

Estudos voltados à investigação da atuação dos miRNAs como reguladores da expressão gênica indicam que um único miRNA pode ter como alvo muitos mRNAs diferentes, enquanto um determinado mRNA pode ser regulado por um conjunto de miRNAs, simultaneamente ou de maneira dependente do contexto (PLOTNIKOVA; BARANOVA; SKOBLOV, 2019). Adicionalmente, miRNAs são reguladores de outros tipos de moléculas de RNAs, e não somente mRNAs como acreditava-se inicialmente, tornando-os importante atores na complexa rede regulatória que governa a expressão gênica (TAY; RINN; PANDOLFI, 2014). Consequentemente, a expressão anômala de miRNAs tem sido associada ao desenvolvimento e progressão de diferentes patologias humanas, inclusive aquelas relacionados ao sistema endócrino e ao câncer (LI; KOWDLEY, 2012; DAVIS-DUSENBERY; HATA, 2010).

Dado o importante papel que miRNAs podem desempenhar nos processos biológicos e no desenvolvimento de doenças, compreender a atuação dos miRNAs através da caracterização das interações entre estes pequenos RNAs e seus respectivos alvos pode auxiliar no diagnóstico e no tratamento de doenças ainda em estágios iniciais. Nos últimos anos, diversos métodos e ferramentas computacionais têm sido desenvolvidos para elucidar os mecanismos de atuação de miRNAs através da identificação de seus alvos (FAN; KURGAN, 2015). Essas ferramentas abrangem diferentes metodologias de predição, desde a modelagem de interações físicas até a incorporação de algoritmos de aprendizado de máquina (AM) (SCHÄFER; CIAUDO, 2020). Devido ao crescimento expo-

[1] A expressão gênica é definida como o processo no qual as instruções contidas no DNA são convertidas em um produto funcional, como proteínas ou moléculas funcionais de RNA.

nencial da quantidade de dados biológicos disponíveis, a predição de alvos de miRNAs com algoritmos de AM tornou-se um caminho bastante promissor (CHEN et al., 2019).

As abordagens baseadas em AM visam tornar o processo de validação experimental de um potencial alvo de miRNA mais rápida e barata através da predição de alvos candidatos utilizando uma série de descritores da interação entre miRNA e mRNA alvos definidos a partir de interações já caracterizadas experimentalmente (PARVEEN et al., 2019). Conforme novos avanços são feitos na caracterização de interações miRNAs-mRNAs alvos, novos descritores (*i.e.,* atributos) são adicionados aos métodos preditivos. Por exemplo, descritores relacionados à complementariedade entre as sequências de miRNA e alvo, especialmente na região de *seed* do miRNA composta pelos nucleotídeos 2–8, à conservação evolutiva destas sequências, à termodinâmica da ligação entre regulador e alvo, e à acessibilidade do sítio de ligação no mRNA são comumente utilizados em ferramentas preditivas (PETERSON et al., 2014). No entanto, de acordo com Peterson et al. (2014), cada método possui limitações em seu poder preditivo em decorrência dos atributos selecionados, como a seleção de regiões comumente suscetíveis à interação. Essa definição manual acaba por capturar apenas parcialmente as inúmeras características que influenciam a efetividade de uma interação entre miRNA e mRNA alvo.

Limitações na identificação de alvos de miRNAs a partir de abordagens baseadas em AM também são introduzidas em nível de algoritmo. Algoritmos de AM são muito dependentes e sensíveis à qualidade dos dados de treinamento e ao conjunto de atributos definidos manualmente para desenvolvimento do modelo (FAN; KURGAN, 2015). Um desafio adicional é o pequeno número de exemplos de pares não funcionais de miRNA e mRNA alvo disponíveis em bancos de dados com dados validados experimentalmente, como MirTarBase (HUANG et al., 2021). Isto gera um desbalanceamento significativo entre exemplos positivos (funcionais) e negativos (não funcionais) nos dados de treinamento usados para o desenvolvimento de modelos preditivos com AM, o que tende a degradar o desempenho dos modelos ao introduzir um viés de aprendizado para a classe majoritária (isto é, positiva). Um dos impactos práticos deste viés é o grande número de falsos positivos nas predições retornadas pelos métodos existentes (PINZÓN et al., 2017).

Assim, a inerente complexidade biológica do problema em questão e os diversos desafios analíticos que surgem na análise computacional destes dados, mantém o problema de predição de alvos de miRNAs como um desafio em aberto na bioinformática. Ao longo dos últimos anos foram desenvolvidos inúmeros métodos de aprendizado de máquina profundo, com diversas aplicações bem sucedidas na biologia e na medicina

(CHING et al., 2018). Dentre estes, destacamos os algoritmos de aprendizado profundo em grafos, conhecidos como *Graph neural networks* (GNNs), que vão se mostrando métodos promissores para a resolução de problemas complexos onde a representação natural dos dados não ocorre em um domínio euclidiano (CAI; ZHENG; CHANG, 2018; ZHOU et al., 2020). O objetivo do presente trabalho é abordar o problema de predição de alvos de miRNAs explorando o potencial de GNNs para análise de padrões em dados complexos e descritos na forma de grafos. Como foco de estudo, investigamos as interações entre miRNAs–alvos associadas a câncer, motivados pelas fortes evidências do papel de miRNAs desregulados na fisiopatologia do câncer e como potenciais biomarcadores de diagnóstico, prognóstico ou como alvos terapêuticos (PENG; CROCE, 2016).

A fim de reduzir as limitações impostas pelos tipos específicos de atributos selecionados, a abordagem proposta visa aprender os padrões entre miRNAs e alvos funcionais diretamente a partir de interações em um grafo, o qual no escopo deste trabalho é definido a partir de relações miRNA–alvo e alvo–alvo. Além da mineração dos padrões relacionados à estrutura do grafo, as GNNs permitem integrar ao aprendizado características específicas dos nós das redes, aqui representados como padrões de expressão diferencial de miRNAs e mRNAs em câncer. Com a representação das relações miRNA–alvo e alvo–alvo na estrutura do grafo temos uma arquitetura heterogênea. Dada essa característica heterogênea do grafo, o modelo de predição foi desenvolvido utilizando um tipo de GNN específico que é capaz de lidar com a heterogeneidade presente nos dados sendo um dos algoritmos mais bem sucedidos na literatura, denominado GraphSage (HAMILTON; YING; LESKOVEC, 2017). Através deste trabalho, espera-se contribuir não só com o desenvolvimento de uma metodologia clara e objetiva para a coleta, preparação, e integração de dados de diferente natureza para predição de miRNAs–alvos com algoritmos de aprendizado em grafos, mas também com a mitigação do problema de falsos positivos reportado na literatura relacionada.

1.1 Justificativa

Apesar da relevância científica e clínica, a predição de alvos de miRNAs ainda é considerado um problema em aberto pelas dificuldades inerentes a esta tarefa analítica. Alguns fatores principais tornam esta tarefa tão difícil. Primeiramente, a multiplicidade de padrões de interações miRNA–alvo, com muitos alvos por miRNAs e muitos miRNAs atuando sobre um único mRNA alvo, inclusive em muitos casos de forma dependente do

contexto. Em segundo lugar, o desbalanceamento entre classes, visto que a grande maioria dos dados validados se referem a interações consideradas funcionais, isto é, exemplos positivos de interações miRNA–alvo. Em terceiro lugar, cada tipo de atributo utilizado como descritor no desenvolvimento de modelos preditivos é capaz de capturar apenas parcialmente as características que tornam um mRNA um alvo funcional (*i.e.,* verdadeiro) de um miRNA.

Desta forma, a premissa que guia o presente trabalho é que o desenvolvimento de uma estratégia computacional baseada em métodos capazes de aprender padrões de regulação de miRNAs a partir da análise de redes de interações poderia fornecer predições mais a robustas a ruídos e deficiências nos dados de treinamentos, bem como melhor capturar a complexidade envolvida na atuação dos miRNAs. No presente trabalho, este método é representado pelo algoritmo GraphSage, o qual é capaz de analisar redes com interações de diferente natureza, como miRNA–alvo e alvo–alvo. Adicionalmente, visando minimizar o efeito negativo da seleção de tipos específicos de atributos descrevendo as características de uma interação miRNA–alvo, o presente trabalho utiliza como hipótese a adição de dados de expressão diferencial de miRNA e mRNAs em câncer adicionalmente à estrutura do grafo.

A proposta de usar dados de expressão tem como objetivo olhar para o aspecto da dinâmica de regulação, a qual é o resultado de um conjunto de fatores moleculares que incluem, dentre outros, complementariedade de sequência, conservação de sequência, termodinâmica favorável e acessibilidade do sítio de ligação do miRNA – todas características que são usualmente integradas de forma individual ou combinada em métodos de predição. Salienta-se ainda que a análise da correlação de perfis de expressão para identificar alvos funcionais de miRNAs associados a contextos específicos, como uma doença de interesse, tem sido um princípio básico da análise integrativa de dados de transcriptoma em diversos trabalhos da bioinformática (DAI; ZHOU, 2010). Esta abordagem tem a vantagem de permitir detectar não somente potenciais mRNAs alvos diretos, isto é, que agem sobre uma interação física entre miRNA e alvo, mas também a relação entre miRNAs e outros alvos secundários que são igualmente impactados pela desregulação da atividades dos miRNAs, mas de forma indireta.

1.2 Descrição da estrutura do trabalho

Este trabalho está organizado da seguinte forma: O Capítulo 2 apresenta a fundamentação teórica, onde descrevemos os conceitos básicos relacionados ao dogma central da biologia molecular, assim como também a relação entre os miRNAs com câncer em humanos. Ainda neste capítulo também apresentamos uma revisão sobre diversos conceitos computacionais, incluindo modelos de aprendizado de máquina e conceitos relacionados à teoria dos grafos. No Capítulo 3 é apresentado um estudo sobre alguns dos trabalhos relacionados. O Capítulo 4 retrata a nossa metodologia aplicada ao longo do desenvolvimento desta proposta, desde a escolha e análise do conjunto de dados até a etapa da definição do algoritmo mais adequado ao nosso problema, bem como a construção do modelo e organização do conjunto de testes propostos. Neste capítulo também estão descritas as decisões tomadas durante a fase do desenvolvimento. Capítulo 5 descreve os resultados alcançados sobre os diversos experimentos propostos realizados, assim como na comparação com um modelo de aprendizado de máquina clássico e trabalhos relacionados. Por fim, no Capítulo 6 são descritas as considerações finais sobre o trabalho realizado.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos biológicos e computacionais envolvidos no presente trabalho e importantes para a compreensão do mesmo. O capítulo inicia com uma introdução a respeito da origem e atuação dos miRNAs, e segue para uma explicação acerca de métodos de AM, teoria de grafos e algoritmos de aprendizado baseado em grafos. Por fim, o capítulo sumariza estratégias de avaliação de desempenho para modelos preditivos.

2.1 MicroRNAs: da biogênese à expressão diferencial

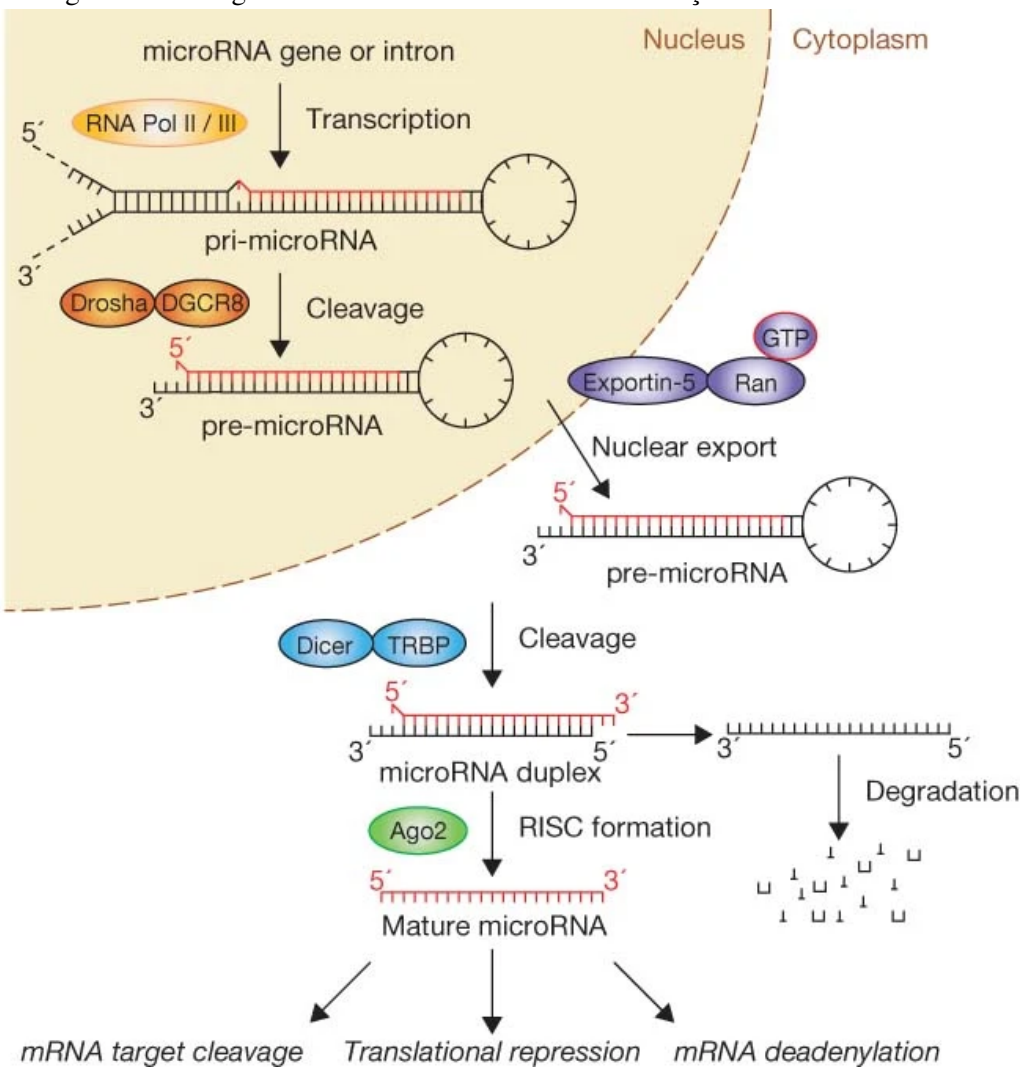
Os microRNAs (miRNAs) são moléculas fundamentais na regulação da expressão gênica em animais e plantas. Eles foram descritos pela primeira vez em *Caenorhabditis elegans* na década de 1990 e são uma classe de pequenos RNAs não-codificantes que têm entre 17 e 25 nucleotídeos de comprimento (LEE; FEINBAUM; AMBROS, 1993; LIGGETT, 2014). Essas pequenas moléculas funcionam como reguladores da expressão gênica de maneira pós-transcricional, alterando a tradução proteica por meio da interação com o mRNA alvo. A produção de miRNA é um processo complexo que envolve a participação de diversas enzimas e complexos protéicos celulares, que regulam todas as etapas até a produção de miRNA maduros capazes de desempenhar sua função. Também é importante ressaltar que a biogênese de miRNAs em animais, como humanos, ocorre de forma diferente do que em plantas. Desta forma, neste trabalho descrevemos a etapa de produção de miRNAs sobre o aspecto humano.

2.1.1 Biogênese dos microRNAs

Existem pelo menos três vias conhecidas para a produção de miRNAs, sendo a via canônica a mais estudada (O'BRIEN et al., 2018). Nesse processo, a transcrição do DNA pode codificar os miRNAs em regiões intragênicas, tanto na fita *sense* quanto na fita *antisense* do DNA. A Figura 2.1 apresenta o processo de geração do miRNA por via canônica. De acordo com Winter et al. (2009), o processamento do miRNA inicia no núcleo da célula, com a produção do transcrito primário de miRNA (pri-miRNA) pela RNA polimerase II ou III e pela clivagem do pri-miRNA pelo complexo microprocessador

Drosha-DGCR8. A molécula em forma de grampo (*i.e., hairpin*) resultante, denominada de precursor de miRNA ou pré-miRNA, é exportada do núcleo. No citoplasma, o pré-miRNA é clivado e a fita funcional do miRNA maduro é carregada junto com as proteínas Argonauta (Ago2) no complexo de silenciamento induzido por RNA (RISC), onde guia o RISC para silenciar os mRNAs alvos através de processos como clivagem do mRNA, repressão translacional ou deadenilação do mRNA.

Figura 2.1 – Biogênese canônica de microRNAs e sua ação sobre o mRNA alvo.



Fonte: Winter et al. (2009).

Para conseguir cumprir sua função, o miRNA se liga por meio do reconhecimento de mRNAs alvos, considerando a complementaridade de sequências, alterando o padrão de tradução desses mRNAs em proteínas (WITKOS; KOSCIANSKA; KRZYZOSIAK, 2011; O'BRIEN et al., 2018). Esse pareamento pode ser total ou parcial, e quanto maior a complementaridade das sequências, mais forte e duradoura parece ser essa interação. Essa ligação ao seu alvo ocorre principalmente na região 3' não traduzida (UTR) e, alternati-

vamente, a outras regiões, como a 5'UTR, ou até em regiões promotoras do mRNA-alvo, o que determina a consequência biológica dessa interação.

Um mRNA pode ser alvo de mais de um miRNA, e é importante considerar a interação entre o mRNA alvo e outros miRNAs para entender os efeitos funcionais de um determinado miRNA (WITKOS; KOSCIANSKA; KRZYZOSIAK, 2011). Um mesmo miRNA pode regular processos opostos em diferentes tipos celulares, e níveis semelhantes de um miRNA específico podem ter efeitos biológicos diferentes dependendo dos alvos e da interação com outras moléculas em um determinado tipo celular (WITKOS; KOSCIANSKA; KRZYZOSIAK, 2011). Atualmente, a versão mais recente do banco de dados miRBase (v22) possui mais de 48 mil miRNAs maduros catalogados para uma base contemplando 271 organismos. Para humanos, por exemplo, a base contém 2654 sequência de miRNAs maduros (KOZOMARA; BIRGAOANU; GRIFFITHS-JONES, 2019).

2.1.2 Relação de microRNAs com doenças humanas

É importante ressaltar que a função de muitos miRNAs ainda não é totalmente compreendida, o que indica a necessidade de mais pesquisas nessa área. No entanto, diversos estudos revelam que variações no padrão de expressão de miRNAs estão associadas a diversos processos biológicos e patológicos (ARDEKANI; NAEINI, 2010). Estima-se que estes pequenos RNAs não-codificantes são responsáveis por regular cerca de 60% dos genes humanos, sendo que muitos destes são encontrados em regiões genômicas envolvidas no desenvolvimento de câncer (SHU et al., 2017; PENG; CROCE, 2016). Uma das primeiras evidências do envolvimento de miRNAs no desenvolvimento de câncer foi publicada por Calin et al. (2002), em que os autores observaram que as sequências codificando o miR-15 e o miR-16 estão dentro de uma pequena região do cromossomo 13q14 que apresenta-se deletada em mais de 65% dos casos observados de leucemia linfocítica crônica. Desde então, alterações na expressão dos miRNAs têm sido detectadas em diversos tumores (DRAGOMIR; KNUTSEN; CALIN, 2022), incluindo câncer de pulmão (ZHONG et al., 2021) e de mama (LOH et al., 2019).

Por meio de tecnologias de sequenciamento de alto desempenho, estudos identificaram que muitos miRNAs apresentam-se desregulados em células cancerígenas, modulando a expressão de genes relevantes, como aqueles que afetam a resposta das células às drogas quimioterápicas, tal que as reduções nos níveis destes miRNAs estão associadas à quimiorresistência (SVORONOS; ENGELMAN; SLACK, 2016). Adicionalmente, já

foi demonstrado que a desregulação nos níveis de expressão de miRNAs afetam diversas das capacidades biológicas adquiridas durante o desenvolvimento do tumor (conhecidos como *hallmarks* do câncer) (HANAHAN, 2022), como sustentação da sinalização proliferativa, evasão dos supressores de crescimento, resistência à morte celular, ativação da invasão e metástase, e indução da angiogênese (PENG; CROCE, 2016). Atualmente, muitos dos estudos sobre os miRNAs relacionados a câncer já classificam os miRNAs como oncogênicos ou supressores de tumor dependendo da função do seu mRNA alvo na carcinogênese, isto é, se o miRNA inibe a expressão de mRNAs supressores de tumor ou de mRNAs oncogênicos, respectivamente (SVORONOS; ENGELMAN; SLACK, 2016). Entender as particularidades de cada miRNA é fundamental para compreender o papel que essas moléculas têm nas doenças de forma geral, e no câncer de forma mais específica (FILHO; KIMURA, 2006; DAVIS-DUSENBERY; HATA, 2010), sendo a investigação de seus alvos um primeiro passo natural em torno da elucidação de seu papel funcional no organismo.

2.1.3 Análise de expressão gênica em larga escala

Transcriptoma é o conjunto de todos os transcritos codificantes (*i.e.,* RNA mensageiro, RNA ribossômico, RNA transportador) e não codificantes (*e.g.,* miRNAs) gerados a partir do código genético de uma célula (WANG; GERSTEIN; SNYDER, 2009). O transcriptoma depende do estágio de desenvolvimento, condições ambientais, estado fisiológico e patológico, e tipo de tecido, assim, um mesmo genoma pode produzir diferentes transcriptomas em diferentes momentos ou circunstâncias (RHODES; CHINNAIYAN, 2005). Visto que o RNA mensageiro (mRNA) é o primeiro produto do processo de expressão gênica, assume-se que o transcriptoma reflete o perfil de expressão gênica do organismo (e por consequência seu estado funcional), e pode, portanto, ser analisado para avaliar e comparar aspectos moleculares subjacentes ao estado funcional dos organismos.

A análise de transcriptoma (também chamada *Gene expression profiling*) é a determinação do padrão de expressão de todos os genes de um organismo, a nível de transcrição, sob circunstâncias específicas ou em células/tecidos específicos, para se obter uma visão global do funcionamento celular (WANG; GERSTEIN; SNYDER, 2009). Esta análise também é capaz de interrogar o conjunto completo de miRNAs, avaliando sua expressão de forma global. De acordo com Gibson (2003), a comparação dos perfis de expressão gênica entre diferentes condições é uma análise de grande interesse, visto que se for obser-

vada uma diferença na abundância de transcritos entre duas ou mais condições, é natural inferir que a diferença pode apontar para um fenômeno biológico interessante.

Técnicas para quantificação de transcriptoma incluem microarranjo de DNA (GIBSON, 2003), o qual mede a atividade relativa de genes alvos previamente anotados, ou por tecnologias de sequenciamento de alto desempenho (RNA-Seq) (WANG; GERSTEIN; SNYDER, 2009) que permitem o monitoramento da expressão de todos os genes ativos, independente de já terem sua sequência conhecida ou não, ao realizarem o sequenciamento das sequências. Ambas tecnologias possuem suas vantagens e desvantagens, e a escolha de qual utilizar ao se planejar um experimento para análise de transcriptoma depende muito de fatores como questão de pesquisa e recursos disponíveis.

Como revisado por Rhodes and Chinnaiyan (2005), centenas de experimentos em larga escala são realizados para gerar perfis quantitativos globais da expressão gênica no câncer, sendo possível utilizá-los para distinguir entre tipos e subtipos de tumor. Adicionalmente, diversos estudos demonstram a utilidade dos perfis de expressão gênica na tomada de decisão clínica durante o manejo de câncer, e na determinação de assinaturas com potencial diagnóstico ou prognóstico.

2.2 Aprendizado de máquina tradicional

De acordo com Norvig e Russell (RUSSELL; NORVIG, 2009), a inteligência artificial (IA) pode ser entendida como uma representação do processo de pensamento ou raciocínio. Seu objetivo é desenvolver técnicas que permitam simular ou expandir a inteligência humana, visando a resolução de problemas. Para atingir esse objetivo e desenvolver novas técnicas, a IA se utiliza de modelos e teorias matemáticas pré-estabelecidas para a análise e interpretação de dados. Com o avanço de tecnologias de processamento e armazenamento de dados e consequente aumento na complexidade dos conjuntos de dados e problemas associados, se faz necessário o desenvolvimento de técnicas capazes de solucionar problemas de forma autônoma, assim originando subáreas como o aprendizado de máquina (AM).

O AM visa detectar padrões ou associações em um grande volume de dados, construindo boas aproximações que sejam úteis para compreender o problema ou para realizar previsões para novos dados (ALPAYDIN, 2014). Desta forma, o AM torna-se extremamente útil para tarefas para as quais não temos um algoritmo pronto para resolvê-la, mas temos muitos dados disponíveis para construção do modelo através de algoritmos de

aprendizado. Os algoritmos de aprendizado são categorizados em dois tipos principais: aprendizado supervisionado (ou modelagem preditiva) e aprendizado não-supervisionado (ou modelagem descritiva). Nas próximas seções, detalharemos os tipos de aprendizado de máquina mais recorrentes e apresentaremos o algoritmos de AM tradicional utilizado neste trabalho – Redes Neurais Artificiais.

2.2.1 Aprendizado supervisionado e não-supervisionado

Em aprendizado de máquina supervisionado, ou preditivo, é realizada uma busca por uma função de aproximação capaz de mapear corretamente entradas e saídas em dados de treinamento a partir de um algoritmo de aprendizado (ALPAYDIN, 2014). Nesse tipo de aprendizado estão inclusas as tarefas descritas como classificação e regressão. Ambas as tarefas podem ser descritas formalmente como o processo de aprender uma função $y = g(x|\Theta)$, onde x representa uma entrada composta por uma ou mais variáveis (*i.e.,* atributos), y representa a saída esperada, $g(.)$ representa a função (*i.e.,* modelo) aprendida e Θ representa o conjunto de parâmetros internos desta função. No caso da classificação, dada uma entrada x, a saída y assume um valor categórico e o algoritmo tem a tarefa de determinar a qual classe k ela pertence. Para o caso da regressão, a partir da entrada x, o modelo tem como objetivo prever um valor contínuo que será dado como a saída y. A função aprendida a partir dos dados de treinamento pode então ser utilizada para prever o valor de saída desconhecido para novas instâncias, nunca antes vistas, usualmente denominadas dados de teste.

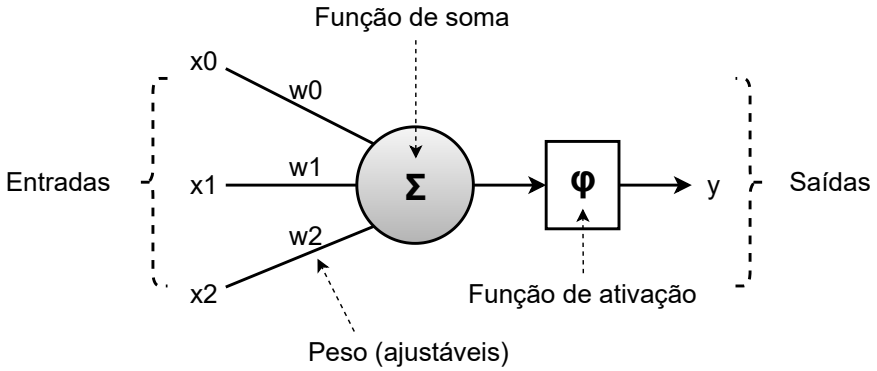
No aprendizado não-supervisionado, o objetivo é descrever os padrões existentes em um conjunto de dados de entrada, em que as saídas não são conhecidas previamente. Uma vez que não se tem rótulos pré-definidos, tenta-se inferir regularidades, associações ou similaridades nestes dados. Uma das abordagens mais conhecidas dentre algoritmos de aprendizado não-supervisionado é a análise de agrupamentos (RUSSELL; NORVIG, 2009). Nas tarefas de agrupamento, utiliza-se o algoritmo de aprendizado para dividir um grande conjunto de dados em vários grupos (*i.e., clusters*) de forma que as instâncias atribuídas aos mesmos grupos sejam mais semelhantes entre si do que em relação a instâncias designadas a outros grupos. Com esta abordagem, é possível descobrir agrupamentos intrínsecos em dados não rotulados que tenham utilidade para entendimento do problema e extração de conhecimento.

2.2.2 Redes Neurais Artificiais

Na ideia de ensinar computadores a processar dados de forma semelhante ao cérebro humano, surgiram as chamadas redes neurais artificiais ou *artificial neural networks* (ANNs). Conforme descrito por Faceli et al. (2011), as ANNs são sistemas distribuídos que aplicam uma função matemática, baseada no funcionamento do sistema nervoso central. As ANNs são compostas por unidades de processamento densamente conectadas que aplicam funções de ativação sobre combinações lineares dos valores de entrada. Essas unidades são conhecidas como neurônios artificiais e suas conexões, como sinapses. Em uma ANN, os neurônios estão distribuídos entre diferentes camadas, incluindo camadas ocultas responsáveis pelo processamento dos dados.

Na Figura 2.2, observamos a representação de um neurônio, a unidade básica de processamento de uma ANN. Este neurônio recebe três entradas numéricas, x_0, x_1 e x_2, cada qual atribuída a um respectivo peso, w_0, w_1 e w_2. Estas entradas passam por uma função de combinação linear, denominada função de soma, e o resultado desta combinação é então passado para uma função de ativação, usualmente de natureza não-linear. Como exemplo de função de ativação comumente utilizada em ANNs, podemos citar a função sigmoide. A saída da função de ativação representa a saída do processamento do neurônio para as entradas informadas.

Figura 2.2 – Representação de um neurônio artificial utilizado como unidade de processamento em redes neurais artificiais.



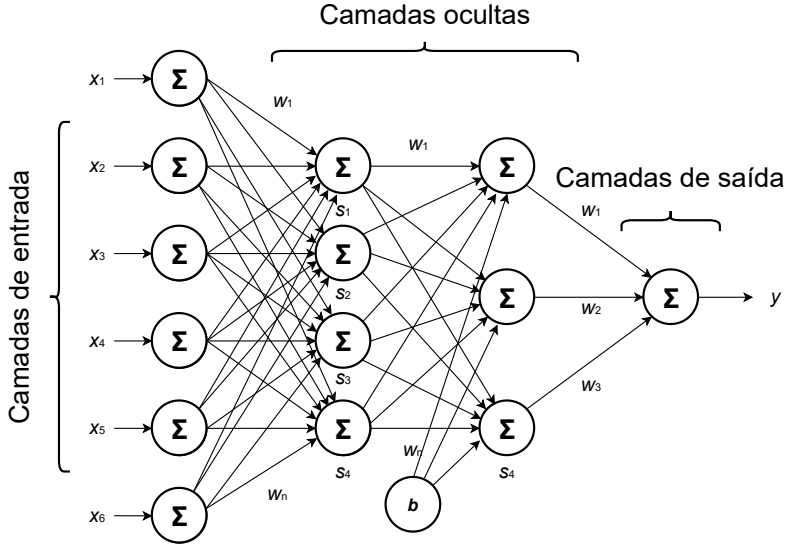
Fonte: O Autor.

Quando as ANNs foram originalmente propostas, modelos simples chamados de Perceptron eram utilizados. Eles contavam apenas com uma camada oculta composta por um número pré-definido de neurônios. Com o aumento da capacidade computacional, essas redes puderam ser aprimoradas e, consequentemente, o número de possíveis aplicações aumentou (TARCA et al., 2007).

As arquiteturas de redes de aprendizagem profunda mais comuns são as de propagação para frente (*feedforward*), também conhecidas como *multilayer perceptrons* (MLPs). As redes *feedforward* utilizam conexões em uma única direção, de forma que um determinado nó recebe como entrada a saída do nó anterior e propaga sua saída para a camada seguinte. Elas são implementadas com uma camada de entrada, uma sequência de camadas ocultas, e uma camada de saída, na qual nenhum neurônio retorna sua saída para ele mesmo (GOODFELLOW; BENGIO; COURVILLE, 2016). Além dessas, existem os modelos descritos como recorrentes, nos quais os elementos de entrada são processados em sequência, e as informações de cada elemento são armazenadas para serem usadas no processamento do elemento seguinte. Isso significa que essas redes processam não só a informação de entrada, mas também são influenciadas pela iteração anterior, funcionando de forma semelhante a uma memória.

A Figura 2.3 ilustra a arquitetura de uma MLP. A camada mais à esquerda é denominada camada de entrada, e cada neurônio nesta camada representa os dados de entrada a serem processados pelo modelo. A camada mais à direita é a camada de saída, contendo os neurônios que irão fornecer a predição final sobre os dados de entrada. A camada de saída pode conter um ou mais neurônios, de acordo com o tipo de tarefa de predição. Já as camadas do meio são denominadas camadas ocultas, e são as responsáveis por processar os dados e aprender os padrões implícitos. Percebe-se que as camadas são densamente conectadas, isto é, cada neurônio de uma camada oculta está conectado a todos os neurônios da camada anterior e da camada seguinte.

Figura 2.3 – Arquitetura de uma rede MLP *feedforward*. Para simplificar a ilustração da rede a função de ativação presente na saída do neurônio foi omitida.



Fonte: O Autor.

O funcionamento geral de uma rede MLP também pode ser compreendido a partir da Figura 2.3. Cada neurônio da camada oculta recebe um conjunto de valores de entrada, representadas pelo símbolo x, cada qual correspondendo a um neurônio na camada de entrada ou a um nerônio na camada oculta anterior. As entradas são multiplicadas pelos respectivos pesos sinápticos, denotados por w, e somadas entre si junto com uma constante chamada de polarização ou bias, representada pelo símbolo b. Essa constante possui o papel de centralizar a curva da função de ativação em um valor conveniente. Caso seja positivo, o movimento do gráfico é realizado para a esquerda, diminuindo o valor do eixo x. Porém, caso seja negativo, o movimento do gráfico é feito para a direita, aumentando o valor do eixo x.

A soma ponderada, que realiza a combinação linear das entradas, gera o potencial de ativação que é utilizado para determinar o valor de saída do neurônio na camada oculta após passar por uma função de ativação. O objetivo desta função de ativação é permitir a modelagem de padrões mais complexos através da introdução de não-linearidade nos dados, bem como limitar a amplitude de saída do neurônio. Ou seja, o valor obtido no somatório é normalizado dentro de um intervalo fechado, como [0,1], podendo em alguns casos ser interpretado também como a probabilidade para um determinado evento.

O processo de aprendizado de ANNs consiste em ajustar iterativamente os seus parâmetros internos, representados pelos pesos das conexões entre os neurônios, através do algoritmo *backpropagation*, visando minimizar uma função que compara a saída predita com a saída real, estimando a diferença entre ambas. Existem diversas formas de mensurar essa diferença (também chamada de erro do modelo), dependendo da natureza do problema. A função que realiza essa quantificação é chamada de função de custo, ou função de perda, pois ela representa o custo, em termos de erro, de usar os parâmetros determinados em uma dada iteração, ou continuar treinando. Existem várias funções de custo e cada uma é adequada a um tipo de base de dados e finalidade da rede neural (AGGARWAL, 2018). No contexto deste trabalho, destacamos a função de custo *binary cross-entropy*.

A *binary cross-entropy* é uma função muito utilizada para problemas de classificação binários. Por exemplo, uma rede neural responsável por identificar se uma imagem contém ou não um cachorro, ou seja, que contém um valor de saída de zero (negativo) ou um (positivo). Considerando nosso contexto, esta função se encaixa para a identificação da existência ou não de uma interação miRNA–mRNA. A função é definida conforme a Equação 2.1:

$$BNE = -\frac{1}{n}\sum_{i=1}^{n}[y_i * \log(y_i') + (1 - y_i) * \log(1 - y_i') \tag{2.1}$$

onde n é o número de valores de saída, y é o valor de saída real e y' é o valor de saída predito. Podemos notar, então, que o valor de saída da função será sempre entre 0 e 1, e que o objetivo da função é retornar valores altos para predições ruins e valores baixos para predições boas. Assim, 0 significa o menor valor possível de diferença entre o valor de saída predito e o valor de saída real e 1 significa o contrário, com o maior erro possível.

2.3 Aprendizado profundo em grafos

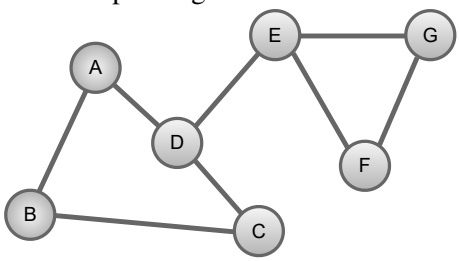
Atualmente, as ANNs podem ser definidas com um grande número de camadas ocultas, permitindo o processamento de funções mais complexas, o que é conhecido como rede neural profunda (*deep learning*) (GOODFELLOW; BENGIO; COURVILLE, 2016). Entretanto, *deep learning* é adequado para capturar padrões ocultos em dados euclidianos (imagens, texto e vídeos), mas conforme descrito por Bronstein et al. (2021). em seu estudo sobre o aprendizado profundo geométrico, "v*ários campos científicos estudam dados com uma estrutura subjacente que é um espaço não euclidiano*". Um bom exemplo dessas aplicações em espaço não euclidiano é a representação de interações complexas, como as redes de interações miRNA–alvo. Nesse sentido, o modelo proposto neste trabalho é baseado em um algoritmo pertencente a uma classe de redes neurais artificiais para processamento de dados que podem ser representados como grafos, as chamadas redes neurais de grafos (*Graph Neural Networks* ou GNNs).

Os grafos são estruturas de dados amplamente utilizadas para modelar problemas onde os dados são gerados a partir de domínios não euclidianos e que geralmente possuem relacionamentos complexos e/ou interdependências entre objetos. O aprendizado profundo geométrico é uma área em desenvolvimento que se concentra em criar redes neurais que exploram explicitamente essa representação não euclidiana. O desempenho de redes neurais de grafos tem chamado bastante atenção atualmente em diversas áreas da biologia (CAI; ZHENG; CHANG, 2018; WU et al., 2021). Com sua utilização, é possível abordar três tarefas principais: 1) classificação de nós em grafos; 2) predição de *links* (*i.e.,* interações) em grafos; e 3) classificação de grafos.

2.3.1 Breve introdução à teoria dos grafos

A parte fundamental do algoritmo empregado neste trabalho baseia-se na teoria dos grafos. Na ciência da computação, um grafo é uma estrutura de dados utilizada para estudar as relações entre uma coleção de entidades. As entidades são representadas por nós no grafo, e as relações são descritas por arestas entre estes nós. Assim, um grafo G é formalmente definido como $G = (V, E)$, onde V é o conjunto de nós e E é o conjunto de arestas (MASON; VERWOERD, 2007). A Figura 2.4 apresenta um grafo que ilustra a definição descrita. Uma aplicação simples de visualizar é a utilização de grafos na representação de uma rede social, onde o conjunto de nós representa os usuários da rede e as arestas representam suas relações de amizade.

Figura 2.4 – Exemplo de grafo com 7 vértices e 8 arestas.



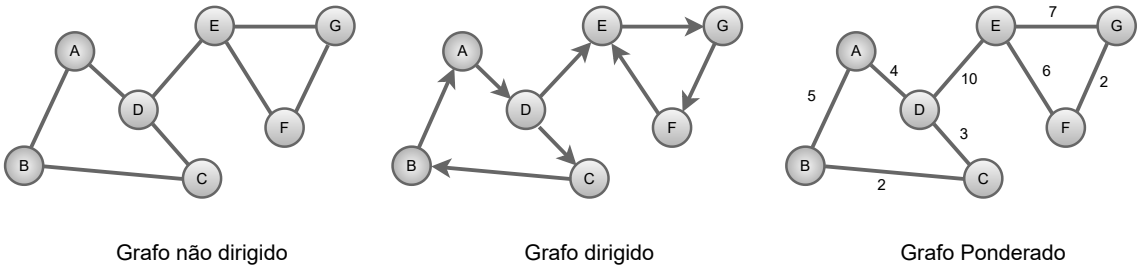
Fonte: O Autor.

A teoria dos grafos é repleta de nomenclaturas e termos técnicos. No entanto, neste capítulo, estamos interessados em apresentar somente algumas definições importantes para o total entendimento deste trabalho proposto. Sendo assim, outro conhecimento necessário neste momento é compreender que um grafo pode ser dirigido ou não dirigido. Em um grafo dirigido, cada aresta conectada tem uma direção, muitas vezes representada por meio de uma seta. Um grafo não dirigido não possui direção. Além da orientação, é possível atribuir um peso para as arestas de acordo com algum conhecimento do domínio modelo. Os grafos que recebem esses valores são chamados de grafos ponderados. A Figura 2.5 mostra exemplos de grafos considerando todas as definições descritas.

Outra característica importante dos nós que um grafo pode possuir é sobre sua representatividade. Em um mesmo grafo, é possível representar nós com características distintas. Grafos com esse aspecto são chamados de grafos heterogêneos. Um ótimo exemplo do uso desta estrutura heterogênea está em redes de proteínas e compostos químicos (BORGWARDT et al., 2005). Partindo desta fundamentação teórica acerca da teoria dos grafos, podemos estabelecer o modelo proposto neste trabalho como um grafo heterogêneo. Os nós caracterizam o conjunto de miRNAs e mRNAs, e as arestas represen-

tam as relações miRNA–mRNA e mRNA–mRNA determinadas a partir de conhecimento prévio.

Figura 2.5 – Exemplos de grafos contendo diferentes características em sua estrutura.



Fonte: O Autor.

2.3.2 Redes neurais de grafos e o algoritmo GraphSAGE

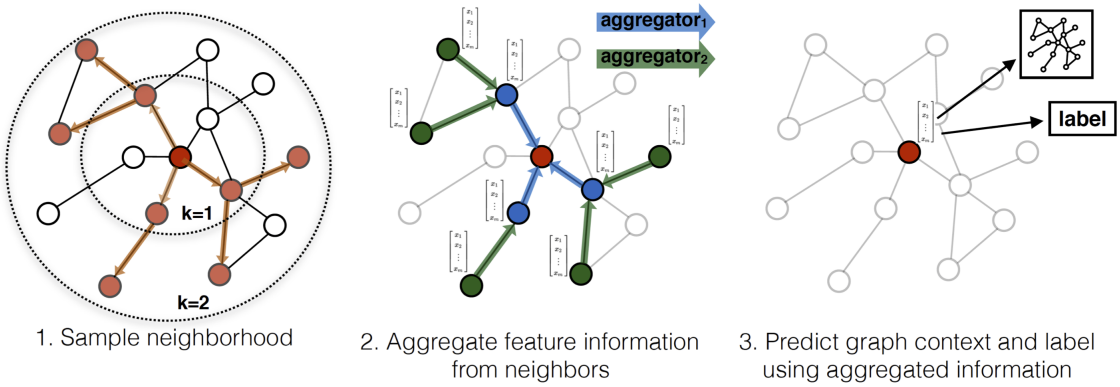
O estado da arte dos algoritmos de redes neurais em grafos é composto por quatro categorias principais: GNNs recorrentes, GNNs convolucionais, *autoencoders* de grafos e GNNs espaço-temporais (ZHOU et al., 2020). Com sua utilização conseguimos realizar três tarefas principais: 1) classificação de nós; 2) predição de links e 3) classificação da rede. Embora haja diversos tipos de algoritmos disponíveis para o aprendizado baseado em redes neurais de grafo, neste trabalho, estamos interessados em identificar um algoritmo capaz de lidar com uma grande quantidade de dados, prever *links* para permitir a predição de interações entre nós e trabalhar com um grafo heterogêneo. De acordo com a revisão de literatura realizada por Zhou et al. (2020), optamos por utilizar o algoritmo indutivo GraphSAGE (HAMILTON; YING; LESKOVEC, 2017).

A ideia geral do GraphSAGE é permitir o aprendizado em um grafo com dados dinâmicos, utilizando informações agregadas dos nós para gerar *embeddings*. Na área de aprendizado de máquina, os *embeddings* são utilizados para transformar informações complexas em estruturas que possam ser aprendidas e diferenciadas, como em sistemas de processamento de linguagem natural para criar representações numéricas de palavras e suas relações.

Uma das características mais relevantes do algoritmo GraphSAGE é a forma como os *embeddings* da rede são gerados. Durante o treinamento do modelo, o algoritmo emprega uma função que gera *embeddings* por amostragem e agregação de atributos da vizinhança local de um nó, em vez de treinar *embeddings* individuais centrados apenas no próprio nó. Esse comportamento pode ser observado na Figura 2.6, onde o valor K re-

presenta a profundidade de busca (ou número de saltos/*hops* a partir de um determinado nó), para amostragem de nós vizinhos e coleta de informações. Embora o parâmetro k possa assumir qualquer valor, em muitos trabalhos relacionados é descrito que valores superiores a 2 podem prejudicar a geração dos *embeddings* aprendidos (ALAMSYAH; RAHARDJO; KUSPRIYANTO, 2013; LO et al., 2021).

Figura 2.6 – Ilustração do funcionamento do GraphSAGE.



Fonte: Hamilton, Ying and Leskovec (2017).

O objetivo do GraphSAGE é aprender uma representação para cada nó com base em alguma combinação de seus nós vizinhos. A quantidade de vizinhos observados pode ser parametrizada, tanto em termos do número de saltos (descrito na Figura 2.6 como k), como em termos do número de nós vizinhos a serem amostrados em cada salto. Adicionalmente, cada nó pode conter uma quantidade de atributos relacionados (*i.e., features*). Como cada nó pode ser definido por seus vizinhos, o *embedding* gerado para um nó pode ser representada por alguma combinação dos vetores de *embedding* extraídos de seus nós vizinhos. Esse processo ocorre por meio da execução do algoritmo GraphSAGE. É importante destacar que o algoritmo GraphSAGE não é empregado somente na predição de arestas, mas também pode ser utilizado para a classificação de nós. Entretanto, esta aplicação está fora do escopo do presente trabalho.

A característica de utilizar a vizinhança para prever o *embedding* de um novo nó no grafo é chamada de aprendizado por indução. Antes do GraphSAGE, a maioria dos modelos de *embedding* de nós era baseada em métodos de decomposição espectral/fatoração de matrizes. O grande problema desses métodos era a incapacidade em trabalhar com dados nunca vistos. Esses métodos necessitam de toda a estrutura do grafo para gerar os *embeddings*, e ao adicionar um novo nó ao grafo, seria necessário realizar o treinamento novamente (ZHOU et al., 2020). Quando trabalhamos com uma grande quantidade de dados, esses métodos acabam sendo muito complexos e demorados.

No entanto, o algoritmo GraphSAGE original não atende totalmente às nossas necessidades devido à sua incompatibilidade com grafos heterogêneos. Por isso, em nossa proposta, empregamos uma variação do algoritmo GraphSAGE, descrita como HinSAGE, e fornecida pela biblioteca de aprendizado de máquina em grafos StellarGraph[1] (DATA61, 2018). A biblioteca realizou uma alteração na matriz de peso utilizada durante o processo de aprendizado, tornando possível o comportamento heterogêneo. As demonstrações matemáticas sobre essas alterações podem ser encontradas na página da biblioteca[2].

Por fim, é importante salientar que redes neurais de grafos podem ser criadas como qualquer outra rede neural, usando camadas totalmente conectadas, camadas convolucionais, camadas de agrupamento, etc. O tipo e o número de camadas dependem do tipo e da complexidade dos dados do grafo e da saída desejada. Devido a todo esse diferencial descrito na utilização, acreditamos que as estruturas das GNNs são capazes de descrever, com certa facilidade, as interações miRNA–alvo.

2.4 Avaliação de modelos preditivos

Tão importante quanto saber escolher um algoritmo de aprendizado de máquina adequado, é avaliar corretamente o desempenho preditivo do modelo gerado, selecionando estratégias de divisão de dados e métricas de desempenho adequadas para o contexto abordado. No âmbito de divisão de dados, as estratégias visam evitar que um modelo seja treinado e avaliado com o mesmo conjunto de dados, tendo em vista que isto introduziria um viés muito otimista na avaliação e impossibilitaria a análise do poder de generalização do modelo para dados novos, ainda não vistos. Por outro lado, diferentes métricas de desempenho refletem tipos de erros e acertos distintos, e podem ser sensíveis ou não a características nos dados, como desbalanceamento de classes. Adicionalmente, a seleção de métricas de desempenho depende do tipo de problema de predição abordado (isto é, supervisionado ou não-supervisionado, e ainda de classificação ou regressão para aprendizado supervisionado). Nesta seção, serão sumarizadas estratégias de avaliação de modelos de AM, focando em tarefas de aprendizado supervisionado.

[1] <https://stellargraph.readthedocs.io/en/stable/>
[2] <https://stellargraph.readthedocs.io/en/stable/hinsage.html>

2.4.1 Divisão de dados com o método Holdout

Um dos métodos mais simples para divisão de dados visando avaliação de modelos preditivos é o *Holdout*. Este método visa fazer uma simples divisão aleatória do conjunto de dados rotulados em conjunto de treinamento e conjunto de teste, usualmente na proporção de 2/3 e 1/3, respectivamente (RASCHKA, 2018). Um modelo é ajustado aos dados de treinamento através de um algoritmo de aprendizado, e posteriormente é aplicado para predizer os rótulos do conjunto de teste. A comparação entre os rótulos preditos e os rótulos reais para o conjunto de teste é a base da avaliação de desempenho do modelo. Entretanto, como muitos algoritmos envolvem uma tarefa de otimização de hiperparâmetros, a qual exige a disponibilidade de um conjunto de dados independentes para avaliação do modelo sob diferentes configurações de hiperparâmetros, o método de *Holdout* foi adaptado para fazer uma divisão dos dados rotulados em três subconjuntos: treinamento, validação, e teste. Denominado *Holdout* de 3 vias (ou *3-way Holdout*), este método permite gerar um conjunto de validação utilizado para avaliar e selecionar a melhor configuração de hiperparâmetros do modelo com dados independentes do conjunto de teste. A melhor configuração de hiperparâmetros é, então, utilizada para treinar um modelo, cujo desempenho final será avaliado com os dados de teste.

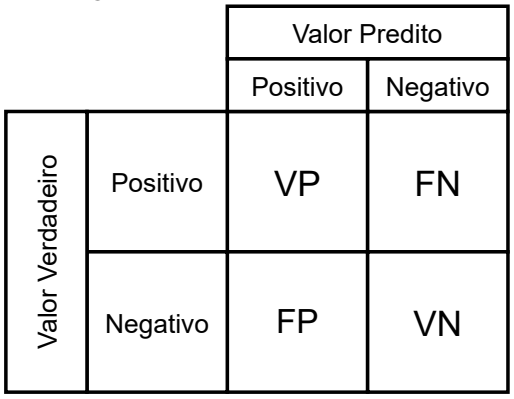
A fim de obter estimativas de desempenho mais robustas e menos sensíveis à forma com que foram divididos os dados entre os conjuntos de treinamento, validação e teste, a avaliação de modelos de aprendizado profundo é usualmente feita a partir da repetição do método *Holdout* r vezes, com diferentes sementes aleatórias (RASCHKA, 2018). Ao final, calcula-se o desempenho médio e a variação de desempenho sobre as r repetições a fim de sumarizar o poder preditivo do modelo para múltiplas execuções com variadas configurações de conjuntos de treinamento, validação e teste.

2.4.2 Métricas de desempenho

Existem inúmeras métricas de desempenho possíveis para avaliação de modelos preditivos, e as mesmas variam de acordo com tarefas de classificação e regressão. Tendo como foco as tarefas de classificação, uma forma elementar de avaliação e que serve como base para a maioria das métricas a serem definidas posteriormente, é a matriz de confusão. Em problemas de classificação binária nos quais assume-se uma classe como positiva (*i.e.,* usualmente a classe de interesse) e outra como negativa, a matriz de confusão consiste em

uma tabela 2x2 que relaciona os rótulos reais com os rótulos preditos para as instâncias de teste. Desta forma, a matriz permite quantificar o número de verdadeiros positivos (VP) e verdadeiros negativos (VN) preditos pelo modelo, que coletivamente formam os acertos da predição, e o número de falsos positivos (FP) e falsos negativos (FN), que representam os erros de predição. A matriz de confusão está ilustrada na Figura 2.7.

Figura 2.7 – Matriz de confusão.



Fonte: O Autor.

A partir dos valores de VP, VN, FP e FN quantificados pela matriz, uma grande variedade de métricas de avaliação de desempenho podem ser definidas. A seguir são enumeradas métricas comumente adotadas na literatura relacionada e de interesse do presente trabalho: acurácia, sensibilidade, especificidade, precisão e F1-score.

- Acurácia: A acurácia reflete quantos dos exemplos de teste foram de fato classificados corretamente, independente da classe à qual pertencem. A acurácia pode ser calculada pela Equação 2.2:

$$\text{Acurácia} = \frac{VP + VN}{VP + FN + VN + FP} \tag{2.2}$$

- Sensibilidade: A sensibilidade, também denominada revocação ou *recall* em inglês, é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos pelo modelo e a quantidade de exemplos que são de fato positivos nos dados de teste. Esta métrica também é por vezes referida como Taxa de Verdadeiros Positivos (TVP) e dá uma maior ênfase para erros por falso positivo. A sensibilidade é calculada conforme a Equação 2.3:

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{2.3}$$

- Especificidade: Ao contrário da sensibilidade, a especificidade avalia a capaci-

dade do modelo de classificar corretamente os casos negativos. A especificidade é usualmente usada para definir a Taxa de Falsos Positivos (TFP) como $TVP = 1 - Especificidade$. A especificidade pode ser calculada conforme a Equação 2.4:

$$\text{Especificidade} = \frac{VN}{VN + FP} \tag{2.4}$$

- Precisão: Esta métrica é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e o total de exemplos classificados como positivos, conforme a Equação 2.5:

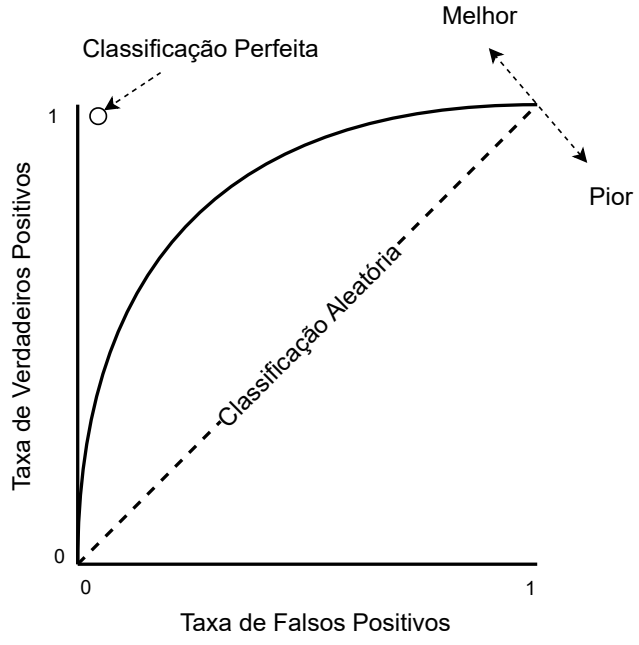
$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.5}$$

- F1-*Score*: Também conhecida como F-*measure*, esta métrica leva em consideração tanto a precisão quanto a sensibilidade (*i.e., recall*), calculando a média harmônica entre ambas as métricas, como mostrado na Equação 2.6:

$$\text{F1-}Score = 2 * \frac{\text{Precisão} * \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \tag{2.6}$$

Outra métrica bastante usual na avaliação de modelos preditivos é a área sob a Curva Receiver Operating Characteristic (ROC), denotada por ROC AUC. Uma ilustração desta curva é apresentada na Figura 2.8.

Figura 2.8 – Exemplo de curva ROC para avaliação de modelos preditivos.



Fonte: O Autor.

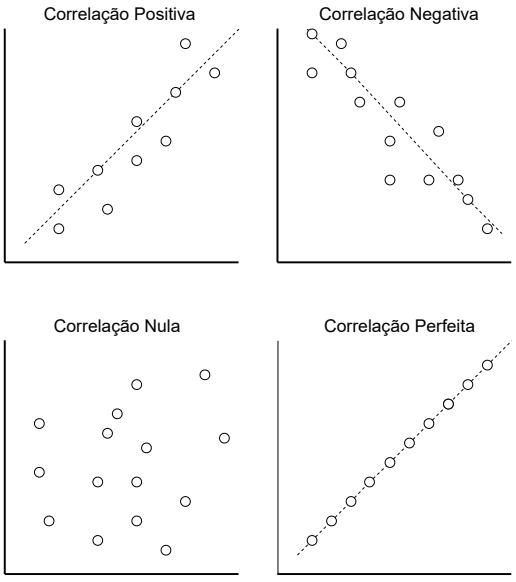
O objetivo da curva ROC é avaliar os valores de TVP (eixo y) e TFP (eixo x) para diferentes limiares de probabilidade usados para a classificação, sumarizando graficamente o desempenho do modelo preditivo (TAN, 2009). A métrica ROC AUC, dada pela área abaixo da curva ROC, pode ser interpretada como a probabilidade de o modelo classificar uma instância positiva escolhida aleatoriamente mais acima do que uma instância negativa escolhida aleatoriamente. Assim, quanto maior o valor de ROC AUC, melhor o modelo.

Enquanto as métricas definidas até o momento se destinam à avaliação de modelos de classificação, existem diversas métricas que se aplicam a modelos de regressão. Neste trabalho, destacaremos o erro quadrático médio (do inglês, *Mean Squared Error* ou MSE), o qual calcula a média da diferença quadrática entre a predição do modelo (p_i) e o valor esperado (y_i), conforme a Equação 2.7:

$$MSE(y, p) = \frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2 \tag{2.7}$$

Adicionalmente, o grau de relacionamento entre o valor predito e o valor real pode ser analisado através da avaliação de correlações, como o coeficiente de correlação linear de Pearson. Este coeficiente mede o grau e o sinal da correlação linear entre duas variáveis, retornando um resultado entre -1 (correlação negativa perfeita) e 1 (correlação positiva perfeita), conforme mostrado na Figura 2.9.

Figura 2.9 – Ilustração dos diferentes tipos de correlações que podem existir entre duas variáveis contínuas.



Fonte: O Autor.

3 TRABALHOS RELACIONADOS

Como mencionado anteriormente, atualmente existem vários métodos computacionais desenvolvidos para a área da biologia, muitos dos quais focados na investigação de processos biológicos envolvidos em diferentes etapas da biogênese de miRNAs e na sua atuação como elementos reguladores da expressão gênica (CHEN et al., 2019). Neste trabalho, temos como foco a etapa de predição de alvos de miRNAs e como interesse principal, métodos baseados em algoritmos de AM. A fim de revisar os trabalhos relacionados, definimos um protocolo de busca com questão principal de pesquisa e critérios de inclusão e exclusão dos trabalhos, conforme apresentado no Apêndice A[1], Tabela A.1. Foram feitas buscas nas bases PubMed e PMC do NCBI, e no Google Scholar, abrangendo publicações de 2000 à 2020 (inclusive). A Tabela A.2 lista os trabalhos relacionados que atendem aos critérios definidos e algumas características básicas extraídas a partir deles, como ano de publicação, algoritmo de AM empregado na tarefa de predição de alvos de miRNAs, origem dos dados de treinamento, dentre outros. A seguir, são descritos os trabalhos mais relevantes para a abordagem proposta nesta pesquisa.

O TargetScan (LEWIS et al., 2004) foi um dos modelos precursores na utilização de aprendizado de máquina para auxiliar na predição de alvos de miRNAs. Inicialmente desenvolvido sem essa característica de treinamento de modelo com algoritmos de aprendizado, o modelo foi aprimorado posteriormente com a adição de regressão linear que procura por locais conservados que correspondem à região *seed* do gene alvo (AGARWAL et al., 2015). Como ferramenta web, é amplamente utilizado e permite a pesquisa por miRNA, gene, locais de conservação e famílias de miRNAs em diferentes espécies. Seus dados são originários de trabalhos científicos validados experimentalmente e do sequenciamento de alto rendimento CLIP-seq. O TargetScanS (LEWIS; BURGE; BARTEL, 2005) é uma variação do TargetScan e também é amplamente utilizado, nesta variação o modelo desenvolvido se utilizada de regressão linear como método de aprendizado de máquina com o objetivo de melhorar as predições realizadas. Entretanto, ambos os métodos possuem algumas limitações, como a utilização implícita de atributos extraídos da região *seed*, o que torna o método sensível a alvos sem boa conservação evolutiva (MIN; YOON, 2010).

O PITA (KERTESZ et al., 2007) oferece uma abordagem diferente para a previsão do alvo do miRNA. A principal característica avaliada por este programa é a acessibili-

[1] As bases de artigos consultadas são: PubMed, PMC de NCBI e Google Scholar. Consulta concluída em 1 de Março de 2023.

dade do sítio de ligação no alvo. Ele avalia a energia livre obtida com a formação do par miRNA–mRNA e o custo de energia para tornar o alvo acessível ao miRNA, e então calcula a diferença entre esses dois parâmetros. Considera também os "*flank sites*", os sítios ao redor da semente, que estão envolvidos na acessibilidade do sítio. O usuário pode impor restrições para reduzir o número de alvos resultantes (ou seja, tamanho mínimo da semente e bases não pareadas). O projeto suporta a predição para as espécies humano, rato, vermes e moscas.

TargetMiner (BANDYOPADHYAY; MITRA, 2009) é um classificador baseado no algoritmo de aprendizado *Support Vector Machine* (SVM), que tem como objetivo identificar potenciais locais de *seed* em genes alvos candidatos a partir de um miRNA fornecido pelo usuário. A saída fornecida descreve a correspondência de *seed*, posição e quantos desses locais são encontrados na sequência. A ferramenta é baseada no aprendizado de máquina, utilizando dados de treinamento positivos e negativos para fornecer previsões de correspondência de *seed* mais precisas entre um miRNA e seu alvo. Os dados de treinamento positivos foram um conjunto de 289 pares de transcritos de miRNA extraídos do banco de dados miRecords (XIAO et al., 2009). Os dados de treinamento negativos foram selecionados a partir de múltiplos algoritmos de predição de alvo, utilizando um conjunto de dados agrupados de pares de miRNAs e alvos preditos por identificação de pares falsos positivos. Pares não-alvo específicos de tecido foram então identificados usando dados de perfil de expressão. Com isso, TargetMiner alcançou a precisão de predição mais balanceada em termos de sensibilidade e especificidade em comparação com os outros métodos, e seu desempenho robusto se deve principalmente ao uso desses exemplos negativos que são considerados de alta qualidade. No entanto, a seleção de um subconjunto de atributos informativos relevantes leva a um modelo mais simples e a utilização de uma base de dados específica como a que foi utilizada pode acabar gerando resultados não satisfatórios quando executados com um conjunto de dados independente. Além disso, é sabido que o modelo SVM costuma apresentar problemas de desempenho quando está trabalhando sobre uma base de dados grande.

RFMirTarget (MENDOZA et al., 2013) é um método baseado no algoritmo de Floresta Aleatória que realiza a predição usando um classificador treinado com atributos extraídos de interações miRNAs–alvos validadas experimentalmente. São definidos 17 atributos provindos da análise das possíveis interações entre miRNA e alvo com o algoritmo miRanda (BETEL et al., 2010). Apesar de possuir resultados encorajadores, existem algumas limitações no método, as quais estão associadas à definição fixa e manual

dos atributos utilizados no modelo. Além do RFMirTarget, o método MiSTAR (PEER et al., 2016) igualmente se utiliza de uma floresta aleatória. No entanto, vai além, adicionando também uma segunda camada empregando regressão logística. Os resultados apresentados superam métodos como TargetScan. Entretanto, o trabalho carece de um treinamento com uma base de dados maior.

DeepTarget (LEE et al., 2016) é uma combinação de métodos de aprendizado supervisionado e não supervisionado. Este método se baseia na aplicação de *autoencoder* para gerar predição de alvo, utilizando um atributo de interação baseado em sequência para treinar o modelo de rede neural recorrente. O DeepTarget tem um alto nível de precisão e elimina a necessidade de atributos selecionados manualmente para predição. Superando modelos bastante estabelecidos como TargetScan em até 26% de melhora na pontuação F1-score. No entanto, nenhum conjunto de dados de teste independente foi usado para a avaliação do modelo.

O projeto miDIP (TOKáR et al., 2017) é uma base de dados construída com o objetivo de catalogar previsões de interações miRNA–alvo, com e sem validação experimental. Construído a partir de uma integração de diferentes projetos, em sua versão 4.1 estão presentes mais de 150 milhões de previsões de alvos miRNA em humanos coletados de 30 atributos diferentes. O processo de aquisição dos dados considerou os trabalhos publicados entre os anos de 2006 e 2017. A aquisição dos dados considerou apenas os atributos cujas previsões são avaliadas por qualquer tipo de medida quantitativa, representando a confiança subjetiva do atribut associado a uma determinada previsão (por exemplo, energia de ligação, significância estatística, etc.). As previsões dos atributos individuais foram recuperadas do site, materiais suplementares da publicação correspondente ou geradas de novo executando o algoritmo de previsão localmente.

Ao final do processo de coleta os dados foram divididos em um conjunto para realização de benchmarking e validação. Posteriormente todas as interações são avaliadas atribuindo uma pontuação de confiança descrevendo como: *very high*, *high*, *medium* e *low*. Esta pontuação foi estatisticamente inferida a partir das previsões obtidas. Os autores também descrevem que a realização desta integração não acumulou viés de previsão.

O miRAW (PLA; ZHONG; RAYNER, 2018) foi uma das primeiras abordagens utilizando um algoritmo de aprendizado de máquina baseado em *Deep Learning*. Sua rede é composta por *autoencoders* e sua arquitetura utiliza *feed-forward*. Os dados para desenvolver o modelo proposto foram obtidos de duas fontes: Diana TarBase v7.0 (VLACHOS et al., 2014) e miRTarBase v6.0 (CHOU et al., 2015), sendo que as anotações

relacionadas a transcrições e locais de ligação de miRNA são originários da realização de referência cruzada de identificadores das bases Diana TarBase com miRBase (KOZOMARA; GRIFFITHS-JONES, 2013). Seu conjunto de dados está composto por mais de 300 mil interações de mRNA descritas como positivas e mais de mil interações descritas como negativas.

Os dados de treinamento e teste do modelo são de diferentes bases de dados, não está claro o uso de um conjunto para validação. O trabalho apresenta no artigo o treinamento e teste sobre um conjunto de dados de miRNA humanos, mas no entanto, é descrito que a metodologia desenvolvida também está preparada para trabalhar com dados de predição de alvos para outras espécies. Os autores descrevem que o desempenho do miRAW quando comparado com outras ferramentas de predição é superior quando se diz respeito a aspectos mais conservadores (que utilizam a região semente como um dos pontos mais importantes para a realização da interação miRNA–alvo). Também é descrito que não utilizar recursos manuais, ou seja, definições manuais de locais alvos no modelo proposto acabou por aumentar o número de falsos positivos. Por fim, é reforçado que a utilização de uma rede neural pode ocasionar uma dificuldade na interpretação e mapeamento das características de classificação.

O modelo DeepMirTar (WEN et al., 2018) trabalha sobre um conjunto relativamente grande de diferentes tipos de atributos para a sua rede neural, incluindo correspondência de *seed*, energia livre, composição de sequência, identidade de nucleotídeos brutos e localização do site, características de conservação e acessibilidade. Sua arquitetura conta com camadas de *Autoencoder*, resultando também em duas saídas, que indicam se a interação é funcional ou não. Entretanto, a comparação realizada entre os diferentes métodos é sobre o seu conjunto de dados e fornece apenas dados de desempenho para um conjunto de dados de teste independente muito limitado de 48 observações positivas extraídas de um experimento CLIP-seq.

Sendo uma das bases de dados de interações miRNA–alvo validadas experimentalmente, o TarBase (KARAGKOUNI et al., 2018) em sua versão 8 conta com mais de 1 milhão de registros, correspondendo a aproximadamente mais de 600 mil pares únicos de miRNA-alvo. As interações são apoiadas por mais de 33 metodologias experimentais, aplicadas a aproximadamente 600 tipos de células/tecidos em aproximadamente 451 condições experimentais. A origem dos dados é de diferentes publicações científicas que utilizam tecnologias experimentais como: CLEAR-CLIP, CLASH, CLIP-chimeric, IMPACT-seq, AGO-IP, RPF-seq, RIP-seq, e outros.

O banco de dados foi preenchido com entradas derivadas da curadoria manual de manuscritos realizando a análise experimental de miRNAs. Os curadores observaram o miRNA, o gene alvo relacionado, bem como informações sobre o experimento, como a linha celular ou tecido utilizado. O projeto disponibiliza uma ferramenta online para a realização de consulta das interações e seus dados estão disponíveis publicamente para download.

3.1 Discussão

Em resumo, conseguimos concluir desta breve revisão sobre alguns dos trabalhos relacionados o quão importante são os dados utilizados durante o treinamento e posteriormente o teste. Na grande maioria dos trabalhos atuais existe uma carência de dados em relação a quantidade assim como também sobre a independência. Outro ponto bastante relevante sobre as dificuldades atuais está relacionado ao viés introduzido por atributos manuais na identificação de regiões de interação miRNA-alvo. Nesse sentido, uma das vantagens na utilização de aprendizado de máquina e, em especial, aprendizado profundo está na possibilidade do seu funcionamento sem a necessidade de definição de atributos manualmente baseado m conhecimento prévio, além de desenvolver o aprendizado usufruindo de informações presentes nos dados. DeepMirTar é um exemplo promissor que apresenta resultados interessantes somente com dados de sequência de entrada, comprovando que é possível prever alinhamento sequências com precisão e prevendo protuberâncias e incompatibilidades. Por fim, observamos que a representação do domínio em trabalhos relacionados ainda permanece inexplorada, além disso, em nenhum dos trabalhos levantados na literatura os modelos foram treinados especificamente para descoberta de miRNAs-alvos em câncer como em nosso modelo proposto.

4 MATERIAIS E MÉTODOS

Este capítulo descreve os materiais e métodos empregados no desenvolvimento do presente trabalho. O capítulo inicia apresentando os dados coletados e as etapas de pré–processamento dos mesmos. Na sequência, são detalhados os aspectos metodológicos da criação de um grafo e do treinamento do modelo a partir desta estrutura com o algoritmo HinSAGE. Por fim, o capítulo apresenta os cenários definidos para os experimentos que foram realizados, cujos resultados serão apresentados no Capítulo 5.

4.1 Coleta e pré-processamento de dados

Conforme discutido na Seção 2.3.2, redes neurais de grafos são algoritmos de aprendizado profundo projetados para operar sobre dados estruturados na forma de grafos. O aprendizado da representação de nós no grafo, isto é, a geração dos *embeddings*, se dá através de atualizações iterativas que agregam as representações dos nós vizinhos e do próprio nó em iterações anteriores. Desta forma, a base do treinamento de uma rede neural de grafo é composta pela estrutura do grafo (*i.e.,* lista de interações entre nós) e, usualmente, por atributos vinculados aos nós ou arestas do grafo. Nesta seção, detalhamos como foram coletadas as interações para definição da estrutura do grafo e os dados de expressão gênica que são utilizados como atributos dos nós no processo de treinamento do modelo.

4.1.1 Dados de interações miRNA–mRNA e mRNA–mRNA

Com o objetivo de construir nosso grafo estruturado a partir de dados significativos para o aprendizado de padrões relacionados a alvos de miRNAs, primeiramente realizamos o levantamento de algumas das principais bases de dados responsáveis em catalogar interações miRNAs–alvo experimentalmente validadas. Algumas das bases tradicionais neste escopo e já empregadas em trabalhos anteriores são a MirTarBase (HUANG et al., 2021) e a TarBase (KARAGKOUNI et al., 2018), que juntas reúnem milhares de interações curadas manualmente da literatura. O MirTarbase release 9.0 possui mais de dois milhões de interações para 37 espécies distintas, enquanto o Tarbase v.8 possui aproximadamente 665 mil interações abrangendo 18 espécies. Entretanto, estas bases possuem

apenas interações miRNAs–alvos, o que poderia tornar o grafo muito esparso tendo em vista os padrões de conexão de miRNAs com seus alvos e o fato de que muitos miRNAs não possuem um número significativo de alvos validados experimentalmente, assim como diversos mRNAs ainda não possuem evidências experimentais de regulação por miRNAs.

Desta forma, optou-se por utilizar uma base de dados alternativa, mas igualmente abrangente na tarefa de catalogar interações gênicas, denominada RNAInter (KANG et al., 2022). O RNAInter v4.0 integra dados de interatoma de RNA coletados a partir da literatura e de outros bancos de dados, incluindo interações validadas experimentalmente e preditas computacionalmente. A versão atual possui mais de 47 milhões de interações anotadas para 156 espécies, incluindo interações do tipo RNA–DNA, RNA–Proteína e RNA–RNA, dentre outras. Adicionalmente, o RNAInter calcula um *score* de confiança para cada interação disponibilizada com base em características como confiança da evidência experimental, confiança da comunidade científica dada pelo número de citações do artigo e número de tecidos ou células distintas em que a interação foi identificada. Estes fatores, segundo os autores (KANG et al., 2022), foram integrados a partir de uma função sigmoide para cálculo do *score*, tal que as interações relatadas em artigos altamente citados, envolvendo detecção em maior número de tecidos/células e realizando experimentos em pequena escala receberiam um *score* de confiança mais alto (KANG et al., 2022).

Todas as interações miRNA–mRNA foram exportadas do RNAInter (KANG et al., 2022), acompanhadas de metadados como espécie, identificador único do miRNA, identificador único do mRNA, categorização do tipo de evidência experimental para a interação entre forte (*strong*) ou fraca (*weak*), bem como se o registro origina-se de uma predição computacional. Adicionalmente, a fim de construir um grafo mais abrangente para o processo de aprendizado, englobando também mRNAs sem interações com miRNAs catalogadas nesta base, fizemos o download da lista de interações mRNA–mRNA disponibilizada pelo RNAInter (KANG et al., 2022). Estas interações mRNA–mRNA possuem os mesmos metadados mencionados anteriormente. O objetivo de incluir interações entre mRNAs está na constatação de que diversos trabalhos relacionados possuem um alto número de falsos positivos em suas predições em decorrência, ao menos parcialmente, da carência de interações negativas. Assim, visamos enriquecer o conjunto de dados utilizado para construção do grafo, propondo uma nova abordagem que mapeia interações entre reguladores e alvos, bem como entre alvos, a fim de tentar aprimorar o poder preditivo do modelo na identificação de padrões de interações miRNAs–alvos.

Salientamos que apenas foram mantidas para análise no presente trabalho as inte-

rações miRNA–mRNA e mRNA–mRNA relacionadas a humanos (*Homo sapiens)*, tendo em vista que nosso objetivo é a predição de alvos de miRNAs relacionados a câncer em humanos. Adicionalmente, os dados foram pré-processados para remover interações que possuíam registros com valores inconsistentes, como identificadores de miRNA ou mRNA indefinidos, e descartar interações preditas computacionalmente. A etapa de remoção de interações derivadas de predição computacional visa melhorar a qualidade dos dados de treinamento, mantendo apenas interações suportadas por evidências experimentais, sejam evidências fortes ou fracas. É importante notar que algumas interações podem aparecer na base de dados associadas a ambos os tipos de evidência experimental. Após estas etapas de limpeza e pré-processamento dos dados, o arquivo final contém interações miRNA–alvo e mRNA–mRNA com identificadores consistentes para miRNAs e mRNAs, e com informações sobre *score* de confiança e tipo de evidência experimental usada na detecção da associação. Na Tabela 4.1 apresentamos os números resultantes da etapa de coleta e pré-processamento de dados de interações miRNA–alvo (a) e mRNA–mRNA (b).

Tabela 4.1 – Quantidade de interações do tipo (a) miRNA–mRNA e (b) mRNA–mRNA após o pré-processamento dos dados obtidos do RNAInter v4.0.

Conjunto de dados	**Quantidade**
Strong	322171
Weak	786127
Strong ∪ *Weak*	1040032
Strong ∩ *Weak*	68266

(a) Interações miRNA–alvo

Conjunto de dados	**Quantidade**
Strong	279
Weak	54458
Strong ∪ *Weak*	54703
Strong ∩ *Weak*	34

(b) Interações mRNA–mRNA

4.1.2 Dados de expressão gênica em câncer

Conforme discutido na Seção 2.3.2, os algoritmos de redes neurais de grafos, como o GraphSAGE e o HinSAGE, utilizam atributos relacionados aos nós do grafo para gerar os *embeddings* a serem utilizados como base da tarefa preditiva. Dado o conhecimento prévio acerca do envolvimento dos miRNAs na regulação de processos biológicos relacionados à carcinogênese e em tecidos tumorais, a hipótese deste trabalho é que a integração de dados de expressão diferencial em câncer no processo de treinamento do modelo poderia aprimorar a detecção de alvos de miRNAs. De fato, inúmeros trabalhos baseados na análise integrativa de dados de transcriptoma para miRNAs e mRNAs demonstram o potencial da análise de correlação de níveis de expressão entre estas mo-

léculas para a descoberta de novas interações miRNA–mRNA (SKOK et al., 2019; LI et al., 2018; XIAO et al., 2012).

Assim, na segunda etapa da coleta de dados, identificamos bases de dados catalogando dados de expressão gênica para miRNAs e mRNAs em variados tipos de tumores. O projeto *The Cancer Genome Atlas* (TCGA) é um dos principais consórcios relacionados à genômica do câncer, tendo gerado um grande volume de dados para a caracterização molecular de mais de 11 mil amostras em 33 tipos de câncer. Os dados do TCGA são acessíveis publicamente e tornaram-se valiosas fontes para a investigação sistemática dos mais variados tipos de câncer, tendo sido utilizados em diversos trabalhos (BLANCO; PAZOS; FERNANDEZ-LOZANO, 2021) e disponibilizados com diferentes níveis de pré–processamento em diferentes bases de dados (SETTINO; CANNATARO, 2018).

No escopo deste trabalho, os dados do TCGA foram obtidos através do portal GDAC FireBrowse[1], disponibilizado pelo *Broad Institute of MIT and Harvard*, o qual fornece uma interface amigável para exploração interativa dos dados e dos resultados gerados pelos métodos e pipelines desenvolvidos no Broad Institute para processar e analisar vários tipos de dados genômicos e proteômicos oriundos do TCGA. O portal oferece a opção de download dos conjuntos de dados do TCGA pré-processados pelo instituto de acordo com protocolos bem estabelecidos de bioinformática[2]. Embora diversos tipos de dados genômicos estejam disponíveis, obtivemos apenas os dados de expressão gênica em larga escala. Estes dados englobam os níveis de expressão de miRNAs e mRNAs para um conjunto de amostras de tumor e amostras não-tumorais (*i.e.,* representando amostras normais), com o tamanho amostral por grupo variando entre os tipos de câncer analisados.

Foram selecionados somente os tipos de câncer que não apresentavam um grande desbalanceamento entre o número de amostras tumorais e não tumorais, ou seja, quantidade de dados próximo ao equivalente para os dois cenários. Os dados clínicos foram utilizados a fim de mapear cada amostra para o seu respectivo tipo (*i.e.,* tumoral ou não-tumoral). Durante o pré-processamento destes dados, foram removidas todas as amostras classificadas como *Recurrent Solid Tumor* e *Metastatic*, bem como registros sem identificadores consistentes para miRNAs e mRNAs. Nosso conjunto final ficou composto por amostras classificadas como *Primary Tumor* (PT) e *Solid Tissue Normal* (TN) para os seguintes tumores: *Bladder Urothelial Carcinoma* (BLCA), *Breast invasive carcinoma* (BRCA), *Cholangiocarcinoma* (CHOL), *Esophageal carcinoma* (ESCA), *Glioma*

[1] <http://firebrowse.org/>

[2] Os arquivos pré-processadores escolhidos cara cada tumor analisado foram: `illuminahiseq_rnaseqv2-RSEM_genes_normalized` e `miRseq_Mature_Preprocess`.

(GBMLGG), *Head and Neck squamous cell carcinoma* (HNSC), *Kidney Chromophobe* (KIPAN), *Liver hepatocellular carcinoma* (LIHC), *Lung adenocarcinoma* (LUAD), *Pancreatic adenocarcinoma* (PAAD), *Prostate adenocarcinoma* (PRAD), *Stomach adenocarcinoma* (STAD), *Stomach and Esophageal carcinoma* (STES), *Thyroid carcinoma* (THCA) e *Uterine Corpus Endometrial Carcinoma* (UCEC). A Tabela 4.2 apresenta os números de amostras por tipo de câncer, grupo amostral, e tipo de dado (mRNASeq ou miRSeq) resultantes do pré-processamento.

Tabela 4.2 – Quantidade de amostras por tipo de câncer, grupo amostral e tipo de dado após a realização do pré-processamento dos dados de expressão gênica.

Tumor	mRNASeq (PT)	mRNASeq (TN)	miRSeq (PT)	miRSeq (TN)
BLCA	408	19	409	19
BRCA	1093	112	1078	104
CHOL	36	9	36	9
ESCA	184	13	184	13
GBMLGG	669	5	512	5
HNSC	520	44	523	44
KIPAN	889	129	873	130
LIHC	371	50	372	50
LUAD	515	59	513	46
PAAD	178	4	178	4
PRAD	497	52	494	52
STAD	416	37	436	41
STES	600	50	620	54
THCA	501	59	502	59
UCEC	546	35	538	33

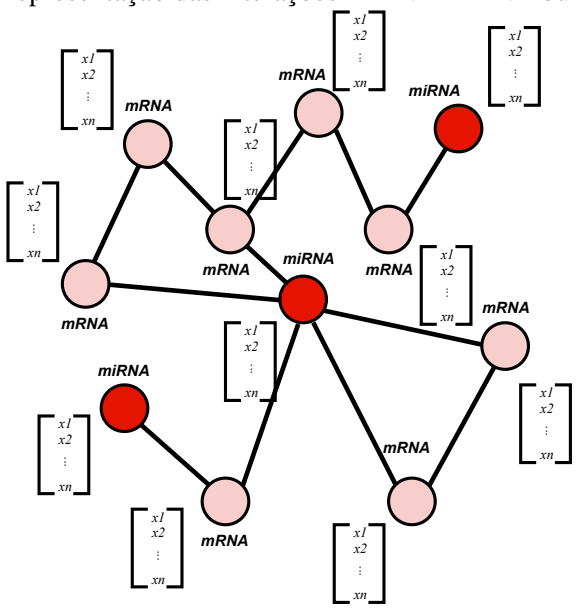
Após o pré-processamento básico dos dados de expressão, os dados foram transformados para a escala log2 e foi calculado um índice de expressão diferencial para cada miRNA e cada mRNA, denominado *log Fold Change* (logFC), associado a cada tipo de tumor. O logFC expressa a razão entre as médias da expressão gênica para um determinado miRNA ou mRNA entre dois grupos amostrais – neste estudo, amostras tumorais vs. amostras não-tumorais para um determinado tipo de câncer. Quanto maior o seu valor absoluto, assume-se que a molécula possui maior relevância biológica no domínio estudado. Após esta etapa, é atribuído um valor de logFC referente a cada tipo de câncer, para cada miRNA e mRNA no conjunto de dados. Tendo em vista que os dados de expressão se referem a 15 tipos de câncer, o processamento de dados de expressão gênica e a sumarização destes padrões através de valores de logFC geram um vetor de atributos de comprimento 15 para cada miRNA e mRNA no grafo.

4.2 Integração de dados para geração do grafo

Após as etapas de coleta e pré-processamento de dados descritas na seção anterior, o processo de construção do conjunto de dados para treinamento do modelo envolveu a integração dos dados de interações obtidas da base de dados RNAInter com os dados de expressão gênica coletados do FireBrowse. Esta integração foi realizada utilizando o identificador único existente para cada miRNA e para cada mRNA na base de dados formada. Durante essa etapa também foram removidas as interações miRNA-alvo sem nenhum registro de expressão identificada. Essa remoção teve como objetivo mitigar a indução de resultados incoerentes durante o treinamento do modelo proposto.

A Figura 4.1 resume a proposta do presente trabalho em relação à integração destes diferentes tipos de dados utilizando a representação de grafos, na qual os nós em vermelho representam miRNAs, os nós em rosa claro representam mRNAs, e as interações podem ser do tipo miRNA–mRNA ou mRNA–mRNA. Além disso, cada nó do grafo (miRNA ou mRNA) possui um vetor de atributos associado, contendo os padrões de expressão diferencial em termos de logFC nos 15 tipos de câncer estudados. A rede de interações final é composta por mais de 100 mil interações miRNA–alvo e quase 30 mil interações mRNA–mRNA. A rede possui um total de 2617 mil nós de miRNAs e 17252 mil nós de mRNAs.

Figura 4.1 – Exemplo da estrutura do grafo proposto, com os nodos em vermelho representando miRNAs e os nodos em cor rosa claro denotando os mRNAs. Cada nodo possui um vetor de atributos composto pelos valores de expressão diferencial nos vários tipos de câncer analisados. As arestas são a representação das interações miRNA–mRNA ou mRNA–mRNA.



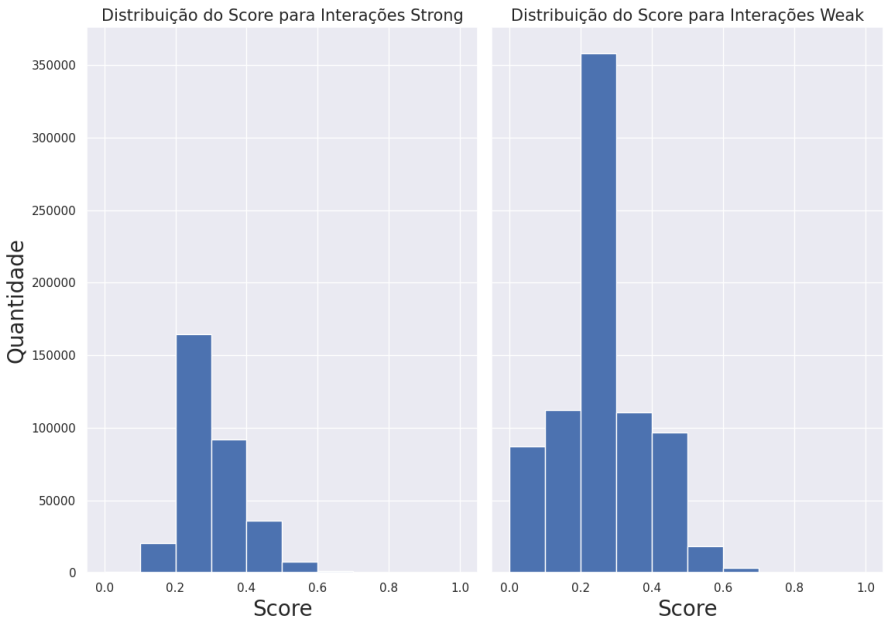
Fonte: O Autor.

4.3 Análise exploratória do conjunto de dados e critérios de filtragem

Com o objetivo de melhor compreender a estrutura e natureza do conjunto de dados criado, bem como identificar possíveis anomalias, foram realizadas algumas análises exploratórias que se mostraram importantes posteriormente para auxiliar na construção dos cenários experimentais propostos para treinamento e avaliação do modelo preditivo.

A Figura 4.2 apresenta um histograma baseado na definição de *score* para cada interação do conjunto de dados coletados, sendo este um dos metadados mais interessantes que a base de dados RNAInter (KANG et al., 2022) provê. Neste histograma, é possível identificar uma notável diferença na distribuição de *scores* para as interações do tipo *Strong* (suportadas por evidência experimental forte) e do tipo *Weak* (suportadas por evidência experimental fraca), bem como no número de interações classificadas como *Strong* e *Weak*. Essa grande quantidade de dados com evidências experimentais descritas como fracas pode ser um dos fatores influenciando o elevado número de interações falso positivas preditas por modelos de AM.

Figura 4.2 – Histograma da distribuição original das interações considerando o *score* de confiança para registros descritos como *Strong* e *Weak*.

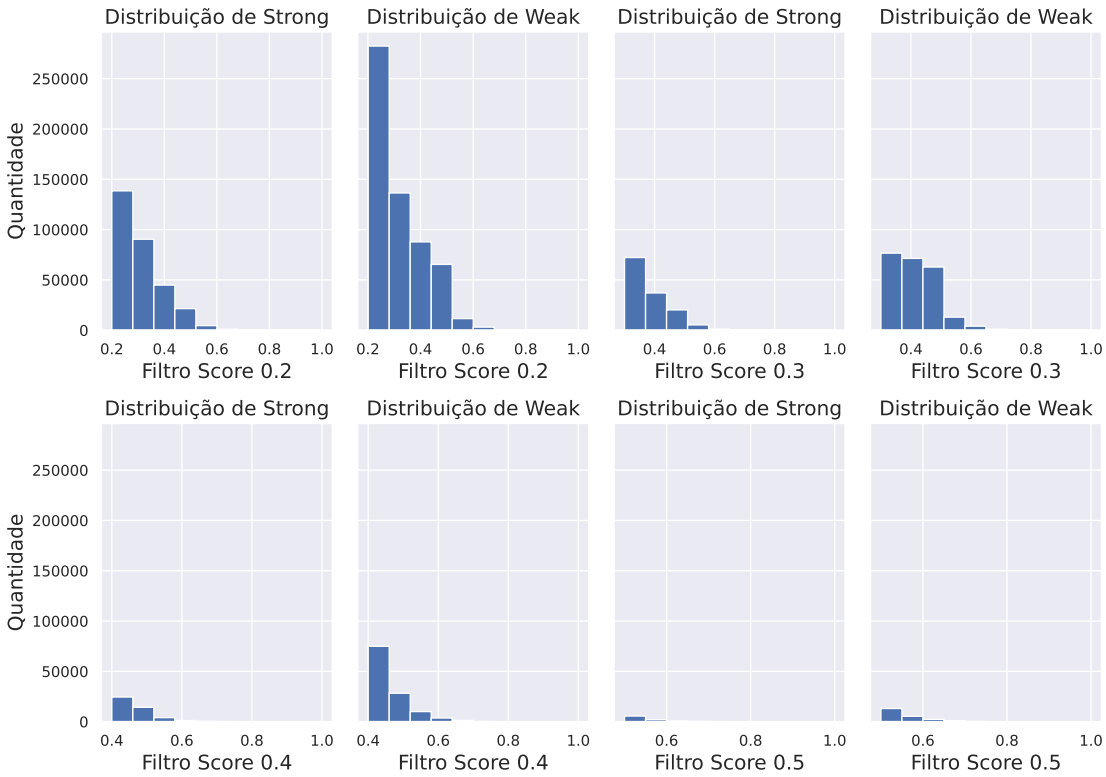


Fonte: O Autor.

Ao identificar essa grande variação de *score* dentre as interações coletadas, optamos em realizar alguns conjuntos de filtros sobre o valor de *score* com o objetivo de posteriormente explorar o efeito destes filtros no poder preditivo do modelo. A intenção é avaliar experimentalmente o impacto de utilizar dados menos ou mais restritivos quanto

ao nível de confiança atribuído às interações utilizadas para treinamento na capacidade do modelo de prever novas interações miRNA–mRNA. Para tanto, a partir do conjunto de dados construído originalmente, produzimos quatro novos conjuntos de dados baseados em filtros aplicados sobre os valores de *score*, utilizando os limiares de 0.2, 0.3, 0.4 e 0.5. Os resultados são mostrados na Figura 4.3.

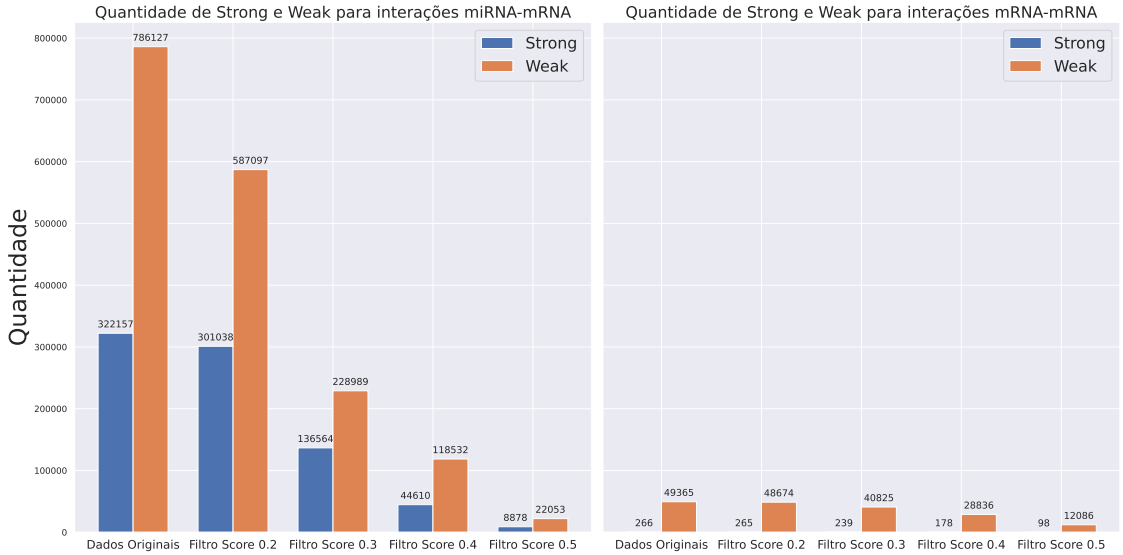
Figura 4.3 – Histograma da distribuição das interações classificadas como *Strong* e *Weak* após a aplicação de filtros sobre o *score* de confiança.



Fonte: O Autor.

Por fim, a última análise gráfica dos dados visou explorar a distribuição de interações classificadas como *Strong* e *Weak* para os dois tipos de interações incluídas no nosso grafo: miRNA–mRNA e mRNA–mRNA. Esta análise sumariza a quantidade de interações para cada nível de evidência experimental, tanto para os dados originais como para os diferentes conjuntos de dados gerados a partir de filtragens sobre o valor de *score*. A Figura 4.4 apresenta os resultados obtidos. A análise destes resultados deixa evidencia o desbalanceamento de interações descritas como *Strong* e *Weak* nos dados coletados, e como a aplicação dos filtros propostos impacta diretamente nessa distribuição original. A construção de diferentes conjuntos de dados para treinamento baseado neste *score*, conforme será detalhado posteriormente na Seção 4.4.1, tem como objetivo investigar a influência desta definição sobre o desempenho do modelo.

Figura 4.4 – Quantidade de interações descritas como forte e fraca sobre os diferentes conjuntos de dados criados sobre o *score* de confiança.



Fonte: O Autor.

4.4 Desenvolvimento do modelo preditivo

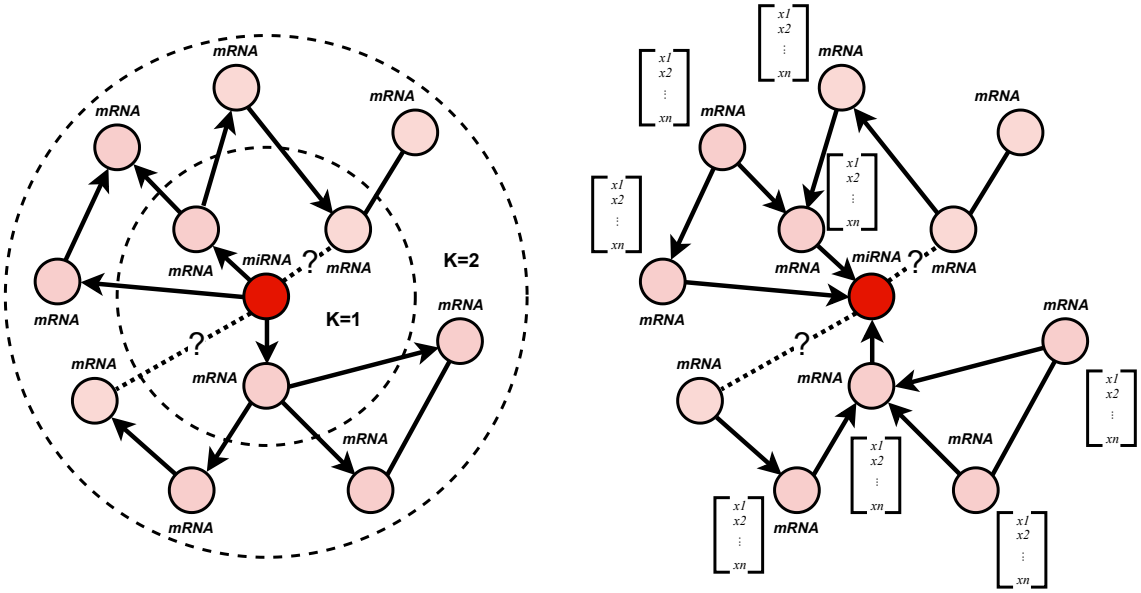
O desenvolvimento dos modelos preditivos foi realizado com a biblioteca em Python StellarGraph[3] (DATA61, 2018), que tem como objetivo facilitar o treinamento de modelos de redes neurais de grafos. Dentre os algoritmos de aprendizado disponibilizados pela biblioteca, foi selecionado o HinSAGE, o qual conforme explicado anteriormente, é uma generalização do algoritmo GraphSAGE para aplicação em grafos heterogêneos.

A modelagem do problema foi realizada com base na tarefa de predição de *links* (*i.e.,* arestas) em grafos, que visa treinar um modelo capaz de prever se uma aresta que ainda não está no grafo deve existir, considerando os padrões aprendidos a partir dos dados. Este procedimento é exemplificado na Figura 4.5, adaptada do exemplo descritivo do funcionamento do algoritmo GraphSAGE (HAMILTON; YING; LESKOVEC, 2017). A linha tracejada na figura representa o momento da indução do algoritmo para predizer novas interações entre os nós existentes no grafo. Essa predição ocorre por meio dos *embeddings* obtidos dos vizinhos observados, com o hiperparâmetro $K = 2$ definindo quantos saltos (*i.e., hops*) devem ser dados a partir do nó analisado para calcular os *embeddings*. Por exemplo, para $K = 2$, os *embeddings* são gerados a partir dos vizinhos diretamente ligados a um determinado nó, e dos vizinhos dos vizinhos destes nós. Ainda, neste caso específico, é possível observar a particularidade da estrutura do grafo

[3]<https://stellargraph.readthedocs.io/en/stable/>

base gerado, o qual contempla diferentes tipos de nós, bem como as diferentes interações existentes (miRNA–mRNA e mRNA–mRNA).

Figura 4.5 – Ilustração do comportamento de indução que ocorre no algoritmo HinSAGE para a predição de links considerando o grafo heterogêneo e a tarefa de predição de alvos de miRNAs.



Fonte: Adaptada de Hamilton, Ying and Leskovec (2017).

Para a criação do grafo utilizando a biblioteca StellarGraph, foi empregado a classe `StellarGraph` disponibilizada pela biblioteca que recebe como entrada, em formato de *dataframe*, o conjunto de nós e arestas definidos a partir da coleta de dados (Seção 4.1). Além disso, em nosso contexto também foi preciso informar a existência de mais de um tipo de aresta no grafo, especificando que apenas estamos interessados em predizer arestas entre nós referentes a miRNAs e mRNAs, isto é, interações miRNA–mRNA. Desta forma, o modelo não será treinado para predição de interações mRNA–mRNA, visto que isto está fora do escopo do presente trabalho. Após esse processo, é retornado um grafo que posteriormente será utilizado como base para geração dos dados a serem usados no treinamento e validação do modelo.

Nota-se que para viabilizar o desenvolvimento e avaliação do modelo, uma etapa imprescindível é a criação de dados de treinamento, validação e teste. As seções a seguir irão descrever a geração destes conjuntos de dados, e os detalhes sobre configuração de hiperparâmetros e avaliação do modelo.

4.4.1 Geração dos dados de treinamento, validação e teste

Considerando o nosso domínio indutivo, o conjunto de treinamento, validação e teste também são grafos obtidos a partir de reduções do grafo original, a fim de gerar dados independentes para as diferentes etapas envolvidas no desenvolvimento do modelo.

A biblioteca StellarGraph disponibiliza um método denominado `EdgeSplitter`, que realiza a operação de divisão do grafo a partir de um grafo fornecido como entrada para criar conjuntos de treinamento e teste em tarefas de predição de *links*. Este método recebe como entrada tanto o grafo a ser subdividido, como uma porcentagem em função do número total de arestas no grafo fornecido que define quantas arestas serão amostradas como exemplos positivos e negativos. As arestas positivas são amostradas a partir do conjunto de arestas reais presentes no grafo, de forma a manter a conectividade do grafo. As arestas negativas são criadas aleatoriamente e amostradas a partir de pares de miRNAs-mRNAs que não estão conectados por uma interação no grafo original. Esta amostra de exemplos negativos tem como objetivo aprimorar a dedução do modelo na identificação de arestas falso positivas.

Um aspecto importante na metodologia de desenvolvimento, portanto, foi definir a proporção de arestas a serem usadas como exemplos positivos e negativos durante o treinamento e validação do modelo. Dado que o hiperparâmetro p que define esta proporção é uma porcentagem calculada em função do número de arestas no grafo original, foi preciso ajustá-lo conforme o conjunto de interações incluído na criação deste grafo, isto é, se é o conjunto de interações completo ou um conjunto com interações filtradas a partir do valor do *score*, conforme discutido na Seção 4.3. Adicionalmente, o valor do hiperparâmetro p precisa ser compatível com o número de interações negativas que podem ser aleatoriamente criadas a partir da estrutura original do grafo de entrada, sendo esta uma restrição aplicada pela própria biblioteca na utilização do método.

Para cada conjunto de dados a ser utilizado nos nossos experimentos, configuramos o valor de p como o maior valor possível suportado pelo método disponibilizado pela biblioteca StellarGraph quando aplicado aos nossos dados. O maior valor possível de p é determinado pelo tamanho do grafo em conjunto da condição do grafo continuar conexo, ou seja, existe um caminho entre qualquer par de nós. Estes valores foram determinados empiricamente, através de múltiplas tentativas, e são sumarizados na Tabela 4.3. Esta tabela também sumariza o número aproximado de arestas utilizadas como exemplos positivos e negativos em cada caso. É importante enfatizar que o conjunto de dados rotulados

é sempre balanceado entre as classes positiva e negativa.

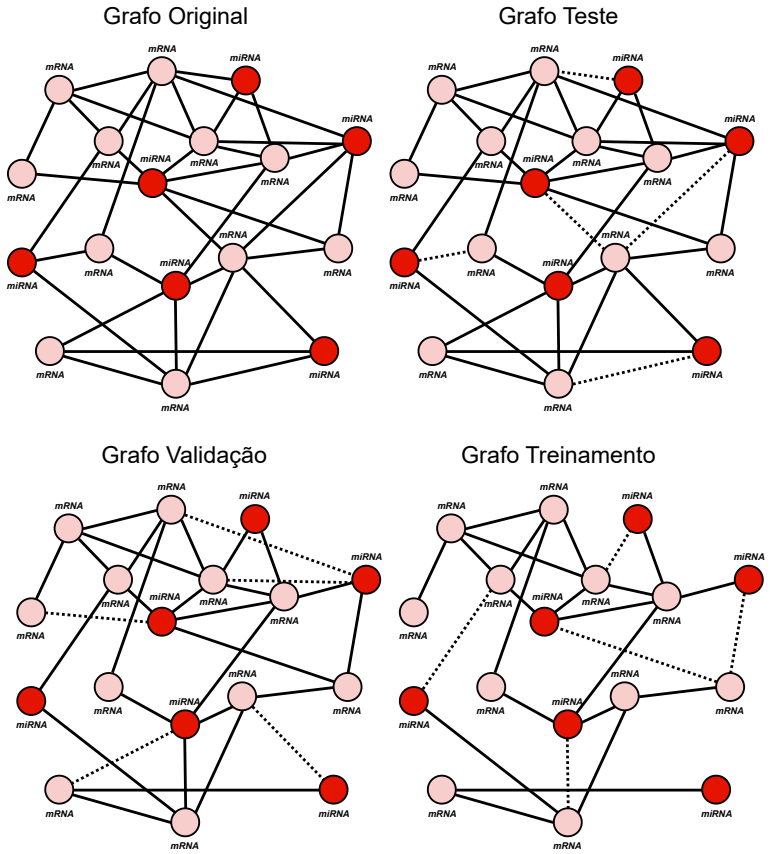
Tabela 4.3 – Apresentação da construção do conjunto de treinamento, validação e teste

Conjunto de dados	# miRNA-mRNA	# mRNA-mRNA	Hiperparâmetro p	# Arestas positivas	# Arestas negativas
Conjunto de dados original, com todas as interações					
Teste	1040018	49601	0.004	4160	4160
Validação	1035858	49601	0.004	4143	4143
Treinamento	1031715	49601	0.004	4126	4126
Conjunto de dados com interações filtradas por *score* >= 0.2					
Teste	820906	48909	0.004	3283	3283
Validação	817623	48909	0.004	3269	3269
Treinamento	814354	48909	0.004	3255	3255
Conjunto de dados com interações filtradas por *score* >= 0.3					
Teste	309892	41039	0.01	3098	3098
Validação	306794	41039	0.01	3067	3067
Treinamento	303727	41039	0.01	3036	3036
Conjunto de dados com interações filtradas por *score* >= 0.4					
Teste	132992	28999	0.01	1329	1329
Validação	131663	28999	0.01	1316	1316
Treinamento	130347	28999	0.01	1303	1303
Conjunto de dados com interações filtradas por *score* >= 0.5					
Teste	22663	12176	0.01	226	226
Validação	22437	12176	0.01	223	223
Treinamento	22214	12176	0.01	220	220

A Figura 4.6 demonstra como a amostragem de arestas positivas e negativas é realizada a partir do grafo original a fim de gerar os conjuntos de treinamento, validação e teste. O método `EdgeSplitter` é aplicado em três etapas sucessivas. Primeiramente, realiza-se a amostragem de arestas para compor o conjunto de teste, utilizando o valor definido para o hiperparâmetro p (Tabela 4.3). As arestas tracejadas no grafo de teste representam a escolha aleatória do método para compor o conjunto de exemplos positivos, enquanto exemplos negativos são criados aleatoriamente a partir de pares de nós que não estão originalmente conectados no grafo, sempre na mesma proporção entre ambos os conjuntos. Por uma questão de visualização, as arestas negativas não são mostradas na figura. As arestas positivas selecionadas aleatoriamente para teste são, então, removidas do grafo original, e o mesmo processo de amostragem de arestas positivas e negativas é aplicado para gerar o conjunto de validação. Nota-se que a amostragem de exemplos positivos e negativos para o conjunto de validação ocorre a partir de um grafo reduzido. Após esta etapa de gerar exemplos de validação, ocorre finalmente a redução do grafo para amostrar exemplos positivos e negativos para a etapa de treinamento.

As arestas amostradas como exemplos positivos para treinamento, validação e teste não serão usadas durante o processo de treinamento como arestas de passagem de informação para criação de *embeddings*, mas sim apenas como arestas de "supervisão" a fim de acompanhar o processo de aprendizado. Essa sequência de construção segue orientações definidas na documentação fornecida pela biblioteca StellarGraph, com exemplos de casos de uso.

Figura 4.6 – Processo de criação de conjuntos de teste, validação e treinamento a partir do grafo original. As arestas tracejadas representam arestas positivas amostradas em cada etapa, as quais são removidas do grafo para as etapas subsequentes.



Fonte: O Autor.

4.4.2 Treinamento e avaliação do modelo baseado em grafos

A biblioteca StellarGraph apresenta um conjunto de exemplos para os diferentes tipos de algoritmos disponibilizados, inclusive para o algoritmo HinSAGE adotado neste estudo, e sugere valores padrões para os hiperparâmetros envolvidos no treinamento dos modelos. Sendo assim, no primeiro momento, optamos em seguir a implementação descrita nos exemplos fornecidos, sem realizar alterações sobre os valores dos hiperparâmetros. Esta decisão tem como principal objetivo explorar e entender o comportamento do modelo GNN sobre os diferentes conjuntos de dados criados e descritos na Seção 4.3.

A configuração inicial dos hiperparâmetros é resumida na Tabela 4.4. Variações nos valores dos hiperparâmetros foram implementadas em uma série de experimentos que serão detalhados mais adiante, na Seção 5.1. O otimizador Adam foi usado em todos os experimentos, variando a taxa de aprendizado em alguns cenários experimentais. A função de custo adotada foi a Binary Cross-Entropy.

Tabela 4.4 – Lista de hiperparâmetros e respectivos valores usados inicialmente nos experimentos.

Hiperparâmetro	Valores iniciais
Tamanho do batch (*batch_size*)	200
Número de épocas (*epochs*)	300
Número de saltos (K)	2
Número de vizinhos amostrados por salto (*num_sample*)	[8, 4]
Número e tamanho das camadas do modelo (*hinsage_layer_size*)	[32, 32]
Taxa de aprendizado (*learning_rate*)	0.001
Função de custo	Binary Cross-Entropy
Função de ativação	ReLU
Otimizador	Adam

Todos os experimentos foram baseados na divisão dos dados em treinamento, validação e teste, conforme explicado na Seção 4.4.1, repetindo-se 10 vezes cada experimento a fim de avaliar o desempenho médio e a variância no desempenho ao longo de múltiplas execuções do algoritmo. Cada execução é baseada em uma *seed* aleatória distinta. A avaliação de desempenho dos modelos treinados com o HinSAGE foi realizada com base nas métricas acurácia, precisão, sensibilidade, F1-Score e ROC AUC, conforme definições apresentadas no Capítulo 2.

Tendo em vista que a maioria das métricas de desempenho utilizadas são baseadas na comparação de rótulos (*i.e.,* classes) preditos e verdadeiros, em alguns casos faz-se necessário definir um limiar ou ponto de corte para as probabilidades por classe preditas pelos modelos de AM. Assim, definimos como limiar de probabilidade padrão o valor de 0.5, o que significa que interações miRNA–mRNA preditas pelo modelo com probabilidade igual ou superior a 0.5 serão classificadas como interações positivas, e interações com probabilidade inferior a 0.5 serão classificadas como negativas.

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta e discute os experimentos e resultados alcançados no presente estudo. Todos os experimentos foram desenvolvidos com base na metodologia apresentada no Capítulo 4. Eventuais modificações metodológicas em decorrência de necessidades específicas de cada experimento serão detalhadas ao longo deste capítulo.

5.1 Definição dos experimentos

Para o desenvolvimento do presente trabalho, foram definidos dois conjuntos de experimentos principais. O primeiro conjunto é centrado na análise do potencial do algoritmo HinSAGE para predição de alvos de miRNAs usando uma abordagem de aprendizado baseado em grafo. O segundo conjunto tem como foco comparar a abordagem desenvolvida neste estudo com outros métodos propostos na literatura. As seções a seguir detalharão cada um dos conjunto de experimentos.

5.1.1 C1: Análise experimental do desempenho preditivo do algoritmo HinSAGE

No primeiro conjunto de experimentos (**C1**), definimos diversos cenários experimentais a partir de variações nos critérios de filtragem do conjunto de dados e nos valores dos hiperparâmetros envolvidos no treinamento do algoritmo. O objetivo é avaliar a influência de aspectos relacionados à construção do conjunto de dados de interações e à configuração do algoritmo de aprendizado sobre os resultados alcançados e determinar uma abordagem de aplicação do algoritmo para tentar alcançar um melhor desempenho preditivo do modelo. Estas variações resultaram em um total de 21 cenários experimentais que estão descritos na Tabela 5.1. Para facilitar o entendimento, organizamos os cenários experimentais em grupos, onde cada grupo possui um objetivo de experimentação bem específico, conforme definido a seguir. Todos os grupos são avaliados conforme o seu impacto no poder preditivo do modelo gerado.

- **Grupo 1 (G1)** : avaliação do impacto do uso de diferentes conjuntos de dados para definição do grafo base a ser usado no treinamento do modelo, construídos com a remoção de interações mRNA–mRNA e com diferentes filtragens de *score*;
- **Grupo 2 (G2)**: avaliação do impacto da remoção de interações mRNA–mRNA

Tabela 5.1 – Definição dos cenários experimentais explorados no conjunto de experimentos **C1**, para explorar o potencial do algoritmo HinSAGE na predição de alvos de miRNAs.

Estruturação dos experimentos								
Grupo	**Nº.**	**Conjunto de dados**	**mRNA–mRNA**	**batch_size**	**epochs**	**num_sample**	**hinsage_layer_size**	**learning_rate**
Grupo 1	00	Original	✓	200	300	[8, 4]	[32, 32]	0.001
	01	Original		200	300	[8, 4]	[32, 32]	0.001
	02	Interações com *score* >= 0.2	✓	200	300	[8, 4]	[32, 32]	0.001
	03	Interações com *score* >= 0.3	✓	200	300	[8, 4]	[32, 32]	0.001
	04	Interações com *score* >= 0.4	✓	200	300	[8, 4]	[32, 32]	0.001
	05	Interações com *score* >= 0.5	✓	200	300	[8, 4]	[32, 32]	0.001
Grupo 2	06	Interações com *score* >= 0.4		200	300	[8, 4]	[32, 32]	0.001
	07	Interações com *score* >= 0.5		200	300	[8, 4]	[32, 32]	0.001
Grupo 3	08	Interações com *score* >= 0.4	✓	200	100	[8, 4]	[32, 32]	0.001
Grupo 4	09	Interações com *score* >= 0.4	✓	1	100	[8, 4]	[32, 32]	0.001
	10	Interações com *score* >= 0.4	✓	50	100	[8, 4]	[32, 32]	0.001
	11	Interações com *score* >= 0.4	✓	100	100	[8, 4]	[32, 32]	0.001
	12	Interações com *score* >= 0.4	✓	300	100	[8, 4]	[32, 32]	0.001
Grupo 5	13	Interações com *score* >= 0.4	✓	300	100	[6, 3]	[32, 32]	0.001
	14	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[32, 32]	0.001
Grupo 6	15	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[64, 64]	0.001
	16	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[128, 128]	0.001
	17	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[16, 16]	0.001
Grupo 7	18	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[32, 32]	0.005
	19	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[32, 32]	0.01
	20	Interações com *score* >= 0.4	✓	300	100	[12, 6]	[32, 32]	0.0001
Grupo 8	21	Interações com *score* >= 0.4	✓	300	300	[12, 6]	[32, 32]	0.001

do grafo base no desempenho dos melhores modelos obtidos nos experimentos do Grupo 1;

- **Grupo 3 (G3)**: avaliação do impacto da redução do número de épocas utilizadas no processo de treinamento do modelo com o algoritmo HinSAGE;
- **Grupo 4 (G4)**: avaliação do impacto do tamanho do *batch* usado no treinamento do modelo com o algoritmo HinSAGE
- **Grupo 5 (G5)**: avaliação do impacto da variação no número de nós vizinhos amostrados como base na geração de *embeddings* pelo algoritmo HinSAGE;
- **Grupo 6 (G6)**: avaliação do impacto da variação no tamanho das camadas ocultas utilizadas no treinamento do modelo com o algoritmo HinSAGE;
- **Grupo 7 (G7)**: avaliação do impacto da variação da taxa de aprendizado utilizada pelo otimizador Adam;
- **Grupo 8 (G8)**: avaliação do impacto do aumento no número de épocas de treinamento com base no melhor modelo geral;

Dado o alto custo computacional envolvido na execução de cada experimento, ao final da execução de cada grupo de cenários experimentais, avaliamos os resultados alcançados a fim de identificar qual dos cenários é mais promissor para ser empregado no próximo grupo. Na tabela 5.1, é possível observar algumas linhas destacadas na coloração cinza, representando os destaques em relação a resultados obtidos com cada cenário executado (os quais serão detalhados e discutidos na Seção 5.2.1). Assim, ressaltamos que cada alteração realizada no treinamento do modelo é incremental, ou seja, as alterações

sobre os experimentos evidenciados são mantidas no próximo grupo de experimentos.

5.1.2 C2: Comparação do modelo baseado no HinSAGE com outras abordagens

No segundo conjunto de experimentos (**C2**), cujo objetivo é comparar o desempenho preditivo do modelo treinado utilizando o algoritmo HinSAGE com outros métodos ou modelos disponíveis na literatura, utilizamos como base o melhor modelo gerado no conjunto **C1** de experimentos, descrito na seção anterior. A partir da revisão da literatura, selecionados algumas abordagens relacionadas para realizar uma comparação em termos de qualidade de predições para conjuntos de teste.

Apesar da existência de muitos trabalhos relacionados abordando a tarefa de predição de alvos de miRNAs, ou de forma análoga, de interações miRNAs–mRNAs, conforme sumarizado no Capítulo 3, este conjunto de experimentos apresentou uma série de desafios importantes. Um dos principais desafios encontrados se deve à dificuldade de identificar, até o momento da escrita deste trabalho, um modelo para predição de alvos de miRNAs baseado no conceito de GNNs, ou seja, adotando abordagem similar à proposta neste trabalho. Salienta-se que GNNs foram previamente aplicadas em tarefas de predição relacionadas a miRNAs, mas com foco na predição da associação de miRNAs com doenças humanas (JI et al., 2021; YU; JU; REN, 2022) – tarefa esta que possui objetivos distintos do problema modelado neste trabalho. Assim, na etapa de comparação do modelo com abordagens prévias, foi inviável realizar uma comparação com outras abordagens equivalentes baseada na análise fim-a-fim de grafos com GNNs.

Uma segunda dificuldade está na grande variedade de atributos utilizados por trabalhos anteriores desenvolvendo modelos preditivos com AM, o que ocasiona uma evidente falta de padronização na construção de dados utilizados para teste dos modelos gerados. Por exemplo, alguns trabalhos focam em atributos obtidos a partir da análise da interação entre miRNAs e alvos candidatos, como conservação da sequência alvo, grau de complementariedade e termodinâmica da ligação, enquanto outros recebem como entrada a sequência do miRNA e de candidatos a gene alvo. Com isso, fornecer ou mesmo construir um conjunto de dados de entrada novo que atenda a especificidade de cada modelo ou ferramenta é um processo lento e bastante árduo. Esta dificuldade é ainda maior para métodos que não são disponibilizados através de ferramentas online, mas apenas através do compartilhamento do código fonte, demandando instalação local da aplicação e suas dependências.

Por fim, salientamos que os trabalhos relacionados envolvendo o uso de AM tradicional ou GNNs no desenvolvimento dos modelos não possuem foco específico na predição de alvos de miRNAs em câncer, mas abordam a tarefa de forma mais genérica, realização a predição de alvos em determinados organismos, como humanos. Desta forma, não foi possível estabelecer uma comparação de desempenho justa em termos de adotar os mesmos dados de treinamento e teste, no mesmo contexto. Entretanto, foram definidas estratégias de comparação a fim de extrair alguns *insights* a respeito do desempenho da abordagem proposta em relação ao estado da arte.

Decidimos realizar a comparação com os trabalhos relacionados mantendo o foco nos resultados das predições dos modelos para os dados de teste, buscando comparar as predições para as interações encontradas em comum entre o conjunto de teste criado neste trabalho e os conjuntos de teste usados em trabalhos relacionados. Para os trabalhos relacionados, as predições usadas foram aquelas disponibilizadas pelos autores dos artigos originais. A fim de não restringir muito o tamanho do conjunto de dados comum em razão da intersecção entre múltiplos métodos, nossa comparação foi realizada par a par com cada trabalho relacionado selecionado. Devido a grande maioria dos trabalhos relacionados não disponibilizarem as probabilidades preditas por classe, optamos por não utilizar a métrica ROC AUC nesta análise comparativa. Assim, as métricas analisadas foram acurácia, sensibilidade, precisão e F1-score. Mais detalhes sobre os trabalhos selecionados serão dados na Seção 5.2.2.1.

5.2 Resultados

Esta seção descreve os resultados alcançados no desenvolvimento deste trabalho. Iniciamos pela análise do desempenho do algoritmo HinSAGE em diversos cenários experimentais definidos a partir de variações no grafo base de treinamento e nos hiperparâmetros (Seção 5.2.1), e na sequência, apresentamos os resultados para a comparação entre o modelo treinado com o algoritmo HinSAGE e outras abordagens propostas na literatura (Seção 5.2.2), incluindo abordagens baseadas em AM tradicional e em aprendizado profundo.

5.2.1 Análise experimental do desempenho preditivo do algoritmo HinSAGE

5.2.1.1 G1: Impacto de variações no conjunto de interações

Em nosso primeiro grupo de experimentos, buscamos explorar e melhor compreender o comportamento do modelo de acordo com diferentes conjuntos de dados construídos para o treinamento. Sendo assim, nosso primeiro cenário (*i.e.,* Experimento 00) emprega o conjunto de dados original incluindo as interações miRNA–mRNA e mRNA–mRNA, e valores padrões para os hiperparâmetros envolvidos no algoritmo HinSAGE. Os resultados obtidos durante o treinamento apresentaram alguns números animadores, com acurácia próxima de 75%. No entanto, observando os gráficos da evolução de acurácia (Figura 5.1) e perda (Figura 5.2) ao longo das épocas de treinamento, é possível identificar que o Experimento 00 possui características da ocorrência de *overfitting*, iniciando a partir da época 50 e bastante evidente na análise de acurácia.

Em nosso segundo cenário proposto (*i.e.,* Experimento 01), queríamos entender se o emprego das interações mRNA–mRNA poderia estar contribuindo para a tendência de *overfitting* apresentada no experimento anteriormente descrito. Neste experimento, nosso modelo é treinado somente com interações miRNA–mRNA, incluindo todas as interações deste tipo coletadas no presente trabalho. Em suma, observando novamente os resultados obtidos ao longo do treinamento para os critérios de acurácia (Figura 5.1) e função de perda (Figura 5.2), não notamos nenhum ganho muito relevante. Deste modo, é possível afirmar que existe uma influência negativa no uso de interações mRNA–mRNA.

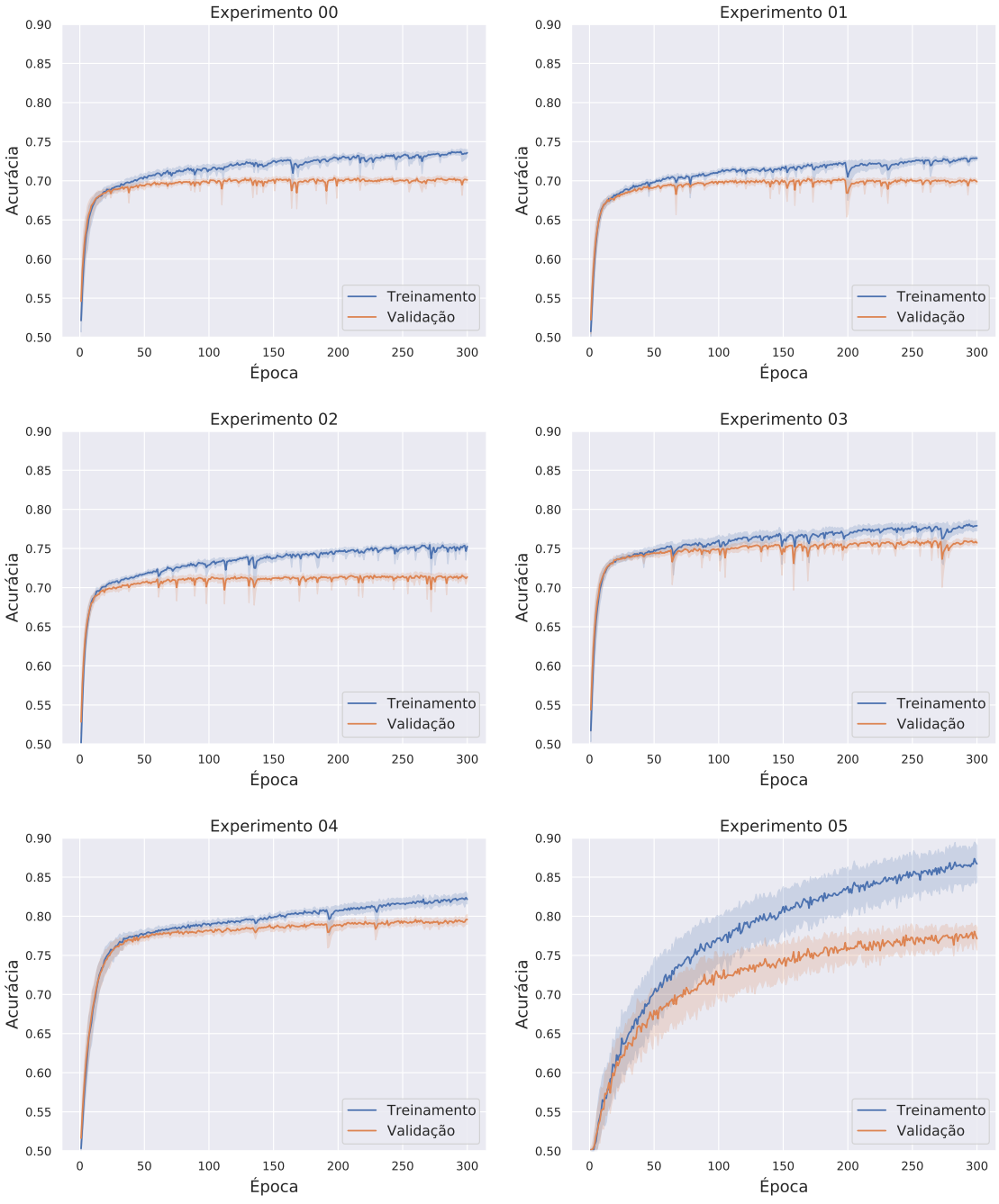
Os demais experimentos do Grupo 1 visaram explorar a influência da utilização dos diversos filtros aplicados sobre o *score* que define a confiabilidade da interação, conforme disponibilizado pela base de dados RNAInter (KANG et al., 2022). Os experimentos 02 a 05 aplicam a filtragem de interações baseada nos limiares de *score* 0.2, 0.3, 0.4 e 0.5. Nesses experimentos, estamos interessados em entender como a quantidade de interações descritas como fraca (*weak*) ou forte (*strong*) poderia influenciar de forma positiva ou negativa no aprendizado do modelo. Utilizamos os dois tipos de interações coletadas, isto é, miRNA–mRNA e mRNA–mRNA. Lembramos que conforme relatado no Capítulo 4, Seção 4.3, cada critério de filtragem gera uma consequente redução no número de interações do grafo, principalmente de arestas com tipo de evidência *Weak*.

Analisando os resultados de acurácia para os Experimentos 02 a 05 (Figura 5.1), notamos uma sequência de melhoras no valor do desempenho para os dados de treinamento e teste ao final das 300 épocas de treinamento, com o filtro inicial (*score* >= 0.2,

no Experimento 02) já apresentando melhoras em relação ao Experimento 00. Entretanto, também observamos que em alguns casos houve um efeito bastante negativo no *overfitting* do modelo, como no Experimento 02 e, principalmente, no Experimento 05.

Dentre os cenários experimentais avaliados no Grupo 1, destacamos os resultados do Experimento 04, o qual apresentou-se como o mais promissor deste grupo. Neste

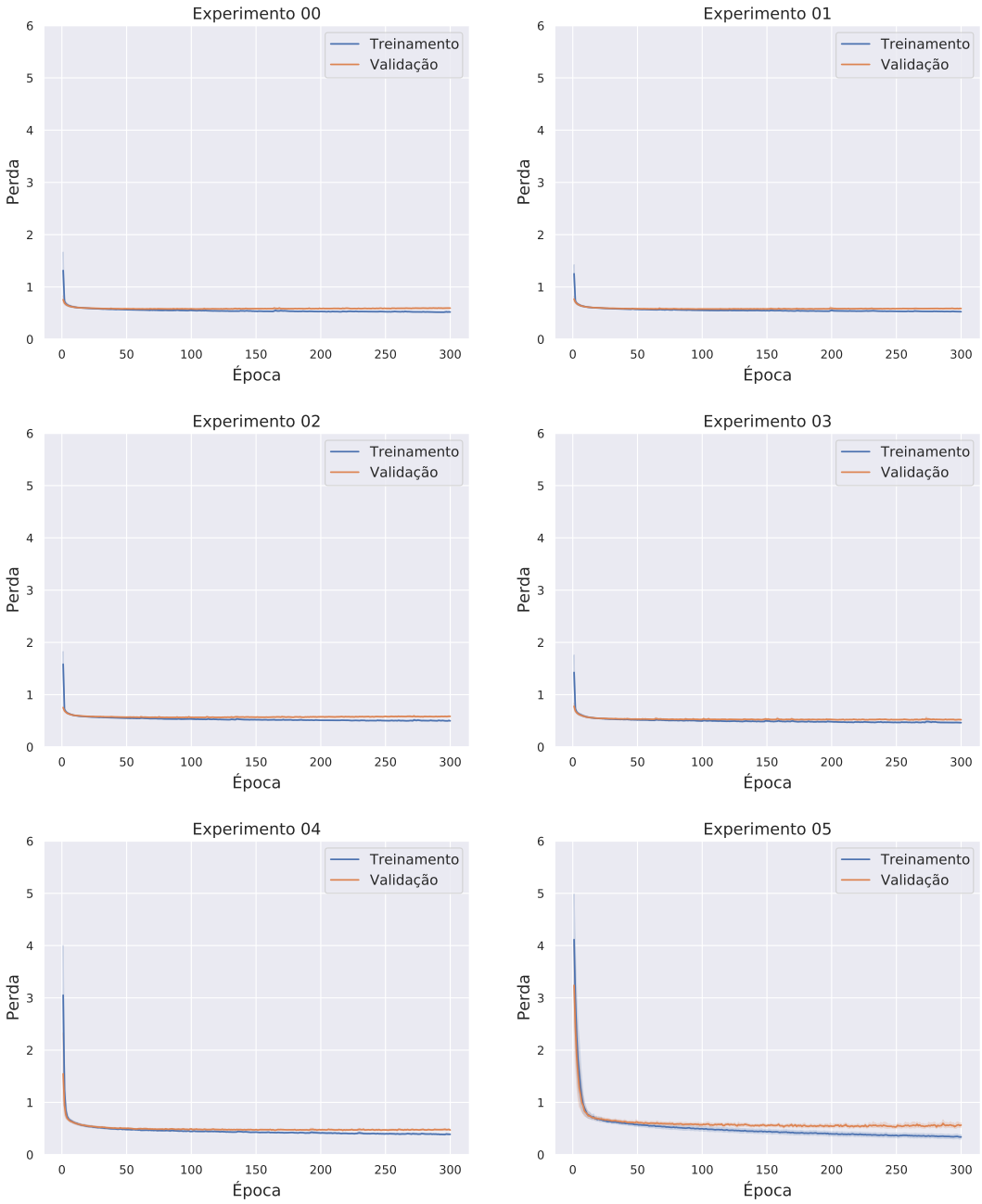
Figura 5.1 – Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 1. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

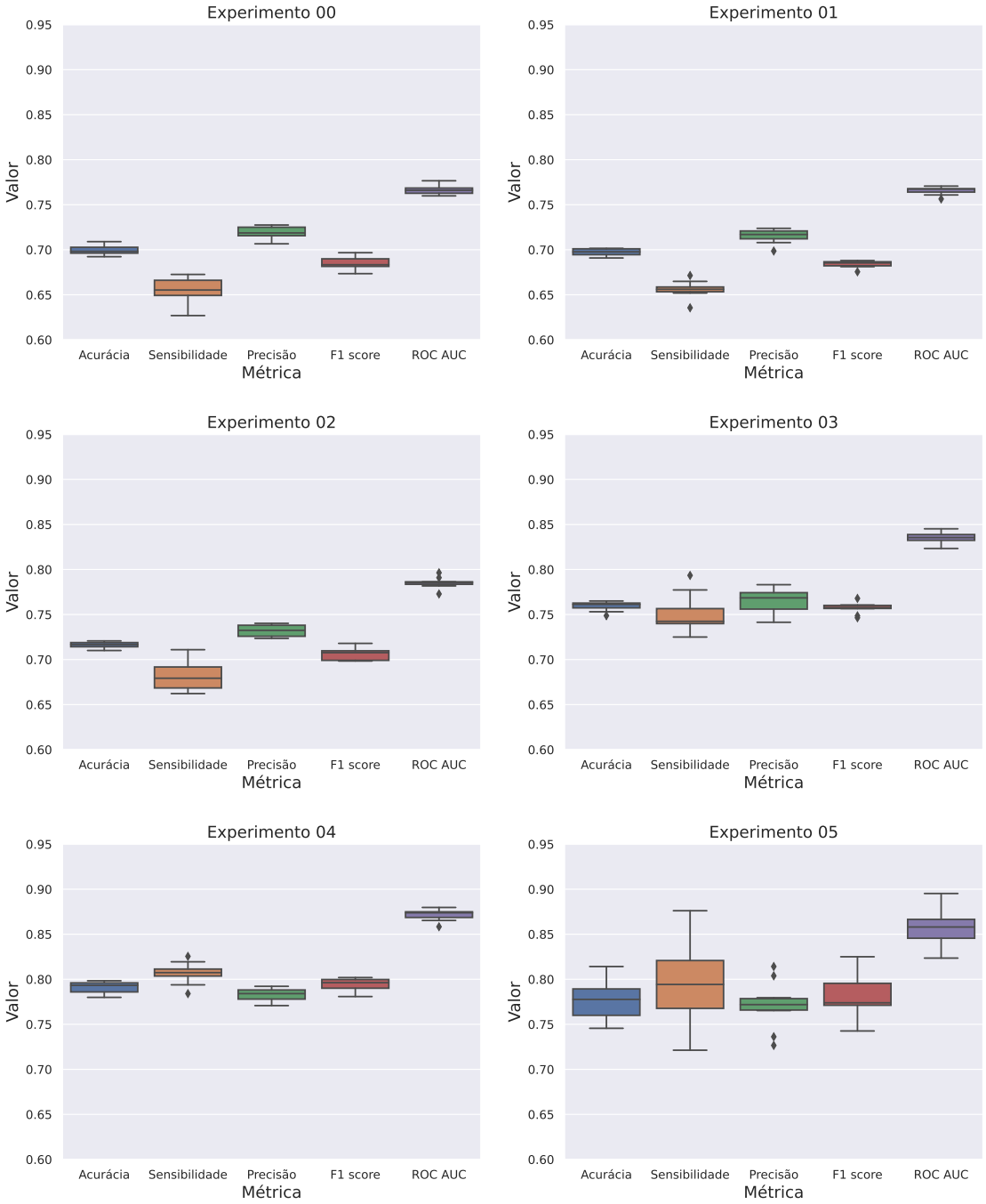
experimento, treinamos um modelo utilizando as interações miRNA–mRNA e mRNA–mRNA, aplicando um filtro para manter apenas interações avaliadas com *score* maior ou igual a 0.4. Assim, nosso conjunto de dados foi reduzido para um total de mais de cento e trinta mil linhas. Os resultados obtidos neste cenário demonstram um aparente ganho, sem uma grande tendência a ocorrência de *overfitting*, situação presente nos demais casos.

Figura 5.2 – Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 1. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

Figura 5.3 – Análise de desempenho nos dados de teste para os experimentos do Grupo 1.



Fonte: O Autor.

Por fim, avaliamos e comparamos as métricas obtidas para o conjunto de teste ao longo das 10 execuções, entre todos os cenários experimentais do Grupo 1. Conforme pode ser visto na Figura 5.3, de uma forma geral, as métricas de desempenho apresentam pouca variação nos resultados entre todas as execuções realizadas. Algumas exceções para esta observação são as métricas relacionadas ao Experimento 05. O Experimento 04 foi o que se destacou neste grupo, obtendo mediana para o F1-score e para a sensibili-

dade próximo a 80%, precisão acima de 75% e ROC AUC superando 85% em todas as execuções.

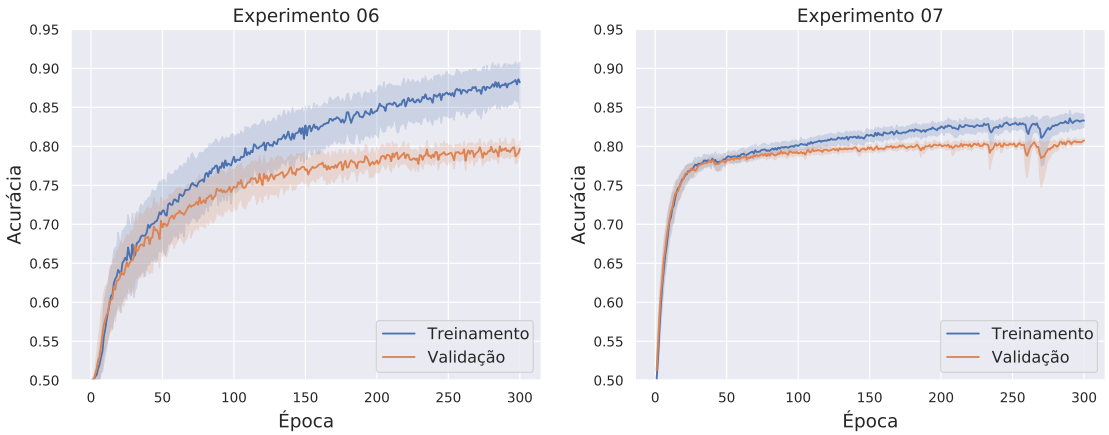
Como mencionado anteriormente, em cada grupo de experimento realizado, elegemos o melhor cenário para seguir com a investigação do modelo proposto, realizando alterações incrementais na configuração utilizada para treinamento do algoritmo. Julgamos o Experimento 04 o mais interessante do Grupo 1, no entanto, também julgamos interessante entender o que pode estar influenciando o comportamento tão discrepante do Experimento 05 proposto. No Experimento 05, o conjunto de dados das interações está filtrado para o score maior ou igual a 0.5, contendo interações miRNA–mRNA e mRNA–mRNA. De antemão, conseguimos concluir que a realização do filtro sobre o *score* realmente influencia os resultados alcançados e que a diminuição da quantidade de interações descritas como fortes (*Strong*) prejudica a predição das interações, ocasionando uma grande dispersão sobre os resultados alcançados. O segundo grupo de experimentos, descrito a seguir, focou em investigar essa questão.

5.2.1.2 G2: Impacto da remoção de interações mRNA–mRNA em grafos filtrados

Os cenários experimentais do Grupo 2 tinham como objetivo novamente identificar se a utilização das interações mRNA–mRNA poderiam estar sendo influenciados de forma positiva ou negativa em relação aos resultados obtidos. Entretanto, nesta etapa a investigação foi centrada nos dados filtrados utilizados para os Experimentos 04 e 05, descritos na seção anterior. Assim, os Experimentos 06 e 07 são, respectivamente, os Experimentos 04 e 05 somente com o emprego das interações miRNA–mRNA para construção do grafo utilizado no aprendizado.

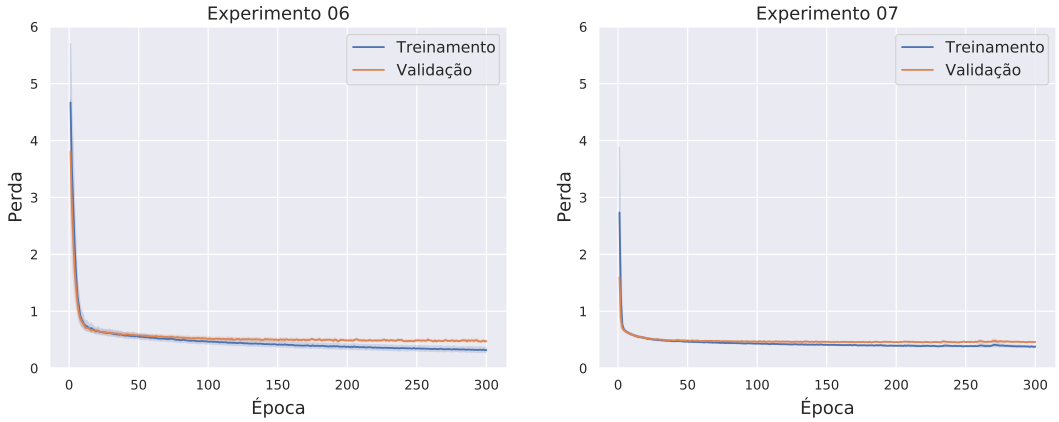
Os resultados obtidos ao longo do treinamento, e após a predição nos dados de teste, sugerem que a utilização das interações mRNA–mRNA de fato influencia sobre os resultados alcançados. Entretanto, como podemos observar nos valores de acurácia (Figura 5.4) e função de perda (Figura 5.5), o impacto não foi padrão entre ambos os experimentos. Comparando os resultados dos experimentos 06 e 04 (Seção 5.2.1.1), notamos que a remoção de interações mRNA–mRNA introduz um *overfitting* no experimento com interações filtradas usando o limiar de *score* de 0.4. Por outro lado, comparando os resultados dos experimentos 07 e 05 (Seção 5.2.1.1), notamos que o inverso ocorreu, e que o treinamento passou a ser mais estável e com menor sinal de *overfitting* após a remoção destas interações.

Figura 5.4 – Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 2. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

Figura 5.5 – Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 2. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.

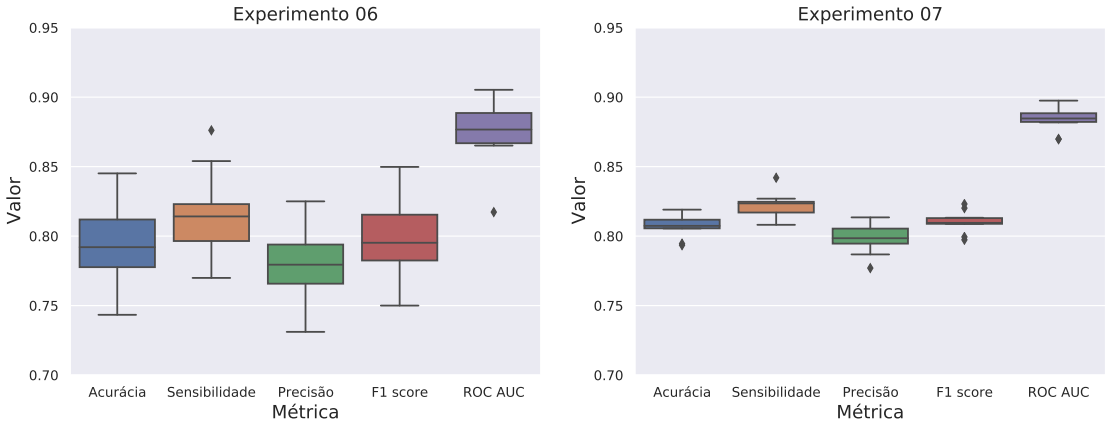


Fonte: O Autor.

Adicionalmente, observando métricas obtidas sobre as predições realizadas para o conjunto de teste (Figura 5.6), é possível observar que o Experimento 06 apresenta uma dispersão maior quando comparado aos resultados obtidos quando o mesmo continha interações mRNA–mRNA. Apesar da melhora observada no Experimento 07, os resultados não são superiores aos obtidos no Experimento 04. Deste modo, decidimos seguir com o cenário do Experimento 04, ou seja, experimento anterior do Grupo 1 contendo interações miRNA–mRNA, assim como interações mRNA–mRNA, e o filtragem de interações com base em um *score* maior ou igual a 0.4.

Por fim, analisando a evolução da acurácia e da função de perda, é possível notar

Figura 5.6 – Análise de desempenho nos dados de teste para os experimentos do Grupo 2.



Fonte: O Autor.

que depois de um certo número de épocas continuar treinando o modelo não o torna mais eficiente. Isto ocorre tanto para o Experimento 06, como também para o Experimento 07. Este último possui um valor de acurácia de validação praticamente estagnado perto de 80% após 50 épocas. Uma observação semelhante pode ser feita para o Experimento 04, no qual o *overfitting* começou a ocorrer a partir de 100 épocas de treinamento. Assim, em nosso próximo grupo de experimentos, optamos em avaliar a diminuição do número de épocas de treinamento, com o objetivo de reduzir *overfitting* e diminuir as chances do modelo predizer interações falso positivas.

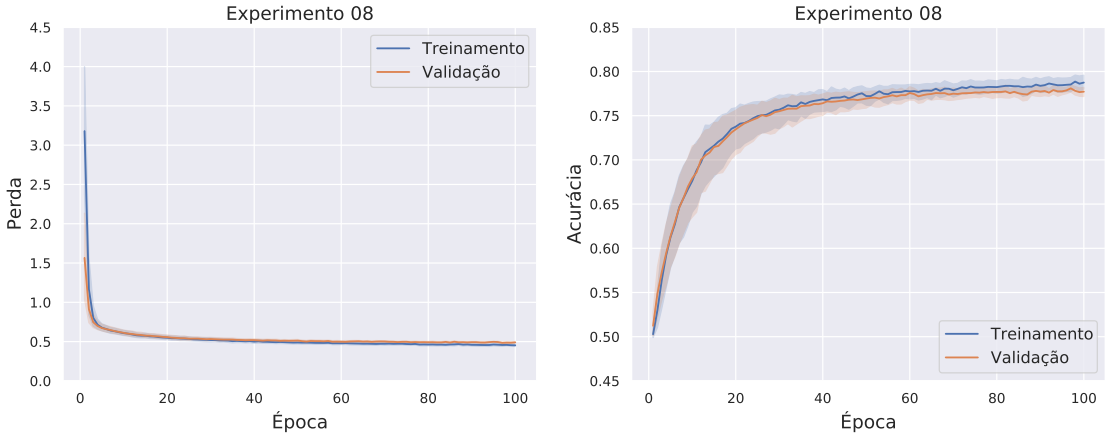
5.2.1.3 G3: Impacto da redução do número de épocas de treinamento

Este grupo de experimentos é composto pelo Cenário 08, o qual tem o número de épocas de treinamento configurado como 100, valor aproximado onde a curva tende ao comportamento indesejado. A análise dos resultados deste cenário pode ser comparada ao Experimento 04, pois as demais configurações se mantém as mesmas entre os dois experimentos. A Figura 5.7 apresenta os resultados obtidos para acurácia e função de perda ao longo do treinamento. Nos gráficos apresentados, é possível observar que as curvas para um treinamento com 100 épocas apresentam um comportamento mais apropriado, com valores mais próximos entre treinamento e validação. Desta forma, a confiança no desempenho do modelo gerado torna-se mais forte.

Adicionalmente, observamos as métricas de desempenho coletadas a partir do conjunto de teste, as quais são bastante satisfatórias, ainda que tenha havido uma pequena redução nos valores em relação ao desempenho nos dados de teste para o modelo gerado no Experimento 04. O resultado para F1-score aproximou-se de 80%, precisão

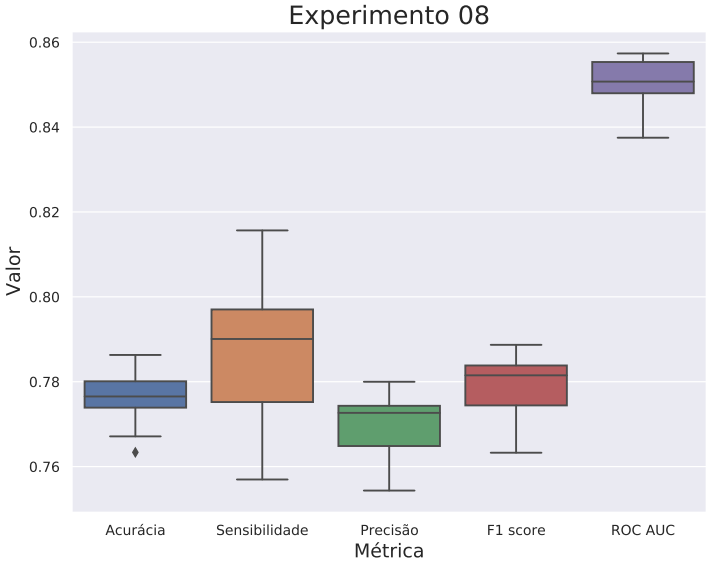
variando na grande maioria dos resultados entre valores acima de 75% e a sensibilidade com resultados bastante próximos de 80%, sendo que em alguns casos ultrapassando essa porcentagem. A ROC AUC, por sua vez, teve uma mediana próxima de 85%. Esse pequeno ajuste sobre as épocas deixa o treinamento do modelo mais adequado por evitar *overfitting*. Portanto, após esta avaliação, decidimos adotar o Experimento 08 como o mais apropriado para seguir nos próximos grupos experimentais.

Figura 5.7 – Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 3. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

Figura 5.8 – Análise de desempenho nos dados de teste para os experimentos do Grupo 3.

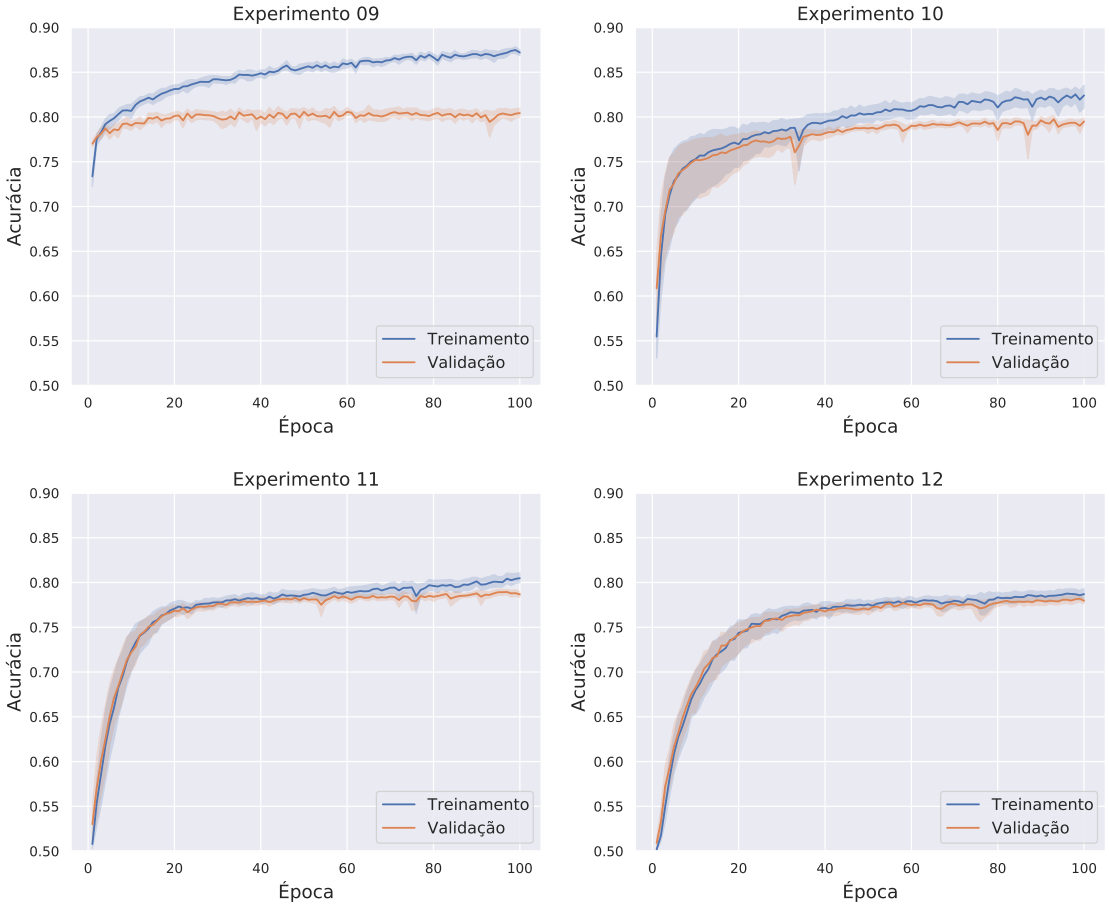


Fonte: O Autor.

5.2.1.4 G4: Impacto da variação no tamanho de batch usado no treinamento

O Grupo 4 tem como finalidade realizar experimentos para iniciar a investigação acerca dos valores definidos para os hiperparâmetros presentes no algoritmo HinSAGE pela biblioteca StellarGraph. Embora os resultados até o momento tenham sido relativamente satisfatórios, estes hiperparâmetros foram definidos sem considerar as particularidades dos dados utilizados e do problema de predição abordado no presente trabalho. Assim, este grupo de experimentos foi desenvolvido para explorar a definição do valor que determina o número de exemplos de treinamento usados em cada interação (hiperparâmetro conhecido como *batch size*), o qual originalmente é configurado como 200. Foram testadas quatro valores distintos: 1 (Experimento 09), 50 (Experimento 10), 100 (Experimento 11) e 300 (Experimento 12).

Figura 5.9 – Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 4. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



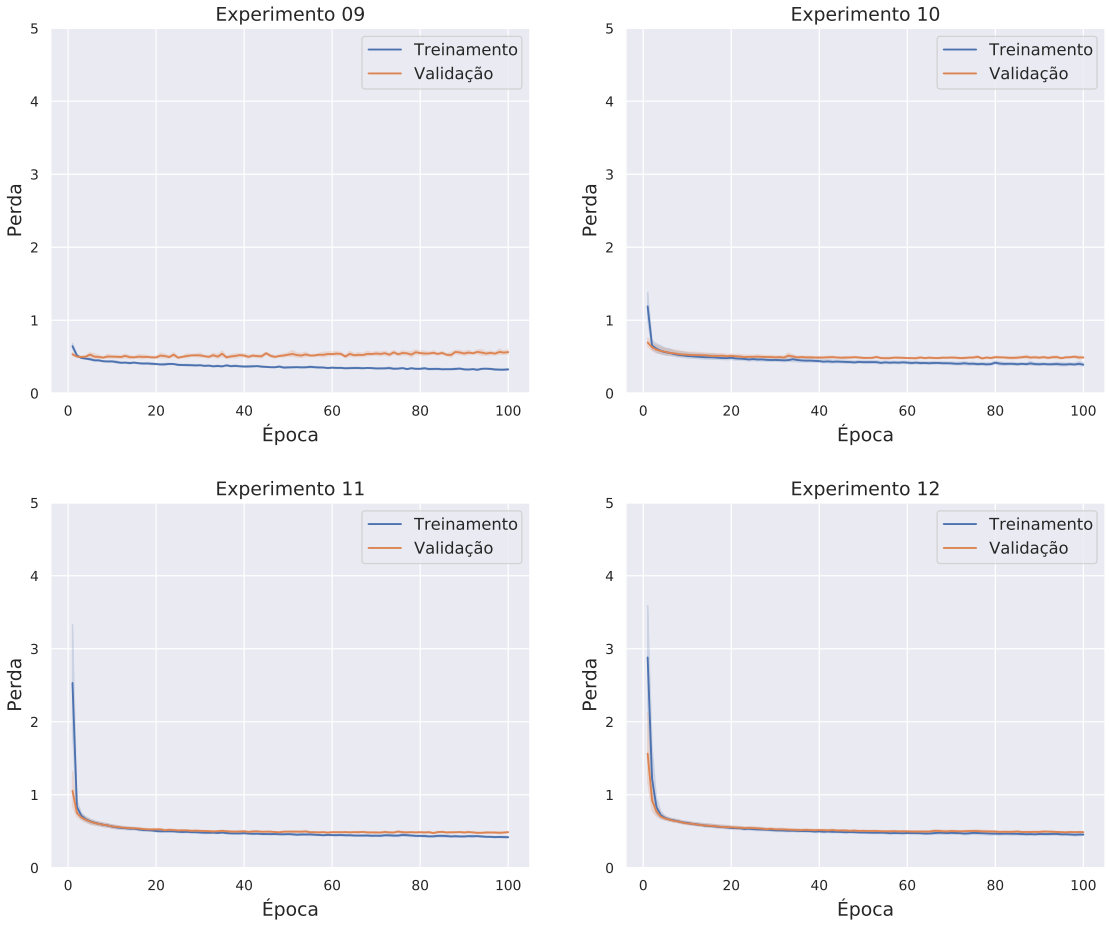
Fonte: O Autor.

No cenário do Experimento 09, com *batch size* igual à 1, produzimos um comportamento de treinamento descrito como estocástico. Analisando os resultados de acurácia (Figura 5.9) e função de perda (Figura 5.10) deste experimento, é possível observar a clássica característica de *overfitting*, quando o erro de validação aumenta enquanto o erro de treinamento acaba caindo. Neste cenário, isto ocorre de forma bem precoce, nas dez primeira épocas de treinamento. Essa característica também parece estar presente sobre o Experimento 10, *batch size* igual à 50.

Os dois últimos cenários analisados, Experimento 11 e Experimento 12, obtiveram resultados mais adequados. Em especial o Experimento 12, utilizando *batch size* igual à 300, o qual obteve uma leve melhora quando comparado ao Experimento 08 definido anteriormente como o mais satisfatório até o momento, principalmente em termos de variação de desempenho ao longo das 10 execuções.

A Figura 5.11 sumariza as métricas de desempenho para os quatro experimentos

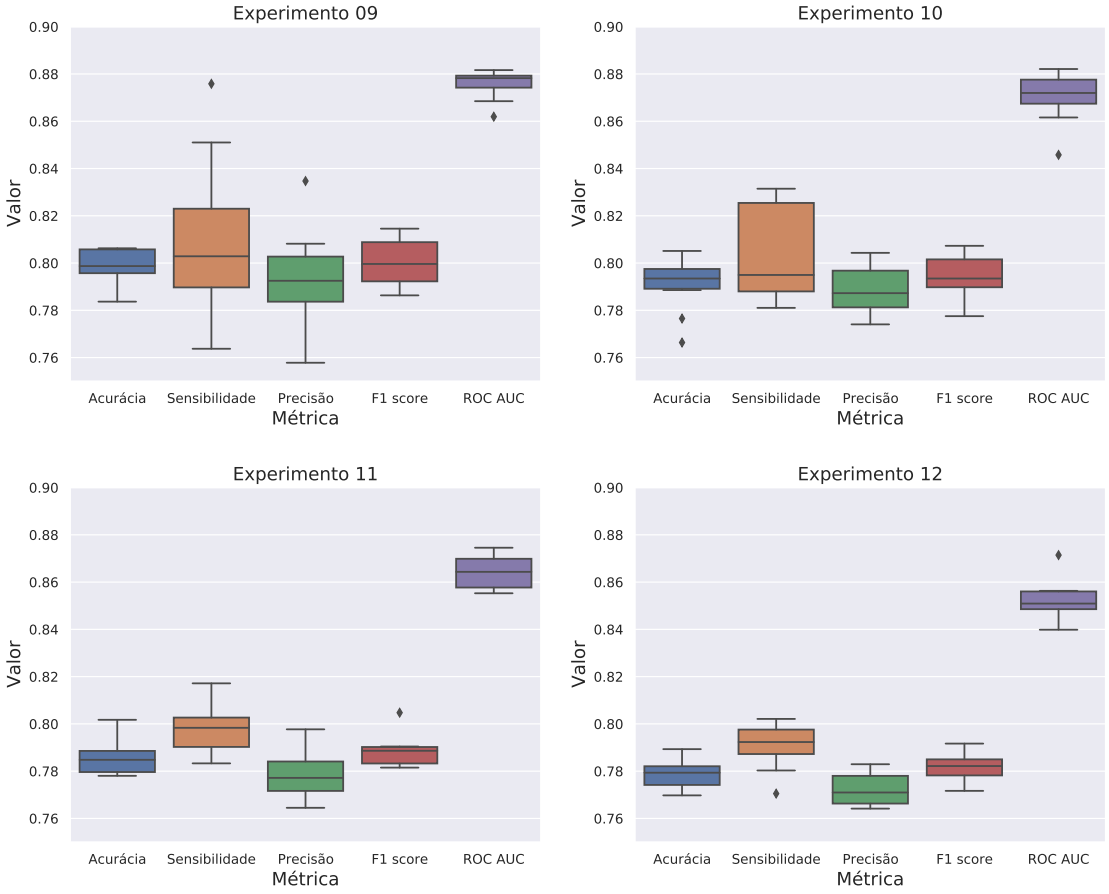
Figura 5.10 – Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 4. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

deste grupo. Pode-se perceber que os Experimentos 09 e 10 possuem dispersões maiores para as métricas analisadas do que os Experimentos 11 e 12. Comparando os resultados do Experimento 12 com o Experimento 08, observamos um desempenho mais consistente neste último experimento realizado, com todas as métricas possuindo uma variação menor ao longo de múltiplas execuções com diferentes *seeds* aleatórias. Adicionalmente, percebemos um sutil aumento na mediana para as métricas de acurácia, precisão e ROC AUC. Assim, optamos por escolher o Experimento 12 como base para o desenvolvimentos dos próximos cenários experimentais.

Figura 5.11 – Análise de desempenho nos dados de teste para os experimentos do Grupo 4.



Fonte: O Autor.

5.2.1.5 G5: Impacto da variação no número de nós vizinhos amostrados

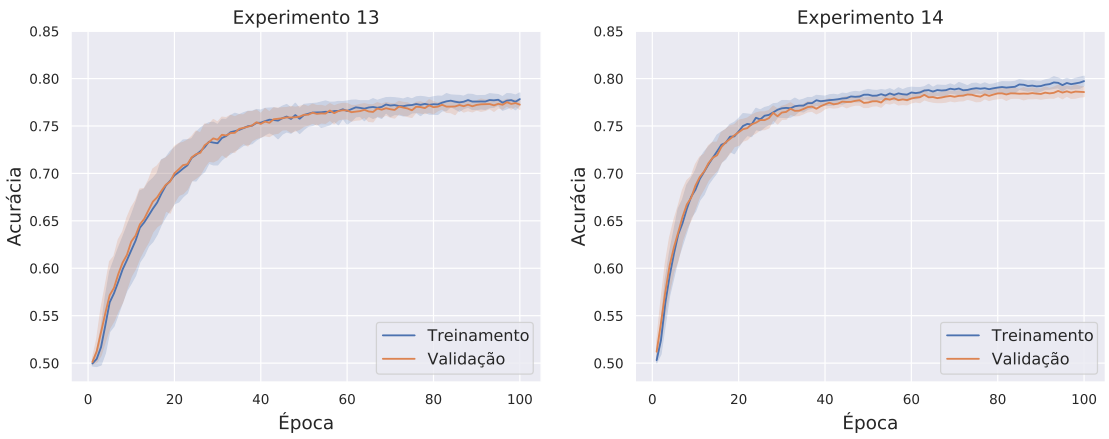
O Grupo 5 possui um conjunto de cenários experimentais voltados a investigar o impacto do hiperparâmetro que define o número de nós vizinhos amostrados pelo algoritmo HinSAGE na geração de *embeddings*. Este hiperparâmetro (*i.e.*, num_sample), recebe como um entrada um vetor de inteiros e determina tanto o valor de K (número de

saltos, ou *hops*, a partir de cada nó) através do comprimento do vetor, como o número de vizinhos a serem amostrados em cada salto através dos valores informados no vetor. Mantivemos o número de saltos fixo em $K = 2$, variando apenas o tamanho da amostragem a ser feita a partir da vizinhança de cada nó. Foram testadas duas variações: o Experimento 13 usa a configuração $[6, 3]$ e o Experimento 14 usa a configuração $[12, 6]$ para o hiperparâmetro num_sample. De forma geral, trabalhos relacionados discutem que quando o número de nós observados na geração do *embedding* é muito grande, os *embeddings* criados podem vir a se tornar menos informativos. Entretanto, como estamos trabalhando com grafos de grande dimensão, é interessante também avaliar uma variação que considere um maior número de nós vizinhos amostrados em cada salto a partir de um determinado nó.

Analisando os resultados de acurácia e função de perda, mostrados nas Figuras 5.12 e 5.13, respectivamente, conseguimos observar que diminuir a quantidade de vizinhos amostrados (Experimento 13) resultou em um aumento da dispersão do desempenho de treinamento e validação do modelo, enquanto aumentar a quantidade de vizinhos amostrados (Experimento 14) resultou em uma diminuição da dispersão destes desempenhos. Adicionalmente, percebemos que em ambos os experimentos, as curvas referentes a treinamento e validação ficaram bastante próximas entre si, indicando que não houve *overfitting* no treinamento do modelo.

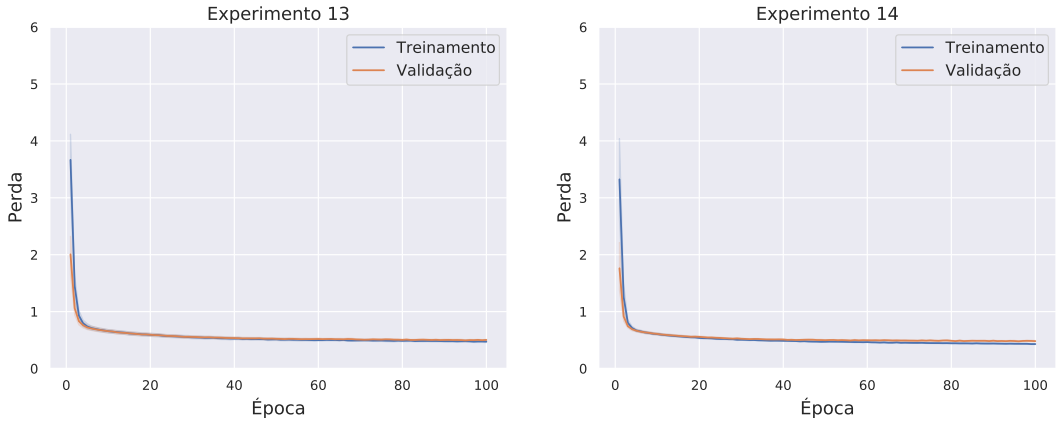
Os valores de desempenho para os dados de teste, ao longo das 10 execuções aleatórias, são apresentados na Figura 5.14. Os diagramas de caixa para os Experimentos 13 e 14 apresentam o mesmo tipo de comportamento descrito para os dados de treinamento

Figura 5.12 – Avaliação da acurácia ao longo do treinamento do modelo para os experimentos do Grupo 5. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



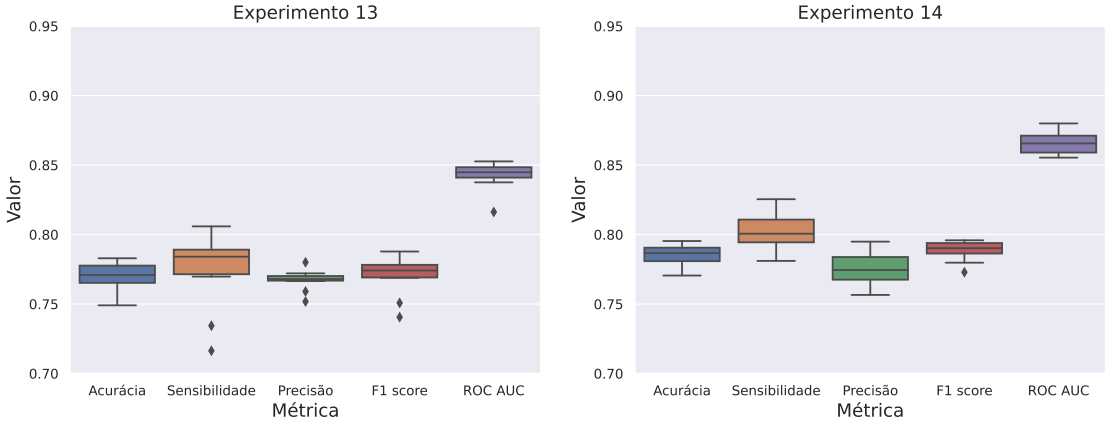
Fonte: O Autor.

Figura 5.13 – Avaliação da perda ao longo do treinamento do modelo para os experimentos do Grupo 5. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

Figura 5.14 – Análise de desempenho nos dados de teste para os experimentos do Grupo 5.



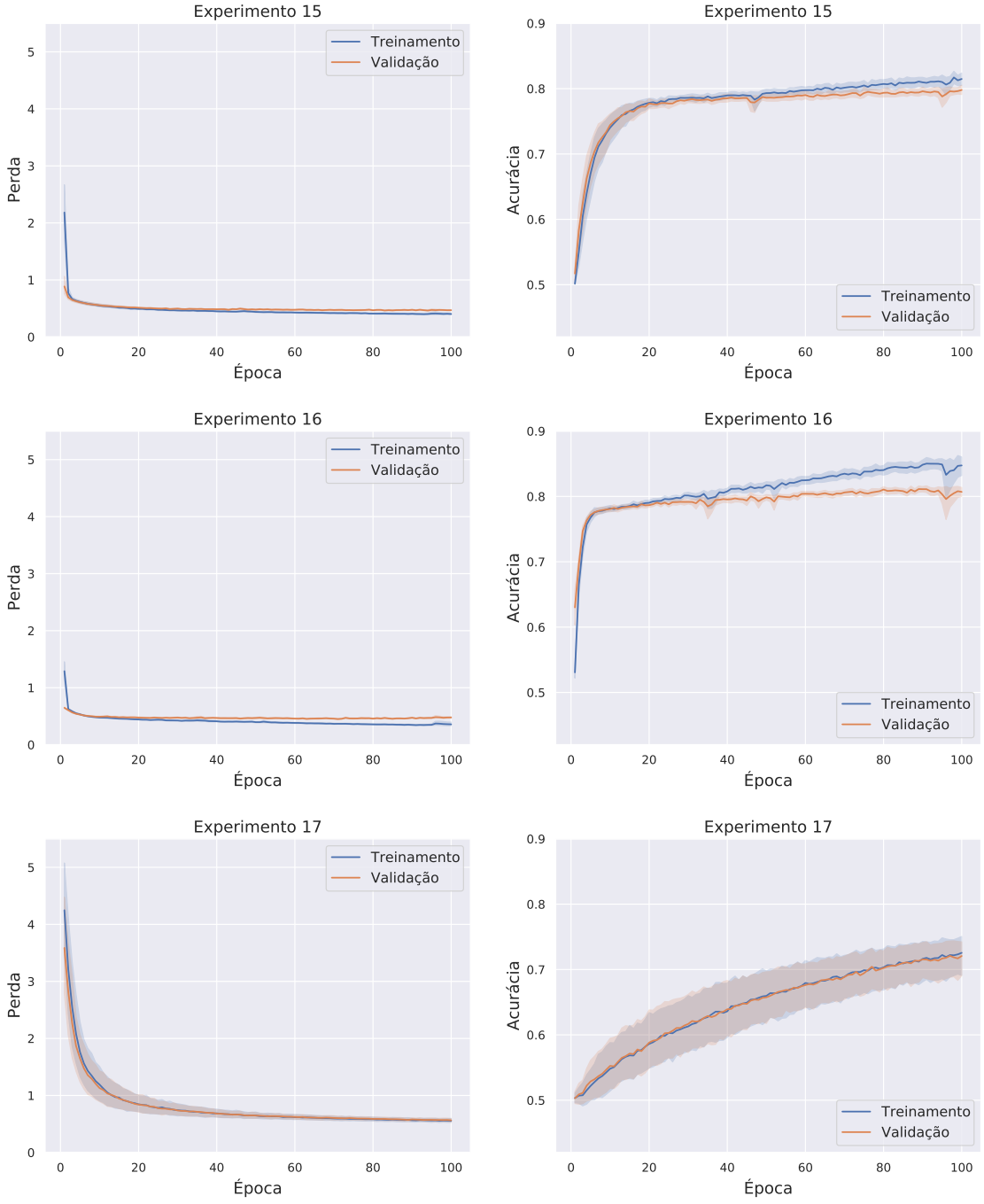
Fonte: O Autor.

e validação, com uma maior dispersão observada para o Experimento 13, no qual foi reduzido o número de nós vizinhos amostrados pelo algoritmo. Já o Experimento 14, além de baixa variância nas métricas de desempenho, também apresentou melhoras em relação ao Experimento 12. As medianas para todas as métricas foram de aproximadamente 80% ou superior. Se observou também um desempenho mais equilibrado entre sensibilidade e precisão, culminando em um aumento na métrica F1-Score. Já a métrica ROC AUC ficou com valores bem próximos a 90%. Estes achados em termos de melhores desempenhos nos dados de treinamento, validação e teste, além do desempenho consistente entre treinamento e validação demonstrado pelas curvas de acurácia e de função de perda, motivaram a escolha do Experimento 14 como o melhor cenário até o momento, servindo como base para os próximos experimentos.

5.2.1.6 G6: Impacto da variação do tamanho das camadas ocultas

Além das alterações sobre hiperparâmetros avaliadas anteriormente, uma experimentação descrita como bastante relevante na grande maioria dos trabalhos no domínio de aprendizado profundo está sobre a quantidade e número de dimensões das camadas ocul-

Figura 5.15 – Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 6. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.

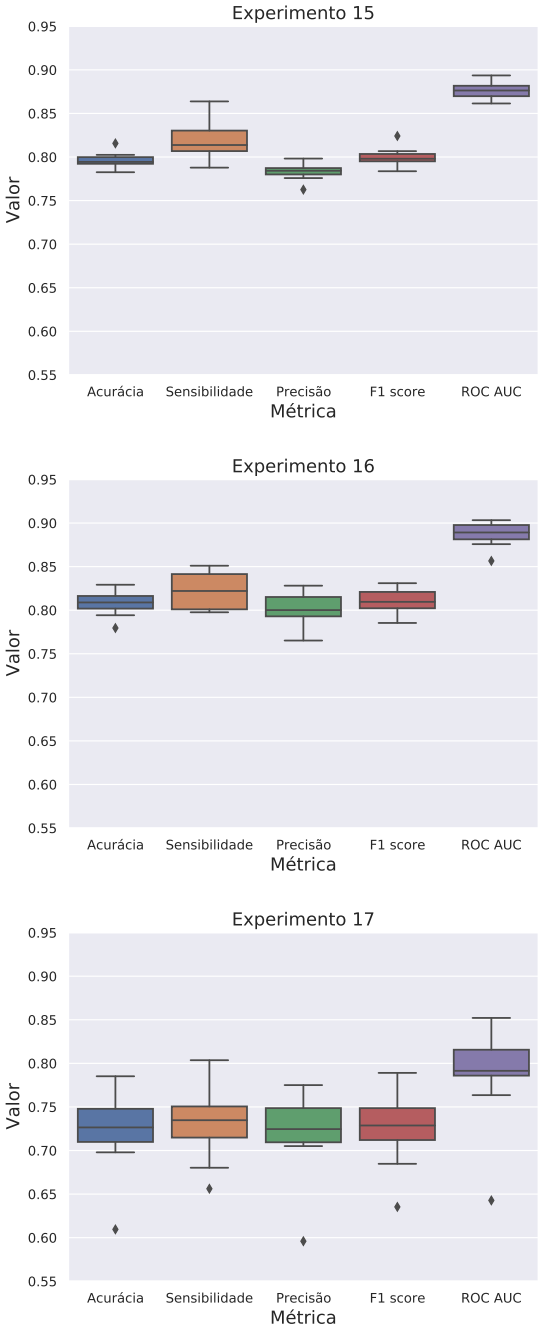


Fonte: O Autor.

tas. Entretanto, para GNNs, no geral 2 ou 3 camadas ocultas são usualmente suficientes para aprender padrões em grafos de acordo com trabalhos anteriores (KIPF; WELLING, 2016). Neste sentido, o Grupo 6 de experimentos teve como objetivo explorar diferentes configurações para o hiperparâmetro $hinsage\_layer\_size$, mantendo sempre 2 camadas ocultas, mas variando suas dimensões.

Os Experimentos 15 e 16 aumentaram as dimensões das camadas ocultas para $[64, 64]$ e $[128, 128]$, respectivamente, enquanto o Experimento 17 diminuiu as dimensões

Figura 5.16 – Análise de desempenho nos dados de teste para os experimentos do Grupo 6.



Fonte: O Autor.

para $[16, 16]$ – sendo o valor original usado de $[32, 32]$. A evolução da acurácia e da função de perda para os três experimentos é mostrada na Figura 5.15. Podemos notar que os experimentos não foram muito satisfatórios. O aumento das dimensões das camadas ocultas aumentou o *overfitting*, principalmente no Experimento 16, e não teve efeito positivo no desempenho máximo obtido pelo modelo. Já o Experimento 17 demonstrou que a redução do tamanho das camadas ocultas fez o modelo perder poder preditivo em relação aos demais cenários avaliados, especialmente o Experimento 14.

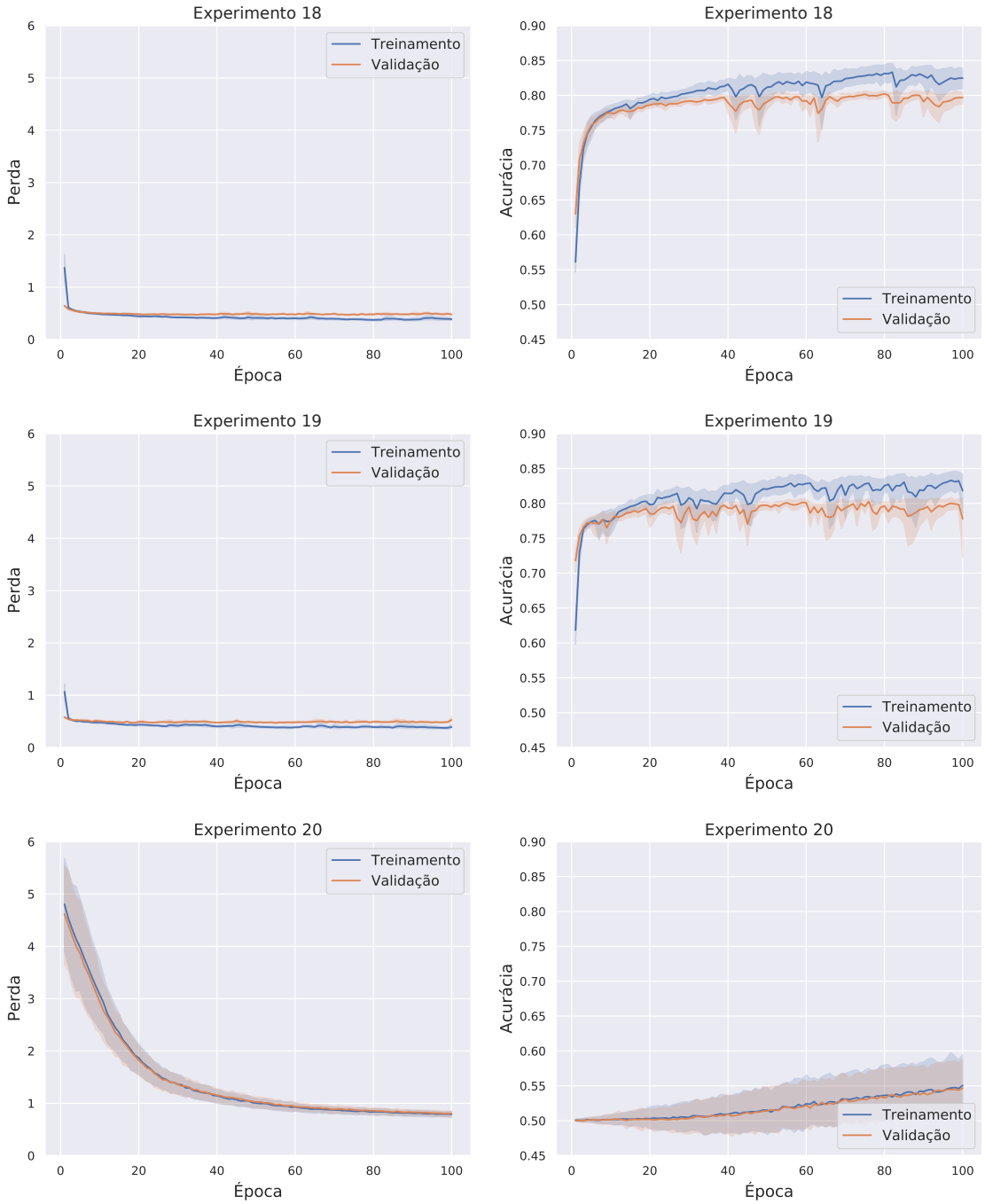
A Figura 5.16 resume as métricas de desempenho no conjunto de teste para estes experimentos e evidencia que não houveram ganhos em relação ao cenário do Experimento 14. Desta forma, decidimos por manter o Experimento 14 como nosso melhor resultado.

5.2.1.7 G7: Impacto da variação na taxa de aprendizado

Um outro experimento relevante de realizar sobre os hiperparâmetros envolvidos no treinamento do modelo com o algoritmo HinSAGE é a variação da taxa de aprendizado. Todos os nossos experimentos foram realizados com o otimizador Adam, bastante conhecido e empregado na literatura. Nos Experimentos 00 a 17, utilizamos a taxa de aprendizado padrão usada na biblioteca StellarGraph, de 0.001. Assim, no Grupo 7, decidimos concentrar os experimentos que visam avaliar o impacto da variação no valor da taxa de aprendizado, empregando um valor mais alto (0.01), um valor intermediário (0.005) e um valor mais baixo (0.0001) em relação ao valor padrão inicial.

Na Figura 5.17 são apresentados os resultados alcançados em termos de acurácia e função de perda para os dados de treinamento e validação. Para os aumentos da taxa de aprendizado, utilizando 0.01 e 0.005, observamos mais sinais de *overfitting* do modelo sem melhora efetiva do desempenho preditivo alcançado. Já no experimento 20, o inverso ocorreu. A diminuição da taxa de aprendizado teve um forte impacto negativo no desempenho do modelo em treinamento e validação, causando inclusive um *underfitting* do modelo. Percebe-se que a evolução do aprendizado foi muito lenta, não convergindo adequadamente dentro das 100 épocas de treinamento.

Figura 5.17 – Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 7. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.

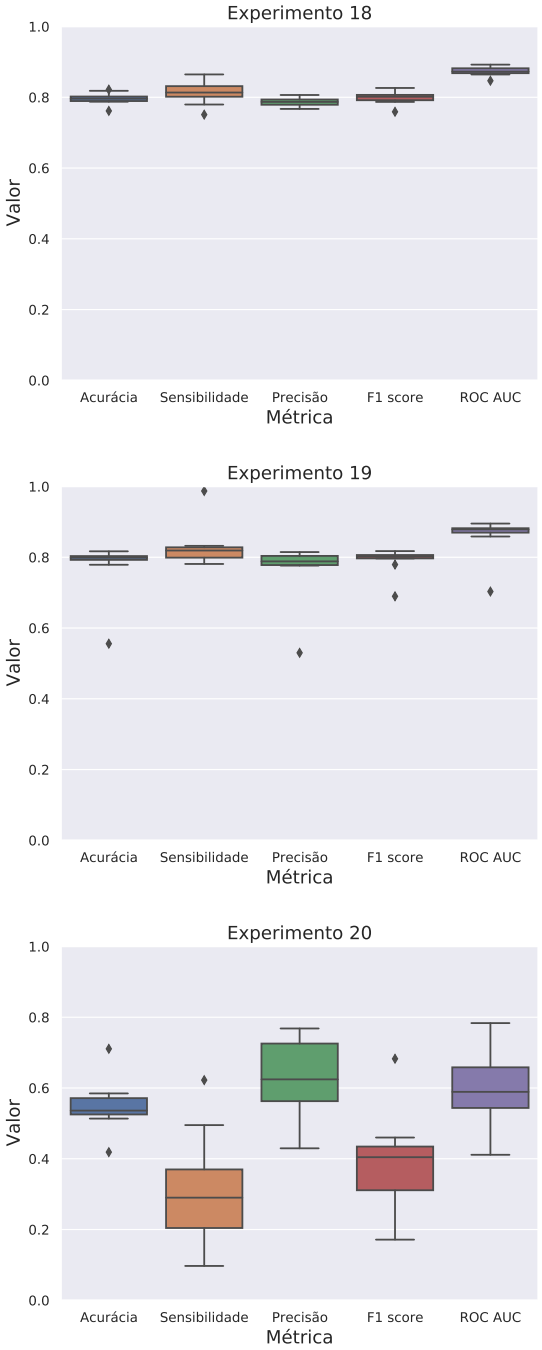


Fonte: O Autor.

Os resultados para o conjunto de teste são mostrados na Figura 5.18. A análise destes gráficos corrobora o fato de que a diminuição da taxa de aprendizado (Experimento 20) teve um efeito prejudicial no desempenho do modelo preditivo. Nos demais cenários, embora as distribuições dos valores das métricas tendam a variar em valores próximos aos obtidos com o Experimento 14, notamos a ocorrência de mais outliers, especialmente

no Experimento 19. Desta forma, avaliamos que o valor da taxa de aprendizado usado inicialmente, de 0.001, pareceu adequado para o nosso problema de predição. Os experimentos realizados neste grupo não forneceram evidências fortes o suficiente para decidir pela mudança na taxa de aprendizado empregada nesta metodologia. Assim, o cenário de destaque continuou sendo o Experimento 14.

Figura 5.18 – Análise de desempenho nos dados de teste para os experimentos do Grupo 7.



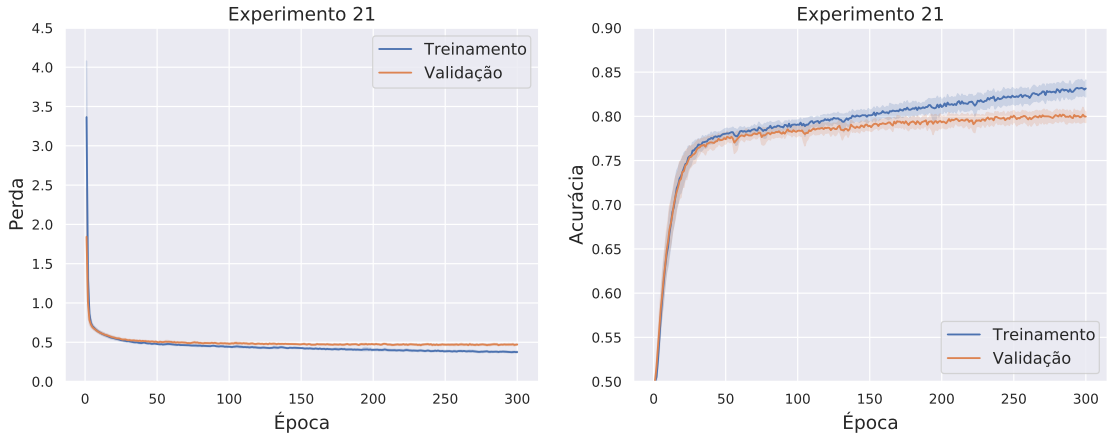
Fonte: O Autor.

5.2.1.8 G8: Impacto do aumento no número de épocas de treinamento com base no melhor modelo

A última etapa da análise experimental do algoritmo HinSAGE visou avaliar se um maior número de épocas poderia trazer benefícios para o treinamento do modelo seguindo as definições do melhor cenário experimental definido até o momento, o Experimento 14. Julgamos importante esta análise, visto que as variações em outros hiperparâmetros feitas ao longo da execução dos grupos experimentais anteriores poderiam interferir na convergência do algoritmo, tornando 100 épocas insuficientes para alcançar um bom desempenho preditivo. Assim, nos experimentos do Grupo 8, visamos identificar se as alterações realizadas sobre os hiperparâmetros descritos anteriormente poderiam ter influências positiva após 100 épocas.

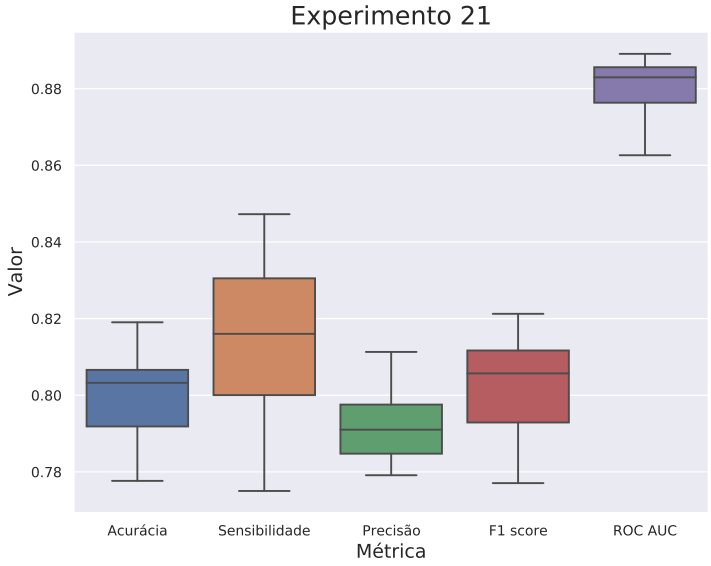
O que observamos na avaliação de desempenho para treinamento e validação (Figura 5.19) foi um modelo que começa a apresentar características de *overfitting*, semelhante ao observado nos primeiros experimentos realizados. Em relação ao desempenho nos dados de teste, mostrados na Figura 5.20, notamos deterioração em termos de maior variância e de medianas mais baixas para todas as métricas analisadas. Assim, optamos por manter o número de épocas de treinamento igual à 100, seguindo a configuração do Experimento 14.

Figura 5.19 – Avaliação da perda e acurácia ao longo do treinamento do modelo para os experimentos do Grupo 8. A linha sólida representa a média, e o sombreado denota o desvio padrão ao longo de 10 execuções.



Fonte: O Autor.

Figura 5.20 – Análise de desempenho nos dados de teste para os experimentos do Grupo 8.



Fonte: O Autor.

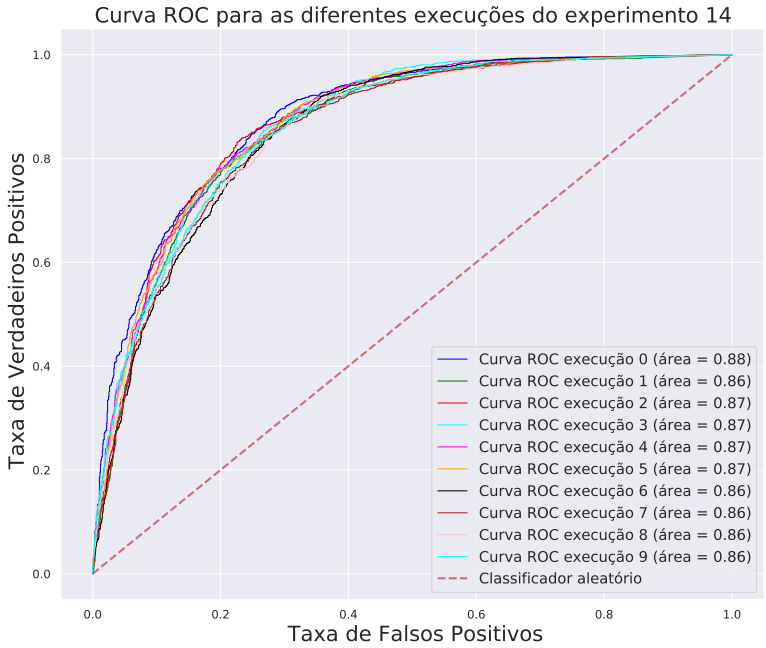
5.2.1.9 Sumário do desempenho do melhor modelo HinSAGE

Entre todos os cenários avaliados no conjunto **C1** de experimentos, baseado em variações em conjuntos de dados e hiperparâmetros descritas na Tabela 5.1, definimos como melhor resultado o modelo originado no Experimento 14. Este modelo é treinado com as interações miRNA–mRNA e mRNA–mRNA filtradas para um *score* maior ou igual a 0.4, utilizando um tamanho de *batch* de 300 e durante 100 épocas. O algoritmo HinSAGE foi aplicado utilizando duas camadas ocultas, cada qual com tamanho igual a 32, e amostragem de nós vizinhos em dois saltos (*i.e., hops*), sendo amostrados 12 nós vizinhos no primeiro salto e 6 nós vizinhos no segundo salto para a geração dos *embeddings*. Por fim, a taxa de aprendizado do otimizador Adam foi configurada como 0.001, valor padrão e que se mostrou melhor opção para o nosso domínio.

A Figura 5.21 mostra as curvas ROC, e os respectivos valores de ROC AUC, para os dados de teste considerando as 10 execuções do Holdout de 3 vias. Podemos observar que o desempenho do modelo parece bastante estável, com ROC AUC variando entre 86% e 88%. Além disso, a curva demonstra um crescimento adequado, tendo um aumento mais acentuado em TVP (eixo y) quando comparado a TFP (eixo x). Por exemplo, para uma TVP igual a 80%, a maioria dos modelos retorna uma TFP próximo de 20%.

Uma análise mais detalhada das métricas de desempenho para os dados de teste é mostrada na Tabela 5.2. São apresentadas as métricas para cada execução, bem como a média e desvio padrão ao longo das 10 execuções aleatórias.

Figura 5.21 – Análise das curvas ROC para as 10 execuções do cenário do Experimento 14.



Fonte: O Autor.

Considerando o problema sobre a alta incidência de falsos positivos, comumente indicado na literatura como um desafio na realização da predição de interações miRNA–mRNA, estamos interessados em analisar principalmente as métricas de sensibilidade, que descreve a proporção de exemplos positivos que conseguimos classificar corretamente, e de precisão, que nos indica a proporção dos exemplos classificados como positivos

Tabela 5.2 – Tabela dos resultados obtidos para todas as execuções do cenário de experimento 14. Nos resultados apresentados destaca-se a execução cinco que julgamos com um bom resultados sobre todos as métricas apresentadas.

Nº	Sensibilidade	Precisão	F1-score	ROC AUC	Acurácia
1	0.8036	0.7847	0.7945	0.8799	0.7915
2	0.8164	0.7716	0.7934	0.8616	0.7874
3	0.7960	0.7948	0.7954	0.8680	0.7953
4	0.8096	0.7729	0.7908	0.8712	0.7859
5	0.8111	0.7811	0.7958	0.8721	0.7919
6	0.7817	0.7907	0.7862	0.8709	0.7874
7	0.8254	0.7565	0.7894	0.8570	0.7799
8	0.7938	0.7661	0.7797	0.8553	0.7757
9	0.7810	0.7649	0.7728	0.8581	0.7705
10	0.7975	0.7759	0.7866	0.8630	0.7836
Média	0.8016	0.7759	0.7884	0.8657	0.7849
Desvio Padrão	0.0136	0.0114	0.0070	0.0075	0.0073

pelo modelo que são realmente positivos de acordo com o rótulo real. Além disso, a métrica F1-score é interessante pois sumariza sensibilidade e precisão através de uma média harmônica. Destacamos na Tabela 5.2 a quinta execução do Experimento 14, o qual apresentou os melhores resultados alcançados, com valor mais alto de F1-score. Esse modelo será posteriormente empregado na comparação com outras abordagens de predição, realizada no conjunto de experimentos **C2**.

5.2.2 Comparação do modelo baseado no HinSAGE com outras abordagens

Nas próximas seções são apresentados os resultados experimentais e uma breve discussão da comparação do nosso modelo proposto, baseado no algoritmo HinSAGE, com trabalhos consolidados da área de previsão de interações miRNA-alvo. Adicionalmente, abordamos uma proposta de comparação para um modelo AM clássico empregando nosso conjunto de dados.

5.2.2.1 Comparação com diferentes trabalhos relacionados

Em razão dos desafios discutidos na Seção 5.1.2, os trabalhos relacionados para os quais foi possível uma comparação direta entre as predições para os dados de teste foram: miRAW (PLA; ZHONG; RAYNER, 2018), TargetScan (LEWIS et al., 2004; AGARWAL et al., 2015) e PITA (KERTESZ et al., 2007). Também utilizamos interações suportadas por predição computacional depositadas em duas bases de dados bem conhecidas, TarBase v8 (KARAGKOUNI et al., 2018) e mirDIP (TOKáR et al., 2017). Ressaltamos que estas interações preditas computacionalmente não se encontram na nossa base de treinamento, visto que analisamos somente as interações suportadas por evidência experimental disponibilizadas no RNAInter.

Para realizar a comparação, realizamos uma intersecção entre os dados de teste utilizados no nosso trabalho (especificamente na quinta execução aleatória do Experimento 14) e resultados de predições de alvos pré-computadas e disponibilizadas por cada ferramenta ou método, par a par. Todo o conjunto de dados de predição foram obtidos nas plataformas oficiais disponibilizadas pelo trabalho relacionado, com acessos realizados em 10 Janeiro de 2023. Entretanto, em alguns casos foram precisos pré-processamentos específicos, conforme detalhado a seguir:

- **miRAW**: O trabalho miRAW disponibiliza o seu código e o conjunto de dados

utilizados com os resultados da predição do modelo desenvolvido[1]. Entre todos os arquivos disponibilizados, escolhemos o arquivo *targetPredictionOutput* que se encontra no caminho $Results\backslash Evaluate\_14K\_FULL\backslash miRAW\_NF$. Este arquivo contém o identificador único do miRNA, identificador único do mRNA, a predição dada pelo modelo, a classe real, assim como outras informações não relevantes para o nosso contexto. Sendo assim, não foi necessário realizar nenhum pré-processamento sobre os dados para possibilitar a comparação com o nosso modelo.

- **TargetScan**: as predições são descritas a partir de um *context++ score* definido pela ferramenta. Alvos com *context++ score* mais baixos são os mais representativos. Entretanto, não há um ponto de corte claramente definido para determinar se uma dada interação miRNA–mRNA deve ser classificada como positiva ou negativa. Assim, seguindo trabalhos anteriores (JACOBSEN et al., 2013), decidimos aplicar três limiares distintos: -0.5, -0.3 e -0.2. Interações com valores de *context++ score* menor que o limiar aplicado foram classificadas como positivas.
- **PITA**: a partir do catálogo de alvos preditos pelo PITA, fizemos o download do arquivo PITA_targets_hg18_3_15_ALL.tab obtido no catalogo $3\backslash 15flankall$[2]. Como o arquivo contém interações preditas como existentes pelo modelo, ainda que com variados níveis de confiança, assumimos que todas as interações contidas no arquivo poderiam ser classificadas como interações positivas durante a comparação.
- **TarBase**: foi feito o download do arquivo TarBase_v8_Human_miR-gene.csv a partir do site da base de dados[3]. Foram utilizadas apenas interações relacionadas a humanos (organismo *H. sapiens*) e classificadas como '*positive*' pela ferramenta, indicando que se trata de uma interação funcional entre miRNA–mRNA.
- **mirDIP**: esta ferramenta não disponibiliza as predições pré-computadas para download, mas fornece uma plataforma web através da qual podem ser realizadas consultas por miRNAs ou por mRNAs específicos. Assim, submetemos o conjunto de miRNAs únicos existentes em nosso modelo para consulta de todas as interações preditas que envolvem estes miRNAs. Cada interação possui um *score* descrito como: *Very High*, *High*, *Medium* e *Low*. Onde *Very High* seria uma alta probabi-

[1]Disponível em: <https://bitbucket.org/bipous/workspace/projects/MIRAW>. Acessado em 2 de Dezembro de 2022

[2]Disponível em: <https://genie.weizmann.ac.il/pubs/mir07/mir07_data.html>. Acessado em 2 de Dezembro de 2022

[3]Disponível em: <https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/download.php?ver=8.0>. Acessado em 2 de Dezembro de 2022

lidade da existência da interação miRNA–mRNA, probabilidade esta que decresce até a definição *Low*. Devido ao grande volume de dados disponibilizado pela base, decidimos trabalhar somente com o conjunto de dados com o valor de *score* descrito como *Very High*, inferindo todas as interações resultantes como positivas.

Os resultados alcançados para as diferentes comparações sobre os trabalhos relacionados podem ser observadas na Tabela 5.3. A tabela apresenta os dados específicos utilizados e cada comparação, em termos de número de miRNAs e número de interações miRNA–mRNA, e os resultados sobre as métricas de desempenho analisadas: sensibilidade, precisão, F1-score e acurácia.

Como mencionado anteriormente, nossa principal métrica avaliativa está sobre F1-score, a qual é definida a partir de sensibilidade e precisão. Analisando os valores de F1-score e suas componentes, conseguimos avaliar o modelo desenvolvido sobre os aspectos relacionados à identificação de verdadeiros positivos e falsos positivos – critérios que julgamos importantes para mitigar o principal desafio a cerca dos desbalanceamento de exemplos positivos e negativos e da consequente tendência de predição de muitos falsos positivos.

Para todos os trabalhos relacionados com os quais foi possível traçar uma comparação direta para as instâncias de teste, o modelo desenvolvido na nossa proposta baseada em GNNs e, mais especificamente, no algoritmo HinSAGE, possui resultados muito próximos ou que se destacam sobre as demais abordagens. Desconsiderando os trabalhos TarBase e mirDIP, que são bases de dados e não métodos de predição baseados em AM, nosso trabalho apresenta resultados bastante animadores. Em quase todas as comparações realizadas, nosso modelo obteve F1-score acima de 90%, indicando que o modelo desenvolvido neste trabalho quando comparado com trabalhos relacionados consegue predizer com um grande grau de confiança satisfatório interações positivas, como também interações negativas.

Salientamos, no entanto, que nenhum dos trabalhos utilizados foca na identificação de interações miRNA–mRNA associadas a câncer. Esta é uma limitação presente na comparação de resultados com a literatura, devido à dificuldade de encontrar conjuntos de dados representativos deste contexto.

Tabela 5.3 – Comparação de desempenho entre o modelo baseado em HinSAGE proposto neste trabalho e outras abordagens da literatura.

Trabalho	Método	Quantidade de miRNAs únicos	Quantidade de interações	Métricas obtidas sobre o mesmo conjunto de dados			
				Sensibilidade	Precisão	F1-score	Acurácia
Comparação do nosso melhor modelo com o trabalho miRAW							
Nosso modelo	GNN	11	12	0.9166	1.0000	0.9565	0.9166
miRAW	Deep Learning	11	12	0.9166	1.0000	0.9565	0.9166
Comparação do nosso melhor modelo com o trabalho TargetScan							
Nosso modelo	GNN	154	533	0.9640	0.9902	0.9769	0.9549
TargetScan (*context++ score* < -0.2)	Correspondência de Semente, Conservação, Energia Livre, Acessibilidade do local,	154	533	0.5037	0.9888	0.6675	0.5028
TargetScan (*context++ score* < -0.3)	Abundância do Local-Alvo, Emparelhamento Compensatório de 3',	154	533	0.2518	0.9851	0.4012	0.2551
TargetScan (*context++ score* < -0.5)	Pares G:U Permitidos na Semente, Conteúdo AU Local.	154	533	0.0246	1.0000	0.0480	0.0337
Comparação do nosso melhor modelo com o trabalho TarBase v.8							
Nosso modelo	GNN	153	423	0.9607	0.9751	0.9679	0.9385
TarBase v.8	Banco de dados	153	423	0.9877	0.9664	0.9769	0.9550
Comparação do nosso melhor modelo com o trabalho PITA							
Nosso modelo	GNN	139	250	0.7175	0.8037	0.7582	0.6760
PITA	Correspondência de semente, Energia Livre, Acessibilidade do Local, Abundância do Local-Alvo, Pares G:U Permitidos na Semente	139	250	1.0000	0.7080	0.8290	0.7080
Comparação do nosso melhor modelo com o trabalho mirDIP							
Nosso modelo	GNN	222	314	0.8175	0.9837	0.8929	0.8152
mirDIP	Banco de dados	222	314	1.0000	0.9426	0.9704	0.9426

5.2.2.2 Comparação com uma rede neural artificial

Para melhor entender a verdadeira contribuição em utilizar uma rede neural de grafo na predição de alvos de miRNAs em relação a outras abordagens de AM, julgamos relevante empregar o conjunto de dados construído neste trabalho para treinamento de um modelo com algoritmo de AM tradicional. O algoritmo escolhido para essa comparação foi o modelo *Multilayer Perceptron* (MLP), devido à similaridade de viés indutivo com as GNNs. Entretanto, algumas particularidades tiveram que ser consideradas. Em nossa metodologia, o modelo GNN aborda uma tarefa de classificação, a fim de predizer se interações miRNA–mRNA são positivas (funcionais) ou negativas (espúrias) utilizando apenas informações da conectividade do grafo e dos atributos relacionados a cada nó. Conforme descrito no Capítulo 4, estes atributos se referem ao padrão de expressão diferencial de cada nó (miRNA ou mRNA) em 15 tipos diferentes de câncer.

Em algoritmos de AM tradicional, um passo necessário é a definição manual de atributos para definir cada instância de treinamento, validação ou teste. Tentando tornar a comparação o mais centrada possível na variação do algoritmo de aprendizado utilizado, visamos remodelar os dados utilizados para serem aplicados ao treinamento de um algoritmo de AM tradicional. Nos dados coletados, os únicos atributos originalmente disponíveis e aptos a serem utilizados em um processo de aprendizado tradicional, baseado em dados estruturados, são os valores de expressão gênica individual. Entretanto, utilizar diretamente estes vetores, concatenando a expressão diferencial do miRNA com a expressão diferencial do mRNA para formação de uma instância, poderia gerar vieses de predição tendo em vista que duas instâncias relacionadas ao mesmo miRNA, mas com dois alvos distintos, teriam metade do vetor de atributos com valores idênticos. Assumindo que uma insteração esteja rotulada originalmente como positiva e a outra como negativa, a tarefa de modelar o padrão através de um modelo de classificação seria bastante difícil.

Desta forma, optamos por utilizar uma informação disponível no conjunto de dados original, dada pelo *score* de confiança associado com cada interação descrita no conjunto de dados RNAInter (KANG et al., 2022). Utilizamos o *score* como o alvo de predição, e abordamos o problema como uma tarefa de regressão. O objetivo é treinar um modelo para prever o *score* de cada interação a partir de uma instância dada pelos padrões de expressão diferencial do miRNA e de um mRNA alvo candidato. Como o nosso intuito não é explorar diferentes arquiteturas para a rede neural, utilizamos uma MLP tradicional, com 1 camada oculta contendo 100 neurônios. A função de ativação utilizada foi a sigmoide, limitando assim os valores de predição para o intervalo [0,1]. No nosso contexto,

valores mais altos significam maior confiança ou probabilidade em uma dada interação miRNA–mRNA. A implementação da MLP foi feita com a biblioteca Scikit-learn (PEDREGOSA et al., 2011).

Nesta comparação, buscamos seguir ao máximo as mesmas definições empregadas em nosso treinamento baseado em grafo. Assim, o conjunto de dados utilizado foi o mesmo que alcançou os melhores resultados em nosso Experimento 14 descrito anteriormente, contendo apenas as interações com *score* maior ou igual a 0.4. Entretanto, apenas interações miRNA–mRNA deste conjunto de dados foram utilizadas nesta etapa. Para treinar a MLP, fizemos uma divisão aleatória das interações entre treinamento e teste, reservando 40% das instâncias para teste. Visto que não foi feita a otimização de hiperparâmetros do algoritmo, não houve necessidade de reservar instâncias para o conjunto de validação. Na Tabela 5.4, é possível visualizar a comparação do tamanho dos conjuntos de dados para treinamento, validação e teste entre o modelo baseado no algoritmo HinSAGE e o modelo treinado com uma MLP. O treinamento foi realizado com 10 execuções aleatórias, utilizando o mesmo valor de semente aleatória.

Tabela 5.4 – Resumo da divisão dos dados em conjuntos de treinamento, validação e teste para a comparação com o algoritmo MLP.

Modelo	Tamanho Conjunto de Treinamento	Tamanho Conjunto de Validação	Tamanho Conjunto de Teste	Tamanho Total do Conjunto
HinSAGE	2606	2632	2658	132992
MLP	79795	-	53197	132992

A fim de entender a qualidade do modelo MLP para esta tarefa de predição, analisamos as predições retornadas pelo modelo de diferentes formas. Iniciamos tentando uma visualização dos valores preditos em comparação com os valores esperados através de um gráfico de dispersão (mostrado no Apêndice B, Figura B.1). Entretanto, através da análise do gráfico gerado, notou-se uma dificuldade de observar uma tendência de concordância entre o *score* previsto e o *score* real, o que pode = ser em razão do grande volume de dados no gráfico. Entretanto, analisando visualmente as diferentes predições para o conjunto de teste, em princípio as duas variáveis parecem não possuir uma relação clara. Portanto, para investigar melhor os resultados alcançados para cada cenário, foi calculado o coeficiente de Pearson entre os valores de saída preditos e os valores esperados. Esta análise da correlação visa investigar se o modelo possui uma tendência em atribuir valores preditos mais altos para as instâncias que possuem valores mais altos de saída (*i.e., score*) esperada. Os resultados são apresentados na segunda coluna da Tabela 5.5. Podemos

notar correlação positiva fraca para as 10 execuções, nunca ultrapassando um valor de correlação de 0.33.

Tabela 5.5 – Resultados do coeficiente de correlação de Pearson e do erro quadrático médio para as predições do modelo MLP nos dados de teste.

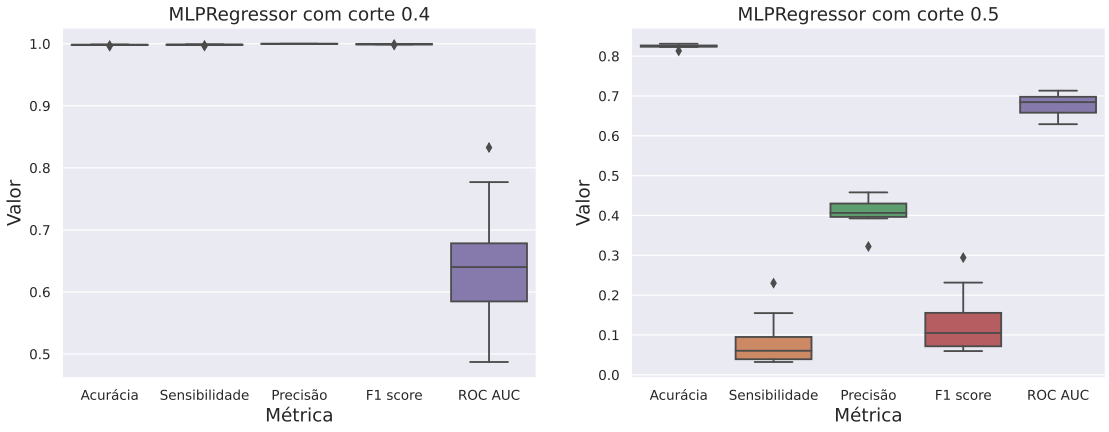
Nº da execução	Coeficiente de Pearson	Erro Quadrático Médio
1	0.1878	0.002655
2	0.2097	0.002647
3	0.2756	0.002524
4	0.2654	0.002571
5	0.2754	0.002505
6	0.3237	0.002378
7	0.1615	0.002696
8	0.3193	0.002391
9	0.3037	0.002496
10	0.2759	0.002588

A Tabela 5.5 também mostra o valor do erro quadrático médio, cuja definição foi fornecida no Capítulo 2. Os valores para esta métrica de avaliação para modelos de regressão foram no geral baixos, atingindo um valor máximo de 0.002696.

A fim de realizar uma comparação direta entre o modelo MLP e o modelo HinSAGE, adotamos pontos de corte no valor de *score* predito pelo modelo a fim de estratificar as interações de teste entre interações preditas como positivas e interações preditas como negativas. Optamos em utilizar dois limiares de *score*, 0.4 e 0.5. A partir da aplicação destes limiares, foi possível calcular as métricas utilizadas originalmente na avaliação do modelo HinSAGE.

A Figura 5.22 apresenta os resultados alcançados pelo modelo MLP considerando os limiares de *score* de 0.4 e 0.5, respectivamente. É possível observar que ao utilizar um corte de 0.4, obtemos uma precisão bastante elevada, indicando que o modelo foi capaz de identificar com um alto grau de assertividade as interações descritas como positivas. Entretanto, esse resultado com um alto grau de assertividade apresenta a característica de *overfitting*, deixando o modelo treinado pouco generalista para dados não conhecidos. Considerando o ponto de corte de 0.5 para definição dos casos como positivo, tanto a precisão como também a sensibilidade acabam por diminuir drasticamente.

Figura 5.22 – Métricas alcançadas para a execução de 10 experimentos sobre o modelo de aprendizado de máquina MLP.



Fonte: O Autor.

6 CONSIDERAÇÕES FINAIS

Neste trabalho, desenvolvemos uma nova abordagem computacional para prever interações miRNA–mRNA alvo associadas a diversos grupos de cânceres em humanos, utilizando uma rede neural de grafos. Nosso objetivo principal era investigar a possibilidade de realizar predições a partir de grafos heterogêneos contendo interações miRNA–mRNA e mRNA–mRNA, usando métodos de aprendizado capazes de analisar diretamente a estrutura do grafo, ao invés de atributos extraídos manualmente a partir de um conjunto de interações conhecidas. Desta forma, nossa hipótese é que vieses humanos acerca da definição de atributos relevantes para a predição de alvos funcionais de miRNAs poderiam ser evitados.

Para construir nosso modelo, coletamos e agregamos dados de diferentes bases de dados responsáveis por catalogar interações entre RNAs (miRNA–mRNA e mRNA–mRNA, especificamente) e padrões de expressão gênica em diferentes tipos de câncer. Estes dados foram integrados através da construção de um grafo, onde os nós representam miRNAs e mRNAs e as arestas representam as interações entre estas moléculas conhecidas na literatura. Cada nó do grafo possui um vetor de atributos associado, definido como o padrão de expressão diferencial do miRNA ou mRNA nos 15 tipos de câncer avaliados neste trabalho.

Adotamos o algoritmo HinSAGE como método de aprendizado na nossa proposta, o qual é uma adaptação do algoritmo GraphSAGE (HAMILTON; YING; LESKOVEC, 2017) para grafos heterogêneos disponibilizado pela biblioteca StellarGraph. A utilização de algoritmos de aprendizado profundo baseado em grafos, além de introduzir a vantagem de explorar de forma muito mais abrangente e robusta a estrutura do grafo em busca de padrões de interações miRNAs–mRNAs, também possibilita lidar de forma eficiente com o problema do desbalanceamento de dados encontrado neste domínio. Os trabalhos relacionados reportam de forma consistente a dificuldade em lidar com o pouco número de exemplos negativos de interações miRNA–alvo registrados em bases de dados, causando um grande viés de predição da classe majoritária (exemplos positivos) nos modelos baseados em AM. Assim, essa escolha nos permitiu minimizar este problema, tendo em vista que o algoritmo HinSAGE, assim como GraphSAGE, realiza a indução de interações negativas durante o processo de treinamento do modelo.

Embora não haja um trabalho equivalente conhecido até o momento que possa servir como *baseline* de comparação para o nosso trabalho, as comparações realizadas

com interações preditas computacionalmente por outros métodos baseados em AM ou depositadas em bases de dados, mostraram que a abordagem baseada em grafos heterogêneos e aplicação do algoritmo HinSAGE é bastante promissora. O desempenho obtido pelo nosso modelo se destacou em diversos cenários, e mostrou um equilíbrio adequado entre sensibilidade e precisão. Assim, acreditamos que o objetivo de explorar o potencial do algoritmo HinSAGE para a tarefa de predição de alvos de miRNAs em câncer foi bem sucedida, e que os resultados alcançados no presente trabalho servem como motivação para aprofundar este estudo em trabalhos futuros.

6.1 Dificuldades Encontradas

Durante o desenvolvimento do trabalho, enfrentamos diversos desafios metodológicos. O primeiro grande desafio foi a dificuldade em obter um conjunto de dados balanceado que contivesse interações miRNA-mRNA alvo positivas e negativas. Essa dificuldade é comum em muitos trabalhos relacionados. Embora haja uma grande disponibilidade de dados biológicos, muitas vezes só são registradas as interações em que o miRNA de fato degrada ou inibe seu mRNA alvo. Isso é problemático em trabalhos que utilizam algoritmos de AM tradicionais para identificar interações gênicas, uma vez que conjuntos de dados desbalanceados podem afetar diretamente os resultados alcançados. O uso de GNNs visou diminuir um pouco esta limitação metodológica.

Também salientamos os desafios encontrados na comparação da nossa proposta com trabalhos relacionados. Estes fatores foram destacados na Seção 5.1.2 e apresentaram uma grande limitação para uma avaliação comparativa mais abrangente do nosso modelo em relação a métodos ou ferramentas disponíveis na literatura. Não só foi inviável comparar o modelo com uma abordagem similar voltada a predição de alvos de miRNAs em câncer, devido à aparente inexistência de um trabalho que aborda especificamente este nicho usando algoritmos de aprendizado de máquina ou aprendizado profundo, como também foi difícil conduzir uma análise comparativa justa, do ponto de vista metodológico, treinando e testando os diferentes modelos com os mesmos conjuntos de dados.

Outro grande desafio foi o fato de as redes neurais de grafos ainda serem uma área de estudo relativamente recente. Embora sua utilização tenha se mostrado promissora na resolução de problemas em que domínios não-Euclidianos, ainda existem lacunas em sua aplicação que precisam ser melhor investigadas. Por exemplo, as estratégias de divisão de dados em treinamento, validação e teste devem ser revisadas, a forma de avaliação

desses modelos precisa ser melhor definida, e a integração de informações como o *score* ou peso atribuído a cada aresta no processo de treinamento ainda parece ser um problema em aberto. Além disso, nosso trabalho parece representar um dos primeiros esforços em predizer interações miRNA–mRNA usando redes neurais de grafo, o que tornou sua avaliação um grande desafio devido à falta de trabalhos diretamente relacionados que utilizassem abordagem similar.

Um outro grande desafio e também limitação está na escolha dos valores dos hiperparâmetros existentes no modelo de aprendizado proposto. Devido ao grande número de combinações possíveis, definir o melhor conjunto de forma manual ou mesmo se utilizando de métodos de otimização é uma tarefa bastante complexa. Deste modo, não é possível garantir que durante a realização dos nossos experimentos foi identificada a melhor configuração de parâmetros para o modelo.

6.2 Trabalhos Futuros

Como trabalhos futuros, identificamos vários aspectos que podem ser explorados de forma mais aprofundada. Primeiramente, seria interessante investigar mais profundamente os resultados obtidos com a utilização da representação baseada em grafo no domínio estudado, principalmente levando em consideração o contexto abordado de miRNAs associados a câncer. Além disso, seria útil entender melhor as percepções que poderiam ser obtidas a partir do grafo construído e das predições geradas pelo modelo, incluindo a investigação de interações positivas preditas pelo modelo que ainda não são conhecidas. Esta análise poderia auxiliar na descoberta de novos conhecimentos relacionados à regulação gênica por miRNAs.

Em segundo lugar, seria importante comparar nossa abordagem com outros algoritmos de redes neurais em grafo, os quais também podem ser aplicados diretamente ao grafo base. Embora o GraphSAGE seja um dos algoritmos mais conhecidos e utilizados, outros algoritmos podem apresentar diferentes características e objetivos que poderiam contribuir na resolução do problema proposto. Entretanto, uma possível dificuldade seria encontrar ou propor adaptações destes algoritmos para grafos heterogêneos.

Por fim, seria interessante realizar uma comparação mais robusta com outros trabalhos que utilizam a mesma abordagem proposta para a predição de interações miRNA-alvo associadas a câncer. Isso permitiria identificar oportunidades de melhoria em relação aos conjuntos de dados utilizados e aprimorar a eficácia da nossa abordagem.

REFERÊNCIAS

AGARWAL, V. et al. Predicting effective microRNA target sites in mammalian mRNAs. **eLife**, v. 4, n. AUGUST2015, p. 1–38, 2015. ISSN 2050084X.

AGGARWAL, C. C. **Neural Networks and Deep Learning**: A textbook. Cham: Springer, 2018. 497 p. ISBN 978-3-319-94462-3.

ALAMSYAH, A.; RAHARDJO, B.; KUSPRIYANTO. **Social Network Analysis Taxonomy Based on Graph Representation**. [S.l.]: arXiv, 2013.

ALPAYDIN, E. **Introduction to Machine Learning**. 3. ed. Cambridge, MA: MIT Press, 2014. (Adaptive Computation and Machine Learning). ISBN 978-0-262-02818-9.

ARDEKANI, A. M.; NAEINI, M. M. The role of microRNAs in human diseases. **Avicenna Journal of Medical Biotechnology**, v. 2, n. 4, p. 161–179, 2010. ISSN 20082835.

BANDYOPADHYAY, S. et al. Mbstar: Multiple instance learning for predicting specific functional binding sites in microrna targets. **Scientific Reports**, v. 5, p. 8004, 01 2015.

BANDYOPADHYAY, S.; MITRA, R. Targetminer: Microrna target prediction with systematic identification of tissue-specific negative examples. **Bioinformatics**, v. 25, p. 2625–31, 09 2009.

BETEL, D. et al. Comprehensive modeling of microRNA targets predicts functional non-conserved and non-canonical sites. **Genome biology**, v. 11, p. R90, 08 2010.

BLANCO, J.; PAZOS, A.; FERNANDEZ-LOZANO, C. Machine learning analysis of TCGA cancer data. **PeerJ Computer Science**, v. 7, p. e584, 07 2021.

BORGWARDT, K. M. et al. Protein function prediction via graph kernels. **Bioinformatics**, Oxford University Press, v. 21, 2005. ISSN 13674811.

BRONSTEIN, M. M. et al. **Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges**. arXiv, 2021. Available from Internet: <https://arxiv.org/abs/2104.13478>.

BUDAK, H.; ZHANG, B. **MicroRNAs in model and complex organisms**. [S.l.]: Springer, 2017. 121–124 p.

CAI, H.; ZHENG, V. W.; CHANG, K. C. C. A comprehensive survey of graph embedding: Problems, techniques, and applications. **IEEE Transactions on Knowledge and Data Engineering**, v. 30, p. 1616–1637, 2018. ISSN 15582191.

CALIN, G. A. et al. Frequent deletions and down-regulation of micro-RNA genes miR15 and miR16 at 13q14 in chronic lymphocytic leukemia. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 99, n. 24, p. 15524–15529, 2002.

CHANDRA, V. et al. Mtar: A computational microrna target prediction architecture for human transcriptome. **BMC bioinformatics**, v. 11 Suppl 1, p. S2, 01 2010.

CHEN, L. et al. Trends in the development of miRNA bioinformatics tools. **Briefings in Bioinformatics**, v. 20, p. 1836–1852, 2019. ISSN 14774054.

CHENG, S. et al. Mirtdl: A deep learning approach for mirna target prediction. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 13, p. 1–1, 12 2015.

CHING, T. et al. Opportunities and obstacles for deep learning in biology and medicine. **Journal of The Royal Society Interface**, v. 15, p. 20170387, 04 2018.

CHOU, C.-H. et al. mirtarbase 2016: Updates to the experimentally validated mirna-target interactions database. **Nucleic Acids Research**, v. 44, p. gkv1258, 11 2015.

DAI, Y.; ZHOU, X. Computational methods for the identification of microRNA targets. **Open Access Bioinformatics**, v. 2, p. 29–39, 05 2010.

DATA61, C. **StellarGraph Machine Learning Library**. [S.l.]: GitHub, 2018. <https://github.com/stellargraph/stellargraph>.

DAVIS-DUSENBERY, B. N.; HATA, A. MicroRNA in cancer: The involvement of aberrant microRNA biogenesis regulatory pathways. **Genes and Cancer**, SAGE Publications Inc., v. 1, p. 1100–1114, 2010. ISSN 19476027.

DING, J.; LI, X.; HU, H. Tarpmir: A new approach for microrna target site prediction. **Bioinformatics**, v. 32, 05 2016.

DRAGOMIR, M. P.; KNUTSEN, E.; CALIN, G. A. Classical and noncanonical functions of miRNAs in cancers. **Trends in Genetics**, Elsevier, v. 38, n. 4, p. 379–394, 2022.

FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2011.

FAN, X.; KURGAN, L. Comprehensive overview and assessment of computational prediction of microRNA targets in animals. **Briefings in Bioinformatics**, Oxford University Press, v. 16, n. 5, p. 780–794, 2015.

FILHO, J. C. R.; KIMURA, E. T. Micrornas: Nova classe de reguladores gênicos envolvidos na função endócrina e câncer. **Arquivos Brasileiros de Endocrinologia e Metabologia**, v. 50, p. 1102–1107, 2006. ISSN 00042730.

GIBSON, G. Microarray analysis: genome-scale hypothesis scanning. **PLoS Biology**, Public Library of Science San Francisco, USA, v. 1, n. 1, p. e15, 2003.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>.

HAMILTON, W.; YING, Z.; LESKOVEC, J. Inductive representation learning on large graphs. **Advances in Neural Information Processing Systems**, Curran Associates, Inc., p. 1024–1034, 2017.

HANAHAN, D. Hallmarks of cancer: new dimensions. **Cancer discovery**, AACR, v. 12, n. 1, p. 31–46, 2022.

HEIKKINEN, L.; KOLEHMAINEN, M.; WONG, G. Prediction of microrna targets in caenorhabditis elegans using a self-organizing map. **Bioinformatics (Oxford, England)**, v. 27, p. 1247–54, 03 2011.

HUANG, H.-Y. et al. miRTarBase update 2022: an informative resource for experimentally validated miRNA–target interactions. **Nucleic Acids Research**, v. 50, n. D1, p. D222–D230, 11 2021. ISSN 0305-1048. Available from Internet: <https://doi.org/10.1093/nar/gkab1079>.

HUANG, J.; MORRIS, Q.; FREY, B. Bayesian inference of microrna targets from sequence and expression data. **Journal of computational biology : a journal of computational molecular cell biology**, v. 14, p. 550–63, 07 2007.

JACOBSEN, A. et al. Analysis of microRNA-target interactions across diverse cancer types. **Nature Structural & Molecular Biology**, Nature Publishing Group US New York, v. 20, n. 11, p. 1325–1332, 2013.

JI, C. et al. Predicting miRNA-disease associations based on heterogeneous graph attention networks. **Frontiers in Genetics**, Frontiers Media SA, v. 12, p. 727744, 2021.

KANG, J. et al. RNAInter v4.0: RNA interactome repository with redefined confidence scoring system and improved accessibility. **Nucleic Acids Research**, v. 50, 10 2022.

KARAGKOUNI, D. et al. DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA–gene interactions. **Nucleic Acids Research**, Oxford University Press, v. 46, n. D1, p. D239–D245, 2018.

KERTESZ, M. et al. The role of site accessibility in microRNA target recognition. **Nature genetics**, v. 39, n. 10, p. 1278—1284, October 2007. ISSN 1061-4036. Available from Internet: <https://doi.org/10.1038/ng2135>.

KIPF, T. N.; WELLING, M. **Semi-Supervised Classification with Graph Convolutional Networks**. arXiv, 2016. Available from Internet: <https://arxiv.org/abs/1609.02907>.

KOZOMARA, A.; BIRGAOANU, M.; GRIFFITHS-JONES, S. miRBase: from microRNA sequences to function. **Nucleic Acids Research**, Oxford University Press, v. 47, n. D1, p. D155–D162, 2019.

KOZOMARA, A.; GRIFFITHS-JONES, S. miRBase: annotating high confidence microRNAs using deep sequencing data. **Nucleic Acids Research**, v. 42, n. D1, p. D68–D73, 11 2013. ISSN 0305-1048. Available from Internet: <https://doi.org/10.1093/nar/gkt1181>.

KYROLLOS, D. et al. Rpmirdip: Reciprocal perspective improves mirna targeting prediction. **Scientific Reports**, v. 10, p. 11770, 07 2020.

LEE, B. Deep learning-based microrna target prediction using experimental negative data. **IEEE Access**, v. 8, p. 197908–197916, 2020.

LEE, B. et al. DeepTarget: End-to-end learning framework for MicroRNA target prediction using deep recurrent neural networks. **ACM-BCB 2016 - 7th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics**, n. March, p. 434–442, 2016.

LEE, R. C.; FEINBAUM, R. L.; AMBROS, V. The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. **Cell**, Elsevier, v. 75, n. 5, p. 843–854, 1993.

LEWIS, B.; BURGE, C.; BARTEL, D. Conserved Seed Pairing, Often Flanked by Adenosines, Indicates that Thousands of Human Genes are MicroRNA Targets. **Cell**, v. 120, p. 15–20, 02 2005.

LEWIS, B. et al. Prediction of mammalian MicroRNA Targets. **Cell**, v. 115, p. 787–98, 01 2004.

LI, L. et al. New support vector machine-based method for microrna target prediction. **Genetics and molecular research : GMR**, v. 13, p. 4165–4176, 06 2014.

LI, X. et al. Integrated analysis of MicroRNA (miRNA) and mRNA profiles reveals reduced correlation between MicroRNA and target gene in cancer. **BioMed Research International**, Hindawi, v. 2018, 2018.

LI, Y.; KOWDLEY, K. V. MicroRNAs in common human diseases. **Genomics, Proteomics & Bioinformatics**, Elsevier, v. 10, n. 5, p. 246–253, 2012.

LIGGETT. Micrornas: History, biogenesis, and their evolving role in animal development and disease. **Bone**, v. 23, p. 1–7, 2014. ISSN 15378276.

LIN, K. et al. Predicting mirna's target from primary structure by the nearest neighbor algorithm. **Molecular diversity**, v. 14, p. 719–29, 11 2010.

LIU, W.; WANG, X. Prediction of functional microrna targets by integrative modeling of microrna binding and target expression data. **Genome Biology**, v. 20, 01 2019.

LO, W. W. et al. E-graphsage: A graph neural network based intrusion detection system for iot. 3 2021. Available from Internet: <http://arxiv.org/abs/2103.16329>.

LOH, H.-Y. et al. The regulatory role of microRNAs in breast cancer. **International Journal of Molecular Sciences**, MDPI, v. 20, n. 19, p. 4940, 2019.

MAJI, R.; KHATUA, S.; GHOSH, Z. A supervised ensemble approach for sensitive microrna target prediction. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, PP, p. 1–1, 07 2018.

MASON, O.; VERWOERD, M. Graph theory and networks in biology. **IET Systems Biology**, IET, v. 1, n. 2, p. 89–119, 2007.

MENDOZA, M. R. et al. RFMirTarget: Predicting Human MicroRNA Target Genes with a Random Forest Classifier. **PLoS ONE**, v. 8, n. 7, 2013. ISSN 19326203.

MENOR, M. et al. Mirmark: A site-level and utr-level classifier for mirna target prediction. **Genome biology**, v. 15, p. 500, 10 2014.

MIN, H.; YOON, S. Got target? Computational methods for microRNA target prediction and their extension. **Experimental and molecular medicine**, v. 42, p. 233–44, 02 2010.

MIN, S.; LEE, B.; YOON, S. TargetNet: functional microRNA target prediction with deep neural networks. **Bioinformatics**, v. 38, n. 3, p. 671–677, 10 2021. ISSN 1367-4803. Available from Internet: <https://doi.org/10.1093/bioinformatics/btab733>.

MITRA, R.; BANDYOPADHYAY, S. Multimitar: A novel multi objective optimization based mirna-target prediction method. **PloS one**, v. 6, p. e24583, 09 2011.

MOUSAVI, R.; EFTEKHARI, M.; HAGHIGHI, M. A new approach to human microrna target prediction using ensemble pruning and rotation forest. **Journal of Bioinformatics and Computational Biology**, v. 13, 05 2015.

O'BRIEN, J. et al. Overview of microrna biogenesis, mechanisms of actions, and circulation. **Frontiers in Endocrinology**, v. 9, p. 1–12, 2018. ISSN 16642392.

PARVEEN, A. et al. Applications of machine learning in miRNA discovery and target prediction. **Current Genomics**, Bentham Science Publishers, v. 20, n. 8, p. 537–544, 2019.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEER, G. V. et al. MiSTAR: MiRNA target prediction through modeling quantitative and qualitative miRNA binding site information in a stacked model structure. **Nucleic Acids Research**, v. 45, 12 2016.

PENG, Y.; CROCE, C. M. The role of MicroRNAs in human cancer. **Signal Transduction and Targeted Tpherapy**, Nature Publishing Group, v. 1, n. 1, p. 1–9, 2016.

PETERSON, S. M. et al. Common features of microRNA target prediction tools. **Frontiers in Genetics**, v. 5, p. 1–10, 2014. ISSN 16648021.

PINZÓN, N. et al. microRNA target prediction programs predict many false positives. **Genome Research**, Cold Spring Harbor Lab, v. 27, n. 2, p. 234–245, 2017.

PLA, A.; ZHONG, X.; RAYNER, S. miRAW: A deep learning-based approach to predict microRNA targets by analyzing whole microRNA transcripts. **PLOS Computational Biology**, v. 14, p. e1006185, 07 2018.

PLOTNIKOVA, O.; BARANOVA, A.; SKOBLOV, M. Comprehensive analysis of human microRNA–mRNA interactome. **Frontiers in Genetics**, Frontiers Media SA, v. 10, p. 933, 2019.

RASCHKA, S. Model evaluation, model selection, and algorithm selection in machine learning. **arXiv**, abs/1811.12808, 2018. Available from Internet: <http://arxiv.org/abs/1811.12808>.

RECAMONDE-MENDOZA, M. et al. Rfmirtarget: Predicting human microrna target genes with a random forest classifier. **PloS one**, v. 8, p. e70153, 07 2013.

RECZKO, M. et al. Accurate microrna target prediction using detailed binding site accessibility and machine learning on proteomics data. **Frontiers in genetics**, v. 2, p. 103, 01 2011.

RHODES, D. R.; CHINNAIYAN, A. M. Integrative analysis of the cancer transcriptome. **Nature Genetics**, Nature Publishing Group US New York, v. 37, n. Suppl 6, p. S31–S37, 2005.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: a modern approach**. 3. ed. [S.l.]: Pearson, 2009.

SAETROM, O.; SNøVE, O.; SAETROM, P. Weighted sequence motifs as an improved seeding step in microrna target prediction algorithms. **RNA (New York, N.Y.)**, v. 11, p. 995–1003, 08 2005.

SCHÄFER, M.; CIAUDO, C. Prediction of the miRNA interactome – Established methods and upcoming perspectives. **Computational & Structural Biotechnology Journal**, v. 18, p. 548–557, 2020. ISSN 20010370.

SETTINO, M.; CANNATARO, M. Survey of main tools for querying and analyzingTCGA data. In: IEEE. **2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)**. [S.l.], 2018. p. 1711–1718.

SHU, J. et al. Dynamic and modularized MicroRNA regulation and its implication in human cancers. **Scientific Reports**, Springer, v. 7, n. 1, p. 1–17, 2017.

SKOK, D. J. et al. The integrative knowledge base for miRNA-mRNA expression in colorectal cancer. **Scientific Reports**, Springer, v. 9, n. 1, p. 1–9, 2019.

STURM, M. et al. Targetspy: A supervised machine learning approach for microrna target prediction. **BMC bioinformatics**, v. 11, p. 292, 05 2010.

SVORONOS, A. A.; ENGELMAN, D. M.; SLACK, F. J. OncomiR or tumor suppressor? The duplicity of microRNAs in cancer. **Cancer Research**, AACR, v. 76, n. 13, p. 3666–3670, 2016.

TAN, P.-N. Receiver operating characteristic. **Encyclopedia of Database Systems**, Springer, v. 1, 2009.

TARCA, A. L. et al. Machine learning and its applications to biology. **PLoS computational biology**, v. 3, n. 6, 2007. ISSN 15537358.

TAY, Y.; RINN, J.; PANDOLFI, P. P. The multilayered complexity of ceRNA crosstalk and competition. **Nature**, Nature Publishing Group, v. 505, n. 7483, p. 344–352, 2014.

TOKáR, T. et al. MirDIP 4.1 - Integrative database of human microRNA target predictions. **Nucleic Acids Research**, v. 46, 11 2017.

VLACHOS, I. et al. Diana-tarbase v7.0: Indexing more than half a million experimentally supported mirna:mrna interactions. **Nucleic acids research**, v. 43, 11 2014.

WANG, X. Improving microrna target prediction by modeling with unambiguously identified microrna-target pairs from clip-ligation studies. **Bioinformatics (Oxford, England)**, v. 32, 01 2016.

WANG, Z.; GERSTEIN, M.; SNYDER, M. RNA-Seq: a revolutionary tool for transcriptomics. **Nature Reviews Genetics**, Nature Publishing Group UK London, v. 10, n. 1, p. 57–63, 2009.

WEN, M. et al. DeepMirTar: A deep-learning approach for predicting human miRNA targets. **Bioinformatics**, v. 34, n. 22, p. 3781–3787, 2018. ISSN 14602059.

WINTER, J. et al. Many roads to maturity: microRNA biogenesis pathways and their regulation. **Nature Cell Biology**, Nature Publishing Group UK London, v. 11, n. 3, p. 228–234, 2009.

WITKOS, T.; KOSCIANSKA, E.; KRZYZOSIAK, W. Practical Aspects of microRNA Target Prediction. **Current Molecular Medicine**, v. 11, p. 93–109, 02 2011.

WU, Z. et al. A comprehensive survey on graph neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 32, p. 4–24, 2021. ISSN 21622388.

XIAO, F. et al. miRecords: An integrated resource for microRNA-target interactions. **Nucleic Acids Research**, v. 37, n. SUPPL. 1, p. 105–110, 2009. ISSN 03051048.

XIAO, Y. et al. Prioritizing cancer-related key miRNA–target interactions by integrative genomics. **Nucleic Acids Research**, Oxford University Press, v. 40, n. 16, p. 7653–7665, 2012.

YANG, Y.; WANG, Y.-P.; LI, K.-B. Mirtif: A support vector machine-based microrna target interaction filter. **BMC bioinformatics**, v. 9 Suppl 12, p. S4, 02 2008.

YOUSEF, M. et al. Naive bayes for microrna target predictions machine learning for microrna targets. **Bioinformatics (Oxford, England)**, v. 23, p. 2987–92, 12 2007.

YU, L.; JU, B.; REN, S. HLGNN-MDA: Heuristic Learning Based on Graph Neural Networks for miRNA–Disease Association Prediction. **International Journal of Molecular Sciences**, MDPI, v. 23, n. 21, p. 13155, 2022.

ZHENG, X. et al. Prediction of mirna targets by learning from interaction sequences. **PLOS ONE**, v. 15, p. e0232578, 05 2020.

ZHONG, S. et al. miRNAs in lung cancer. A systematic review identifies predictive and prognostic miRNA candidates for precision medicine in lung cancer. **Translational Research**, Elsevier, v. 230, p. 164–196, 2021.

ZHOU, J. et al. Graph neural networks: A review of methods and applications. **AI Open**, Elsevier Ltd, v. 1, p. 57–81, 2020. ISSN 26666510. Available from Internet: <https://doi.org/10.1016/j.aiopen.2021.01.001>.

APÊNDICE A — INFORMAÇÕES ADICIONAIS DA REVISÃO DA LITERATURA

Tabela A.1 – Protocolo definido para a busca de trabalhos relacionados.

Objetivo	Nosso objetivo foi o levantamento da literatura aplicada à previsões de alvos de miRNA usando aprendizado de máquina
Questão principal	Qual é o estado da arte no desenvolvimento de métodos de previsão de alvos usando aprendizado de máquina?
Termo de busca	((("microRNA target"[Title/Abstract] OR "miRNAtarget"[Title/Abstract]) AND ("prediction"[Title/Abstract] OR "identification"[Title/Abstract]))) AND("machine learning"[Title/Abstract] OR "supervisedlearning"[Title/Abstract] OR "classification"[Title/Abstract] OR "deep learning"[Title/Abstract] OR "unsupervisedlearning"[Title/Abstract])
Questões específicas	1. Qual método(s) de aprendizado de máquina foi empregado? 2. Como foi gerada a base de dados utilizada no desenvolvimento? 3. Como ocorreu a divisão entre a base de treino, validação e teste? 4. Quais foram os critérios de avaliação do modelo? 5. Como funciona a reprodutividade do trabalho desenvolvido?
Critérios de inclusão	1. O estudo desenvolvido menciona previsão de alvos miRNAs. 2. O artigo emprega a utilização métodos de aprendizado computacionais. 3. O artigo esclarece o método de aprendizado utilizado. 4. O artigo descreve como ocorreu a aquisição e manipulação sobre os dados. 5. O estudo descreve quais foram os métodos de avaliação empregados.
Critérios de exclusão	1. O trabalho não indica o desenvolvimento de métodos para previsão de alvos 2. O trabalho não utiliza métodos de aprendizado de máquina 3. O artigo não esclarece como foi definida a divisão entre treino, validação e teste 4. O artigo não apresenta os critérios de avaliação

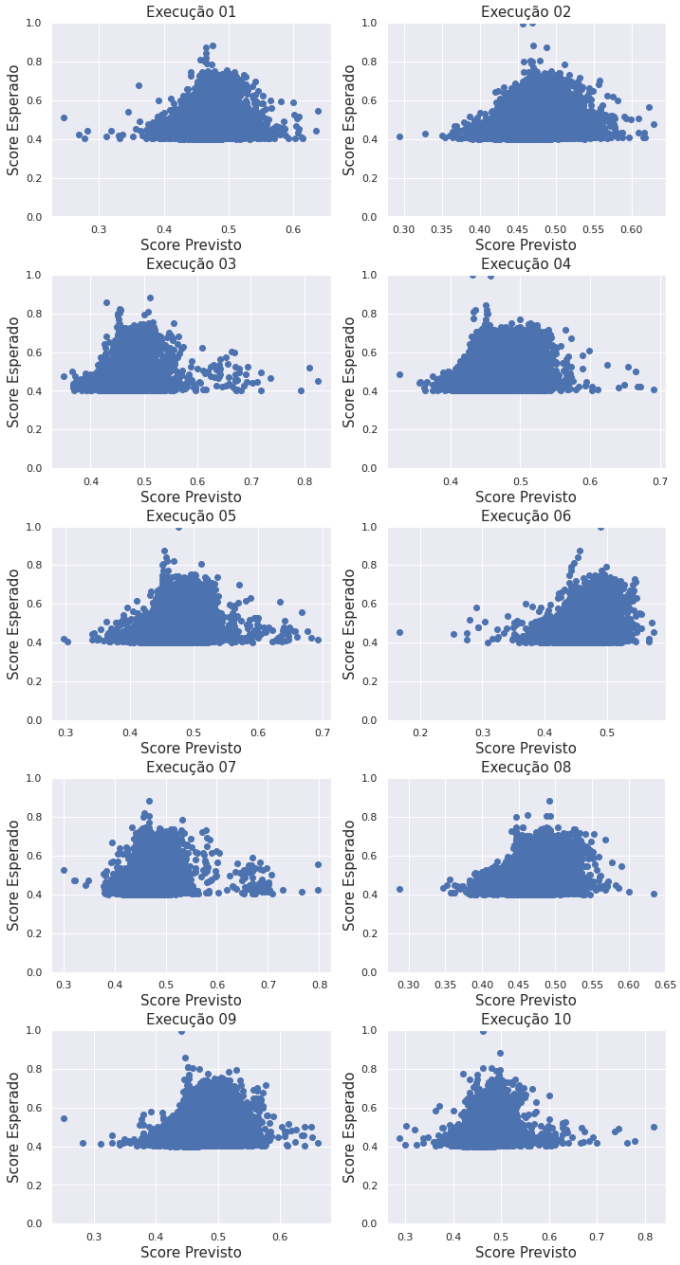
Tabela A.2 – Revisão da Literatura

Referência	Qt. Citações	Ano da Publicação	Método Comp..	Origem dos Dados	Repositorio/Site
Lee (2020)	2	2020	Deep Learning	-	-
Min, Lee and Yoon (2021)	0	2021	Residual neural network	DIANA-TarBase; MirTarBase;	https://github.com/mswzeus/TargetNet
Maji, Khatua and Ghosh (2018)	4	2020	KNN / XGB	AGO PAR-CLIP; CLASH; mirBase;	http://bicresources.jcbose.ac.in/zhumur/mirtpred/
Zheng et al. (2020)	10	2020	Rede Neural Convolucional Multicamadas	CLASH; miRBase; miRTarBase; Diana TarBase;	https://github.com/zhengxueming/cnnMirTarget
Kyrollos et al. (2020)	6	2020	Perspectiva recíproca (RP) / XGB	mirDIP; DIANA-TarBase; mirTarBase;	https://doi.org/10.5683/SP2/LD8JKJ https://cu-bic.ca/RPmirDIP/
Liu and Wang (2019)	310	2019	Support Vector Machine (SVM)	miRDB; NCBI GEO (perfil de RNA-seq);	http://mirdb.org/
Wen et al. (2018)	44	2018	Rede neural multicamada		https://github.com/Bjoux2/DeepMirTar_SdA
Pla, Zhong and Rayner (2018)	43	2018	Deep Learning	Diana TarBase; MirTarBase; CLIP-Seq; CLASH; iPAR-CLIP;	https://bitbucket.org/account/user/bipous/projects/MIRAW
Peer et al. (2016)	23	2017	Regressão Logística e Florestas Aleatórias	-	http://mi-star.org/
Wang (2016)	208	2016		-	http://mirdb.org
Ding, Li and Hu (2016)	104	2016	Florestas aleatórias	CLASH; PAR-CLIP; HEK293;	http://hulab.ucf.edu/research/projects/miRNA/TarPmiR/
Lee et al. (2016)	68	2016	Redes Neurais Recorrentes Profundas		http://data.snu.ac.kr/pub/deepTarget/
Cheng et al. (2015)	61	2016	Rede Neural Convolucional (CNN)	TarBase; miRBase; NCBI	http://nclab.hit.edu.cn/ccrm
Mousavi, Eftekhari and Haghighi (2015)	9	2015	Ensemble Pruning (GA); Rotation Forest (EP-RTF);	Yan et al. (Conjunto de dados I); Ahmadi et al. (Conjunto de dados II); Yu et al. (Conjunto de dados III); Mendoza et al. (Conjunto de dados IV);	-
Bandyopadhyay et al. (2015)	58	2014	Florestas aleatórias	UCSC Genome Browser; miRBase; miRecords; TargetMiner para obter exemplos negativos	http://www.isical.ac.in/~bioinfo_miu/MBStar30.htm
Menor et al. (2014)	37	2014	Floresta aleatória; Support Vector Machine (SVM); Regressão Logística	miRecords e mirTarBase	https://github.com/lanagarmire/MirMark
Li et al. (2014)	4	2014	Support Vector Machine (SVM)	miRecords; pSILAC; TarBase; miRBase; Gene Expression Omnibus (GEO);	http://hpabws.s87.cnaaa7.com/
Recamonde-Mendoza et al. (2013)	38	2013	Florestas aleatórias	-	-
Reczko et al. (2011)	48			-	http://diana.cslab.ece.ntua.gr/DianaTools/
Heikkinen, Kolehmainen and Wong (2011)	21	2011	Rede Neural Artificial (Self-organizing map)	Harris et al 2010; miRBase; WormBase;	Disponibilidade do código
Mitra and Bandyopadhyay (2011)	51	2011	Support Vector Machine (SVM)	miRBase; UCSC; pSILAC;	https://www.isical.ac.in/~bioinfo_miu/multimitar.htm
Betel et al. (2010)	1523	2010	Support Vector Regression (SVR)	Grimson et al;	http://www.microrna.org/
Lin et al. (2010)	6		Nearest neighbour algorithm (KNN)	-	-
Chandra et al. (2010)	75	2010	Rede Neural Artificial	miRBase; Tarbase; miRecords; NCBI RefSeq; Biomart;	-
Sturm et al. (2010)	181	2010	MultiBoost with decision stumps	UCSC Genome Database; RefSeq; FlyBase; miRBase; Galaxy;	http://webclu.bio.wzw.tum.de/targetspy
Bandyopadhyay and Mitra (2009)	218	2009	Support Vector Machine (SVM)	-	-
Yang, Wang and Li (2008)	73	2008	Support Vector Machine (SVM)	-	-
Yousef et al. (2007)	173	2007	Naïve Bayes	-	-
Huang, Morris and Frey (2007)	147	2007	Modelo Bayesiano	-	-
Saetrom, Snøve and Saetrom (2005)	149	2005	Algoritmos genéticos	-	-

APÊNDICE B — ANÁLISE DE CORRELAÇÃO PARA PREDIÇÕES REALIZADAS PELO MODELO MLP

A Figura B.1 mostra o valor de *score* previsto (eixo x) pelo modelo MLP (descrito na Seção 5.2.2.2) e o valor de *score* esperado (eixo y) de acordo com informações da base RNAInter, através de um gráfico de dispersão. São mostrados os resultados para os dados de teste ao longo das 10 execuções do algoritmo.

Figura B.1 – Correlação do *score* predito e do valor esperado para cada execução realizada do modelo MLP.



Fonte: O Autor.