

ERICH LACERDA MALINOWSKI

**CRITÉRIOS DE PARADA PARA PREDIÇÃO
NÃO OBRIGATÓRIA EM NÓS-FOLHA EM
PROBLEMAS DE CLASSIFICAÇÃO
HIERÁRQUICA LOCAL**

Curitiba - PR, Brasil

2020

ERICH LACERDA MALINOWSKI

**CRITÉRIOS DE PARADA PARA PREDIÇÃO NÃO
OBRIGATÓRIA EM NÓS-FOLHA EM PROBLEMAS DE
CLASSIFICAÇÃO HIERÁRQUICA LOCAL**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGIa

Orientador: Júlio Cesar Nievola

Coorientador: Carlos Nascimento Silla Jr

Curitiba - PR, Brasil

2020

Malinowski, Erich Lacerda

CRITÉRIOS DE PARADA PARA PREDIÇÃO NÃO OBRIGATÓRIA EM NÓS-FOLHA EM
PROBLEMAS DE CLASSIFICAÇÃO HIERÁRQUICA LOCAL

ERICH LACERDA MALINOWSKI. – Curitiba - PR, Brasil, 2020-
143 p. : il. (algumas color.) ; 30 cm.

Orientador: Júlio Cesar Nievola

Coorientador: Carlos Nascimento Silla Jr

Projeto de Tese de Doutorado –

Pontifícia Universidade Católica do Paraná - PUCPR

Programa de Pós-Graduação em Informática - PPGLa, 2020.

1. Critérios de Parada. 2. Classificação Hierárquica Local. 3. Predição de Função de
Proteínas. 4. Bioinformática. I. Nievola, Júlio Cesar. II. Silla Jr, Carlos Nascimento.

III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

ERICH LACERDA MALINOWSKI

CRITÉRIOS DE PARADA PARA PREDIÇÃO NÃO OBRIGATÓRIA EM NÓS-FOLHA EM PROBLEMAS DE CLASSIFICAÇÃO HIERÁRQUICA LOCAL

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Doutor em Informática.

Trabalho aprovado. Curitiba - PR, Brasil, 05 de fevereiro de 2020:

Prof. Dr. Júlio Cesar Nievola
Orientador

Prof. Dr. Carlos Nascimento Silla Jr
Coorientador

Prof. Dr. Emerson Cabrera Paraíso
Convidado 1

**Prof(a). Dr(a). Myriam Regattieri de
B. da S. Delgado**
Convidada 2

**Prof. Dr. Luiz Henrique de Campos
Merschmann**
Convidado 3

Curitiba - PR, Brasil
2020

Dedico este trabalho à minha família que sempre me motivou para que eu conseguisse atingir mais essa etapa.

Agradecimentos

A Deus, por todas as bênçãos em minha vida.

Um agradecimento especial aos meus orientadores ao longo destes anos de pesquisa, Prof. Dr. Júlio Cesar Nievola e Prof. Dr. Carlos Nascimento Silla Junior, pela disponibilidade e paciência. Pessoas que admiro muito pelo profissionalismo e dedicação com a profissão.

À minha família, minha mãe, irmã e esposa pelo incentivo e compreensão nos momentos de ausência, durante todo este período.

Aos meus amigos de doutorado pelo apoio e momentos de estudo partilhados.

E a todos que direta ou indiretamente colaboraram na execução desta pesquisa.

Resumo

Muitas pesquisas nas áreas de mineração de dados, aprendizagem de máquina, reconhecimento de padrões, entre outras, abordam apenas a classificação plana. Porém, existe uma variedade enorme de problemas, cujas classes estão dispostas em uma hierarquia, como por exemplo na predição de função de proteína, a qual a Bioinformática vem tratando destes respectivos problemas. A dificuldade deste tipo de aplicação é oriunda da própria estrutura de organização da proteína, que, por sua vez, descreve suas funções por meio de hierarquia no formato de árvore ou grafo acíclico dirigido. Além disso, a complexidade aumenta pelo fato de que muitas destas proteínas são rotuladas apenas parcialmente nestas bases de dados. Neste caso, uma determinada solução para este problema deve levar em consideração o formato da estrutura hierárquica que estas proteínas estão rotuladas, bem como propiciar que uma determinada proteína possa ser predita em qualquer nível da hierarquia. Estes requisitos devem ser atendidos para obter uma melhor qualidade de predição. Em muitas soluções da literatura a profundidade de rotulação não é levada em consideração. Nesta tese foram propostos diversos métodos de critérios de parada para predição opcional de nós-folha. Os quatro primeiros métodos - LCPNwFOE, LCPNwVOE, LCPNwFME e LCPNwVME - utilizaram a Entropia como critério de parada. O quinto método LCPNwLC definiu o critério de parada por meio de um Classificador de Nível. Por fim, o sexto método LCPNwINT realizou a interseção de dois ou mais métodos de avaliação diferentes para a devida predição da classe de uma determinada hierarquia. A avaliação dos resultados foi obtida por meio das métricas de avaliação precisão hierárquica, revocação hierárquica e medida-F hierárquica em bases de dados de função de proteínas com duas representações diferentes: Descritores Locais (LD) e FunCat. Os resultados mostraram que os métodos propostos de critérios de parada obtiveram melhores resultados em comparação com os principais métodos do estado da arte.

Palavras-chave: Critérios de Parada; Classificação Hierárquica Local; Predição de Função de Proteínas; Bioinformática.

Abstract

Much research in the fields of data mining, machine learning, pattern recognition, among others, addresses only flat classification. However, there are many problems, whose classes are arranged in a hierarchy, as in protein function prediction, which Bioinformatics has been dealing with these respective problems. The difficulty of this type of application comes from the protein's organization structure, which, in turn, describes its functions through a hierarchy in the shape of a tree or directed acyclic graph. Furthermore, the complexity increases due to the fact that many of these proteins are only partially labeled in these databases. In this case, a particular solution to this problem must consider the format of the hierarchical structure that these proteins are labeled and allow that a particular protein can be predicted at any level of the hierarchy. These requirements must be met to obtain a better quality of prediction. Despite that, in many solutions in the literature, the depth of labeling is not taken into account. In this thesis, several methods of stopping criteria for optional leaf node prediction have been proposed. The first four methods - LCPNwFOE, LCPNwVOE, LCPNwFME and LCPNwVME - used Entropy as a stopping criterion. The fifth method, LCPNwLC, defined the stopping criterion through a Level Classifier. Finally, the sixth method, LCPNwINT, performed the intersection of two or more different evaluation methods to predict the class of a given hierarchy. The evaluation of the results was obtained through the metrics of hierarchical precision evaluation, hierarchical recall, and hierarchical F-measure in protein function databases with two different representations: Local Descriptors (LD) and FunCat. The results showed that the proposed methods of stopping criteria can lead to better results than the main methods of state of the art.

Keywords: Stopping Criteria; Local Hierarchical Classification; Protein Function Prediction; Bioinformatics.

Lista de ilustrações

Figura 1 – Etapas de um classificador	25
Figura 2 – Modelo não linear de um neurônio	26
Figura 3 – Rede neural alimentada adiante ou acíclica com uma camada oculta	27
Figura 4 – Rede recorrente sem laços de auto realimentação e sem neurônios ocultos	28
Figura 5 – Rede recorrente com neurônios ocultos	28
Figura 6 – Arquitetura CNN	30
Figura 7 – Exemplo de funcionamento de uma CNN	31
Figura 8 – Diferença entre as CNN's 1D e 2D	32
Figura 9 – Exemplo de uma estrutura hierárquica de classes baseada em árvore	33
Figura 10 – Exemplo de uma estrutura hierárquica de classes em DAG	33
Figura 11 – Exemplo de predição em nós-folha obrigatória	34
Figura 12 – Exemplo de predição em nós-folha não obrigatória	34
Figura 13 – Exemplo da abordagem classificação hierárquica plana	36
Figura 14 – Exemplo de classificação hierárquica local por nó	37
Figura 15 – Exemplo de classificação hierárquica local por nó pai	38
Figura 16 – Exemplo de classificação hierárquica local por nível	40
Figura 17 – Exemplo de classificação hierárquica global	41
Figura 18 – Exemplo de DAG independente da profundidade	44
Figura 19 – Exemplo de Medida independente da profundidade em árvore	44
Figura 20 – Exemplo de Medida independente da profundidade em árvore	45
Figura 21 – Exemplo de hierarquia de classes em DAG - O círculo sólido G representa a classe correta de classificação	47
Figura 22 – Ordem Decrescente de Probabilidades (DOP)	52
Figura 23 – Multiplicação de probabilidades (MP)	52
Figura 24 – Predição obrigatória de nós-folha (MLNP) Top-Down	54
Figura 25 – Exemplo de critério de parada com retorno da melhor ou última classe	55
Figura 26 – Exemplo da hierarquia do método nLLCPN	57
Figura 27 – Método LCPNB	58
Figura 28 – As 10 regiões usadas pela técnica do Descritor Local, usadas em Davies et al. (2008) e Tong e Tammi (2008)	63
Figura 29 – Modelo CNN	65
Figura 30 – Teste Estatístico Nemenyi dos métodos do Estado da Arte com classificador MLP	71
Figura 31 – Teste Estatístico dos métodos do Estado da Arte com classificador MLP (2 camadas ocultas)	72
Figura 32 – Teste Estatístico dos métodos do Estado da Arte com classificador CNN	73

Figura 33 – Cálculo da Entropia do Nó-Pai Raiz	77
Figura 34 – Exemplo com probabilidades e entropia	78
Figura 35 – Comparação pareada do método proposto LCPNwFOE com métodos da literatura por meio da Métrica hF	80
Figura 36 – Teste Nemenyi da Métrica hF do método LCPNwFOE	81
Figura 37 – Comparação pareada do método proposto LCPNwFOE com métodos da literatura por meio da Métrica hP	82
Figura 38 – Teste Nemenyi da Métrica hP com classificadores MLP (1 e 2 camadas ocultas) e CNN	83
Figura 39 – Exemplo com probabilidades e entropia	84
Figura 40 – Exemplo com probabilidades e entropia	87
Figura 41 – Comparação pareada do método proposto LCPNwVOE com métodos da literatura	88
Figura 42 – Teste Nemenyi da Métrica hF com classificadores MLP (1 e 2 camadas ocultas) e CNN	89
Figura 43 – Comparação pareada do método proposto LCPNwVOE com métodos da literatura por meio da Métrica hP	91
Figura 44 – Teste Nemenyi da Métrica hP com classificadores MLP (1 e 2 camadas ocultas) e CNN	92
Figura 45 – Exemplo para escolha da porta correta	93
Figura 46 – Exemplo cálculos da Entropia para determinadas probabilidades	94
Figura 47 – Exemplo ajuste de Entropia H'	94
Figura 48 – Comparativo Entropia Original x Modificada	95
Figura 49 – Exemplo com probabilidades e entropia	97
Figura 50 – Comparação pareada do método proposto LCPNwFME com métodos da literatura	98
Figura 51 – Teste Nemenyi para comparação dos resultados da Métrica hF do método LCPNwFME	99
Figura 52 – Comparação pareada do método proposto LCPNwFME com métodos da literatura por meio da Métrica hP	101
Figura 53 – Teste Nemenyi para comparação dos resultados da Métrica hP do método LCPNwFME	102
Figura 54 – Exemplo com probabilidades e entropia	103
Figura 55 – Exemplo com probabilidades e entropia	105
Figura 56 – Comparação pareada do método proposto LCPNwVME com métodos da literatura	107
Figura 57 – Teste Nemenyi da Métrica hF dos métodos proposto LCPNwVME e da literatura	108

Figura 58 – Comparação pareada do método proposto LCPNwVME com métodos da literatura por meio da Métrica hP	109
Figura 59 – Teste Nemenyi da Métrica hP dos métodos LCPNwVME e da literatura	110
Figura 60 – Método para reconhecimento de componentes conexos de dígitos manuscritos	111
Figura 61 – Treinamento dos Classificadores LCPN e de Nível η	112
Figura 62 – Rotulação nó-interno baseado no Classificador de Nível	113
Figura 63 – Comparação pareada do método proposto LCPNwLC com métodos da literatura	115
Figura 64 – Teste Nemenyi da Métrica hF para comparação do método proposto LCPNwLC	116
Figura 65 – Comparação pareada do método proposto LCPNwLC com métodos da literatura por meio da Métrica hP	117
Figura 66 – Teste Nemenyi da Métrica hP para comparação estatística do método LCPNwLC	118
Figura 67 – Exemplo do método LCPNwINT para Critério de Parada	119
Figura 68 – Comparação pareada do método proposto LCPNwINT com métodos da literatura	121
Figura 69 – Teste Nemenyi da Métrica hF para comparação do método proposto LCPNwINT	122
Figura 70 – Comparação pareada do método proposto LCPNwINT com métodos da literatura por meio da Métrica hP	123
Figura 71 – Teste Nemenyi da Métrica hP para comparação estatística do método LCPNwINT	124
Figura 72 – Exemplo da Avaliação de Erros por Nível	124
Figura 73 – Avaliação da Média de Erros por Nível com classificador MLP	125
Figura 74 – Avaliação da Média de Erros por Nível dos métodos com classificador MLP (2 camadas ocultas)	126
Figura 75 – Avaliação da Média de Erros por Nível com classificador CNN	126
Figura 76 – Teste de Nemenyi com métrica hF com MLP 1 e 2 camadas	133
Figura 77 – Teste de Nemenyi com métrica hF com CNN	133
Figura 78 – Teste de Nemenyi com métrica hP com MLP 1 e 2 camadas	134
Figura 79 – Teste de Nemenyi com métrica hP com CNN	134

Lista de tabelas

Tabela 1 – Base de dados FunCat	61
Tabela 2 – Características das bases de dados Descritores Locais (LD)	63
Tabela 3 – Comparativo dos Métodos com abordagem MLNP	68
Tabela 4 – Legenda das Bases de Dados	69
Tabela 5 – Comparativo dos Métodos com abordagem NMLNP	69
Tabela 6 – Resultados dos métodos da literatura com Classificador MLP	70
Tabela 7 – Resultados dos métodos da literatura com Classificador MLP com 2 camadas ocultas	71
Tabela 8 – Comparativo dos Métodos do Estado da arte com Classificador CNN	72
Tabela 9 – Resultados da métrica hF dos métodos LCNPwFOE, Top-Down e Ganho de Informação	79
Tabela 10 – Resultados da métrica hP dos métodos LCNPwFOE, Top-Down e Ganho de Informação	81
Tabela 11 – Comparativo de limiares dos métodos LCPNwFOE e LCPNwVOE	85
Tabela 12 – Resultados da métrica HF dos métodos LCNPwVOE, Top-Down e Ganho de Informação	88
Tabela 13 – Resultados da métrica hP dos métodos LCNPwVOE, Top-Down e Ganho de Informação	90
Tabela 14 – Resultados da métrica hF dos métodos LCNPwFME, Top-Down e Ganho de Informação	98
Tabela 15 – Resultados da métrica hP dos métodos LCNPwFME, Top-Down e Ganho de Informação	100
Tabela 16 – Resultados da métrica hF dos métodos LCNPwVME, Top-Down e Ganho de Informação	106
Tabela 17 – Resultados da métrica hP dos métodos LCNPwVME, Top-Down e Ganho de Informação	108
Tabela 18 – Resultados da métrica de avaliação hF dos métodos LCNPwLC, Top- Down e LCNPwVME	114
Tabela 19 – Resultados da métrica de avaliação hP dos métodos LCNPwLC, Top- Down e LCNPwVME	116
Tabela 20 – Resultados da métrica hF dos métodos LCNPwINT, Top-Down e LCNPwVME	120
Tabela 21 – Resultados da métrica hP dos métodos LCNPwINT, Top-Down e LCNPwVME	123
Tabela 22 – Resultados hF com otimização de limiares - Classificador MLP com 1 camada oculta	128

Tabela 23 – Resultados hP com otimização de limiares - Classificador MLP com 1 camada oculta	129
Tabela 24 – Resultados hF com otimização de limiares - Classificador MLP com 2 camadas ocultas	130
Tabela 25 – Resultados hP com otimização de Limiares - Classificador MLP com 2 camadas ocultas	130
Tabela 26 – Resultados hF com otimização de limiares - Classificador CNN	131
Tabela 27 – Resultados hP com otimização de limiares - Classificador CNN	132

Lista de Siglas

DAG	Grafo Acíclico Dirigido
MPL	Multi-Path of Labels
SPL	Single Path of Labels
MLNP	Mandatory Leaf Node Prediction
NMLNP	non-Mandatory Leaf Node Prediction
LCPNwFOE	Local Classifier per Parent Node with Fixed Original Entropy
LCPNwVOE	Local Classifier per Parent Node with Variable Original Entropy
LCPNwFME	Local Classifier per Parent Node with Fixed Modified Entropy
LCPNwVME	Local Classifier per Parent Node with Variable Modified Entropy
LCPNwLC	Local Classifier per Parent Node with Level Classifier
LCPNwINT	Local Classifier per Parent Node with Intersection
CNN	Convolutional Neural Networks
LCN	Local Classifier per Node
LCPN	Local Classifier per Parent Node
PCT	Predictive Clustering Trees
SC	Single Label Classification
HSC	Hierarchical Single Label Classification
HMC	Hierarchical Multilabel Classification
GMNB	GLobal Model Naive Bayes
GMNBwU	GLobal Model Naive Bayes with Usefulness
LMNBwU	Local Model Naive Bayes with Usefulness
AIS	Artificial Imune Systems
FunCat	Functional Catalogue
ELHNB	Extended Local Hierarchical Naive Bayes classifier
SPL	Single Path of Labels
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Networks
HL	Camada Oculta (do inglês, Hidden Layer)
CC	Componentes conexos de dígitos manuscritos
LCPNwFOE	Local Classifier per Parent Node with Fixed Original Entropy
LCPNwVOE	Local Classifier per Parent Node with Variable Original Entropy

LCPNwFME Local Classifier per Parent Node with Fixed Modified Entropy
LCPNwVME Local Classifier per Parent Node with Variable Modified Entropy
LCPNwLC Local Classifier per Parent Node with Level Classifier
LCPNwINT Local Classifier per Parent Node with Intersection

Lista de símbolos

T	Limiar baseado na entropia (do inglês, Threshold)
I	Instância de uma imagem de dígitos manuscritos
\mathcal{L}	Modelo Classificador de Tamanho
η	Modelo Classificador de Nível
φ	Classe atribuída de acordo com o nível

Sumário

1	INTRODUÇÃO	19
1.1	Motivação	20
1.2	Objetivos	21
1.2.1	Objetivo Geral	21
1.2.2	Objetivos Específicos	21
1.3	Contribuições científicas	22
1.4	Organização do Trabalho	23
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	Classificadores	24
2.1.1	Redes Neurais Artificiais	25
2.1.2	Aprendizagem Profunda	29
2.1.2.1	Redes Neurais Convolucionais	29
2.2	Classificação Hierárquica	32
2.2.1	Classificação Hierárquica Plana	35
2.2.2	Classificação Hierárquica Local	36
2.2.2.1	Classificação Hierárquica Local por Nó	36
2.2.2.2	Classificação Hierárquica Local por Nó Pai	38
2.2.2.3	Classificação Hierárquica Local por Nível	39
2.2.2.4	Classificação Hierárquica Global	40
2.3	Métricas de Avaliação de Classificadores Hierárquicos	42
2.3.1	Métricas de Avaliação Baseada em Distância Independente da profundidade	43
2.3.2	Métricas de avaliação baseada em distância dependente da profundidade	44
2.3.3	Métricas de avaliação baseada na Descendência	45
2.3.4	Métricas de avaliação baseada em Ancestralidade	46
2.4	Entropia - Teoria Matemática da Informação	48
3	ESTADO DA ARTE	51
3.1	Predição Obrigatória em Nós-Folha	51
3.1.1	Ordenação, Soma e Multiplicação	51
3.1.2	Média	53
3.1.3	Top-Down	53
3.2	Predição Não-Obrigatória (Opcional) em Nós-folha	54
3.2.1	TopDown com Limiar (Threshold)	54
3.2.2	Redução de Limiares, Votação Restrita e Limiares multiplicativos estendidos	55
3.2.3	Medida de Utilidade	56

3.2.4	Replicação do Nó-Interno	56
3.2.5	Ganho de Informação	59
4	EXPERIMENTOS	60
4.1	Materiais e Configuração Experimental	60
4.1.1	Bases de Dados FunCat	61
4.1.2	Bases de Dados Descritores Locais (LD)	62
4.2	Configuração Experimental	63
4.2.1	Pré-processamento	63
4.2.2	Política de Seleção de Exemplos	64
4.2.3	Classificadores Bases	64
4.2.3.1	Configuração dos Classificadores MLP	64
4.2.3.2	Configuração dos classificadores CNN	65
4.3	Métricas de Avaliação	66
4.4	Avaliação Estatística	66
5	ANÁLISE EXTENSIVA DOS MÉTODOS DO ESTADO DA ARTE	68
5.0.1	Resultados computacionais dos principais métodos MLNP e NMLNP da literatura	70
5.1	Considerações Finais	73
6	MÉTODOS PROPOSTOS	75
6.1	Método LCPNwFOE	75
6.1.1	Resultados computacionais do método LCPNwFOE	78
6.2	Método LCPNwVOE	83
6.2.1	Resultados computacionais do método LCPNwVOE	87
6.3	Método LCPNwFME	92
6.3.1	Motivação para ajuste na Fórmula da Entropia	92
6.3.2	Resultados Computacionais do método LCPNwFME	97
6.4	Método LCPNwVME	102
6.4.1	Resultados Computacionais do método LCPNwVME	105
6.5	Método LCPNwLC	110
6.5.1	Resultados computacionais	113
6.6	Método LCPNwINT	118
6.6.1	Resultados computacionais	120
6.7	Avaliação de erros por nível	124
6.8	Avaliação dos resultados com otimização do Limiar (Threshold) por Base de Dados	127
6.8.1	Resultados da otimização de Limiares com classificador MLP com 1 camada oculta	127

6.8.2	Resultados da otimização de Limiares com classificador MLP com 2 camadas ocultas	129
6.8.3	Resultados da otimização de Limiares com classificador CNN	131
6.8.4	Avaliação estatística com otimização de Limiares	132
7	CONCLUSÃO	135
7.1	Trabalhos futuros	136
	REFERÊNCIAS	138

1 Introdução

A mineração de dados é um campo interdisciplinar que permite utilizar diferentes tipos de tarefas para a descoberta de padrões, tais como, agrupamento, classificação, associação e regressão. A classificação é uma tarefa comum na aprendizagem de máquina e consiste em prever se uma determinada instância pertence a uma ou mais classes, em um conjunto de classes previamente definidas.

Muitos trabalhos da literatura tratam somente da classificação plana, na qual uma instância da base de treinamento está relacionada a apenas uma determinada classe ou várias classes em um único nível. No entanto, existe uma variedade enorme de problemas cujas classes estão dispostas em uma hierarquia, como por exemplo: na categorização de textos (DUMAIS; CHEN, 2000); (SUN; LIM, 2001), classificação de gêneros musicais (SILLA; FREITAS, 2009b); (DECORO; BARUTCUOGLU; FIEBRINK, 2007), reconhecimento de imagens (BARUTCUOGLU; DECORO, 2006);(BINDER; KAWANABE; BREFELD, 2009) e predição de função de proteínas no contexto de bioinformática (SECKER et al., 2007);(CERRI; BARROS; CARVALHO, 2014);(BORGES; NIEVOLA, 2012);(FREITAS; CARVALHO, 2007).

Tais problemas precisam levar em consideração a taxonomia hierárquica das classes, tal como uma árvore ou grafo acíclico dirigido (DAG, do inglês *Directed Acyclic Graph*). A diferença principal da disposição da taxonomia, ou seja, entre estrutura de árvore e estrutura de DAG, consiste que nesta última, cada nó pode ter um ou mais nó pai. Já na estrutura de árvore, um nó filho possui somente um nó pai, exceto o nó "Raiz". Além disso, estes problemas hierárquicos apresentam outra característica com relação ao caminho de rotulação. Caso a instância possa obter múltiplos caminhos da hierarquia, denomina-se de MPL (do inglês Multi-path of Labels) ou se cada instância tem apenas um caminho único de rotulação - SPL (Single Path of Labels) - na hierarquia de classes. Esta distinção é análoga aos problemas de classificação com rótulo único e multirrótulo na classificação "plana"(SILLA; FREITAS, 2011b).

Outro fator que distingue a classificação hierárquica da classificação plana é como a predição das classes é realizada de acordo com a profundidade de rotulação na hierarquia. As duas categorias são: predição obrigatória em nós-folha e predição não-obrigatória (opcional) em nós-folha. Na predição obrigatória em nós-folha (MLNP - Mandatory Leaf Node Prediction) todos os exemplos classificados devem ser associados com classes representadas por nós-folha. Já na predição não-obrigatória em nós-folha (NMLNP - non-Mandatory Leaf Node Prediction) uma instância pode ser predita em um nó não-folha como sua classe mais específica, sendo esta abordagem o foco principal deste trabalho

(SILLA; FREITAS, 2011b). Portanto, o problema hierárquico tratado nesta tese é da estrutura no formato de árvore (Tree), predição dos rótulos de classe em um único caminho (SPL) e a predição não-obrigatória de nós-folha (NMLNP).

Neste contexto, inúmeras abordagens para a resolução destes problemas de classificação hierárquica foram desenvolvidas. De acordo com a literatura, há duas abordagens para resolver problemas de classificação hierárquica, chamadas: Global e Local. A abordagem Global adapta algoritmos tradicionais da literatura para lidar com toda a hierarquia de classes do problema (BORGES; NIEVOLA, 2012);(SILLA; FREITAS, 2009a);(FERRANDIN; ROMAO, 2016);(ROMAO; NIEVOLA, 2012). Já a abordagem Local realiza a redução do problema em diversos problemas menores, diminuindo neste caso a complexidade. Esta abordagem utiliza classificadores tradicionais da literatura para cada sub-problema sem a necessidade de modificação dos mesmos (BLOCKKEEL et al., 2002);(CECI; MALERBA, 2007);(FABRIS; FREITAS, 2014); (HERNANDEZ; SUCAR; MORALES, 2013); (SUN et al., 2004); (HERNÁNDEZ; SUCAR; MORALES, 2014). Dentro dessas abordagens há diversas formas de desenvolvimento, bem como várias formas de avaliação de uma determinada solução para um problema hierárquico (SILLA; FREITAS, 2011b). Portanto, além do escopo do problema hierárquico já comentado anteriormente, a abordagem utilizada para o desenvolvimento da solução hierárquica para aplicação dos métodos de critérios de parada foi a Classificação Hierárquica Local por Nó Pai, pois muitos trabalhos da literatura utilizam-se desta abordagem em seus experimentos e por ter apresentado melhores resultados na análise experimental no domínio avaliado.

Nesta tese foram propostos diversos métodos de critérios de parada para predição opcional em nós-folha. Os quatro primeiros métodos utilizam a entropia como critério de parada. O quinto método define o critério de parada por meio de um classificador de nível. Por fim, o sexto método realiza a interseção de dois ou mais métodos classificação hierárquica para determinar a predição da classe em uma determinada hierarquia.

1.1 Motivação

A predição das funções de proteínas, denominada proteômica, é uma área relevante da bioinformática, porque este tipo de informação pode ser potencialmente utilizada para desenvolver fármacos e para um melhor diagnóstico no tratamento de doenças (SILLA; FREITAS, 2011b). Porém, não é uma tarefa fácil classificar ou determinar as funções das proteínas, pois, proteínas com sequências semelhantes, bem como estruturas semelhantes, podem realizar funções diferentes. Assim como, muitas sequências diferentes, compatíveis com estruturas diferentes, podem ter uma mesma função de proteína (LESK et al., 2008).

Baseado neste contexto, o uso da mineração de dados, por meio da tarefa de classificação, pode auxiliar, por exemplo, no processo de classificação funcional das proteínas.

Tal classificação é um problema importante e desafiador no campo da genômica funcional, pois apresenta-se como um problema de classificação hierárquica, devido às suas classes funcionais estarem relacionadas entre si em uma taxonomia hierárquica de classes.

Outra propriedade importante da definição de um problema hierárquico é relacionado à profundidade de rotulação. Existem abordagens da literatura que realizam a predição obrigatória em nós-folhas (MLNP). Porém, existem problemas mais interessantes e complexos, como nos casos de problemas de rotulagens com profundidade parcial. Em outras palavras, a predição pode ser realizada em qualquer nível da hierarquia, denominado de predição não-obrigatória (opcional) em nós-folhas (NMLNP). Este é um cenário em que a aplicação dessas abordagens enfrenta alguns problemas, principalmente devido a uma relação entre os conceitos de precisão hierárquica e revocação hierárquica e os conceitos de erros de generalização e especialização apresentados por (CECI; MALERBA, 2007). Ou seja, um erro de generalização refere-se ao caso em que a classe predita para um exemplo é mais genérica que a classe verdadeira conhecida associada ao exemplo. Por exemplo, predizer apenas a classe R.1 para um exemplo cuja classe verdadeira mais específica é R.1.1. Por outro lado, um erro de especialização refere-se ao caso em que a classe predita para um exemplo é mais específica do que a classe verdadeira conhecida associada ao exemplo. Por exemplo, predizer a classe R.1.1 para um exemplo cuja classe conhecida mais específica é R.1.

Alguns domínios necessitam de uma melhor especialização de rotulação da classe de acordo com a taxonomia hierárquica. Um desses casos é a predição de função de proteínas, pois evita-se custos operacionais, demanda de tempo em experimentos, especialistas na respectiva área, entre outras variáveis relacionadas à classificação funcional das proteínas. Além disso, uma grande parte das pesquisas em classificação hierárquica está concentrada neste cenário, que apresenta muitos desafios, como um grande número de classes, conjuntos de dados com taxonomias desbalanceadas e em determinadas classes possuem poucos exemplos de treinamento (CARVALHO; BRUNORO; PAPPA, 2011).

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo deste trabalho é desenvolver um método de critério de parada para predição não obrigatória em nós-folha em problemas de classificação hierárquica local.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- Avaliar os principais métodos de critérios de parada para rotulação parcial em problemas de classificação hierárquica da literatura;
- Desenvolver um método de critério de parada que permita a predição ou rotulação em nó-interno da hierarquia em problemas de classificação hierárquica local que seja independente do algoritmo de classificação;
- Avaliar os resultados por meio de métricas de desempenho específicas para a classificação hierárquica em diferentes conjunto de dados relacionados a função de proteínas;
- Avaliar e comparar estatisticamente os resultados obtidos do método proposto com os principais métodos de critérios de parada de rotulação parcial da literatura.

1.3 Contribuições científicas

Nesta tese foi proposta, inicialmente, uma análise extensiva dos principais métodos de critérios de parada da literatura, bem como a avaliação destes no contexto de predição das funções de proteínas, cujas taxonomias são desbalanceadas e a quantidade de instâncias de treinamento para determinadas classes são pequenas.

Outras contribuições desta tese foram os métodos de critérios de parada para predição não obrigatória (opcional) em nós-folha, para aplicação em problemas de Classificação Hierárquica Local.

Os quatro primeiros métodos propostos denominados LCPNwFOE (Local Classifier per Parent Node with Fixed Original Entropy), LCPNwVOE (Local Classifier per Parent Node with Variable Original Entropy), LCPNwFME (Local Classifier per Parent Node with Fixed Modified Entropy) e LCPNwVME (Local Classifier per Parent Node with Variable Modified Entropy), utilizam a entropia como critério de parada. Nestes dois últimos citados, foram aplicados os mesmos métodos de classificação hierárquica e avaliação dos anteriores, porém foi proposto um ajuste no cálculo da entropia, denotando-se esta alteração na fórmula da entropia como uma outra contribuição importante desta tese.

O quinto método LCPNwLC (Local Classifier per Parent Node with Level Classifier), define o critério de parada por meio de um classificador de nível, no qual identifica-se em que nível da árvore deve ser realizada a predição e, por consequência, a parada.

Por fim, no método denominado LCPNwINT (Local Classifier per Parent Node with Intersection), é proposta a combinação ou interseção de dois ou mais métodos de classificação hierárquica.

1.4 Organização do Trabalho

Esta tese está organizada em 7 capítulos. O Capítulo 1 apresenta a introdução, motivação, os objetivos e a contribuição científica no desenvolvimento deste trabalho. O Capítulo 2 descreve os principais conceitos sobre classificação hierárquica, as variações das abordagens para classificação hierárquica e conceitos de Bioinformática, mais especificamente sobre proteínas. O Capítulo 3 apresenta os trabalhos relacionados ao tema desta tese. O Capítulo 4 mostra a configuração experimental, materiais, classificadores base e as métricas de avaliação utilizadas. O Capítulo 5 descreve uma análise extensiva dos métodos do estado da arte, bem como a metodologia utilizada para desenvolvimento e avaliação dos métodos. O Capítulo 6 apresenta os métodos propostos, bem como os resultados computacionais comparados com os métodos base. O Capítulo 7 relata as conclusões e os trabalhos futuros deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

No presente capítulo apresentar-se-ão os principais conceitos sobre a tarefa de Classificação. Na seção 2.1 citam-se os principais conceitos sobre a classificação de dados. Na seção 2.2 são apresentados os fundamentos e as principais abordagens de Classificação Hierárquica. Na seção 2.3 são mostradas as principais métricas de avaliação para Classificação Hierárquica. Por fim, na seção 2.4 descreve os conceitos relacionados a teoria matemática da informação, mais especificamente a teoria da Entropia.

2.1 Classificadores

A mineração de dados é uma etapa no processo de descoberta de conhecimento, que consiste na realização da análise dos dados e na aplicação de algoritmos de descoberta que, sob certas limitações computacionais, produzem um conjunto de padrões (ou modelos) sobre os dados analisados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). São utilizadas tarefas para extrair conhecimento a partir dos dados de experiências anteriores, sem que haja necessidade de criação ou inserção de conhecimento prévio de um especialista. Isto possibilita a descoberta de novas regras e padrões, que dificilmente seriam identificadas por um especialista humano. Dentre estas, destacam-se: o agrupamento, a classificação, a regressão e a associação.

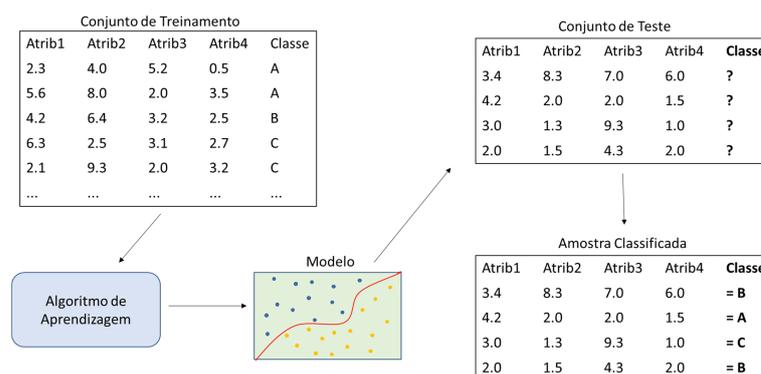
Os algoritmos de aprendizagem de máquina podem ser classificados basicamente em três categorias: aprendizagem supervisionada, não supervisionada e semisupervisionada. Na aprendizagem supervisionada, os classificadores tradicionais utilizam apenas dados rotulados para o treinamento e teste. No entanto, casos rotulados são muitas vezes mais difíceis, caros ou demorados para se obterem. Estes casos exigem esforços de anotações por humanos experientes. Enquanto isso, os dados não rotulados podem ser relativamente fáceis de se obter, mas existem poucas maneiras de usá-los. Neste caso, utiliza-se a aprendizagem não supervisionada para a identificação e agrupamento das instâncias mais similares. Alguns trabalhos na literatura utilizam aprendizagens semisupervisionadas. Nesta abordagem, os algoritmos usam grande quantidade de dados não rotulados, em conjunto com dados rotulados, para a construção de classificadores melhores (ZHU, 2005).

De acordo com Freitas (2013), por meio do conhecimento prévio extraído de uma base de dados de treinamento, procura-se descobrir com qual das classes existentes, novos dados serão rotulados. Esta é uma tarefa de classificação, que está associada com a predição. Dado um conjunto de instâncias, cada instância de dados pertence a uma determinada classe que é indicada pelo valor de um atributo principal. A definição de classificadores,

apresentada por [Costa et al. \(2007\)](#), é uma tarefa de mapear uma classe a partir dos valores de outras variáveis, denominados atributos. Considerando um conjunto de dados constituído por n pares (x_i, y_i) , onde x_i é uma coleção de dados (atributos) e y_i representa sua classe, um algoritmo de classificação deve encontrar uma função que mapeie os atributos às suas respectivas classes corretamente. De uma forma sucinta, classificação é uma tarefa que define uma função f responsável por mapear cada conjunto de atributos x para um dos rótulos de classes em y ([TAN; STEINBACH; KUMAR, 2009](#)).

Na literatura existem diversos algoritmos de aprendizagem de máquina. Esses algoritmos, por meio de um conjunto de treinamento, realizam a etapa de treinamento do classificador, gerando um modelo. Esse modelo (classificador) será responsável por rotular novas instâncias (conjunto de teste) por meio do modelo gerado na etapa de treinamento, conforme exemplo mostrado na Figura 1.

Figura 1 – Etapas de um classificador



Fonte: Adaptado de [Borges \(2012\)](#)

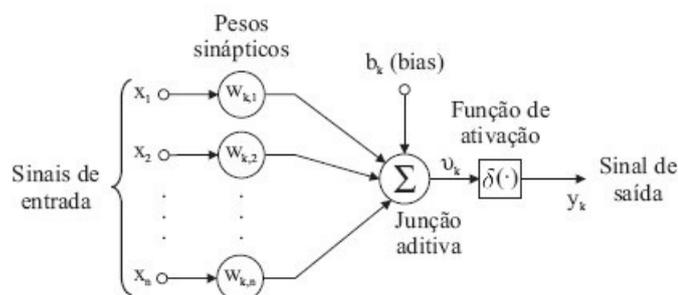
2.1.1 Redes Neurais Artificiais

Uma rede neural é uma rede ou circuito de neurônios, ou, no sentido moderno, uma rede neural artificial, composta por neurônios ou nós artificiais. Assim, uma rede neural é uma rede neural biológica, composta de neurônios biológicos reais, ou uma rede neural artificial, para resolver problemas de inteligência artificial. As conexões do neurônio biológico são modeladas como pesos. Um peso positivo reflete uma conexão excitatória, enquanto valores negativos significam conexões inibitórias. Todas as entradas são modificadas por um peso e somadas. Esta atividade é referida como uma combinação linear. Por fim, uma função de ativação controla a amplitude da saída ([HOPFIELD, 1982](#)). Por exemplo, um intervalo aceitável de saída é geralmente entre 0 e 1, ou pode ser -1 e 1.

De acordo com [Haykin \(2007\)](#), “um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural”. A Figura 2 mostra

um neurônio utilizado como base para o projeto de redes neurais artificiais.

Figura 2 – Modelo não linear de um neurônio



Fonte: Haykin (2007)

Pode-se citar três elementos básicos do modelo neuronal (HAYKIN, 2007):

1. Um conjunto de sinapses, cada uma caracterizada por um peso. Ou seja, um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} .
2. Um somador ou combinador linear, para somar os sinais de entrada, ponderados pelas respectivas sinapses do neurônio;
3. Uma função de ativação para restringir a amplitude da saída de um neurônio. Também pode ser chamada de função restritiva. Ela limita o intervalo permissível de amplitude do sinal de saída a um valor finito.

Neste modelo neuronal inclui-se também um bias, representado por b_k . O bias tem o efeito de aumentar (se positivo) ou diminuir (se negativo) a entrada líquida (cálculo ponderado do valor das entradas e dos pesos associados) da função de ativação. Em termos matemáticos, um neurônio k pode ser definido pela Equação 2.1 e Equação 2.2.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.1)$$

$$y_k = \delta(u_k + b_k) \quad (2.2)$$

Os sinais de entrada são as variáveis x_1, x_2, \dots, x_m ; Os pesos sinápticos são as variáveis $w_{k1}, w_{k2}, \dots, w_{km}$ de um neurônio k ; u_k é a saída do combinador linear devido aos sinais de entrada; bias é representado por b_k ; a função de ativação é $\delta(\cdot)$ e o sinal de saída do neurônio é representado por y_k (HAYKIN, 2007). O uso do bias b_k tem o efeito

de aplicar uma transformação à saída u_k do combinador linear no modelo da Figura 2, conforme mostrado na Equação 2.3.

$$y_k = u_k + b_k \quad (2.3)$$

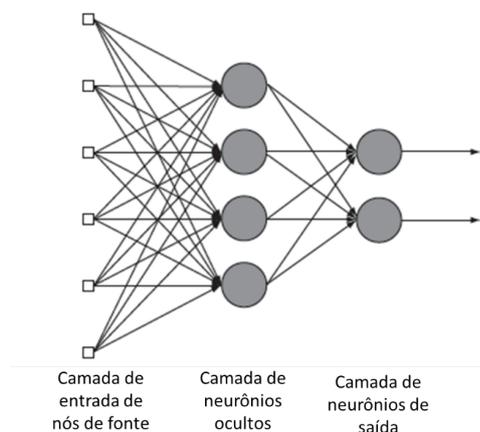
Uma rede neural padrão (NN) consiste em muitos processadores simples conectados, denominados de neurônios, cada um produzindo uma sequência de ativações. Os neurônios de entrada são ativados através de sensores que percebem o ambiente, outros neurônios são ativados através de conexões ponderadas de neurônios previamente ativos (SCHMIDHUBER, 2015).

De acordo com Haykin (2007), pode-se identificar três classes de arquiteturas de rede fundamentalmente diferentes: Redes alimentadas adiante com camada única, Redes alimentadas diretamente com múltiplas camadas e Redes recorrentes.

As redes alimentadas adiante com camada única, possuem uma camada de entrada de nós de fonte que se projeta sobre uma camada de saída de neurônios (nós computacionais), mas não vice-versa. Em outras palavras, esta rede é estritamente do tipo alimentada adiante ou acíclica.

A segunda arquitetura se distingue pela presença de uma ou mais camadas ocultas, cujos nós computacionais são chamados de neurônios ocultos. Adicionando-se uma ou mais camadas ocultas, a rede torna-se capaz de extrair estatísticas de ordem elevada. A Figura 3 mostra uma rede neural de múltiplas camadas alimentadas adiante para o caso de uma única camada oculta.

Figura 3 – Rede neural alimentada adiante ou acíclica com uma camada oculta

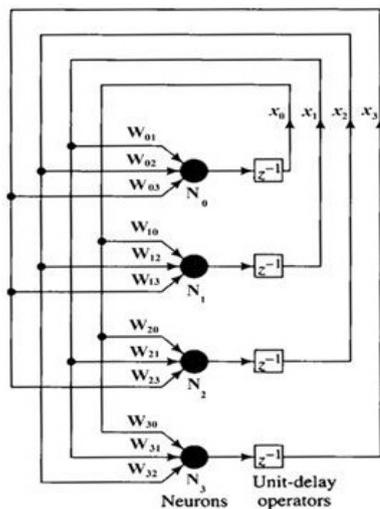


Fonte: Haykin (2007)

Uma rede neural recorrente possui, pelo menos, um laço de realimentação. Por exemplo, essa rede pode consistir de uma única camada de neurônios, cada neurônio

alimenta seu sinal de saída de volta para as entradas de todos os outros neurônios, conforme mostrado na Figura 4. Esta rede não possui neurônios ocultos.

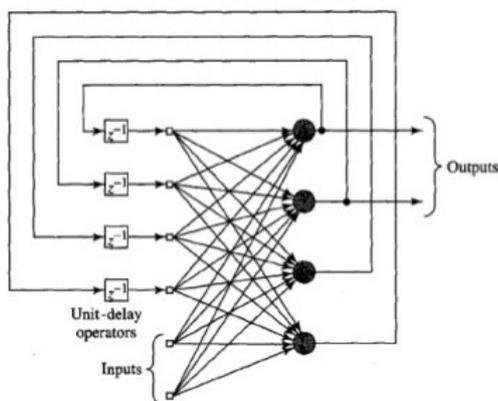
Figura 4 – Rede recorrente sem laços de auto realimentação e sem neurônios ocultos



Fonte: Haykin (2007)

Uma outra classe de redes recorrentes com neurônios ocultos é mostrada na Figura 5. As conexões de realimentação desta rede se originam dos neurônios ocultos, bem como dos neurônios de saída.

Figura 5 – Rede recorrente com neurônios ocultos



Fonte: Haykin (2007)

2.1.2 Aprendizagem Profunda

A aprendizagem profunda (Deep Learning) permite modelos computacionais que são compostos de várias camadas de processamento para aprender representações de dados com múltiplos níveis de abstração. Tais métodos mostraram bons resultados em vários domínios, como por exemplo: no reconhecimento de fala, reconhecimento de objetos visuais, detecção de objetos, diagnóstico de doenças, genômica e entre outros (LECUN; BENGIO; HINTON, 2015).

Além disso, a aprendizagem profunda é um termo utilizado para um conjunto de técnicas de aprendizagem de máquina. Em redes neurais, ele simplesmente denota redes com muitas camadas (em oposição as redes neurais planas). O objetivo principal do uso deste tipo de arquitetura é aprender uma hierarquia de recursos, em que cada camada processa a entrada e provê uma melhor representação dos dados para a próxima camada. De uma certa forma, cada camada adicionada pode aumentar exponencialmente o número de possíveis representações do estado da rede (SERRE et al., 2007).

Segundo LeCun, Bengio e Hinton (2015), a aprendizagem profunda permite descobrir estruturas complexas em grandes conjuntos de dados, usando uma variante do algoritmo de retropropagação (backpropagation) para indicar como uma rede deve alterar os parâmetros internos usados para calcular a representação em cada camada da representação na camada anterior. As redes neurais convolucionais (CNN) trouxeram grandes avanços no processamento de imagens, vídeo, voz e áudio(LECUN; BENGIO; HINTON, 2015).

2.1.2.1 Redes Neurais Convolucionais

As redes convolucionais, ou em inglês, Convolutional Neural Networks (CNN) são um tipo de rede bastante utilizado em Deep Learning. As CNNs fazem parte da família de redes neurais multicamadas, inicialmente desenvolvidas para a utilização de dados bidimensionais, tais como imagens e vídeos. Ainda segundo os autores, essa é a primeira abordagem de Deep Learning que obteve verdadeiramente sucesso (AREL et al., 2010).

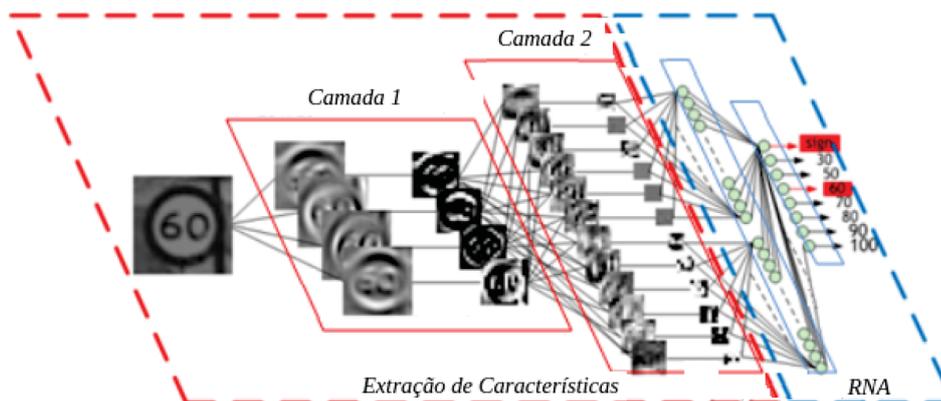
Posteriormente, LeCun et al. (1998) acompanhou essa ideia e treinou essas redes usando gradiente de erro, obtendo e mantendo atualizações de ponta em várias tarefas de visão. A compreensão moderna da fisiologia do sistema visual é consistente com o estilo de processamento encontrado nas redes convolucionais, pelo menos para o reconhecimento rápido de objetos, isto é, sem o benefício da atenção e conexões de realimentação de cima para baixo. Sistemas de visão, baseados em redes neurais convolucionais, proporcionam sistemas com bons desempenhos. Isto foi claramente demonstrado para o reconhecimento de caracteres manuscritos, apresentados no trabalho de LeCun et al. (1998), que tem sido utilizado como referência de aprendizagem de máquina por muitos anos (BENGIO et al.,

2009).

Uma CNN tende a demandar um nível mínimo de pre-processamento quando comparada a outros algoritmos de classificação de imagens. Isso significa que a rede "aprende" os filtros que em um algoritmo tradicional precisariam ser implementados manualmente. Essa independência de um conhecimento a priori e do esforço humano no desenvolvimento de suas funcionalidades básicas pode ser considerada a maior vantagem de sua aplicação (LECUN; BENGIO; HINTON, 2015).

Basicamente, existem duas fases importantes em uma rede convolucional (CNN). A primeira é responsável por extrair características de uma matriz de dados, comumente uma imagem, fornecida como entrada para a rede. Sua arquitetura é composta de camadas convolucionais as quais implementam uma sequência de filtros (kernel), ativações e subamostragens. Com isso mapeia-se os dados de entrada em novos mapas de características (*feature maps*). Ao final desta etapa tem-se um modelo que transforma uma matriz de entrada em um vetor de características. Já na segunda fase, o vetor de características produzido na etapa anterior pelas camadas convolucionais, é aplicado a um algoritmo de aprendizagem de máquina, como uma Rede Neural Artificial ou um SVM, realizando, neste caso, a tarefa de classificação da aprendizagem de máquina (HOCHULI et al., 2019). Um exemplo de arquitetura completa da CNN é mostrado na Figura 6.

Figura 6 – Arquitetura CNN



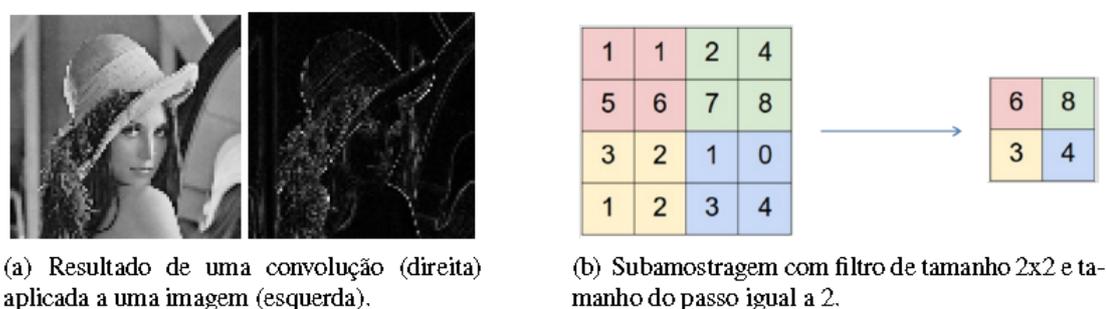
Fonte: Hochuli et al. (2019)

Para entender essa operação de convolução, considere que a imagem de entrada contém 28×28 pixels. Por simplicidade, considere ainda que essas imagens de entrada estão em uma escala de cinza. Em processamento de imagens, uma convolução sobre uma imagem I corresponde a aplicar o produto, entre a matriz de pixels de I e na outra matriz, denominada núcleo da convolução (convolution kernel). Por exemplo, a Figura 7(a) apresenta o resultado da aplicação de um tipo particular de convolução (direita) sobre

uma imagem (esquerda) (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Outra operação importante, utilizada em uma CNN, é a subamostragem. Em processamento de imagens, a subamostragem de uma imagem envolve reduzir a sua resolução, sem, no entanto, alterar significativamente o seu aspecto. No contexto de uma CNN, a subamostragem reduz a dimensionalidade de um mapa de característica, fornecido como entrada e produz outro mapa de característica, uma espécie de resumo do primeiro. Há várias formas de subamostragem aplicáveis a um mapa de características: selecionar o valor máximo (*maxpooling*), a média (*averagepooling*) ou a norma do conjunto (*L2 – pooling*), entre outras (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Figura 7 – Exemplo de funcionamento de uma CNN



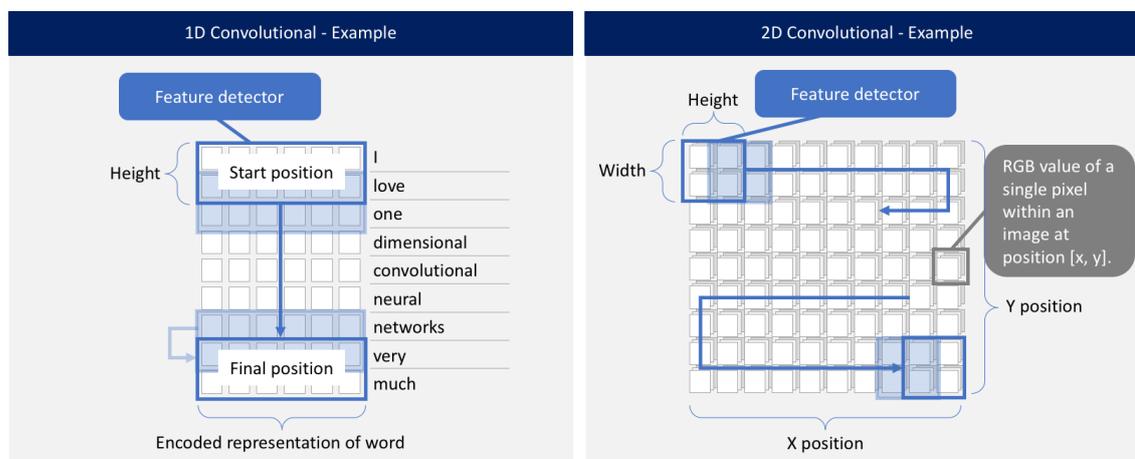
Fonte: Albawi, Mohammed e Al-Zawi (2017)

A ilustração (b) da Figura 7, que corresponde às ativações das unidades de um mapa de característica de tamanho 4x4, apresenta outro mapa de característica, resultante da operação de subamostragem com o uso de filtros de tamanho igual a 2x2 e tamanho do passo igual a 2.

As CNNs compartilham as mesmas características e seguem a mesma abordagem, independentemente de ser 1D, 2D ou 3D. A principal diferença é a dimensionalidade dos dados de entrada e como o detector de recursos (ou filtro) desliza pelos dados. A figura 8 apresenta um exemplo com a diferença entre as CNN's 1D e 2D.

No exemplo da Figura 8, a CNN 1D é aplicado ao processamento de linguagem natural. Este exemplo possui uma frase composta por 9 palavras e cada palavra é um vetor que define uma palavra como uma representação de baixa dimensão. O filtro aplicado (kernel) sempre cobrirá a palavra inteira e a altura determina quantas palavras são consideradas para o treinamento na etapa de extração de características. Neste exemplo a altura é dois e o filtro (kernel) deslizará 8 vezes nos dados de entrada com passo de 1. Já no exemplo da CNN 2D aplicado na visão computacional, cada pixel na imagem é representado por sua posição x e y, além de três valores (RGB). O filtro (kernel) tem uma dimensão de 2 x 2 neste exemplo. O filtro (kernel) agora deslizará em duas posições

Figura 8 – Diferença entre as CNN's 1D e 2D



Fonte: Ackermann (2019)

(horizontal e vertical) pela imagem (ACKERMANN, 2019).

De acordo com LeCun, Bengio e Hinton (2015), a aprendizagem profunda terá ainda mais sucesso, pois requer pouca engenharia de representação, ademais ela pode utilizar-se, de forma eficiente, do aumento dos recursos computacionais e dos dados disponíveis. Novas arquiteturas e algoritmos de aprendizagem, que estão sendo desenvolvidos atualmente para redes neurais profundas, só acelerarão esse progresso.

2.2 Classificação Hierárquica

Na literatura a maioria dos problemas de classificação envolve a classificação plana, onde cada exemplo é atribuído a uma classe a partir de um conjunto finito de classes. No entanto, existem problemas de classificação mais complexos, nos quais as classes a serem preditas estão hierarquicamente relacionadas (FREITAS; CARVALHO, 2007).

Estas hierarquias (taxonomias) são estruturas de dados comumente usadas para organizar um grande volume de informações de uma base de dados em diversos domínios. Como por exemplo, Funcat para organização de sequências genômicas (RUEPP et al., 2004), ImageNet para organização de imagens (DENG et al., 2009) e o Diretório Web do Yahoo para organização de documentos. Dada a hierarquia de classes, a classificação hierárquica (HC) trata os problemas com modelos de aprendizagem, utilizando (ou ignorando) a estrutura hierárquica para classificar automaticamente instâncias de teste não rotuladas (exemplos) em classes relevantes (NAIK; RANGWALA, 2016).

Nestes problemas, as classes estão dispostas em uma taxonomia de árvore (Tree) ou de um Grafo Acíclico Dirigido (DAG - do inglês *Directed Acyclic Graph*), mostrados

Figura 9 – Exemplo de uma estrutura hierárquica de classes baseada em árvore

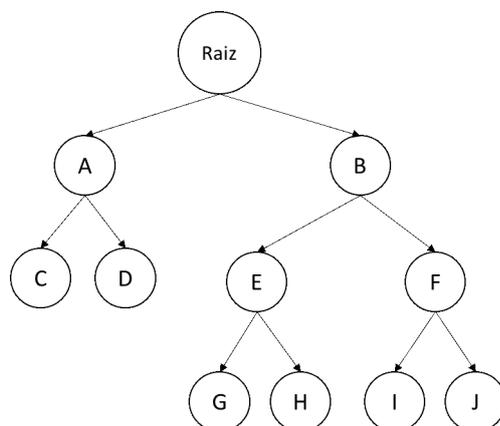
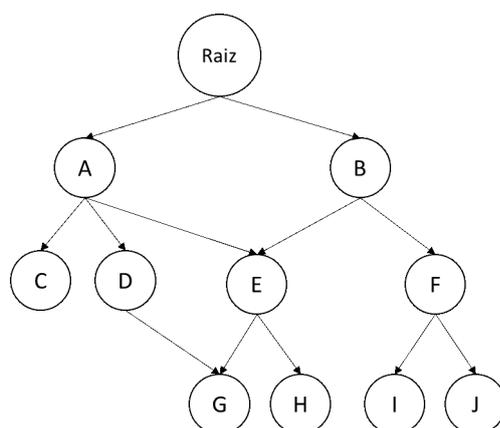


Figura 10 – Exemplo de uma estrutura hierárquica de classes em DAG

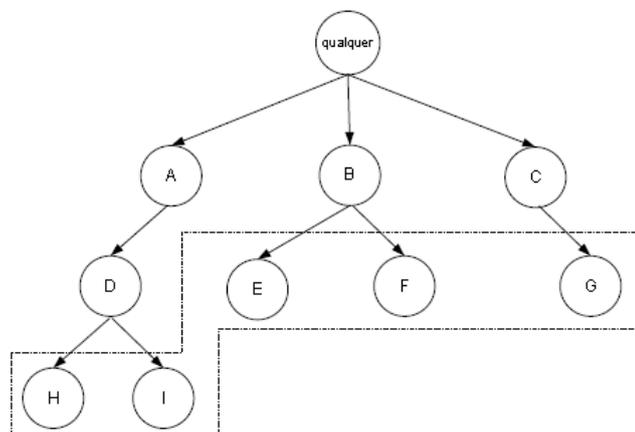


respectivamente nas Figura 9 e Figura 10. Estes problemas são conhecidos na literatura de Aprendizagem de Máquina como problemas de classificação hierárquica, bem como também são denominados na literatura como classificação estruturada (SILLA; FREITAS, 2011b). A diferença principal da disposição da taxonomia, ou seja, entre estrutura de árvore e de DAG, consiste que nesta, cada nó pode ter um ou mais nó pai. Já na estrutura de árvore, um nó filho possui somente um nó pai.

Um outro fator que distingue a classificação hierárquica da classificação convencional, consubstancia-se em como a predição das classes é realizada na hierarquia. As duas categorias são: predição obrigatória em nós-folha e predição opcional em nós-folha (SILLA; FREITAS, 2011b). Na predição obrigatória em nós-folha, todos os exemplos classificados devem ser associados com classes representadas por nós-folha. Neste caso, ao predizer uma classe que está em um nível mais específico (nó-folha), estará também predizendo todas as classes ancestrais (classes que estão nos níveis acima) do nó-folha predito. Conforme mostrado na Figura 11, somente as classes denominadas de H, I, E, F, e G devem ser

preditas (SILLA; FREITAS, 2011b).

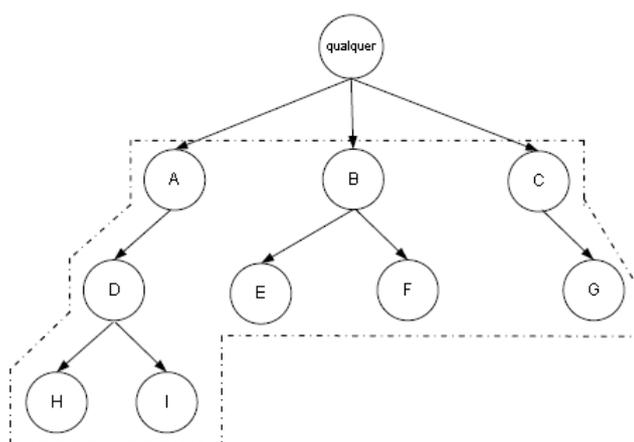
Figura 11 – Exemplo de predição em nós-folha obrigatória



Fonte: Silla e Freitas (2011b)

Já na predição opcional em nós-folha qualquer nó da hierarquia pode ser predito. Ou seja, não existe a obrigatoriedade de que a predição ocorra em nós-folha. “Dessa forma, os exemplos podem ser associados a classes que são representadas por qualquer nó interno da hierarquia de classes e seus ancestrais” (SILLA; FREITAS, 2011b). A Figura 12 mostra que qualquer nó da hierarquia pode ser predito nesta categoria.

Figura 12 – Exemplo de predição em nós-folha não obrigatória



Fonte: Silla e Freitas (2011b)

Uma padronização, apresentada por SILLA; FREITAS, mostra que problemas de classificação hierárquica podem ser descritos pela tupla (Υ, Ψ, Ω) . Onde $\Upsilon \in \{\text{Árvore}, \text{DAG}\}$, define a disposição da taxonomia de classe (Árvore ou DAG, respectivamente). E $\Psi \in$

$\{MPL, SPL\}$ indica, respectivamente, se pelo menos uma instância tem caminhos múltiplos de rótulos (MPL – Multi Path of Labels) ou se cada instância tem apenas um caminho único de rótulos (SPL – Single Path of Labels) na taxonomia de classe. Esta distinção é análoga aos problemas de classificação com rótulo único e multirrótulo na classificação "plana". $\Omega \in \{PD, FD\}$ mostra se pelo menos uma instância tem rotulagem de profundidade parcial (PD – Partial Depth Labeling) ou se todas as instâncias devem possuir a rotulagem de profundidade total (FD - Full Depth Labeling). A rotulagem FD requer que a classe mais específica de todas as instâncias seja um nó-folha, enquanto na rotulagem PD uma instância pode ter um nó não-folha como sua classe mais específica.

A classificação hierárquica pode ser classificada como: local ou global. Os algoritmos de classificação hierárquica local treinam vários algoritmos de classificação tradicionais (planos), usando alguma abordagem para transformar o problema de classificação hierárquica em uma classificação plana durante a fase de treino. Após a etapa de treinamento, a estrutura hierárquica das classificações planas são recuperadas na etapa de predição. Já na classificação hierárquica global é construído um único modelo de classificação para toda a hierarquia e não exige qualquer transformação do problema (SILLA; FREITAS, 2011b).

2.2.1 Classificação Hierárquica Plana

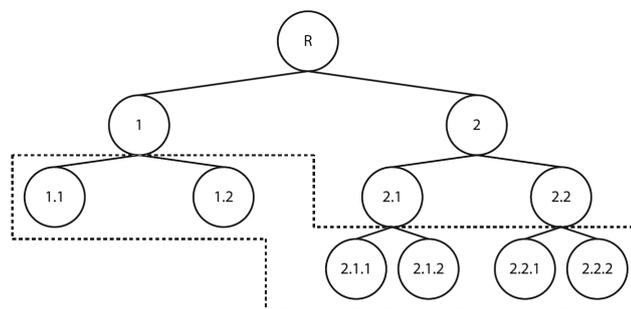
A abordagem de classificação plana, que é a mais simples de lidar com problemas de classificação hierárquica, consiste em ignorar completamente a hierarquia de classes, geralmente predizendo apenas as classes folhas. Esta abordagem se comporta como um algoritmo de classificação tradicional durante o treinamento e testes. No entanto, ele fornece uma solução indireta para o problema da classificação hierárquica, porque, quando uma classe folha é atribuída a um determinado exemplo, pode-se considerar que todas as suas classes ancestrais também são atribuídas implicitamente a essa instância (SILLA; FREITAS, 2011b). A Figura 13 ilustra esta abordagem de classificação plana usando um algoritmo de classificação multiclasse para realizar a predição de um nó-filho.

No entanto, esta abordagem muito simples tem a séria desvantagem de ter que construir um classificador discriminando um grande número de classes (todas as classes de folhas), sem explorar informações sobre relações entre pais e filhos presentes na hierarquia de classes.

Várias abordagens que reestruturam a hierarquia foram desenvolvidas no passado. Level flattening, proposta por Wang e Lu (2010), é uma das abordagens utilizadas em trabalhos anteriores de modificação hierárquica, onde alguns dos níveis são achatados (removidos) da hierarquia original antes de aprender os modelos (NAIK; RANGWALA, 2016).

Os autores Babbar et al. (2013) propuseram uma estratégia baseada na margem

Figura 13 – Exemplo da abordagem classificação hierárquica plana



Fonte: [Silla e Freitas \(2011b\)](#)

máxima para a modificação da hierarquia. Esse método remove seletivamente alguns dos nós inconsistentes na hierarquia com base nas margens ao invés de remover os níveis completos. Neste procedimento foi considerada uma perspectiva global da hierarquia para seleção de limiar para identificar o conjunto inconsistente de nós. Esta abordagem se baseia em toda a hierarquia durante o processo de aprendizagem e classificação, o qual impede um achatamento excessivo dos nós com a realização apenas nas decisões locais.

2.2.2 Classificação Hierárquica Local

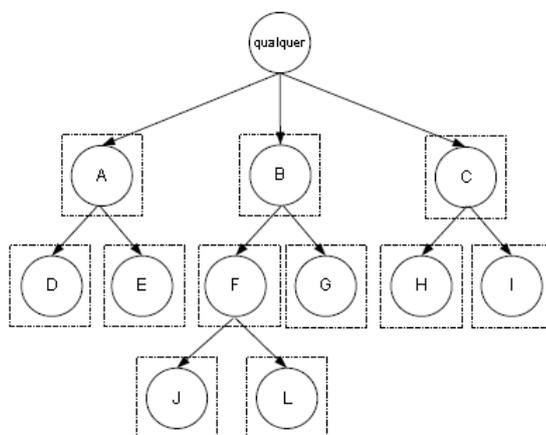
A forma mais popular para explorar a estrutura hierárquica, em problemas binários ou multiclases, é a abordagem local, também chamada na literatura de abordagem de cima para baixo (top-down). A fase de treinamento pode ser realizada de três maneiras diferentes: (1) usando classificador binário por nó, exceto o nó raiz, (2) usando classificador multiclasse por nó pai, e (3) usando classificador multiclasse por nível. Na fase de classificação, o primeiro classificador decide em qual classe o exemplo pertence e passa o exemplo para o classificador do nível imediato. Este procedimento é repetido até que o exemplo atinja um nó folha ou um nó interno, dependendo da abordagem do algoritmo ([HERNANDEZ; SUCAR; MORALES, 2013](#)) ([SILLA; FREITAS, 2011b](#)).

2.2.2.1 Classificação Hierárquica Local por Nó

A abordagem de classificação hierárquica local por nó (LCN), consiste em treinar um classificador binário para cada nó da hierarquia de classes, exceto o nó raiz ([SILLA; FREITAS, 2011b](#)). A Figura 14 representa a abordagem de classificação hierárquica local por nó. Os círculos representam as classes e os quadrados representam classificadores binários.

Conforme citado por [Silla e Freitas \(2011b\)](#), existem diferentes maneiras de definir o conjunto de exemplos positivos e negativos para o treinamento dos classificadores

Figura 14 – Exemplo de classificação hierárquica local por nó



Fonte: [Silla e Freitas \(2011b\)](#)

binários. A política de seleção de exemplos, apresentado por [Fagni e Sebastiani \(2007\)](#), é o método mais comum utilizado na literatura, no qual utiliza como exemplos positivos para treinamento todos os exemplos pertencentes ao nó de classe atual e todos os exemplos das suas classes descendentes, embora outras políticas de seleção de exemplos possam ser utilizados. Além disso, essa é uma das abordagens mais utilizadas na literatura, como por exemplo em [Koller e Sahami \(1997\)](#), [Barutcuoglu e DeCoro \(2006\)](#), [Dumais e Chen \(2000\)](#), entre outros.

Durante a fase de testes, independentemente de como exemplos positivos e negativos foram definidos, a saída de cada classificador binário será uma predição, a qual indicará se um determinado exemplo de teste pertence ou não a classe predita pelo classificador. Uma vantagem desta abordagem é que de imediato abrange o problema de múltiplos caminhos, pelo fato de possibilitar a predição de várias classes por nível. Entretanto, para o caso de problemas de caminho único, pode-se adotar a maior probabilidade de acordo com as predições inferidas para cada classe da hierarquia em um determinado nível ([SILLA; FREITAS, 2011b](#)).

No trabalho de [Merschmann e Freitas \(2013\)](#) são utilizados classificadores binários na etapa de treinamento, por meio dos algoritmos Naive Bayes, para cada nó da hierarquia. Este método foi denominado de ELHNB (Extended Local Hierarchical Naive Bayes). Já na fase de teste uma nova instância é avaliada de forma global, a qual é inferida em todos os classificadores da hierarquia. Para todos os caminhos possíveis da hierarquia a média geométrica das probabilidades são calculadas. O caminho escolhido como resultado será o que tiver maior média geométrica.

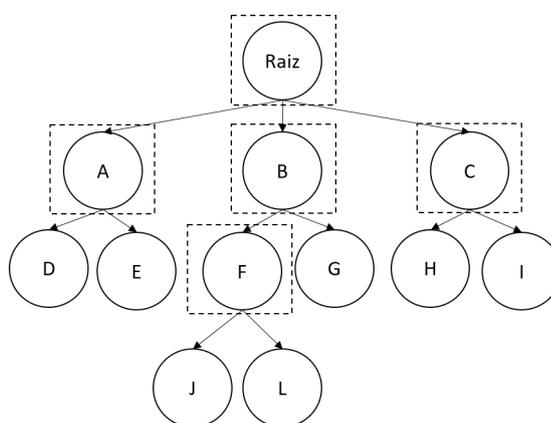
Os autores [Fabris e Freitas \(2014\)](#) propuseram uma extensão do método ELHNB.

Porém, ao invés de realizar a avaliação global e escolher o caminho com a maior média geométrica, os autores adotaram a estratégia Top-Down. Os resultados alcançados foram equivalentes ao método ELHNB.

2.2.2.2 Classificação Hierárquica Local por Nó Pai

A classificação hierárquica local por nó pai (LCPN) é a abordagem em que, para cada nó-pai da hierarquia, um classificador multiclasse é treinado para distinguir entre seus nós-filho (SILLA; FREITAS, 2011b). A Figura 15 mostra um exemplo da classificação local por nó pai, onde os círculos representam as classes e os quadrados representam classificadores multiclasse para a realização das predições entre seus nós-filhos.

Figura 15 – Exemplo de classificação hierárquica local por nó pai



Na etapa de treinamento dos classificadores desta abordagem a política de seleção de exemplos "irmãos", bem como a política de "irmãos exclusivos", ambas citadas no trabalho de Silla e Freitas (2011b), podem ser utilizadas.

Durante a fase de teste, essa abordagem geralmente é associada à abordagem de classificação Top-Down, porém outras abordagens podem ser utilizadas para prever uma determinada classe da hierarquia. Considerando a abordagem para classificação de uma determinada classe como Top-Down e o mesmo exemplo de árvore de classes da Figura 15 e suponha que o classificador de primeiro nível atribua o exemplo à classe B. O classificador de segundo nível, que foi treinado apenas com os filhos do nó da classe B, neste caso F e G, fará sua atribuição de classe (e assim por diante, se houver classificadores de nível mais profundo disponíveis), evitando assim o problema de fazer predições inconsistentes e respeitar as restrições naturais da associação de classe.

Normalmente, na abordagem LCPN o mesmo algoritmo de classificação é utilizado ao longo de toda a hierarquia de classe. Todavia, no trabalho de Secker et al. (2007), os autores propuseram um método de classificador seletivo. Este método treina diversos classificadores para cada nó-pai e tem por objetivo melhorar os resultados em comparação com

a utilização de somente um classificador por nó-pai. Neste caso, diferentes classificadores são treinados usando o conjunto de treinamento e na sequência são avaliados no conjunto de validação. O classificador escolhido para cada classe nó-pai é aquele com a maior precisão da classificação no conjunto de validação. Uma melhoria em relação à abordagem do classificador seletivo foi proposta por [Holden e Freitas \(2008\)](#), onde um algoritmo de otimização de colônia de formigas foi usado para realizar a seleção do classificador. Outra melhoria em relação à abordagem do classificador seletivo foi proposta por [Silla e Freitas \(2009b\)](#), a partir dos quais, o melhor classificador e o melhor tipo de representação de exemplo são selecionados para cada classificador do nó pai.

2.2.2.3 Classificação Hierárquica Local por Nível

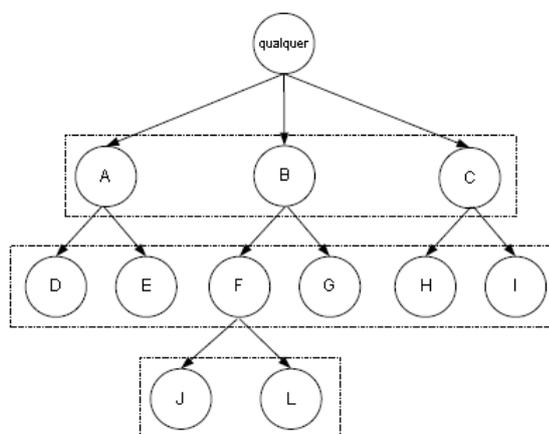
Nesta abordagem um ou mais classificadores são construídos em cada nível da árvore de classes e cada classificador funciona como um classificador plano nesse nível. Uma instância será primeiramente classificada pelo classificador do primeiro nível (mais genérico) em uma ou mais classes de nível inferior. Após, será então classificada pelo classificador do nível inferior até atingir uma classe final, que poderá ser um nó-folha ou um nó interno da hierarquia, dependendo do método adotado de critério de parada ([SUN; LIM, 2001](#)).

A Figura 16 mostra um exemplo da classificação hierárquica local por nível, na qual três classificadores multiclasse são treinados, um classificador para cada nível de classe, para prever uma ou mais classes (dependendo se o problema é de caminho único ou múltiplos caminhos) em seu nível de classe correspondente. A política de seleção de exemplos, para o conjunto de treinamento, podem ser executada da mesma forma como na abordagem com classificador local por nó-pai ([SILLA; FREITAS, 2011b](#)).

Uma maneira possível (embora muito ingênua) de classificar exemplos de teste usando classificadores treinados por essa abordagem é a seguinte. Quando um novo exemplo de teste é apresentado ao classificador, obtém-se a saída de todos os classificadores (um classificador por nível) e utiliza-se essas informações como a classificação final. A principal desvantagem dessa abordagem de predição de classe está sendo propensa a inconsistência na associação de classe. De acordo com a Figura 16, ao treinar classificadores diferentes para cada nível da hierarquia, é possível obter resultados como a classe B no primeiro nível, a classe D no segundo nível e a classe L no terceiro nível, gerando inconsistência. Portanto, se essa abordagem for usada, ela deverá ser complementada por um método de pós-processamento que tente corrigir a inconsistência da predição ([SILLA; FREITAS, 2011b](#)).

Algumas técnicas para correção de inconsistências desta abordagem são encontradas na literatura. Por exemplo, no trabalho de [Freitas e Carvalho \(2007\)](#) é utilizada uma estratégia Top-Down, onde aplica-se a instância no primeiro classificador (Nível 1) e

Figura 16 – Exemplo de classificação hierárquica local por nível



Fonte: Silla e Freitas (2011b)

verifica-se qual classe tem maior probabilidade. No próximo nível são mantidas somente as classes pertencentes à classe com maior probabilidade do nível anterior e o mesmo processo se repete até que chegue em um nó-folha.

Outra abordagem para tentar corrigir inconsistências foi trazida por Paes, Plastino e Freitas (2012). Nesta, realiza-se uma avaliação global, na qual, por meio da predição de todas as probabilidades das classes inferidas pelos classificadores, em todos os níveis, é realizada a correção do caminho. Este método foi denominado de SWV (*Sum of Weighted Votes*), e por meio da soma das probabilidades de todas as classes do caminho, realiza-se a predição no caminho que contém a maior soma das probabilidades.

No trabalho realizado por Cerri, Barros e Carvalho (2014) foi proposto um método de classificação baseado em redes neurais locais, que treina incrementalmente uma rede neural perceptron de múltiplas camadas para cada nível da hierarquia. As predições feitas por uma rede neural em um determinado nível, são usadas como entradas para a rede neural responsável pela predição no próximo nível.

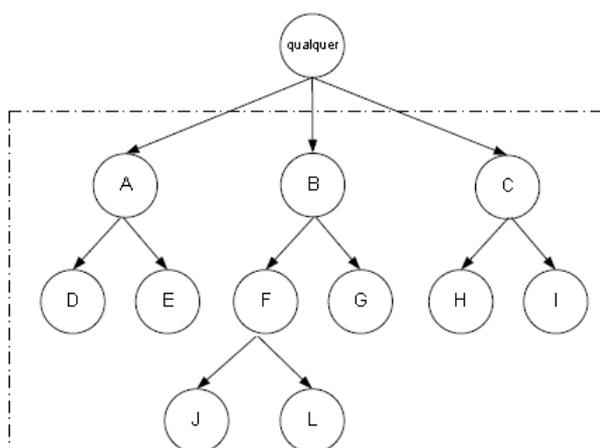
2.2.2.4 Classificação Hierárquica Global

Na classificação hierárquica global, também chamada na literatura como *big-bang*, apenas um único modelo é usado no processo de classificação. Dada uma instância, o classificador irá atribuí-la a uma ou mais classes na hierarquia de classes. As classes atribuídas podem ser nós internos ou nós-folhas, dependendo da estratégia utilizada para os problemas hierárquicos suportados pelo método (SILLA; FREITAS, 2011b).

Desenvolver um modelo global único para todas as classes, possui a vantagem de que o tamanho total do modelo de classificação global torna-se menor, em comparação

com o tamanho de todos modelos locais desenvolvidos por qualquer uma das abordagens de classificadores hierárquicos locais (SUN; LIM, 2001). A Figura 17 mostra um exemplo de classificação hierárquica global.

Figura 17 – Exemplo de classificação hierárquica global



Fonte: Silla e Freitas (2011b)

Comparada com a abordagem baseada em níveis, a abordagem global só pode usar as informações transportadas pela estrutura de classes durante a fase de treinamento, mas não na fase de classificação. Outra questão na abordagem global, é que o classificador construído pode não ser flexível o suficiente para atender a mudanças na estrutura de classes. Ou seja, o classificador precisa ser treinado novamente uma vez que a estrutura da classe seja alterada (SUN; LIM, 2001).

No trabalho proposto por Vens et al. (2008), foi desenvolvido um modelo de classificação hierárquica global utilizando indução de árvores de decisão. Esse algoritmo de indução da árvore de decisão é baseado no framework Predictive Clustering Trees (PCT). Os autores abordam três técnicas de classificação: a classificação de uma única classe SC (Single Label Classification), a classificação hierárquica de uma única classe HSC (Hierarchical Single Label Classification) e a classificação hierárquica de múltiplas classes HMC (Hierarchical Multilabel Classification). Estas técnicas estão implementadas no sistema CLUS e consistem em gerar uma única árvore de decisão para toda a hierarquia.

Os autores Silla e Freitas (2009a) propuseram métodos de classificação hierárquica baseados em Naive Bayes. Dois métodos são globais, denominados de GMNB e GMNBwU. A diferença entre eles é que o segundo tolera mais os erros de classificação em níveis maiores da árvore do que erros cometidos em níveis menores. Esta tolerância é inferida por meio de uma medida de utilidade (usefulness). Da mesma forma que o método GMNBwU, o classificador LMNBwU penaliza menos os erros em níveis maiores da hierarquia. Estes

métodos foram avaliados em oito bases de dados, mais especificamente, quatro bases de proteínas GPCR e quatro bases de enzimas e comparados entre si. Os resultados mostraram que o método GMNBwU foi estatisticamente melhor em comparação com os outros métodos, entretanto, o GMNBwU teve maior revocação hierárquica; já o método GMNB obteve maior precisão hierárquica.

No trabalho de [Alves, Delgado e Freitas \(2010\)](#), apresentou-se um método global usando Artificial Imune Systems (AIS), para a geração de regras do Classificador Hierárquico Multirrótulo. O procedimento de Cobertura Sequencial, iterativamente, chama o procedimento Evolução da Regra até que cada instância de treinamento (antígenos) seja coberta pelas regras descobertas. O procedimento Evolução da Regra, evolui regras de classificação (anticorpos) que são empregados para classificar as instâncias. O melhor anticorpo é adicionado ao conjunto de regras descobertas.

No trabalho de [Borges e Nievola \(2012\)](#) dois algoritmos foram propostos para a construção de classificadores hierárquicos globais aplicados na estrutura de DAG. O primeiro método é um classificador multirrótulo hierárquico baseado em uma rede neural competitiva, chamado de MHC-CNN (Multi-label Hierarchical Classification using a Competitive Neural Network). Uma segunda versão do classificador MHC-CNN, denominada de MHC-ES é desenvolvida e utiliza como técnica de aprendizagem da rede neural a estratégia evolucionária. A diferença entre os dois está na etapa de treinamento da rede.

2.3 Métricas de Avaliação de Classificadores Hierárquicos

Na classificação plana, a avaliação do desempenho de um modelo por meio das métricas tradicionais, é baseada nas contagens de registros de testes, corretamente e incorretamente, preditas pelo modelo. A taxa de acerto (TA) e a taxa de erro (TE) de um classificador são obtidas por meio das contagens de valores de Verdadeiro Positivo (VP), Falso Positivo (FP), Verdadeiro Negativo (VN) e Falso Negativo (FN) ([SUN; LIM, 2001](#)).

Existem outras medidas, além da taxa de acerto e da taxa de erro, que são utilizadas para avaliar a qualidade de um classificador como: Precisão, Revocação, Especificidade e Medida-F ([OLSON; DELEN, 2008](#)).

A maioria dos pesquisadores avaliam sistemas de classificação hierárquicos baseados em medidas "planas". Como por exemplo: acurácia e erro, bem como precisão e revocação. No entanto, estas medidas não são adequadas para categorização hierárquica, uma vez que não diferenciam entre os diversos tipos de erros de classificação. Intuitivamente, uma classificação errada, para um irmão ou um nó pai de uma classe correta é muito melhor do que a classificação errada para um nó mais distante ([KIRITCHENKO et al., 2006](#)).

Em uma classificação hierárquica todas as classes devem ser consideradas. Além

disso, as classes estão relacionadas com as ligações entre pai, filho e irmão. Por exemplo, na classificação de textos, duas classes podem ser semelhantes quando compartilham muitos documentos comuns; ou seja, as categorias de Programação e Engenharia de Software podem ter várias características em comuns, que permitem que os documentos sejam classificados em ambos. Portanto, se um documento não é classificado para a categoria correta, deve-se considerar o grau de erro da classificação. Em outras palavras, classificar erroneamente um documento em uma categoria pai ou filho é considerado melhor do que classificá-lo em categorias que estão longe da categoria correta (SUN; LIM, 2001).

Foram propostas medidas específicas para classificadores hierárquicos, como alternativas às medidas de avaliação convencionais para problemas de classificação. Como por exemplo: medidas de avaliação baseadas na hierarquia e medidas de avaliação à distância. Enquanto as medidas baseadas na hierarquia são fundamentadas apenas pela estrutura de classes hierárquicas (suas subclasses e superclasses), as medidas baseadas na distância também consideram a distância entre as classes preditas e verdadeiras na estrutura da hierarquia (CERRI et al., 2015).

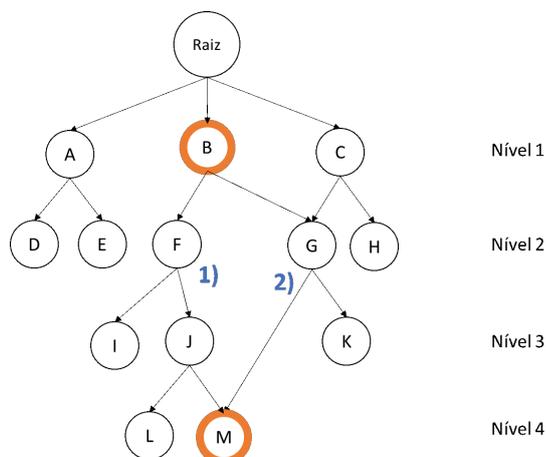
2.3.1 Métricas de Avaliação Baseada em Distância Independente da profundidade

Na métrica de avaliação baseada em distância independente da profundidade, uma distância entre uma classe verdadeira e a classe predita é atribuída, distância (x, y) . Esta distância é o comprimento do caminho do nó x para o nó y em uma árvore hierárquica. Esta medida de distância provê penalidades diferentes para a classificação errada em uma classe vizinha ou distante. No entanto, tem algumas desvantagens. A primeira desvantagem desta medida, mostra que não é facilmente extensível às hierarquias do DAG, na qual podem existir vários caminhos entre duas classes (KIRITCHENKO et al., 2006). Conforme verifica-se na Figura 18, há dois caminhos entre o nó B e o nó M. No caminho 1, a distância entre o nó verdadeiro e o nó predito é três. Já no caminho 2, a distância entre o nó verdadeiro e o nó predito é 2.

Neste caso, algum critério deve ser utilizado para selecionar o caminho entre o nó de classe verdadeira e o nó de classe predita. Um critério que pode ser adotado é o do menor caminho. Seguindo este critério no exemplo abaixo, o caminho selecionado seria o 2 cuja distância é igual a dois.

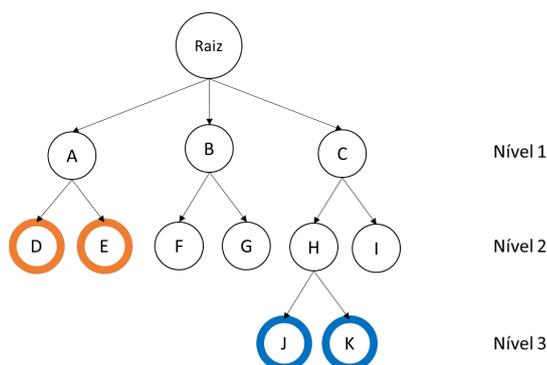
A segunda desvantagem é que esta medida independe da profundidade. Por exemplo, um erro de classificação entre dois irmãos de uma classe de nível superior e um erro de classificação entre irmãos situados na profundidade do 10º nível são considerados o mesmo tipo de erro (distância de 2). No entanto, um erro no nível 10 pode ser muito menos prejudicial do que um erro no nível superior (KIRITCHENKO et al., 2006). Considerando

Figura 18 – Exemplo de DAG independente da profundidade



o exemplo da árvore da Figura 19.

Figura 19 – Exemplo de Medida independente da profundidade em árvore



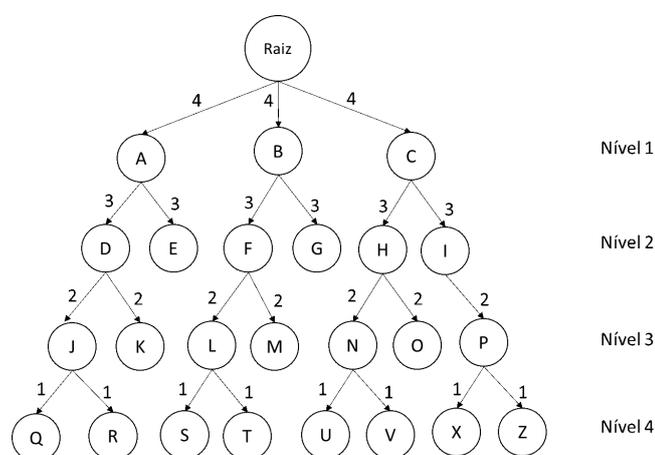
Na instância 1 (laranja) - a classe verdadeira é a letra D e a classe predita foi a letra E. Já na instância 2 (azul) - a classe verdadeira é a letra J e a classe predita foi a letra K. Nestes casos, observa-se que a distâncias entre a classe D até E, bem como da classe J até K na hierarquia nessa estrutura é igual a 2. Porém, um erro no nível 3 poderá ser melhor do que um erro no nível 2 em determinados domínios.

2.3.2 Métricas de avaliação baseada em distância dependente da profundidade

Esta abordagem, proposta por [Blockeel et al. \(2002\)](#), tem como objetivo, considerar o nível das classes para o cálculo da medida baseada em distância. Nesta medida, a distância ponderada, calculada entre a classe verdadeira e a classe predita, é avaliada de acordo com o número de ligação entre as duas classes e a profundidade em que se encontram ambas as classes na hierarquia. Esta abordagem por sua vez, fornece penalizações maiores para erros de classificação nos níveis superiores em relação aos erros de classificação nos

níveis mais profundos na hierarquia, pelo fato de que é mais simples classificar classes mais genéricas (nível superior) do que classes mais específicas. Para cada nível da hierarquia um peso é atribuído nas ligações entre as classes, conforme mostrado na Figura 20. Nesta atribuição de pesos, as ligações que são mais profundas recebem um peso menor do que as ligações dos níveis superiores. Os pesos podem ser ajustados de acordo com o problema em questão. Na abordagem utilizada por [Blockeel et al. \(2002\)](#) e [Holden e Freitas \(2009\)](#), os pesos foram diminuídos exponencialmente na medida em que o nível de profundidade aumentava.

Figura 20 – Exemplo de Medida independente da profundidade em árvore



Ao levar em consideração algumas situações esta abordagem apresentará alguns problemas. Um deles é quando esta árvore apresenta um grande desbalanceamento. Neste caso, um erro de classificação onde um nó está mais distante da raiz, terá uma penalização maior em relação ao nó que está mais próximo da raiz. [Lord et al. \(2003\)](#) mostra outro problema, quando duas classes estão em diferentes subárvores ligadas pelo nó-raiz: uma classe no nível superior, poderá ser mais específica do que uma classe em outro ramo da árvore de profundidade maior. Ou seja, a associação de pesos às ligações, considerando apenas a profundidade e não tendo em vista a informação associada a cada classe, em alguns casos pode apresentar problemas. Outro ponto a ser avaliado é que nos problemas estruturados por DAG, pode haver múltiplos caminhos, sendo necessário adotar critérios para determinar a distância ponderada adequada de acordo com o caminho percorrido.

2.3.3 Métricas de avaliação baseada na Descendência

A medida baseada na descendência, apresentada por [Ipeirotis, Gravano e Sahami \(2001\)](#), utiliza o conceito de classes descendentes no processo de avaliação, considerando as subárvores enraizadas na classe predita e da classe verdadeira. Ou seja, cada subárvore é formada pela classe em questão e por suas classes descendentes. O cálculo das medidas

de precisão hierárquica (hP) e revocação hierárquica (hR) é calculada pela intersecção dessas subárvores. A medida de precisão hierárquica é calculada com o número de classes pertencentes à intersecção das subárvores (C_v – Classe verdadeira e C_p – Classe predita), dividido pelo número de classes pertencentes à subárvore enraizada na classe predita, como mostrado na Equação 2.4.

$$hP = \frac{|Desc(C_v) \cap Desc(C_p)|}{|Desc(C_p)|} \quad (2.4)$$

Nessa equação, $Desc(C)$ representa o conjunto de classes contidas na subárvore, cuja raiz é a classe C , incluindo C . O cálculo da revocação hierárquica, conforme mostrado na Equação 2.5, se perfaz pela divisão do número de classes pertencentes à intersecção das subárvores pelo número de classes pertencentes à subárvore enraizada na classe verdadeira.

$$hR = \frac{|Desc(C_v) \cap Desc(C_p)|}{|Desc(C_v)|} \quad (2.5)$$

Após o cálculo da predição e da revocação hierárquica, o cálculo da Medida-F Hierárquica pode ser realizado, conforme mostrado na Equação 2.6. O valor de β geralmente é 1.

$$Medida_hF = \frac{(\beta^2 + 1) * hP * hR}{hP + hR} \quad (2.6)$$

Essa medida apresenta um problema, pois ela assume que a classe predita é uma classe descendente ou uma superclasse da classe verdadeira. Quando essas classes estão no mesmo nível, o valor reportado pelas medidas é igual a 0. Pois a intersecção entre as respectivas subárvores é nula.

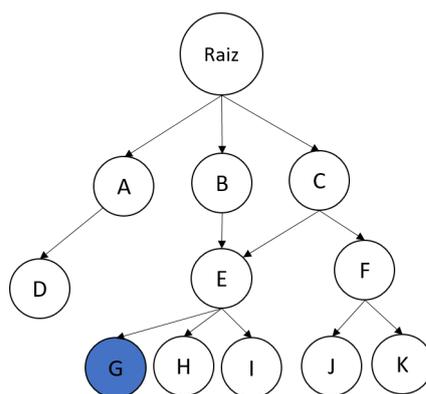
2.3.4 Métricas de avaliação baseada em Ancestralidade

Um método de avaliação proposto por Kiritchenko et al. (2006) é similar ao método baseado na descendência. Porém, o mesmo utiliza a ancestralidade (classes ancestrais) em vez de utilizar classes descendentes para os cálculos de hP, hR e Medida_hF. Segundo Kiritchenko et al. (2006), para expressar as propriedades desejadas de uma medida de avaliação hierárquica, formularam-se os seguintes requisitos, conforme Figura 21:

1. A medida provê crédito a uma classificação parcialmente correta. Por exemplo, contendo como classe correta a classe G , a classificação incorreta no nó I deve ser penalizada menos do que classificação errada no nó D . Pois o nó I é irmão de G , ao contrário do nó D .

2. A medida pune erros distantes mais pesadamente: A medida proporciona uma avaliação mais elevada para classificar corretamente um nível abaixo em comparação com a permanência no nó pai. Por exemplo, a classificação no nó E é melhor do que a classificação no nó C, uma vez que E está mais próximo da classe correta, no caso G; A medida provê uma avaliação inferior para classificar incorretamente um nível abaixo em comparação com a permanência no nó pai. Por exemplo, a classificação no nó F é pior do que a classificação em seu pai C, pois F é mais distante de G.
3. A medida pune os erros em níveis mais elevados de uma hierarquia mais pesadamente. Ou seja, a classificação incorreta no nó I quando a classe correta é seu irmão G é menos grave do que a classificação errada no nó C quando a classe correta é seu irmão A.

Figura 21 – Exemplo de hierarquia de classes em DAG - O círculo sólido G representa a classe correta de classificação



Fonte - Kiritchenko et al. (2006)

De acordo com Kiritchenko et al. (2006), as medidas hierárquicas baseadas na distância satisfazem o segundo princípio, mas nem sempre o primeiro e o terceiro. Neste contexto, foi apresentada uma nova medida de avaliação hierárquica que satisfaça os três princípios. Estas medidas são chamadas de hP (precisão hierárquica) e hR (recall hierárquico), onde cada exemplo pertence não somente à sua classe, mas também a todas as classes ascendentes em um grafo hierárquico, exceto o nó raiz.

A medida-F Hierárquica (hF), definida na Equação 2.9, é calculada como a média harmônica das medidas precisão hierárquica (hP) e revocação hierárquica (hR). A precisão hierárquica é definida na Equação 2.7 e a revocação hierárquica é definida na Equação 2.8, onde C_p representa o conjunto de classes previstas e C_v é o conjunto de classes verdadeiras.

$$hP = \frac{|Ance(C_v) \cap Ance(C_p)|}{|Ance(C_p)|} \quad (2.7)$$

$$hR = \frac{|Ance(C_v) \cap Ance(C_p)|}{|Ance(C_v)|} \quad (2.8)$$

$$Medida_hF = \frac{(\beta^2 + 1) * hP * hR}{hP + hR} \quad (2.9)$$

Segundo [Silla e Freitas \(2011b\)](#), embora nenhuma medida de classificação hierárquica possa ser considerada melhor em todos os cenários e aplicações possíveis da classificação hierárquica, recomenda-se o uso das medidas baseadas na ancestralidade hP , hR e $Medida_hF$, pois podem ser aplicadas em diversos cenários de classificação hierárquica. Entre eles, estruturas de árvore e DAG, caminho único, múltiplos caminhos, predição não obrigatória em nós-folha e predição obrigatória em nós-folha. Porém, apresenta algumas ressalvas.

Considerando um exemplo da classificação hierárquica na predição das funções de proteínas, onde as classes são termos da base de dados da FunCat. Nesta aplicação, diversas proteínas estão anotadas atualmente com classes mais genéricas. No entanto, isso não significa que a proteína realmente não possui classes mais específicas. Isto pode demonstrar que apenas as classes mais específicas da proteína não são conhecidas atualmente, mas que futuramente estas classes podem ser descobertas por biólogos. Por exemplo, se uma classe conhecida mais específica de um exemplo é R.1, e a classe predita pelo classificador for R.1.1, pode-se dizer que a predição estava correta no primeiro nível. Já a predição para o segundo nível (R.1.1) pode ser correta ou não, pois pode ser que a classe verdadeira no segundo nível ainda é desconhecida. Portanto, neste tipo de aplicação, há um argumento para modificar a definição de hP , de tal forma que as predições sobre especializados, não devem ser consideradas como erros, portanto, não é penalizada ([SILLA; FREITAS, 2011b](#)).

Apesar do problema citado acima, em geral, as medidas de hP , hR e $Medida_hF$ parecem medidas eficazes de classificação hierárquica para avaliação em diversos cenários, o que justifica sua recomendação ([SILLA; FREITAS, 2011b](#)).

2.4 Entropia - Teoria Matemática da Informação

Na área de Teoria da Informação, a entropia é definida como sendo uma maneira de mensurar o grau médio de incerteza a respeito de fontes de informação, o que conseqüentemente permite a quantificação da informação presente que flui no sistema.

De acordo com [Shannon \(1948\)](#), que fez o seguinte questionamento em seu trabalho: Podemos definir uma quantidade que irá medir, em certo sentido, quanta informação é “produzida” por tal processo, ou melhor, qual a taxa informação é produzida?

Segundo [Cunha e Velho \(2003\)](#), suponha-se existir um conjunto de eventos possíveis cujas probabilidades de ocorrência são p_1, p_2, \dots, p_n . Estas probabilidades são conhecidas,

mas isso é tudo o que se sabe sobre qual evento ocorrerá. Pode-se encontrar uma medida que mensure a quantidade de “escolha” envolvida na seleção do evento ou de quão incertos somos do resultado? Se houver tal medida, digamos $H(p_1, p_2, \dots, p_n)$, é razoável exigir dela as seguintes propriedades:

1. H deve ser contínuo no p_i .
2. Se todos os p_i são iguais, $p_i = \frac{1}{n}$, então H deveria ser uma função crescente monotônica de n . Com eventos igualmente prováveis, há mais escolha, ou incerteza, quando há mais eventos possíveis.
3. Se a escolha for dividida em duas escolhas sucessivas, o H original deve ser a soma ponderada dos valores individuais de H .

A fim de chegar na formulação matemática da entropia, imagine por exemplo uma variável aleatória X , que pode assumir dois valores distintos x_1 e x_2 com probabilidades p_1 e p_2 , respectivamente. Seguindo a notação definida na seção: Variáveis aleatórias discretas, temos: $X = x_1, x_2$ e $p(X) = p_1, p_2$. A informação de Shannon (1948) associada a cada um dos valores é: $h_1 = \log_2 \frac{1}{p_1}$ e $h_2 = \log_2 \frac{1}{p_2}$.

Na prática, geralmente o pesquisador não se interessa em saber da surpresa de um valor em particular que uma variável aleatória possa assumir, e sim da surpresa associada com todos os possíveis valores que essa variável aleatória pode ter. De modo a obter a surpresa associada a todos possíveis valores que X pode assumir, defini-se a entropia $H(X)$ como a informação de Shannon média (Equação 2.10):

$$H(X) = p_1 h_1 + p_2 h_2 = p_1 \log_2 \frac{1}{p_1} + p_2 \log_2 \frac{1}{p_2} = \sum_{i=1}^2 p_i \log_2 \frac{1}{p_i} \quad (2.10)$$

Caso X , possa assumir m valores, a expressão anterior pode ser escrita de modo mais geral, conforme equação 2.11.

$$H(X) = - \sum_{i=0}^m p_i \log_2 p_i \quad (2.11)$$

Perceba que a entropia nada mais é do que a informação média associada às observações relativas à variável aleatória. Também é possível dizer que a entropia é uma medida da incerteza média associada à variável (COVER; THOMAS, 2012).

A quantidade H tem um número de propriedades interessantes que mais adiante substanciam isto como um razoável medida de escolha ou informação (SHANNON, 1948).

1. $H = 0$ se e somente se todos os p_i , porém um é igual a zero, este tendo o valor único. Assim, somente quando se está certo do resultado que H desaparece. Caso contrário, H é positivo.
2. Para um dado n , H é um máximo e igual a $\log_2 n$, quando todos os p_i são iguais (ou seja, $\frac{1}{n}$). Isto é também intuitivamente a situação mais incerta.

3 Estado da arte

Inserem-se neste capítulo um levantamento dos principais métodos do estado da arte para predição obrigatória e não obrigatória (opcional) de nós-folha em problemas de classificação hierárquica.

3.1 Predição Obrigatória em Nós-Folha

Nesta seção apresentar-se-ão os trabalhos relacionados dos principais métodos de predição obrigatória de nós-folha (MLNP) relacionados à classificação hierárquica.

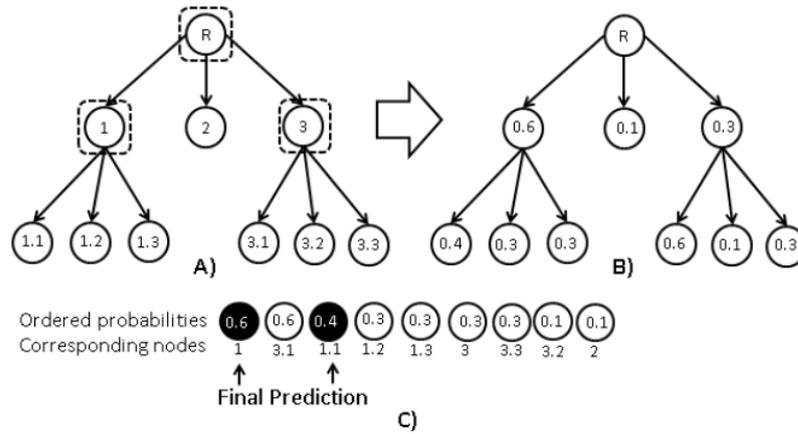
3.1.1 Ordenação, Soma e Multiplicação

No trabalho de [Hernandez, Sucar e Morales \(2013\)](#) foi proposta uma abordagem alternativa para a classificação hierárquica, que pode ser considerado como um método multidimensional. A ideia principal é aprender uma série de classificadores locais e depois combinar seus resultados a fim de obter um conjunto de classes consistentes; ou seja, um conjunto de caminhos do nó-raiz até um nó-folha na hierarquia. Na fase de treinamento, para cada nó-pai da hierarquia é criado um classificador multi-classe. Na fase de classificação, em contraste com as abordagens tradicionais de cima para baixo, todos os classificadores são aplicados simultaneamente a cada instância, e para cada classificador local, é estimada uma probabilidade para cada classe.

A partir das probabilidades de todos os classificadores, um conjunto de classes consistentes, de acordo com a hierarquia, é obtido. Foram propostas três alternativas para construir o conjunto consistente: (I) ordenar todas as classes de acordo com suas probabilidades e selecionar o primeiro subconjunto consistente nesta ordem, conforme mostrado na Figura 22; (II) multiplicar as probabilidades de todos os possíveis caminhos da raiz até um nó folha na hierarquia, e selecionar a trajetória com maior valor, conforme mostrado na Figura 23; e (III), similar a alternativa (II), porém utilizou-se a soma das probabilidades, ao invés da multiplicação.

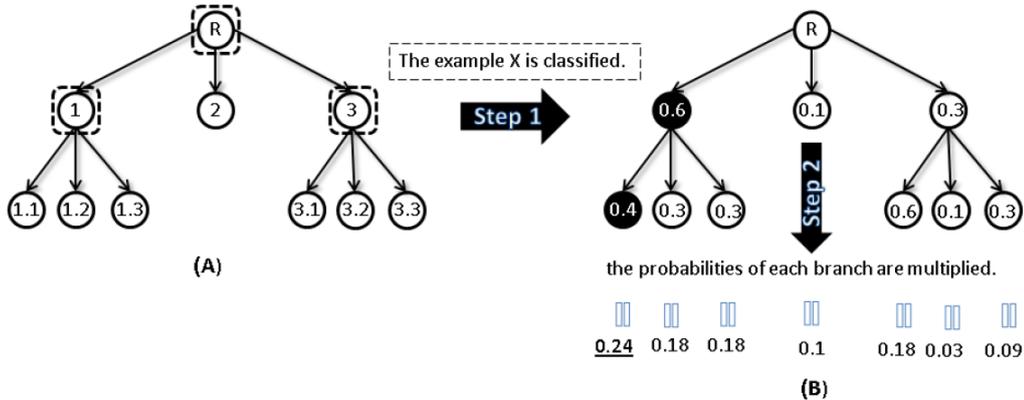
Conforme hierarquia apresentada na Figura 22, os círculos representam as classes e os quadrados tracejados representam classificadores multiclases. Para a predição desta primeira abordagem, para cada nó (exceto a raiz), $P(i)$ corresponde a probabilidade predita de cada classe. Posteriormente, as classes são classificadas em ordem decrescente de acordo com sua probabilidade, o primeiro subconjunto de classes consistentes de acordo com esta ordem são selecionados (marcados).

Figura 22 – Ordem Decrescente de Probabilidades (DOP)



Fonte: Hernandez, Sucar e Morales (2013)

Figura 23 – Multiplicação de probabilidades (MP)



Fonte: Hernandez, Sucar e Morales (2013)

Já na segunda abordagem proposta, conforme Figura 23, tendo a mesma hierarquia da primeira abordagem, cada nó (exceto a raiz) representa a probabilidade predita de sua classe. As probabilidades em cada trajetória na árvore - por exemplo $P(1) \times P(1.1)$ - são multiplicadas e os resultados são mostrados logo abaixo da Figura. Posteriormente a trajetória com o produto mais alto é selecionada (sublinhada).

A solução proposta por Dumais e Chen (2000) considera uma árvore com dois níveis de categorias para classificação de documentos da web, sendo o primeiro nível superior mais genérico e o segundo nível mais específico (nós-folha). Foram apresentados dois métodos de combinar as probabilidades do primeiro e do segundo nível para a abordagem hierárquica.

Neste primeiro método, foi definido um limiar de probabilidade no nível superior onde apenas os documentos aceitos de acordo com este limite passam para o nível inferior da hierarquia. Para ser classificada corretamente, uma instância de teste deve satisfazer ambas as restrições, no primeiro e segundo nível. Este método é bastante eficiente, pois um grande número de categorias de segundo nível não precisam ser testadas. Já no segundo método, multiplica-se a probabilidade da classe do nível superior com a probabilidade da classe do nível inferior, possibilitando a classificação da instância mesmo que a pontuação em um determinado nível esteja abaixo do limite.

3.1.2 Média

No trabalho de [Merschmann e Freitas \(2013\)](#), foi proposto o método ELHNB (Extended Local Hierarchical Naive Bayes classifier), no qual foram treinados um classificador Naive Bayes para prever o vetor de rótulo de classe $C = c_1, c_2, \dots, c_m$ para uma nova instância $X = x_1, x_2, \dots, x_n$, ou seja, aprender um preditor $f : X \mapsto [c_1, c_2, \dots, c_m]$. A abordagem proposta tem duas fases. A primeira é uma fase de classificação local, onde foi treinado um classificador binário para cada classe. Cada classificador binário realiza uma classificação probabilística para cada classe C_i , ou seja, calcula $P(C_i = 1|X)$ e $P(C_i = 0|X)$. Na segunda fase, denominada fase de classificação global, foram usadas as probabilidades $P(C_i = c_i|X)$ computadas na primeira fase. Para obter uma classificação consistente, para cada caminho possível na hierarquia do nó raiz para o nó i , calcula-se a média geométrica (Equação 3.1) das probabilidades $P(C_i = 1|X)$ ao longo do caminho, onde L_p é o conjunto de classes no caminho p .

$$GA_p = |L_p| \sqrt{\prod_{C_i \in L_p} P(C_i = 1|X)} \quad (3.1)$$

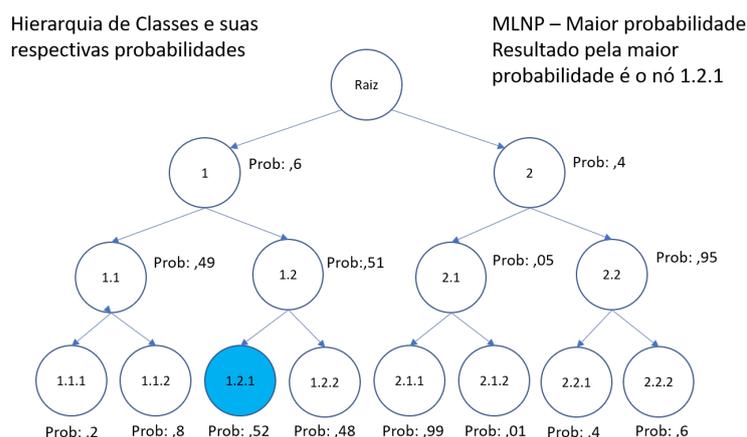
Como solução final, a instância X é atribuída ao vetor de classe C , onde as classes C_i contidas no caminho com o maior GA são definidas com 1 e o restante com 0.

3.1.3 Top-Down

No trabalho de [Fabris e Freitas \(2014\)](#), estes propuseram uma extensão do trabalho de [Merschmann e Freitas \(2013\)](#). Porém, os autores não utilizaram a abordagem da média geométrica para encontrar o caminho mais provável na hierarquia de classes, em vez disso, usaram uma estratégia Top-Down que é: primeiro encontrar a classe mais provável que é um filho do nó raiz, para então, encontrar recursivamente o filho mais provável desse nó, e assim por diante, até que um nó folha seja atingido. Essa abordagem teve melhores resultados do que a estratégia original de usar a média geométrica em seus experimentos, embora estatisticamente os resultados não apresentaram diferenças significativas entre estes métodos.

Por meio do critério de seguir o caminho pela maior probabilidade de um determinado nó, começando a partir do nó-raiz e percorrendo um nível de cada vez até chegar em um nó-folha, a Figura 24 mostra o caminho percorrido na árvore desde a raiz, descendo para a classe com maior probabilidade dos seus respectivos nós-filhos, até chegar em um nó-folha. Neste exemplo o caminho que será percorrido é Raiz, 1, 1.2 e 1.2.1.

Figura 24 – Predição obrigatória de nós-folha (MLNP) Top-Down



3.2 Predição Não-Obrigatória (Opcional) em Nós-folha

Nesta seção mostrar-se-ão os trabalhos relacionados dos principais métodos de critérios de parada do estado da arte para predição não-obrigatória de nós-folha (NMLNP).

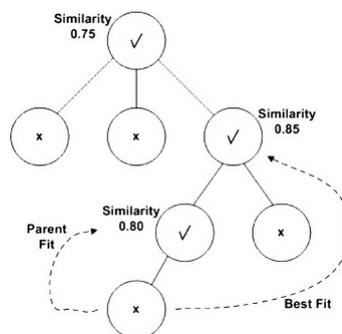
3.2.1 TopDown com Limiar (Threshold)

O trabalho de [Ceci e Malerba \(2007\)](#) apresenta uma maneira de lidar com o problema NMLNP com a utilização de um limiar em cada nó de uma classe. Se a pontuação de confiança ou probabilidade em um determinado nó, em uma nova instância de teste, é inferior ao limiar adotado, a classificação para em um nó-interior. Baseado neste contexto foi criado um método para calcular automaticamente esses limiares.

O trabalho de [Moskovitch et al. \(2006\)](#) apresenta uma abordagem de classificação de documentos (diretrizes) clínicos em categorias hierárquicas predefinidas. Dada uma instância a classificação é baseada na aplicação de uma função de similaridade em cada categoria. Se o critério do teste de classificação for atendido, examina-se cada sub-árvore do nó da categoria atual, e a mesma operação de classificação será executada recursivamente, seguindo assim da raiz até as folhas. O processo de classificação pode parar em um nó folha ou em uma categoria intermediária da hierarquia, caso nenhum dos testes de similaridade das subcategorias seja satisfatório devido ao limiar. Para a interrupção da classificação em

uma determinada categoria da hierarquia, dois métodos foram propostos: retornar com o último nó-pai (Parent-Fit) ou retornar o melhor resultado de todo o caminho percorrido (Best-Fit), conforme mostrado na Figura 25.

Figura 25 – Exemplo de critério de parada com retorno da melhor ou última classe



Fonte: [Moskovitch et al. \(2006\)](#)

O critério de parada Parent-Fit classifica o novo documento na última categoria em que o teste é verdadeiro, enquanto o critério de parada Best-Fit classifica o documento na categoria em que o valor da medida de similaridade foi o mais alto de todo o percurso.

3.2.2 Redução de Limiares, Votação Restrita e Limiares multiplicativos estendidos

Outras estratégias de bloqueio foram discutidas em [Sun et al. \(2004\)](#), como um método de redução de limiares, método de votação restrito e limiares multiplicativos estendidos. As três estratégias são detalhadas a seguir:

- Método de redução de Limite: consiste em reduzir os limiares dos classificadores de sub-árvore. A ideia por trás dessa abordagem é que, ao reduzir os limiares isso permitirá mais exemplos a serem passados para os classificadores em níveis mais baixos. O desafio associado a esta abordagem é a forma de definir o limiar de cada sub classificador. Este método pode ser facilmente utilizado com ambas estruturas hierárquicas: árvore ou DAG.
- Votação restrita: consiste na criação de um conjunto de classificadores secundários que irão ligar um nó e seu nó avô (dois níveis acima). A motivação para esta abordagem é que, embora o método de redução do limiar seja capaz de passar mais exemplos para os classificadores nos níveis mais baixos, ainda é possível ter exemplos erradamente rejeitados pelos classificadores de subárvore de alto nível. Portanto, a abordagem de votação restrita dá aos classificadores de baixo nível a oportunidade de acessar esses exemplos, antes que sejam rejeitados.

- Limites multiplicativos estendidos: é uma extensão direta do limiar multiplicativo proposto por Dumais e Chen (2000) (explicado na Seção 4.1.), o qual originalmente só trabalhava para uma hierarquia de três níveis. A extensão consiste simplesmente em estabelecer limiares de forma recursiva para cada dois níveis.

Sun et al. (2017) formularam a tarefa de classificação como um problema de seleção de caminhos, onde cada caminho começa na raiz e termina em uma folha ou nó interno. Eles usaram mínimos quadrados parciais para resolver o problema de predição de rótulos como um problema de predição de caminho ótimo. Cada predição é um subgrafo conectado que pode ser formado por um pequeno número de caminhos. O método de avaliação global encontra os caminhos ótimos, e a predição final é dada pela união desses caminhos.

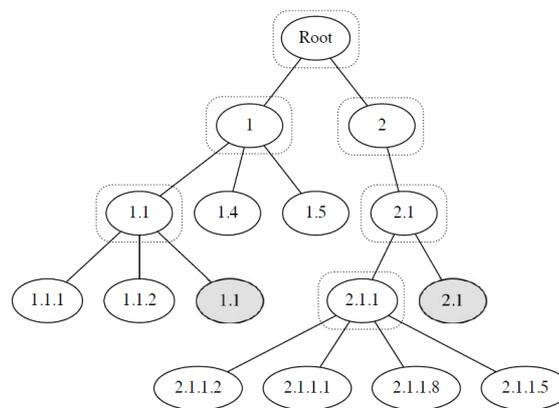
3.2.3 Medida de Utilidade

No trabalho de Silla e Freitas (2009a), são utilizadas as medidas de utilidade e as probabilidades das classes atual e filhas como critério para decidir em qual nível interromper a classificação durante o processo de classificação de cima para baixo. Como tem-se a medida de utilidade de uma determinada classe predita, usa-se o seguinte critério de parada: se $p(c_i) * utilidade(c_i) > p(c_j) * utilidade(c_j)$ para todas as classes (c_j) que são filhas da classe atual (c_i), então para a classificação no nó atual. Em outras palavras, se a probabilidade a posteriori multiplicada pela utilidade computada pelo classificador no nó da classe atual for maior que a probabilidade a posteriori multiplicada pela utilidade computada para cada um dos nós da classe filha, a classificação é interrompida no nó da classe atual - isto é, retorna esta classe como mais específica (mais profunda), predita para o exemplo de teste atual.

3.2.4 Replicação do Nó-Interno

No trabalho de Nakano et al. (2017), foi investigada a classificação de Elementos Transponíveis como um problema de Classificação Hierárquica, usando Aprendizado de Máquina. Elementos transponíveis são sequências de DNA que podem se mover de um lugar para outro dentro do genoma de uma célula. Eles são importantes para a variabilidade genética e podem modificar a funcionalidade dos genes. Além disso, os autores propuseram duas novas estratégias NMLNP. Na fase de treinamento da primeira estratégia, denominada nLLCPN, segue o mesmo procedimento adotado pela abordagem LCPN, onde para cada nó tracejado na Figura 26 um classificador de várias classes é treinado. No entanto, quando casos como o ilustrado na Figura 26 ocorrem na hierarquia - por exemplo: quando existem instâncias que podem ser classificadas na classe 2.1 como sendo sua classe mais específica - o método nLLCPN replica o nó interno, e a classe 2.1 aparece como uma subclasse de si mesmo. Desta forma, pode-se ter uma classificação final com um nó interno da hierarquia.

Figura 26 – Exemplo da hierarquia do método nLLCPN



Fonte: Nakano et al. (2017)

Para realizar a classificação de uma determinada instância neste método nLLCPN, uma estratégia top-down convencional é adotada. No entanto, se um classificador de um nó atual realizar a predição em seu nó replicado, a classificação será interrompida neste nó interno da hierarquia, realizando neste caso a NMLNP. No caso em que o problema requer uma classificação obrigatória de nó-folha, as estratégias LCPN e nLLCPN são equivalentes.

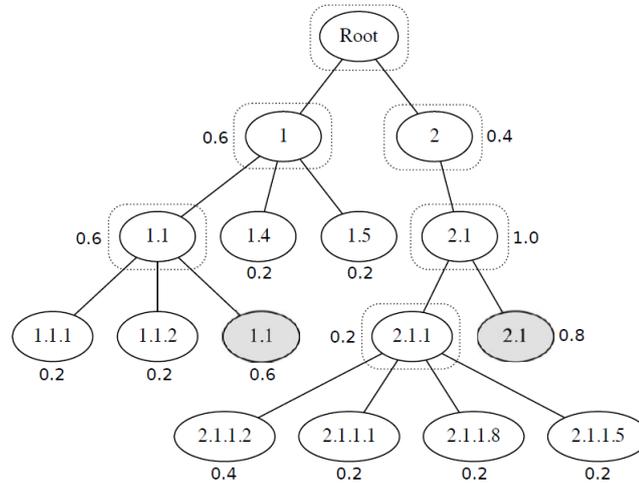
A segunda estratégia é um Classificador Local por Nó Pai e Ramos (LCPNB), onde o mesmo utiliza o procedimento de replicação dos nós-internos proposto no nLLCPN. Ou seja, a taxonomia é modificada (replicação dos nós internos) para permitir a classificação de um nó não-folha, e cada nó tracejado é associado a um classificador de várias classes.

No entanto, em vez de usar uma estratégia convencional top-down para classificar uma nova instância, a instância é fornecida como entrada para todos os classificadores da hierarquia e as probabilidades são obtidas para todas as classes. Posteriormente, para cada caminho da raiz até os nós-folha, as probabilidades médias são calculadas considerando as probabilidades associadas a cada classe no caminho. A classificação final é dada pelo caminho com a maior probabilidade média. A lógica por trás dessa estratégia é que possíveis erros em níveis mais altos podem ser corrigidos pelas probabilidades das classes mais profundas.

A Figura 27 ilustra o procedimento de classificação do LCPNB para uma determinada instância. Observe que, para cada nó, um valor de probabilidade atribuído pelo classificador associado ao pai tracejado é mostrado. Vamos nos concentrar nas instâncias da classe 1.1. Uma estratégia LCPN prediria a classe 1.1 como a classe final se estivesse usando uma estratégia de redução de limiar, ou continuaria a classificação até que um nó folha fosse alcançado (classes 1.1.1 ou 1.1.2). Isso ocorre porque a estratégia top-down convencional segue o caminho com probabilidades mais altas, e a classificação final seguiria

o caminho começando na classe 1 na Figura 27.

Figura 27 – Método LCPNB



Fonte: Nakano et al. (2017)

As probabilidades dos caminhos possíveis são mostradas abaixo:

- Classificação de nó folha não obrigatório:

$$1(0,6) \rightarrow 1.1(0,6) \rightarrow 1.1(0,6) \mid \text{Média} = (0,6 + 0,6 + 0,6)/3 = 0,6;$$

- Classificação obrigatória de nó folha:

$$1(0,6) \rightarrow 1.1(0,6) \rightarrow 1.1.1 \text{ ou } 1.1.2(0,2) \mid \text{Média} = (0,6 + 0,6 + 0,2)/3 = 0,47.$$

No caso do LCPNB, considerando-se o máximo das probabilidades médias de todos os caminhos possíveis, a classificação final seria alterada. O LCPNB considera as probabilidades médias de todas as classificações finais possíveis, e a classificação final seria dada pelo caminho que leva à classe 2.1:

$$2(0,4) \rightarrow 2.1(1,0) \rightarrow 2.1(0,8) \mid \text{Média} = (0,4 + 1,0 + 0,8)/3 = 0,73.$$

Uma vez que a classificação de Elementos Transponíveis é um problema de classificação não obrigatória em nós-folha (NMLNP), as abordagens LCL e LCPN precisam de algum procedimento para permitir classificações em um nó interno da hierarquia. No entanto as estratégias nLLCPN e LCPNB não necessitam de tal procedimento, uma vez que podem lidar naturalmente com predições não obrigatória em nós-folha por meio da replicação dos nós de determinadas classes. (NAKANO et al., 2017).

3.2.5 Ganho de Informação

No trabalho de [Hernández, Sucar e Morales \(2014\)](#) é apresentada uma abordagem NMLNP baseada no Ganho de Informação. O melhor caminho é definido pela avaliação global por meio da multiplicação de todas as probabilidades das classes de todos os caminhos. Segundo os autores, a exploração de toda a hierarquia em uma avaliação global utilizando abordagens de classificação hierárquica local implica no cálculo, por exemplo na soma, média ou multiplicação, de todos os resultados (probabilidades) de todas as classes em cada caminho. O caminho com o melhor resultado é avaliado utilizando o Ganho de Informação para realizar a predição em qualquer nível da hierarquia.

Posteriormente, o ganho de informação é aplicado a cada classificador local por nó pai, no caminho selecionado (avaliação global), de maneira ascendente (bottom-up). O cálculo do ganho de informação é definido na Equação 3.2, onde E representa a entropia, $ch_j(c_i)$ é o filho j do nó i , e W é o peso de cada classe filho do nó i , que corresponde ao número proporcional de exemplos que pertence a cada filho.

$$IG(c_i) = E(c_i) - \sum_{j=1}^{ns} w_{ch_j(c_i)} * E(ch_j(c_i)) \quad (3.2)$$

Se o ganho de informações for menor que zero para o classificador inferior, a decisão será definida como desconhecida para esse classificador e o próximo classificador, um nível acima, será analisado. Caso contrário, se o ganho de informação for maior que zero, a predição do classificador local será aceita e a análise será encerrada. Se o classificador de nível superior (Raiz) for alcançado, o processo também será encerrado, aceitando a predição deste classificador.

No caso da predição não obrigatória de nós-folha, o caminho retornado é seguido em ordem inversa, do Nó-Folha ao Nó-Raiz da árvore ou grafo. Se um nó com vários pais for encontrado na trajetória, o caminho original (com a maior probabilidade) será seguido em direção à raiz, sem considerar caminhos alternativos.

Segundo [Hernández, Sucar e Morales \(2014\)](#), a utilização do ganho de informação para a NMLNP obteve bons resultados, mas que há a necessidade de fazer mais experimentos para fortalecer os resultados que foram apresentados no trabalho. Além disso, os autores apontaram que não provaram seus métodos em taxonomias desbalanceadas.

4 Experimentos

Neste capítulo apresentam-se os materiais utilizados, os métodos de classificação e a configuração experimental. Nos materiais, estão descritas as bases de dados utilizadas nos experimentos (Datasets), importantes para analisar os critérios de paradas dos classificadores diante de diferentes problemas. Também estão descritas as configurações usadas nos experimentos, para fins de detalhamento e possibilidade de reprodução.

4.1 Materiais e Configuração Experimental

A Bioinformática ou Biologia Computacional é uma das áreas de conhecimento derivada da aplicação da Ciência da Computação no armazenamento, análise e geração de dados biológicos.

Um grande desafio da Bioinformática é de auxiliar na predição da função de proteínas. Os trabalhos relativos a esse assunto determinam uma nova fase para as pesquisas genéticas, denominada proteômica. Este estudo abrange a determinação das funções fisiológicas e patológicas das proteínas, bem como a identificação de todas as proteínas expressas pelo genoma (BITTENCOURT, 2005).

A predição da função das proteínas é importante, haja vista que este tipo de informação pode ser potencialmente utilizada para desenvolver remédios e para um melhor diagnóstico no tratamento de doenças, onde muitas delas são causadas pelo mau funcionamento de proteínas (SILLA; FREITAS, 2011b).

As ontologias denotam uma forma para representar um conhecimento, de modo que sejam permitidos o compartilhamento e o reaproveitamento desse conhecimento adquirido. De uma forma sucinta, a ontologia é uma especificação explícita de uma conceitualização (GRUBER, 1993), e é definida como um conjunto de representações primitivas responsáveis por modelar um conhecimento de um determinado domínio (SAM; CHATWIN, 2015).

Uma ontologia que foi criada para a classificação da função de proteínas é a *Functional Catalogue*, chamada geralmente de FunCat (RUEPP et al., 2004). Essa ontologia foi criada objetivando descrever as proteínas do ponto de vista funcional. A estrutura utilizada para representar as funções de proteínas nesta base de dados é uma hierarquia no formato de árvore.

Nestes conjuntos de dados são usadas duas famílias de proteínas diferentes: EC (*Enzyme Commission*) e GPCRs (*G-Protein-Coupled Receptors*). Enzimas são catalisadores que aceleram reações químicas, enquanto os GPCRs são proteínas envolvidas na sinalização e são particularmente importantes em aplicações médicas, pois acredita-se que 40-50% dos

medicamentos atuais atinjam a atividade do GPCR (FILMORE, 2004).

Cada assinatura (*motifs*) é representada por um atributo binário, que indica a presença ou ausência da assinatura em cada proteína. Este tipo de representação proteica incorpora implicitamente um grande conhecimento sobre as proteínas disponíveis na literatura, uma vez que as assinaturas proteicas foram criadas e aperfeiçoadas por biólogos especialistas e bioinformáticos (HIGGS; ATTWOOD, 2013).

Para avaliar os resultados dos métodos propostos em comparação com os principais métodos de critérios de parada da literatura, foram utilizadas duas representações diferentes de proteínas.

4.1.1 Bases de Dados FunCat

Este conjunto de dados apresenta duas famílias de proteínas diferentes: EC (*Enzyme Commission*) e GPCRs (*G-Protein-Coupled Receptors*). Para cada família, EC e GPCR, quatro conjuntos de dados foram construídos, com a presença de assinaturas (*motifs*) como características. Quatro assinaturas foram usadas: Interpro, Pfam, Prosite as Prints. Quando existe a presença da assinatura, o atributo recebe 1, caso contrário, o atributo é 0. Há também a presença de dois atributos contínuos: comprimento da sequência e peso molecular.

Estes conjuntos de dados utilizam uma estrutura hierárquica em um formato de árvore, independente, flexível e escalável, permitindo a descrição funcional de proteínas de qualquer organismo. Cada um desses ramos funcionais é organizado em uma hierarquia, com base na estrutura da árvore. Este conceito tem sido mantido desde a anotação do genoma da levedura e provou ser adequado para a anotação de outros genomas. Esses ramos abrangem as características gerais, como transporte celular, metabolismo e regulação da atividade proteica (RUEPP et al., 2004). A Tabela 1 apresenta as características de cada conjunto de dados da FunCat.

Tabela 1 – Base de dados FunCat

<i>Base de Dados</i>	<i>Instâncias</i>	<i>Atributos</i>	<i>Classes</i>	<i>Classes por Nível</i>
ECinterpro	14036	1216	331	6/41/96/188
ECpfam	13995	708	334	6/41/96/191
ECprints	14038	382	352	6/45/92/209
ECprosite	14048	585	324	6/42/89/187
GPCRinterpro	7461	450	198	12/54/82/50
GPCRpfam	7077	75	192	12/52/79/49
GPCRprints	5422	282	179	8/46/76/49
GPCRprosite	6261	128	187	9/50/79/49

4.1.2 Bases de Dados Descritores Locais (LD)

No trabalho feito por [Silla e Freitas \(2011a\)](#), foram criadas cinco diferentes representações dos conjuntos de dados de proteínas. Estes conjuntos foram originalmente desenvolvidos por [Holden e Freitas \(2009\)](#). Uma destas representações, e que foi utilizada para avaliação e comparação deste trabalho, é denominada Local Descriptors (Descritores Locais).

Os descritores locais são também conhecidos como descritores de sequências de proteínas globais. Existem três tipos de descritores locais: Composição, Transição e Distribuição, que são calculados com base na variação de ocorrência de grupos funcionais de aminoácidos dentro da sequência primária da proteína. Os grupos funcionais utilizados foram: hidrofóbicos (aminoácidos CVLIMFW), neutros (aminoácidos GASTPHY) e polares (aminoácidos RKEDQN) ([SILLA; FREITAS, 2011a](#)).

A composição é responsável pelo percentual (frequência relativa) de um grupo funcional particular dentro da sequência de aminoácidos. Portanto, existem três características de composição, uma para cada grupo funcional de aminoácidos.

As características de transição representam a frequência relativa em que um aminoácido de um grupo funcional particular é seguido por um aminoácido de outro grupo funcional. Mais precisamente, as seguintes transições são consideradas: Polar \rightarrow Neutro ou Neutro \rightarrow Polar; Polar \rightarrow Hidrofóbico ou Hidrofóbico \rightarrow Polar; e Neutro \rightarrow Hidrofóbico ou Hidrofóbico \rightarrow Neutro.

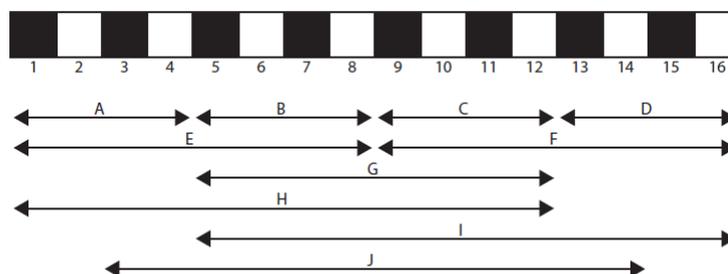
As características de distribuição são calculadas com base na percentagem de quantos aminoácidos de um grupo funcional particular estão presentes nos primeiros 25%, 50%, 75% e 100% da sequência dos aminoácidos.

No total, haveria 21 características (3 composição, 3 transição, 15 distribuição) se fossem calculadas a partir de toda a sequência de aminoácidos. No entanto, em [Davies et al. \(2008\)](#) e [Tong e Tammi \(2008\)](#), os autores dividiram a sequência proteica em 10 regiões descritoras (A-J), conforme segue: as regiões A, B, C e D são obtidas dividindo toda a sequência proteica em quatro regiões de comprimento igual. As regiões E e F são obtidas dividindo a sequência proteica em duas regiões de comprimento igual. A região G representa o meio com 50% da sequência. A região H representa os primeiros 75% da sequência, a região I equivale aos 75% final da sequência e a região J corresponde ao meio com 75% da sequência. Para cada região, os 21 descritores locais são extraídos, resultando em um vetor de característica 210. Essas regiões estão ilustradas na [Figura 28](#).

A [Tabela 2](#) apresenta as características de cada conjunto de dados com Descritores Locais (LD).

Destaca-se que nestas duas representações de proteínas, utilizadas neste trabalho, que as características do problema hierárquico presentes nestes conjuntos de dados são de

Figura 28 – As 10 regiões usadas pela técnica do Descritor Local, usadas em [Davies et al. \(2008\)](#) e [Tong e Tammi \(2008\)](#)



Fonte: [Silla e Freitas \(2011a\)](#)

Tabela 2 – Características das bases de dados Descritores Locais (LD)

Base de Dados	Instâncias	Atributos	Classes	Classes por Nível
LD-ECinterpro	14027	212	330	6/41/96/187
LD-ECpfam	13987	212	333	6/41/96/190
LD-ECprints	12487	212	351	6/45/92/208
LD-ECprosite	14041	212	324	6/42/89/187
LD-GPCRinterpro	7444	212	198	12/54/82/50
LD-GPCRpfam	7053	212	192	12/52/79/49
LD-GPCRprints	5404	212	179	8/46/76/49
LD-GPCRprosite	6246	212	187	9/50/79/49

taxonomia hierárquica no formato árvore, apresentam instâncias com rotulação parcial e o número de caminhos para predição é de caminho único SPL.

4.2 Configuração Experimental

4.2.1 Pré-processamento

Um pré-processamento foi realizado nos dois últimos atributos das bases de dados EC e GPCR da Funcat, mais especificamente nos atributos Comprimento da Sequência e Peso Molecular.

A técnica utilizada foi a Min-Max, conforme mostrado na Equação 4.1, a qual faz a transformação linear dos valores v de um atributo A mapeando os valores mínimo e máximo originais (min_A e max_A) do conjunto do atributo A para valores mínimo e

máximo ($nMin_E$ e $nMax_E$) da nova escala.

$$v' = \frac{v - min_A}{max_A - min_A}(nMax_E - nMin_E) + nMin_E \quad (4.1)$$

4.2.2 Política de Seleção de Exemplos

Neste trabalho, a política de seleção dos exemplos positivos e negativos, adotados para o treinamento de cada classificador Nó-Pai, foi a política de "Irmãos", definida por Fagni e Sebastiani (2007) e Ceci e Malerba (2007). Essa política determina que os exemplos positivos são as instâncias do nó da classe atual e todas as suas classes descendentes.

A estratégia de validação cruzada com 10 subconjuntos (*Folds*) foi utilizada para gerar os conjuntos de treinamento e teste. Além disso, foi usado o conjunto de validação para auxiliar na definição do limiar para cada conjunto de dados, bem como para determinação do momento de interrupção de treinamento dos classificadores para evitar o sobreajuste. Este conjunto foi retirado de um valor total de 20% do conjunto de treinamento. Ademais, todos os conjuntos foram recuperados de maneira estratificada. Esse parâmetro estratificado faz uma divisão para que a proporção de valores na amostra produzida seja a mesma que a proporção de valores na amostra original. Por exemplo, se a variável y for uma variável binária com valores 0 e 1 e houver 25% de zeros e 75% de uns, o uso de estratificação garantirá que sua divisão aleatória tenha 25% de zeros e 75% de uns.

4.2.3 Classificadores Bases

Foram utilizados dois algoritmos para os experimentos desta pesquisa. O primeiro algoritmo é denominado Perceptron Multi-Camadas (MLP) e o segundo consiste no algoritmo de Aprendizagem Profunda denominado Redes Neurais Convolucionais (CNN). As configurações destes algoritmos serão detalhadas abaixo.

4.2.3.1 Configuração dos Classificadores MLP

Foram utilizados classificadores MLP, respectivamente com 1 e 2 camadas ocultas e avaliados individualmente. A quantidade de neurônios das camadas ocultas são equivalentes à média entre o total de atributos de entrada e a quantidade de classes filhas de um respectivo nó-pai. Ou seja, a quantidade de neurônios varia de acordo com cada Classificador nó-pai, em decorrência do fato de que cada nó-pai de uma determinada hierarquia pode apresentar quantidades diferentes de nós-filho.

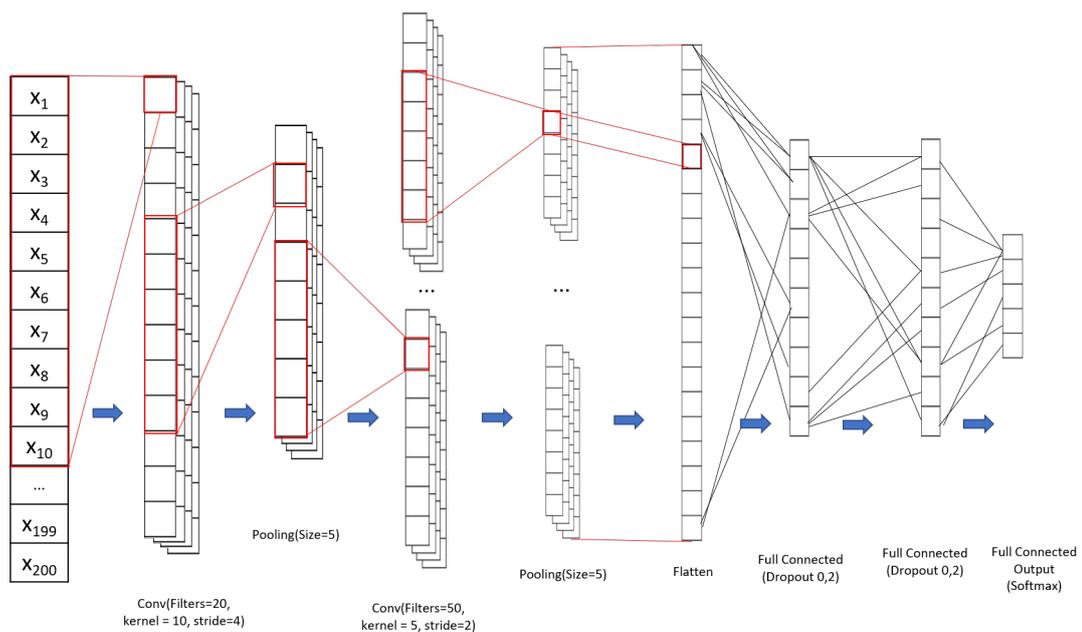
Foram definidas as mesmas configurações da rede para todos os nós-pai da hierarquia para as duas abordagens LCPN que utilizaram rede MLP, respectivamente, 1 e 2 camadas ocultas.

Na etapa de treinamento dos classificadores, tanto MLP, quanto CNN, o conjunto de validação foi utilizado para interromper o processo de treinamento, para evitar o sobreajuste da rede, com uma tolerância de 10 épocas. Ou seja, o treinamento é interrompido, caso o erro (função loss) do conjunto de validação não diminua dentro das quantidades de épocas ajustadas no parâmetro de tolerância.

4.2.3.2 Configuração dos classificadores CNN

Em conformidade com os classificadores MLP, foi utilizada a mesma configuração da rede CNN para todos os nós-pai da hierarquia. Nas camadas Densas (totalmente conectadas), a mesma quantidade de neurônios foi utilizada, sendo a média entre o total de atributos de entrada e a quantidade de classes filhas de um respectivo nó-pai. A figura 29 apresenta o modelo da rede CNN utilizada nos experimentos.

Figura 29 – Modelo CNN



Para ambos os classificadores de base (MLP e CNN), a função de ativação Softmax foi adicionada à camada de saída de cada classificador. Esta função Softmax normaliza as saídas de cada nó-filho para que a soma das probabilidades de todas as classes seja 1. Foi utilizado o otimizador Adam para o treinamento da rede com a taxa de aprendizagem padrão (0.001).

Embora tais configurações dos classificadores MLP, bem como CNN, tenham sido definidos com estes parâmetros, vale ressaltar que esta pesquisa visa avaliar diferentes métodos de critérios de parada para abordagem de predição não obrigatória em nós-folha, independente do classificador adotado como classificador base. Tanto os principais métodos

de critérios de parada da literatura, quanto os métodos propostos, foram testados nos três algoritmos de classificação. Seus resultados serão apresentados separadamente pelo fato de que esta pesquisa não busca encontrar um melhor algoritmo classificador, tampouco uma melhor configuração destes, mas sim propor métodos de critérios de parada independentes do classificador.

4.3 Métricas de Avaliação

De acordo com [Silla e Freitas \(2011b\)](#), a razão primordial para preferência quanto às medidas hP, hR e hF reside em sua capacidade de serem aplicadas em quaisquer contextos de classificação hierárquica, em que pese nenhuma medida de desempenho utilizada para classificação hierárquica poder ser considerada a melhor em todos os cenários e aplicações possíveis dentro das características dos problemas hierárquicos, quais sejam: Árvore ou DAG, SPL ou MPL, MLNP ou NMLNP.

Na fase de testes, as predições das abordagens foram realizadas de forma top-down e global, dependendo do método aplicado. Na abordagem top-down, a avaliação inicia pela predição do nó-pai (Raiz) e segue para seus respectivos nós-filho dos níveis inferiores. Já na abordagem de avaliação Global, aplica-se a instância para todos os classificadores nós-pai da hierarquia e a partir das probabilidades de todos os nós-filho, adota-se uma forma de percorrer o caminho na hierarquia e emprega-se algum tipo de critério de parada, neste caso, nas abordagens NMLNP.

Após o retorno do nó-predito, que pode ser um nó-interno ou um nó-folha, dependendo do critério de parada utilizado, os resultados das métricas de avaliação hP, hR e hF, descritas na Seção 2.3.4, são calculadas por meio do nó-predito e do nó-verdadeiro, bem como das suas respectivas classes ancestrais dos métodos avaliados.

4.4 Avaliação Estatística

Para a comparação dos métodos avaliados neste trabalho, foram aplicados testes não paramétricos de Friedman e o teste pós-hoc de Nemenyi. O teste de Friedman é um procedimento estatístico não paramétrico que permite determinar se existe ou não alguma diferença entre mais de duas amostras, sendo equivalente ao teste paramétrico de medidas repetidas de análise de variância ANOVA ([FRIEDMAN, 1937](#)).

Caso o resultado do teste de Friedman seja positivo, ou seja, caso ele indique que há alguma diferença entre as amostras, isto evidencia que pelo menos uma das amostras é diferente das demais; entretanto, ele não permite identificar qual amostra (ou quais amostras) é (são) diferentes do conjunto. Neste caso, [Demšar \(2006\)](#) indica o teste de Nemenyi, que calcula a distância crítica entre os resultados, o qual foi usado nesta pesquisa

para mostrar quais métodos apresentam diferença significativa de desempenho, bem como seus ranqueamentos.

5 Análise extensiva dos métodos do estado da arte

Neste capítulo foram avaliados os principais métodos do estado da arte para problemas de classificação hierárquica, bem como uma análise extensiva dos métodos de critérios de parada e uma abordagem comparativa com outros trabalhos da literatura.

Para avaliar os principais métodos de critérios de parada do estado da arte, bem como obter uma forma de investigar se uma determinada abordagem é melhor ou não, comparados com outros trabalhos da literatura (com base nas conclusões dos autores de cada trabalho), foram propostas duas tabelas. A primeira apresenta os métodos que

Tabela 3 – Comparativo dos Métodos com abordagem MLNP

				MLNP TopDown	Classifier Selection	GMNB	ELHNB	Class. Plana	NMLNP TopDown	LCL - Sim- ple Prune	LCL - Sim- ple Prune - 80%Limiar	LCL SWV	LCL SWV - 80%Limiar
				Koller and Sabami, 1997	Secker et. Al, 2007	Silla, 2009	Merschmann e Freitas, 2013	Ceci e Ma- lerba, 2007	Ceci e Ma- lerba, 2007	Corri e Car- valho 2016a	Corri e Carvalho 2016b	Pae, Plas- tino e Frei- tas 2012	Pae, Plas- tino e Frei- tas 2012
Trabalho	Dataset	Abordagem HC	Método de critério de parada	LCPN	LCPN	GC	LCN	Flat	LCPN	LCL	LCL	LCL	LCL
Hernandez, Sucar, Moralez, 2013	RC	LCPN	Ordenação descen- dente	~	~								
Hernandez, Sucar, Moralez, 2013	FC	LCPN	Ordenação Descen- dente	↑	~								
Hernandez, Sucar, Moralez, 2013	IAPR	LCPN	Ordenação Descen- dente	↑	↑								
Hernandez, Sucar, Moralez, 2013	RC	LCPN	Multiplicação Path	~	~								
Hernandez, Sucar, Moralez, 2013	FC	LCPN	Multiplicação Path	~	↑								
Hernandez, Sucar, Moralez, 2013	IAPR	LCPN	Multiplicação Path	~	↑								
Hernandez, Sucar, Moralez, 2013	RC	LCPN	Soma Path	~	~								
Hernandez, Sucar, Moralez, 2013	FC	LCPN	Soma Path	~	↑								
Hernandez, Sucar, Moralez, 2013	IAPR	LCPN	Soma Path	~	↑								
Merschmann e Freitas, 2013	GPCR	LCN	Média Geométrica Path	~		~							
Merschmann e Freitas, 2013	EC	LCN	Média Geométrica Path	~		↑							
Merschmann e Freitas, 2013	GENE	LCN	Média Geométrica Path	~		~							
Fabris e Freitas, 2014	GPCR	LCN	MLNP Top-Down	~			~						
Fabris e Freitas, 2014	EC	LCN	MLNP Top-Down	~			↑						
Fabris e Freitas, 2014	GENE	LCN	MLNP Top-Down	~			~						
Hernandez, Sucar, Morales, 2014	RC	LCPN	Multiplicação Path	~	~								
Hernandez, Sucar, Morales, 2014	FC	LCPN	Multiplicação Path	~	↑								
Hernandez, Sucar, Morales, 2014	IAPR	LCPN	Multiplicação Path	~	↑								
Hernandez, Sucar, Morales, 2014	MIREX	LCPN	Multiplicação Path	~	↑								

realizam a predição somente em nós-folhas (MLNP), conforme apresentado na Tabela 3. Tais tabelas foram separadas pelos autores, as bases de dados utilizadas conforme mostrado na Tabela 4, o tipo da abordagem hierárquica utilizada e o método de predição adotado.

Nestas tabelas os símbolos $\uparrow, \downarrow, \sim$ representam se cada abordagem (correspondente a uma determinada linha na tabela) obteve um melhor (\uparrow), pior (\downarrow) ou similar (\sim) desempenho preditivo do que a abordagem mostrada na coluna correspondente. Destaca-se que os nomes das abordagens nas linhas e colunas desta tabela são abreviados como se segue: LCN é o classificador locais por nó, LCPN é o Classificador local por nó pai, LCL é o Classificador local por nível e GC é o Classificador Global. Outrossim, a falta de quaisquer comparações nas devidas células da linha de uma determinada abordagem, não deve ser interpretado de não haver comparações com outros métodos. Pode ser, no entanto, que os

Tabela 4 – Legenda das Bases de Dados

Sigla	Domínio	Autor
RC	Texto	(Yang 1999); (Hernandez; Sucar; Morales, 2014)
FC	Proteína	(Ruepp et al. 2004) (Hernandez; Sucar; Morales, 2014)
IAPR	Imagem	(Escalante et al. 2010) (Hernandez; Sucar; Morales, 2014)
GPCR	Bioinformática	(Merschmann, 2013); (Carvalho, Brunoro e Pappa, 2011)
EC	Bioinformática	(Merschmann, 2013)
GENE	Bioinformática	(Merschmann, 2013)
YAH	Texto	(Ceci e Malerba, 2007)
DMOZ	Texto	(Ceci e Malerba, 2007)
CPG	Textos Clínicos	(Moskovitch et al., 2006)
PGSB	Bioinformática	(Nakano et al. 2017)
REP	Bioinformática	(Nakano et al. 2017)
PG-RE	Bioinformática	(Nakano et al. 2017)
MIREX	Música	(Hernandez; Sucar; Morales, 2014)

autores compararam apenas variações diferentes da sua própria abordagem (por exemplo parâmetros de ajuste), sem a comparação com outras abordagens.

Tabela 5 – Comparativo dos Métodos com abordagem NMLNP

				MLNP TopDown	Classifier Selection	GMNB	ELHNB	Class. Plana	NMLNP TopDown	LCL - Simple Prune	LCL - Simple Prune - 80%Limiar	LCL SWV	LCL SWV - 80%Limiar
				Koller and Sahami, 1997	Secker et. Al, 2007	Silla, 2009	Merschmann e Freitas, 2013	Ceci e Malerba, 2007	Ceci e Malerba, 2007	Cerri e Carvalho 2016a	Cerri e Carvalho 2016b	Pae, Plastino e Freitas 2012	Pae, Plastino e Freitas 2012
Trabalho	Dataset	Abordagem HC	Método de critério de parada	LCPN	LCPN	GC	LCN	Flat	LCPN	LCL	LCL	LCL	LCL
Ceci e Malerba, 2007	YAH	LCPN	NMLNP Top-Down	■									
Ceci e Malerba, 2007	DMOZ	LCPN	NMLNP Top-Down	■									
Ceci e Malerba, 2007	RC	LCPN	NMLNP Top-Down	■				↑					
Moskovitch et al, 2006	CPG	LCPN	Best Fit e Parent Fit	■									
Carvalho, Brunoro e Pappa, 2011	GPCR	LCPN	NMLNP Top-Down	■									
Sun et. al, 2004	RC	LCPN	Método de redução de Limiares	■					↑				
Sun et. al, 2004	RC	LCPN	Método de votação restrita	■					↑				
Sun et. al, 2004	RC	LCPN	Limiares multiplicativos estendidos	■					↑				
Nakano; Pinto; Pappa e Cerri, 2017	PGSB	LCPN	nLLCPN	■	~				~	~	~	~	~
Nakano; Pinto; Pappa e Cerri, 2017	PGSB	LCPN	LCPNB	■	↑				↑	↑	↑	↑	↑
Nakano; Pinto; Pappa e Cerri, 2017	REP	LCPN	nLLCPN	■					↑		↑		↑
Nakano; Pinto; Pappa e Cerri, 2017	REP	LCPN	LCPNB	■	~				~	~	~	~	~
Nakano; Pinto; Pappa e Cerri, 2017	PG-RE	LCPN	nLLCPN	■	↑				↑	↑	↑	↑	↑
Nakano; Pinto; Pappa e Cerri, 2017	PG-RE	LCPN	LCPNB	■	↑				↑	~	↑	↑	↑
Hernandez; Sucar; Morales, 2014	RC	LCPN	Ganho de Informação	■	↑	↑							
Hernandez; Sucar; Morales, 2014	FC	LCPN	Ganho de Informação	■	↑	↑							
Hernandez; Sucar; Morales, 2014	IAPR	LCPN	Ganho de Informação	■	↑	↑							
Hernandez; Sucar; Morales, 2014	MIREX	LCPN	Ganho de Informação	■	↑	↑							

Já a Tabela 5 apresenta os métodos que realizam a predição opcional de nós-folha (NMLNP). Nesta tabela identifica-se os autores, as bases de dados utilizadas conforme mostrado na Tabela 4, o tipo da abordagem hierárquica utilizada e o método de critério de parada utilizado.

Pelo fato de existirem diversos métodos de critérios de parada na literatura, bem como diversas abordagens para resolver diferentes problemas hierárquicos, pode ser difícil saber qual o melhor método para ser aplicado em um determinado problema. Neste caso optou-se por implementar tais métodos da literatura e avaliar seus desempenhos no domínio do problema em questão que é a predição de função de proteínas.

Os métodos de critérios de parada avaliados nesta pesquisa foram testados de forma a utilizar três algoritmos de classificação, MLP (Versões com uma e duas camadas ocultas) e CNN. Os resultados foram apresentados separadamente para uma melhor comparação entre os métodos de critérios de parada, pelo fato desta pesquisa propor novos métodos de critérios de parada independente do algoritmo de classificação utilizado para indução. Tais métodos foram implementados para predição obrigatória de nós-folha (MLNP) e não obrigatória de nós-folha (NMLNP) e comparados por meio da métrica de avaliação Medida-F Hierárquica (HF) nas bases de dados descritas na seção 4.1.

5.0.1 Resultados computacionais dos principais métodos MLNP e NMLNP da literatura

Nesta seção são apresentados os resultados dos principais métodos do estado da arte, tanto abordagens MLNP, quanto NMLNP, bem como são avaliados estatisticamente de uma forma geral tais métodos para o ranqueamento dos mesmo perante a métrica de avaliação HF.

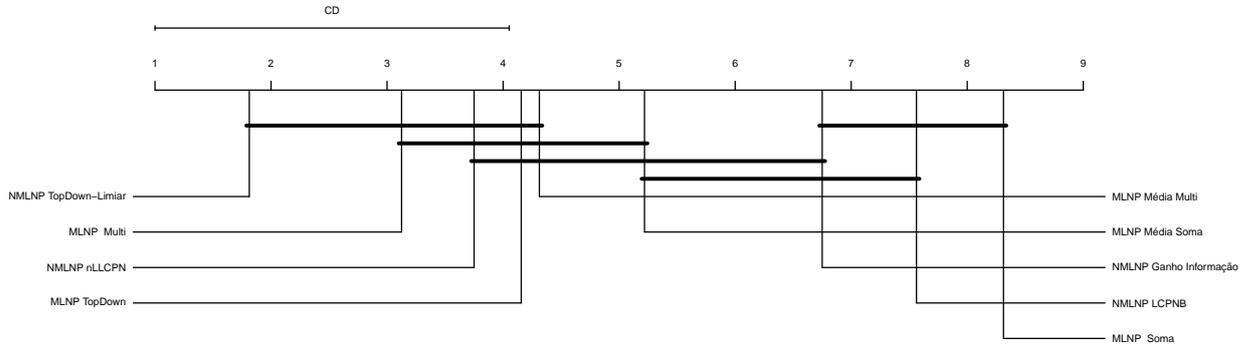
A Tabela 6 apresenta os resultados da métrica de avaliação HF dos principais métodos de critérios de parada da literatura com aplicação do algoritmo de classificação MLP com 1 camada oculta.

Tabela 6 – Resultados dos métodos da literatura com Classificador MLP

Datasets	Métodos Base MLNP						Métodos Base NMLNP			
	TopDown	Multiplicação	Soma	Média Multiplicação	Média-Soma	TopDown-Limiar	nLLCPN	LCPNB	Ganho Informação	
ECinterpro	93,24 ±1,62	93,26 ±1,70	92,42 ±1,84	93,25 ±1,72	93,10 ±1,79	94,01 ±1,62	93,60 ±1,68	91,60 ±1,54	86,85 ±1,78	
ECpfam	92,38 ±1,51	92,01 ±1,81	91,35 ±1,88	92,02 ±1,81	92,19 ±1,72	93,12 ±1,47	92,46 ±2,30	91,39 ±1,78	85,70 ±1,98	
ECprints	90,94 ±1,05	90,46 ±1,34	89,45 ±1,07	90,43 ±1,32	90,89 ±1,10	91,60 ±1,10	91,28 ±1,17	88,92 ±1,25	79,97 ±8,31	
ECprosite	93,09 ±0,97	91,49 ±3,32	91,61 ±1,16	91,47 ±3,33	93,09 ±0,95	93,58 ±1,15	93,49 ±1,18	91,33 ±1,20	86,31 ±1,79	
GPCRinterpro	77,60 ±1,99	75,90 ±5,65	47,11 ±1,88	75,74 ±5,62	74,51 ±5,84	81,15 ±2,12	77,25 ±1,98	74,66 ±5,27	73,08 ±7,06	
GPCRpfam	59,81 ±1,13	55,70 ±1,18	40,07 ±1,04	51,42 ±2,36	46,24 ±1,15	66,52 ±0,79	59,84 ±0,93	48,14 ±3,23	51,52 ±2,33	
GPCRprints	74,35 ±2,14	72,90 ±5,93	49,61 ±2,01	72,34 ±6,20	65,63 ±6,68	78,66 ±2,13	74,37 ±2,27	63,05 ±7,31	70,51 ±5,88	
GPCRprosite	55,19 ±1,53	51,36 ±2,35	37,53 ±1,38	48,48 ±2,20	46,71 ±3,76	60,97 ±1,44	55,38 ±1,44	43,25 ±4,41	48,01 ±2,89	
SM-LD-ECinterpro	48,13 ±3,26	50,30 ±3,34	46,01 ±3,26	50,35 ±3,44	49,10 ±2,04	49,51 ±3,12	47,48 ±3,30	45,33 ±2,71	49,96 ±3,09	
SM-LD-ECpfam	48,53 ±2,66	50,30 ±3,36	46,04 ±3,31	50,24 ±3,37	49,49 ±2,66	49,78 ±2,74	48,54 ±3,97	46,61 ±2,94	49,30 ±3,79	
SM-LD-ECprints	55,19 ±4,42	56,92 ±4,45	51,91 ±5,58	56,80 ±4,41	54,86 ±4,60	56,32 ±4,31	54,35 ±2,94	49,71 ±3,16	56,15 ±4,20	
SM-LD-ECprosite	49,32 ±4,06	52,33 ±3,42	46,59 ±3,58	52,25 ±3,51	51,31 ±2,56	50,53 ±4,07	52,68 ±3,92	51,04 ±2,44	52,09 ±3,08	
SM-LD-GPCRinterpro	73,22 ±2,14	73,72 ±1,84	49,37 ±2,74	72,18 ±2,31	73,58 ±2,35	74,35 ±2,07	72,91 ±1,49	71,58 ±1,50	69,74 ±2,32	
SM-LD-GPCRpfam	74,15 ±1,76	74,74 ±2,29	50,22 ±3,05	73,86 ±2,20	73,74 ±1,58	75,52 ±1,64	73,81 ±2,72	72,75 ±2,87	70,51 ±2,74	
SM-LD-GPCRprints	74,30 ±1,82	74,73 ±1,77	53,92 ±2,41	74,34 ±2,59	74,36 ±1,55	75,53 ±1,89	71,57 ±1,49	69,59 ±1,65	70,98 ±1,85	
SM-LD-GPCRprosite	73,43 ±2,03	74,02 ±2,48	52,97 ±3,60	73,35 ±2,36	72,53 ±2,36	74,82 ±1,77	72,13 ±2,28	71,47 ±2,01	69,72 ±2,81	

A partir destes resultados o teste de Friedman foi aplicado com confiança de 95% para verificar se há diferença entre os métodos implementados. O valor de p retornado por este teste foi de 7.32e-14. Neste caso rejeitou-se a hipótese nula mostrando que há diferença estatística entre os métodos avaliados. Na sequência foi aplicado o teste de Nemenyi para realizar um ranqueamento dos métodos e mostrar graficamente quais métodos apresentam diferença estatística entre si, conforme mostrado na Figura 30.

Figura 30 – Teste Estatístico Nemenyi dos métodos do Estado da Arte com classificador MLP



O teste de Nemenyi apresentado na Figura 30 mostra que o método NMLNP Top-Down ficou em primeiro lugar, seguidos por MLNP Multiplicação, nLLCPN, MLNP Top-Down e MLNP Média Multiplicação, embora o gráfico mostre que, entre estes cinco primeiros métodos da literatura, não houve diferença estatística entre estes avaliados de uma forma geral, pela utilização dos algoritmos de classificação MLP com 1 camada oculta.

Com a finalidade de verificar os resultados com outras configurações de rede, tais métodos foram avaliados, conforme mostrado na Tabela 7. Esta tabela mostra os resultados da métrica de avaliação HF dos principais métodos de critérios de parada da literatura com aplicação do algoritmo de classificação MLP com 2 camadas ocultas.

Tabela 7 – Resultados dos métodos da literatura com Classificador MLP com 2 camadas ocultas

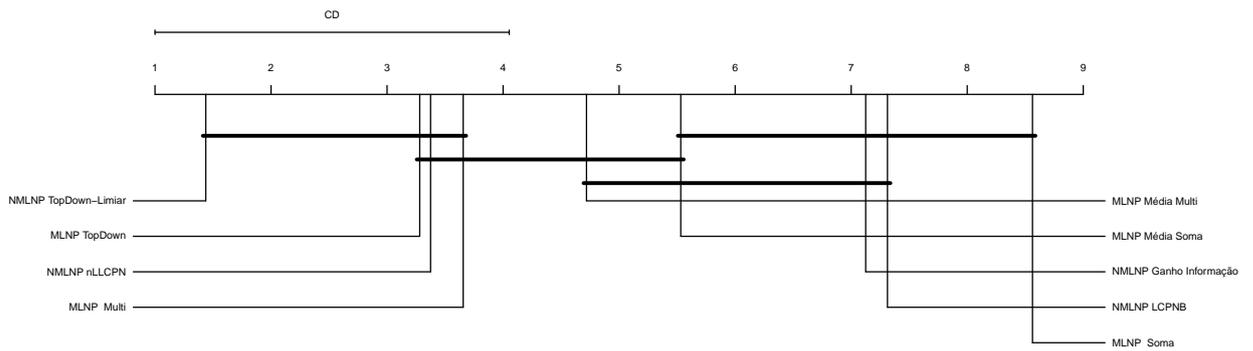
Datasets	Métodos Base MLNP					Métodos Base NMLNP			
	TopDown	Multiplicação	Soma	Média Multiplicação	Média-Soma	TopDown-Limiar	nLLCPN	LCPNB	Ganho Informação
ECinterpro	93,31 ±1,60	93,26 ±1,58	91,17 ±2,99	93,23 ±1,59	93,20 ±1,67	93,84 ±1,76	93,57 ±1,60	91,75 ±1,76	84,38 ±1,89
ECpfam	92,08 ±1,93	92,07 ±1,95	91,08 ±1,90	92,07 ±1,93	92,03 ±2,00	92,50 ±1,84	92,62 ±2,18	91,69 ±1,72	86,21 ±1,56
ECprints	90,65 ±0,96	89,92 ±2,14	88,94 ±1,20	89,98 ±2,09	90,55 ±1,11	91,26 ±1,06	91,07 ±1,09	87,67 ±0,92	82,01 ±2,23
ECprosite	92,78 ±1,10	92,15 ±2,18	90,96 ±1,78	92,20 ±2,15	92,78 ±1,08	93,22 ±1,20	93,32 ±1,16	91,34 ±0,93	85,05 ±2,45
GPCRinterpro	77,67 ±1,87	74,69 ±6,01	49,57 ±2,56	74,48 ±5,90	73,70 ±6,26	81,32 ±2,05	77,42 ±2,09	74,08 ±5,29	70,09 ±8,20
GPCRpfam	59,53 ±0,61	52,79 ±2,87	40,30 ±0,63	48,99 ±3,62	45,73 ±1,25	66,41 ±0,62	59,48 ±0,85	47,91 ±2,97	50,70 ±2,06
GPCRprints	73,95 ±2,21	70,27 ±7,11	50,02 ±1,98	70,25 ±7,01	73,70 ±2,69	78,28 ±2,24	74,31 ±2,29	73,21 ±3,20	68,60 ±6,90
GPCRprosite	55,80 ±1,54	45,69 ±2,69	37,27 ±0,98	43,52 ±3,38	47,37 ±2,74	61,20 ±1,49	54,75 ±1,00	40,89 ±4,15	43,16 ±2,78
SM-LD-ECinterpro	46,68 ±3,61	48,84 ±3,34	42,72 ±3,29	48,80 ±3,35	45,32 ±4,21	48,32 ±3,72	46,68 ±4,10	43,52 ±3,33	47,65 ±3,26
SM-LD-ECpfam	47,71 ±3,86	49,30 ±3,35	43,84 ±3,54	49,35 ±3,18	47,09 ±2,44	49,70 ±3,84	46,95 ±3,14	42,68 ±3,41	48,11 ±3,23
SM-LD-ECprints	53,35 ±3,06	54,27 ±3,76	49,62 ±3,67	54,21 ±3,70	50,78 ±2,98	55,06 ±3,05	53,08 ±4,68	47,70 ±4,95	52,92 ±3,57
SM-LD-ECprosite	47,30 ±6,03	49,81 ±5,11	43,33 ±5,02	49,77 ±5,21	46,64 ±3,38	49,11 ±6,28	50,13 ±3,29	46,58 ±3,27	48,57 ±5,07
SM-LD-GPCRinterpro	67,99 ±2,18	67,61 ±2,61	43,51 ±2,03	65,32 ±3,82	66,81 ±1,87	70,19 ±2,16	67,99 ±1,37	65,98 ±1,35	64,49 ±2,00
SM-LD-GPCRpfam	69,68 ±1,99	69,34 ±2,88	47,96 ±2,84	67,78 ±3,04	67,89 ±2,35	71,92 ±1,71	68,60 ±3,15	67,36 ±3,15	64,81 ±3,08
SM-LD-GPCRprints	68,80 ±2,35	69,14 ±2,71	50,96 ±1,94	68,68 ±2,21	67,99 ±2,19	71,75 ±2,24	67,30 ±2,10	64,91 ±2,03	65,91 ±2,11
SM-LD-GPCRprosite	66,62 ±2,79	67,68 ±2,64	48,26 ±3,08	65,79 ±2,94	65,89 ±2,70	69,73 ±2,48	65,04 ±2,02	63,60 ±2,40	63,25 ±2,49

No teste de Friedman para avaliação dos métodos com classificador MLP com duas camadas ocultas o valor de p retornado foi 2.2e-16. Rejeitando-se a hipótese nula e

portanto mostra que há diferença estatística entre os métodos.

Posteriormente o teste de Nemenyi foi aplicado nestes resultados com a finalidade de ranquear os métodos e mostrar quais deles apresentam diferença estatística, conforme apresentado na Figura 31.

Figura 31 – Teste Estatístico dos métodos do Estado da Arte com classificador MLP (2 camadas ocultas)



O teste de Nemenyi apresentado na Figura 31 mostra que o método NMLNP Top-Down ficou em primeiro lugar, seguidos por MLNP Top-Down, nLLCPN e MLNP Média Multiplicação, embora o gráfico mostre que, entre estes quatro primeiros métodos, não houve diferença estatística entre estes avaliados de uma forma geral com aplicação do algoritmo de classificação MLP com 2 camadas ocultas.

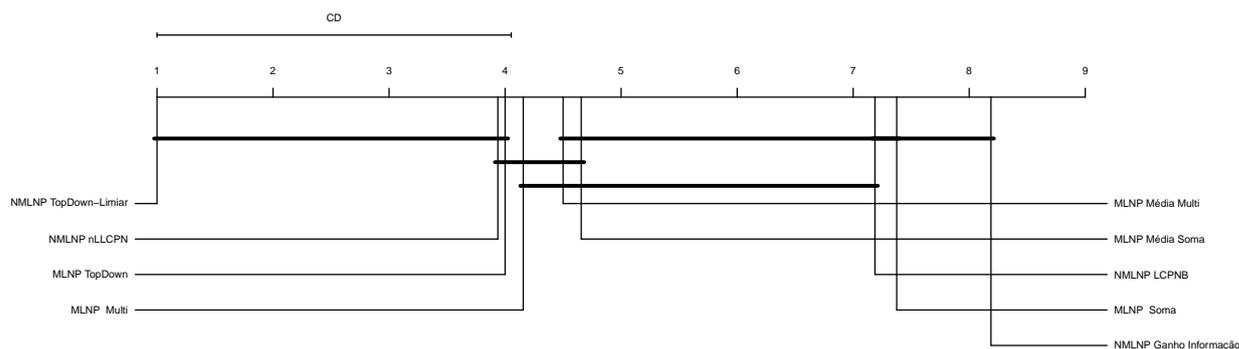
Por fim, a tabela 8 apresenta os resultados da métrica de avaliação HF dos principais métodos de critérios de parada da literatura com aplicação do algoritmo de classificação CNN.

Tabela 8 – Comparativo dos Métodos do Estado da arte com Classificador CNN

Datasets	Métodos Base MLNP					Métodos Base NMLNP				
	TopDown	Multiplicação	Soma	Média Multiplicação	Média-Soma	TopDown-Limiar	nLLCPN	LCPNB	Ganho Informação	
ECinterpro	92,71 ±1,7	76,11 ±13,8	89,54 ±1,4	77,05 ±13,1	92,64 ±1,7	93,59 ±1,7	93,06 ±1,7	89,30 ±1,7	70,16 ±12,1	
ECpfam	91,90 ±1,9	89,73 ±3,3	89,18 ±1,8	89,83 ±3,1	91,83 ±1,9	92,37 ±1,8	92,23 ±1,7	88,88 ±1,6	85,85 ±2,9	
ECprints	89,89 ±0,9	89,77 ±0,9	88,72 ±1,0	89,74 ±0,9	89,66 ±0,9	91,37 ±1,0	90,11 ±1,0	86,85 ±1,1	84,09 ±1,3	
ECprosite	92,44 ±1,3	91,81 ±1,4	90,28 ±1,3	91,76 ±1,4	92,38 ±1,2	93,14 ±1,3	92,58 ±1,4	90,24 ±1,0	84,14 ±2,6	
GPCRinterpro	77,29 ±1,7	72,65 ±5,1	47,96 ±2,3	74,31 ±5,2	64,35 ±2,2	80,82 ±1,6	77,48 ±1,9	68,69 ±6,4	66,35 ±6,4	
GPCRpfam	59,65 ±0,9	44,44 ±9,5	36,82 ±2,2	42,75 ±8,2	43,59 ±1,3	65,76 ±0,6	60,44 ±1,1	47,25 ±3,4	32,00 ±9,4	
GPCRprints	74,18 ±2,2	71,84 ±4,4	47,67 ±1,5	72,99 ±3,8	58,66 ±2,8	78,65 ±2,2	74,20 ±2,2	59,49 ±2,7	67,84 ±3,6	
GPCRprosite	56,40 ±1,8	41,88 ±1,9	35,32 ±1,8	40,53 ±2,7	45,92 ±4,3	61,17 ±1,4	56,02 ±1,5	44,36 ±4,3	37,47 ±2,5	
SM-LD-ECinterpro	58,43 ±3,5	59,26 ±3,6	58,55 ±3,4	59,24 ±3,7	59,39 ±3,7	59,49 ±3,5	57,15 ±3,0	56,38 ±3,0	57,08 ±3,4	
SM-LD-ECpfam	59,26 ±3,5	60,37 ±3,5	59,77 ±2,9	60,36 ±3,5	60,44 ±3,1	60,52 ±3,3	58,18 ±2,9	57,60 ±2,5	58,03 ±3,6	
SM-LD-ECprints	63,70 ±3,0	64,31 ±3,3	64,07 ±3,2	64,36 ±3,4	64,36 ±3,2	64,94 ±2,9	64,48 ±3,3	63,17 ±2,8	61,41 ±3,1	
SM-LD-ECprosite	60,04 ±3,0	60,73 ±3,2	60,11 ±3,1	60,73 ±3,2	60,88 ±3,2	61,14 ±3,0	59,44 ±2,9	57,59 ±2,6	58,39 ±3,4	
SM-LD-GPCRinterpro	79,60 ±1,9	80,05 ±1,8	51,25 ±2,2	79,78 ±1,9	79,43 ±2,4	80,17 ±1,8	79,51 ±1,5	78,57 ±1,4	71,51 ±2,1	
SM-LD-GPCRpfam	81,18 ±1,2	81,45 ±1,4	52,87 ±2,4	81,36 ±1,2	80,58 ±1,4	81,94 ±1,2	79,76 ±2,0	78,52 ±2,1	70,21 ±1,6	
SM-LD-GPCRprints	80,27 ±1,3	80,87 ±1,5	58,43 ±1,9	80,43 ±1,7	80,25 ±1,1	81,47 ±1,2	79,24 ±2,4	77,35 ±2,8	71,02 ±2,2	
SM-LD-GPCRprosite	80,31 ±1,5	80,76 ±1,7	56,19 ±2,5	80,59 ±2,0	80,16 ±1,6	81,13 ±1,3	79,09 ±1,8	78,04 ±1,8	68,36 ±2,4	

No teste de Friedman, para avaliação dos métodos com classificador CNN, o valor de p retornado foi $4.827e-15$. Rejeitando-se a hipótese nula e portanto mostra que há diferença estatística entre os métodos. Portanto, o teste de Nemenyi foi aplicado nestes resultados com a finalidade de ranquear os métodos e mostrar quais métodos apresentam diferença estatística, conforme apresentado na Figura 32.

Figura 32 – Teste Estatístico dos métodos do Estado da Arte com classificador CNN



O teste de Nemenyi apresentado na 32 mostra que o método NMLNP Top-Down ficou em primeiro lugar, seguidos por nLLCPN e MLNP Top-Down. e MLNP Média Multiplicação, embora o gráfico mostre que, entre estes três primeiros métodos, não houve diferença estatística entre estes avaliados de uma forma geral, pela utilização do algoritmo de classificação CNN. .

5.1 Considerações Finais

Apesar de um grande avanço nas técnicas e abordagens apresentadas de Classificação Hierárquica, podemos observar neste capítulo que os problemas relacionados ao critério de parada em abordagens hierárquicas ainda permanecem em aberto. Além disso, baseando-se nos experimentos executados nesta análise extensiva referente aos métodos de critérios da parada do estado da arte, bem como dada as propriedades do problema hierárquico para predição de função de proteínas, adotamos o método NMLNP Top-Down como baseline para a comparação dos métodos propostos, bem como motivou a implementação dos métodos, apresentados na próxima seção, utilizando a estratégia Top-Down, pelo fato desta estratégia ter apresentado melhores resultados em relação aos demais métodos da literatura avaliados neste capítulo.

Outra consideração importante desta análise extensiva é que o caminho que será percorrido até a predição de um determinado nó, delimitado pelo tipo de avaliação Global/Top-Down, têm forte influência nos resultados de predição, principalmente em taxonomias desbalanceadas, nas quais a escolha do caminho por meio da soma, multi-

plicação ou a média pela estratégia de avaliação global pode beneficiar caminhos mais longos ou que apresentam predições inconsistentes em níveis mais inferiores e que possam interferir na escolha do caminho a ser predito. Por exemplo, em problemas que são MLNP e que as taxonomias são balanceadas, adotar estratégias globais pode apresentar melhores resultados, conforme relatado em trabalhos da literatura na seção 3. Porém, para problemas que necessitam de NMLNP e que apresentam taxonomias desbalanceadas, no caso deste trabalho, sugere-se a escolha de uma abordagem Top-Down, pois apresentaram melhores resultados em comparação com as avaliações globais avaliadas nesta pesquisa.

Baseado neste contexto, um determinado critério de parada aplicado a uma escolha de caminho incorreto pode apresentar resultados insatisfatórios. Neste sentido, para o desenvolvimento de uma solução de Classificação Hierárquica Local, na etapa de classificação, tanto a escolha do caminho a ser percorrido, quanto o tipo de critério de parada, apresentam peculiaridades que devem ser analisadas para a aplicação destes métodos em determinados problemas hierárquicos, já citados anteriormente.

6 Métodos Propostos

Neste capítulo serão apresentados os diferentes métodos propostos para lidar com critérios de parada para Predição Não Obrigatória de Nós-Folha (NMLNP) em problemas de Classificação Hierárquica Local. Os quatro primeiros métodos propostos denominados LCPNwFOE, LCPNwVOE, LCPNwFME e LCPNwVME, apresentados respectivamente nas Seções 6.1, 6.2, 6.3 e 6.4, utilizam a entropia como critério de parada. Nestes dois últimos, foi proposta uma nova fórmula baseada na entropia. O quinto método LCPNwLC, apresentado na Seção 6.5, define o critério de parada por meio de um classificador de nível, no qual identifica-se em que nível da árvore deve ser realizada a predição e, por consequência, a parada. Finalizando, o método LCPNwINT, apresentado na Seção 6.6, é proposto a combinação ou interseção de dois ou mais métodos de avaliação diferentes (MLNP ou NMLNP) para a devida predição da classe de uma determinada hierarquia.

6.1 Método LCPNwFOE

No trabalho proposto por [Hernández, Sucar e Morales \(2014\)](#), foi apresentada a utilização do Ganho de Informação para avaliar se a predição feita por um determinado Nó-Pai é aceita ou, no caso de descarte, passa para avaliação do Nó-Pai em um nível acima. Este método foi aplicado usando uma avaliação Bottom-Up e, caso o Ganho de Informação não seja aceitável (menor que zero) para este nó-pai, desconsidera-se a predição desse, subindo para o próximo classificador Nó-Pai, para uma nova avaliação, até que a condição seja satisfeita ou o Nó-Raiz seja alcançado.

Os autores apontaram que o Ganho de Informação para a Predição Não Obrigatória em Nó-Folha obteve bons resultados, mas que há a necessidade de fazer mais experimentos para fortalecer os resultados que foram alcançados. Outro ponto em questão é que seus métodos não foram testados em taxonomias desbalanceadas (sendo esta objeto desta pesquisa).

Fazendo uma reflexão para avaliar o Ganho de Informação para analisar a descida ou a subida em um nó da hierarquia utilizando abordagem de Classificação Hierárquica Local por Nó-Pai pode apresentar inconsistências de propagação de erros na hierarquia, principalmente em taxonomias desbalanceadas e que apresentam uma quantidade reduzida de instâncias para treinamento nos níveis inferiores. Além disso, um outro problema da NMLNP pelo ganho de informação é pelo fato de que as probabilidades que foram atribuídas aos seus nós-filho são de um respectivo classificador (nó-pai) e a probabilidade atribuída ao nó-pai é determinada por um outro classificar acima deste nível. Ou seja, avaliar o ganho de informação do nó-pai em relação aos seus nós-filhos utilizando classificadores diferentes

para a inferência das probabilidades, pode não apresentar uma relação consistente para que seja possível realizar um avanço dentro desta hierarquia. Além disso, a classificação pode ser interrompida pelo fato de uma classe filha, mesmo com probabilidade alta, porém inferior à probabilidade de seu Nó-Pai, não apresentar Ganho de Informação suficiente.

Outro ponto importante para o desenvolvimento do método de critério de parada baseado somente na entropia foi determinada pelo contexto no qual, segundo [Shannon \(1948\)](#), a entropia consegue medir o grau de incerteza de uma variável aleatória, conforme apresentado na Seção 2.4. A Equação da entropia, já apresentada anteriormente, é definida pela Equação 6.1.

$$H(X) = - \sum_{i=0}^m p_i \log_2 p_i \quad (6.1)$$

Este primeiro método proposto, denominado LCPNwFOE (do inglês, Local Classifier per Parent Node with Fixed Original Entropy), aplica-se a Entropia como critério de parada para abordagem NMLNP e para um melhor entendimento do cálculo da entropia aplicado ao contexto de Classificação Hierárquica Local por Nó-Pai, conforme hierarquia mostrada na Figura 33, temos:

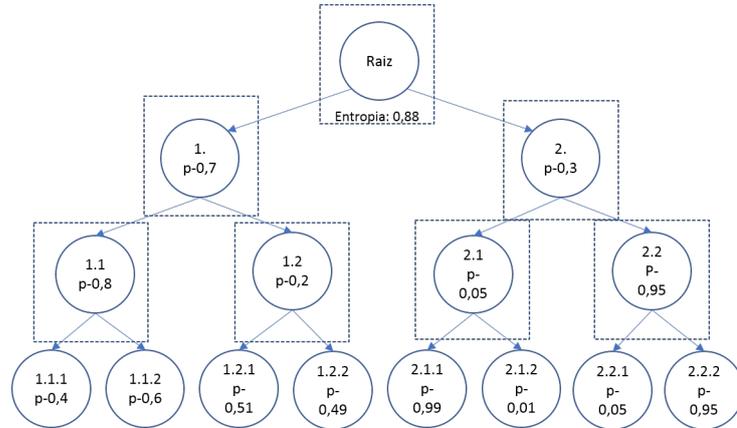
- Dada uma determinada instância de teste X e seguindo a abordagem de avaliação de Cima para Baixo (Top-Down), esta é inferida primeiramente no classificador Nó-Pai (Raiz): C_{Raiz} .
- Este determinado classificador gera duas probabilidades de acordo com a quantidade das suas classes filhas $p(C_{Raiz}(X)) = \{p_1, p_2\}$, sendo $p_1 = 0.7$ e $p_2 = 0.3$, respectivamente.
- Em seguida é feito o cálculo da entropia (H), onde temos: $H(p_1, p_2) = -(0.7 * \log_2 0.7 + 0.3 * \log_2 0.3)$ e resultando em $H(p_1, p_2) = 0.8813$.

Recursivamente são realizados estes cálculos, de forma Top-Down, até atingir um nó-Folha ou até que a predição seja interrompida em um nó-interno, neste caso algum critério de parada seja satisfeito. Neste primeiro método proposto, um limiar baseado na entropia T (Threshold) fixo é utilizada como critério de parada para um determinado nível da hierarquia.

Na etapa de treinamento, cria-se um classificador para cada classe nó-pai, no qual utiliza-se o mesmo classificador base (algoritmo), bem como os mesmos parâmetros para todos os classificadores da hierarquia. Os classificadores que foram avaliados, bem como suas configurações, foram discutidos na Seção 4.2.3.

Na etapa de avaliação deste método proposto, calcula-se a entropia para cada nó-pai da hierarquia. Caso a entropia calculada para cada nó-pai em relação às probabilidades

Figura 33 – Cálculo da Entropia do Nó-Pai Raiz



de suas classes filhas seja maior que o Limiar T , é retornado o nó-atual como resultado. Caso contrário, percorre-se o caminho para o próximo nó-filho de maior probabilidade. Recursivamente, calcula-se a entropia deste novo Nó-Pai com a finalidade de verificar se a condição de parada é atendida ou até que um Nó-Folha seja predito, conforme Algoritmo 1.

Algoritmo 1 Predição recursiva do método LCPNwFOE

```

1: procedure LCPNwFOE_RECURSIVO(noAtual)
2:   if noAtual ≠ noFolha then
3:     probs ← retornarProbabilidadesdasClassesFilhas(noAtual)
4:     h ← calcularEntropia(probs)
5:     if  $h \leq \text{limiarEntropiaFixa}$  then
6:       noAtual ← retornaNoFilhoComMaiorProbabilidade(noAtual)
7:       noAtual ← LCPNwFOE_Recursivo(noAtual)
8:     end if
9:   end if
10:  return noAtual
11: end procedure

```

A determinação do Limiar T da Entropia foi definida a partir de um protocolo empírico realizado no conjunto de validação, no qual foi avaliado e adotado o limiar com a maior média geral da métrica hF entre todas as bases de dados avaliadas nesta pesquisa.

A Figura 34 apresenta um exemplo das probabilidades e do cálculo da entropia a partir do Nó-Raiz até a predição do Nó-Interno (1.1). Neste exemplo a classificação não avançou até o último nível pois a Entropia do respectivo Nó-pai (1.1) apresentou um valor maior que o Limiar T de 0.9. Logo abaixo, foram detalhados os cálculos da entropia de cada Nó-Pai: $H'(Raiz)$, $H'(1)$ e $H'(1.1)$:

$$H(Raiz) = -(0.7 * \log_2 0.7 + 0.3 * \log_2 0.3) \rightarrow H(Raiz) = 0.8813$$

$$H(\text{Raiz}) \leq 0.9(\text{True}) \rightarrow \text{Probabilidade maior} \rightarrow \text{Nó} : (1.)$$

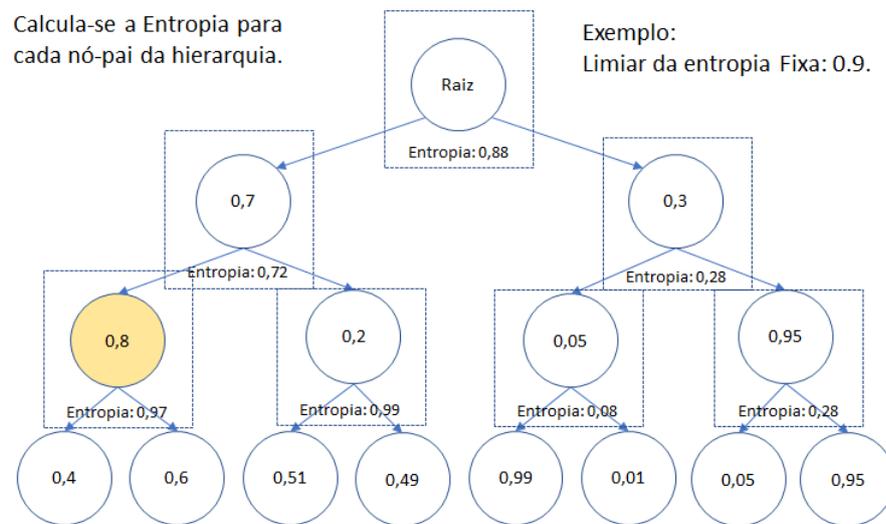
$$H(1) = -(0.8 * \log_2 0.8 + 0.2 * \log_2 0.2) \rightarrow H(1.) = 0.7219$$

$$H(1) \leq 0.9(\text{True}) \rightarrow \text{Probabilidade maior} \rightarrow \text{Nó} : (1.1)$$

$$H(1.1) = -(0.4 * \log_2 0.4 + 0.6 * \log_2 0.6) \rightarrow H(1.1) = 0.9710$$

$$H(1.1) \leq 0.9(\text{False}) \rightarrow \text{Falso} \rightarrow \text{Retorna Nó} : (1.1)$$

Figura 34 – Exemplo com probabilidades e entropia



Verifica-se neste exemplo que, adotando como limiar da entropia o valor de 0.9, o resultado deste método de critério de parada é a classe 1.1, pois a entropia do classificador nó-Pai 1.1, de acordo com as probabilidades geradas das suas classes filhas, é de 0.97. Ou seja, a entropia calculada do respectivo nó é maior que o limiar adotado. Neste caso, é interrompida a etapa de classificação e retorna-se como resultado da predição o respectivo Nó-Pai.

6.1.1 Resultados computacionais do método LCPNwFOE

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNw-FOE, além disso, análises estatísticas das métricas hF e hP são realizadas para a verificação se há diferença estatística entre os métodos avaliados.

Uma questão importante ao lidar com questões de classificação hierárquica é como estabelecer um método de linha de base significativo. Como utilizamos limiares para determinar a parada usando entropia, delimitamos os métodos base Top-Down com Limiar (CECI; MALERBA, 2007) pois apresentou resultados melhores na avaliação preliminar dos métodos do estado da arte (Capítulo 5), bem como o Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014) pois utiliza implicitamente no seu cálculo a entropia.

A Tabela 9 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre o método proposto LCPNwFOE e os métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas e CNN.

Tabela 9 – Resultados da métrica hF dos métodos LCPNwFOE, Top-Down e Ganho de Informação

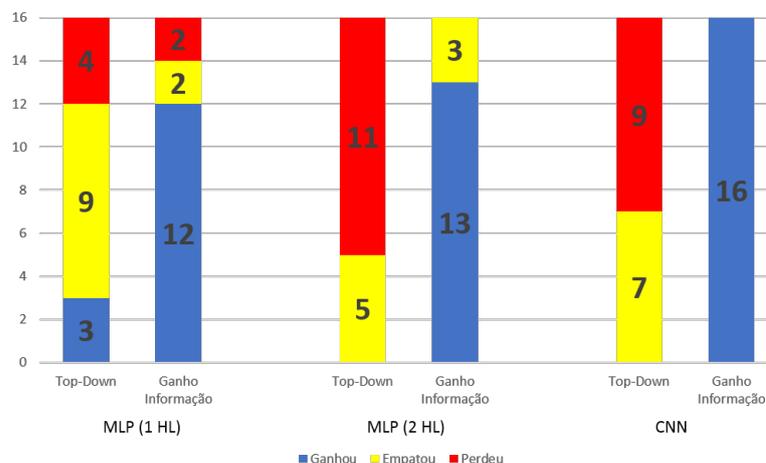
Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCNPw FOE	NMLNP Top-Down	NMLNP Ganho Informação	LCNPw FOE	NMLNP Top-Down	NMLNP Ganho Informação	LCNPw FOE	NMLNP Top-Down	NMLNP Ganho Informação
ECinterpro	94,50	94,01	86,85	93,52	93,84	84,38	93,28	93,59	70,16
ECpfam	93,24	93,12	85,70	92,32	92,50	86,21	92,16	92,37	85,85
ECprints	91,91	91,60	79,97	90,88	91,26	82,01	91,00	91,37	84,09
ECprosite	93,75	93,58	74,03	93,06	93,22	85,05	92,90	93,14	84,14
GPCRinterpro	81,13	81,15	73,08	81,36	81,32	70,09	80,83	80,82	66,35
GPCRpfam	65,54	66,52	51,52	66,27	66,41	50,70	65,83	65,76	32,00
GPCRprints	78,50	78,66	70,51	78,22	78,28	68,60	78,63	78,65	67,84
GPCRprosite	59,52	60,97	48,01	60,92	61,20	43,16	60,97	61,17	37,47
SM-LD-ECinterpro	48,97	49,51	49,96	48,02	48,32	47,65	59,40	59,49	57,08
SM-LD-ECpfam	49,42	49,78	49,30	49,30	49,70	48,11	60,30	60,52	58,03
SM-LD-ECprints	56,17	56,32	56,15	54,57	55,06	52,92	64,73	64,94	61,41
SM-LD-ECprosite	50,18	50,53	52,09	48,60	49,11	48,57	61,05	61,14	58,39
SM-LD-GPCRinterpro	74,22	74,35	69,74	69,99	70,19	64,49	80,12	80,17	71,51
SM-LD-GPCRpfam	75,43	75,52	70,51	71,84	71,92	64,81	81,85	81,94	70,21
SM-LD-GPCRprints	75,41	75,53	70,98	71,52	71,75	65,91	81,33	81,47	71,02
SM-LD-GPCRprosite	74,64	74,82	69,72	69,54	69,73	63,25	81,09	81,13	68,36
Média Geral	72,66	72,87	66,22	71,24	71,49	64,22	76,59	76,73	65,24

A abordagem proposta LCPNwFOE foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hF. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hF, para cada base de dados. Na Figura 35 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 35, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwFOE com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwFOE foi melhor em 3 bases de dados em comparação com o método Top-Down que ganhou em 4 bases e 9 empates. Já na comparação com o método Ganho de informação o

Figura 35 – Comparação pareada do método proposto LCPNwFOE com métodos da literatura por meio da Métrica hF



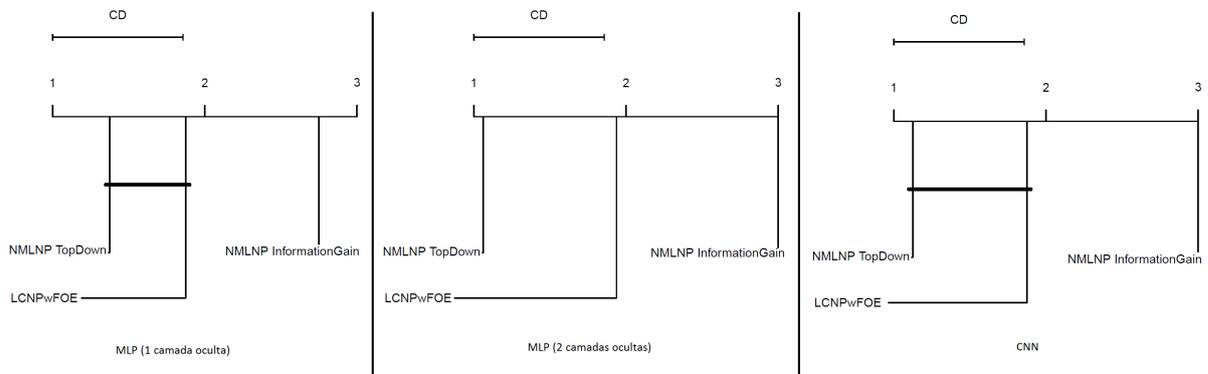
método proposto ganhou em 12 bases de dados, perdeu em 2 bases e empatou com 2 bases.

- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwFOE empatou em 5 bases de dados em relação ao método Top-Down que ainda ganhou em 11 bases. Já o método proposto, ganhou em 13 e empatou em 3 bases de dados, em relação ao método Ganho de Informação.
- Com classificador CNN, o método proposto LCPNwFOE empatou em 7 bases de dados em comparação ao método Top-Down que ganhou em 11 bases. Já o método proposto ganhou nas 16 bases em comparação ao método Ganho de Informação.

Todos os testes estatísticos aplicados nesta pesquisa, Friedman, Wilcoxon e Nemenyi, foram aplicados com confiança de 95%. Pelo fato do teste de Friedman ter mostrado que há diferença estatística entre os resultados das amostras, aplicou-se o teste de Nemenyi, conforme apresentado na Figura 36, para avaliação estatística do desempenho relacionado à métrica de avaliação hF do primeiro método proposto LCPNwFOE em relação aos métodos da literatura NMLNP Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Analisando o teste de Nemenyi dos resultados da métrica de avaliação hF, Figura 36, o método base Top-Down ficou ranqueado em primeiro nas abordagens dos três algoritmos de classificação avaliados. Além disso, não apresentou diferença estatística com o método proposto LCPNwFOE nas abordagens com utilização dos algoritmos MLP (1 camada

Figura 36 – Teste Nemenyi da Métrica hF do método LCPNwFOE



oculta) e CNN. Já o método Ganho de Informação foi inferior estatisticamente aos demais métodos avaliados.

A Tabela 10 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwFOE e dos métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

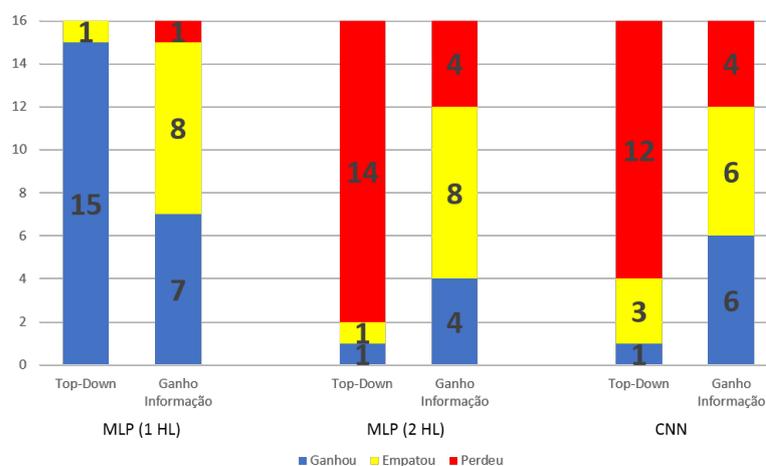
Tabela 10 – Resultados da métrica hP dos métodos LCPNwFOE, Top-Down e Ganho de Informação

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCPNwFOE	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwFOE	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwFOE	NMLNP Top-Down	NMLNP Ganho Informação
ECinterpro	92,99	91,93	93,46	90,73	91,57	93,51	91,20	92,05	76,05
ECpfam	91,33	90,99	92,32	89,59	90,13	92,43	89,74	90,42	90,10
ECprints	90,82	89,95	87,56	88,55	89,61	91,07	90,37	91,56	90,77
ECprosite	92,01	91,63	79,68	90,64	91,30	92,74	91,29	91,96	90,83
GPCRinterpro	87,62	87,67	84,71	86,92	87,60	83,19	87,41	87,91	78,77
GPCRpfam	88,70	87,22	67,03	87,07	87,48	68,71	88,93	89,42	44,67
GPCRprints	87,49	86,31	83,02	86,19	86,06	80,91	87,38	88,09	80,07
GPCRprosite	81,03	76,61	62,70	79,17	77,85	58,05	81,29	81,00	50,74
SM-LD-ECinterpro	55,03	52,73	50,91	50,86	52,69	49,09	60,40	61,28	60,29
SM-LD-ECpfam	54,43	52,30	50,51	52,26	54,71	49,52	61,13	62,19	60,80
SM-LD-ECprints	59,03	57,35	57,62	55,38	57,09	54,72	65,46	66,39	65,07
SM-LD-ECprosite	54,02	52,42	53,61	49,96	52,25	50,19	61,92	62,71	61,34
SM-LD-GPCRinterpro	78,27	76,33	78,63	73,90	74,47	73,73	81,43	81,96	83,98
SM-LD-GPCRpfam	79,34	77,59	80,06	75,49	76,18	75,86	83,14	83,74	85,25
SM-LD-GPCRprints	80,59	77,91	79,47	77,01	77,78	75,40	82,82	83,11	84,39
SM-LD-GPCRprosite	79,31	77,20	78,97	74,96	75,55	73,82	82,12	82,43	84,88
Média Geral	78,25	76,63	73,89	75,54	76,39	72,83	80,38	81,01	74,25

A abordagem proposta LCPNwFOE foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par,

utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 37 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Figura 37 – Comparação pareada do método proposto LCPNwFOE com métodos da literatura por meio da Métrica hP



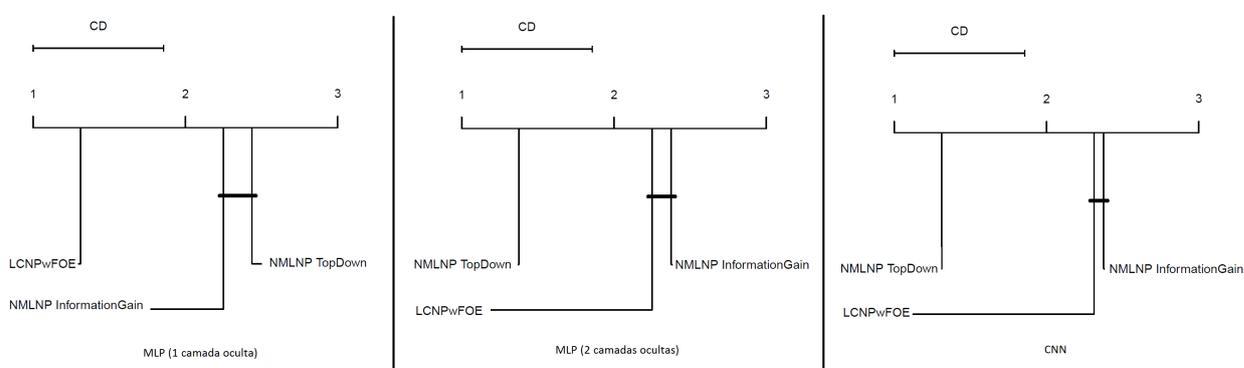
De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 37, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwFOE com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwFOE foi melhor em 15 bases de dados em comparação com o método Top-Down e empatou em apenas 1 base. Já na comparação do método proposto com o método base Ganho de Informação, o método LCPNwFOE foi melhor em 7 bases e 8 empates perante ao método Ganho de Informação que foi melhor em apenas 1 base.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwFOE foi melhor em 1 base de dado e empatou em 1 em comparação com o método Top-Down que ganhou em 14 bases. Já na comparação do método proposto com o método base Ganho de Informação, o método LCPNwFOE foi melhor em 4 e empatou em 8 bases perante ao método Ganho de Informação que foi melhor em 4 bases.
- Com classificador CNN, o método proposto LCPNwFOE foi melhor em apenas 1 base e empatou em 3 em comparação com o método Top-Down que ganhou em

12 bases. Já na comparação do método proposto com o método base Ganho de Informação, o método LCPNwFOE foi melhor em 6 bases e empatou em 6 perante ao método Ganho de Informação que foi melhor em 4 bases.

A Figura 38 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwFOE em relação aos métodos da literatura Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Figura 38 – Teste Nemenyi da Métrica hP com classificadores MLP (1 e 2 camadas ocultas) e CNN



Analisando o teste de Nemenyi pela métrica de avaliação hP, Figura 38, o método proposto LCPNwFOE foi superior estatisticamente comparado com os demais métodos da literatura pela abordagem com classificador MLP com 1 camada oculta. Já nas outras abordagens (MLP com 2 camadas ocultas e CNN) o método proposto LCPNwFOE não apresentou diferença estatística para o método Ganho de Informação. Ainda analisando estas duas abordagens citadas anteriormente, o método Top-Down apresentou superioridade estatística em relação aos métodos LCPNwFOE e Ganho de Informação.

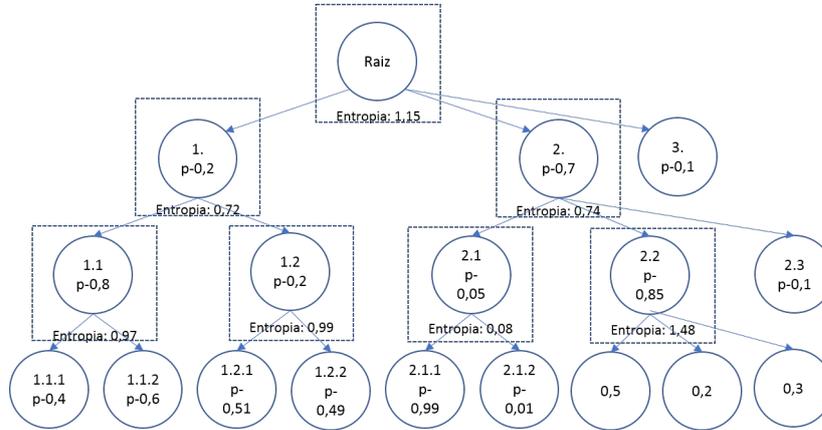
6.2 Método LCPNwVOE

Numa hierarquia de árvore binária ou que apresente quantidades iguais de Nós-Filhos para todos os nós-pais, pode ser interessante adotar o método LCPNwFOE. Porém, para hierarquias onde existem diferentes quantidades de nós-filho em determinados nós-Pai, este método pode não ser eficiente. Isto deve-se ao fato de que a entropia tende a aumentar quando há uma maior distribuição de probabilidades de uma variável aleatória discreta.

De acordo com a Figura 39, podemos verificar nos Nós-Pais ("Raiz", "2." e "2.2") que, com a inserção de mais um filho destes Nós-Pais, em relação ao exemplo anterior,

aumentou o valor da Entropia. Nesse caso, adotar um limiar de Entropia Fixa pode não apresentar a melhor solução para uma taxonomia desbalanceada.

Figura 39 – Exemplo com probabilidades e entropia



Desta forma, neste método proposto, denominado LCPNwVOE (do inglês, Local Classifier per Parent Node with Variable Original Entropy), utiliza-se um limiar (T) baseado na entropia máxima de acordo com a quantidade de classes filhas do nó em questão. Ou seja, devido à oscilação dos valores da entropia em relação à quantidade de filhos em taxonomias desbalanceadas, deve-se avaliar a entropia das probabilidades geradas por este classificador e aferir com o Limiar T ajustado por uma porcentagem da entropia máxima para cada nó-pai, ao contrário do método de Entropia Fixa, que define um Limiar T Fixo para todos os nós-pai da hierarquia.

Para a definição da entropia máxima, no caso de haver N eventos equiprováveis ($\frac{1}{N}$), a entropia associada com a observação desta variável é $\log_2(N)$ (SHANNON, 1948). Ou seja, com o número de classes filhas de um respectivo nó pai (CQ), a entropia máxima é definida por:

$$H(X) = \sum_{i=1}^{CQ} p_i \log_2 p_i = -CQ * \frac{1}{CQ} * \log_2 \frac{1}{CQ} \tag{6.2}$$

$$H(X) = -\log_2 \frac{1}{CQ} = -\log_2 CQ^{-1} = \log_2 CQ$$

A Equação 6.3 mostra a fórmula para a realização do cálculo do limiar baseado na entropia máxima, de acordo com a quantidade de classes filhas e do percentual (porcentagemAjuste) que será adotado da entropia máxima.

$$T = \log_2 qtdeFilhos * porcentagemAjuste \tag{6.3}$$

A Tabela 11 mostra um comparativo da definição de limiares de acordo com a quantidade de probabilidades das classes dos métodos LCPNwFOE e LCPNwVOE, baseado em 90% (porcentagem de ajuste) da entropia máxima.

Tabela 11 – Comparativo de limiares dos métodos LCPNwFOE e LCPNwVOE

Quantidade de classes Filhas	Limiar (T)		
	$h(\text{Max})$	LCPNwFOE (90%)	LCPNwVOE (90%)
2	1,000	0,90	0,900
3	1,585	0,90	1,426
4	2,000	0,90	1,800
5	2,322	0,90	2,090
6	2,585	0,90	2,326
7	2,807	0,90	2,527
8	3,000	0,90	2,700
9	3,170	0,90	2,853
10	3,322	0,90	2,990

Desta forma, adota-se os seguintes passos para definir o limiar (T), baseado na entropia máxima de um determinado nó-pai da hierarquia:

- Primeiramente calcula-se o valor da entropia máxima de acordo com a quantidade total de classes filhas desse respectivo nó.
- Em seguida, para a realização do cálculo do Limiar (T), aplica-se uma porcentagem em relação a entropia máxima, resultando, neste caso, em um limiar dinâmico (variável) de um determinado nó-pai.

No Algoritmo 2 é mostrada a função que calcula o limiar (T) por meio da entropia máxima, de acordo com a quantidade de classes filhas de um determinado nó-pai da hierarquia e do percentual de ajuste adotado. Por exemplo: caso seja acionada esta função com a quantidade de classes filhas igual a 3 e com o percentual de ajuste de 90%, o limiar retornado por essa função será 1,426.

Algoritmo 2 Definição do limiar baseado na entropia máxima

- 1: **procedure** CALCULARLIMIARLCPNwVOE(*qtdeFilhos*, *porcentagemAjuste*)
 - 2: $hMaxima \leftarrow \log_2 qtdeFilhos$
 - 3: $T \leftarrow hMaxima \times porcentagemAjuste$
 - 4: **return** T ▷ Retorna o limiar da entropia de um nó-pai
 - 5: **end procedure**
-

Já o Algoritmo 3 apresenta a etapa de predição de uma determinada instância de teste, realizada de forma Top-Down. O caminho que será seguido pela hierarquia é executado de forma recursiva, o qual inicia-se pelo classificador nó-pai Raiz e segue em direção até um nó-folha ou até que um critério de parada, neste caso baseado em um percentual da entropia máxima, seja atingido.

Algoritmo 3 Predição do método LCPNwVOE baseada na entropia máxima

```

1: procedure RETORNARNoLCPNwVOE_RECURSIVO(noAtual, porcAjuste)
2:   if noAtual = noFolha then
3:     return noAtual
4:   end if
5:   probasFilhos ← retornarProbabilidadesClassesFilhas(noAtual)
6:   H ← calcularEntropia(probasFilhos)
7:   Q ← quantidadeFilhos(noAtual)
8:   T ← calcularLimiarLCPNwVOE(Q, porcAjuste)
9:   if H ≤ T then
10:    noAtual ← retornaNoFilhocomMaiorProbabilidade(noAtual)
11:    noAtual ← retornarNoLCPNwVOE_Recursivo(noAtual, porcAjuste)
12:   end if
13:   return noAtual
14: end procedure

```

Para a realização da predição em um nó-interno (NMLNP) deste método proposto LCPNwVOE, conforme descrito no Algoritmo 3, adotam-se as seguintes premissas:

- Caso a entropia calculada do nó-pai atual, referente às probabilidades dos seus nós-filhos, seja menor que o limiar ajustado (da porcentagem da entropia máxima daquele respectivo nó-pai), então caminha para o nó-filho com o maior valor de probabilidade; e
- Caso a entropia calculada do nó-pai atual, referente às probabilidades dos seus nós-filhos, seja maior que o limiar ajustado (da porcentagem da entropia máxima daquele respectivo nó-pai), então não caminha na hierarquia e retorna o nó-pai atual, realizando o critério de parada, baseado na entropia dinâmica.

A Figura 40 apresenta uma hierarquia com suas respectivas probabilidades e seus limiares de Entropia Variável. O cálculo da entropia máxima nos nós: Raiz, 2. e 2.2 tem o valor de: 1,584, pois apresentam 3 classes filhas em cada nó-pai. Já nos demais nós, onde contém apenas duas classes filhas, a entropia máxima é 1. Na sequência, estipula-se uma porcentagem desse valor máximo da entropia para cada nó-pai. Para a porcentagem de 90% do valor máximo, o limiar (T) dos nós RAIZ, 2 e 2.2 é de 1,42; e nos demais nós é de 0,9.

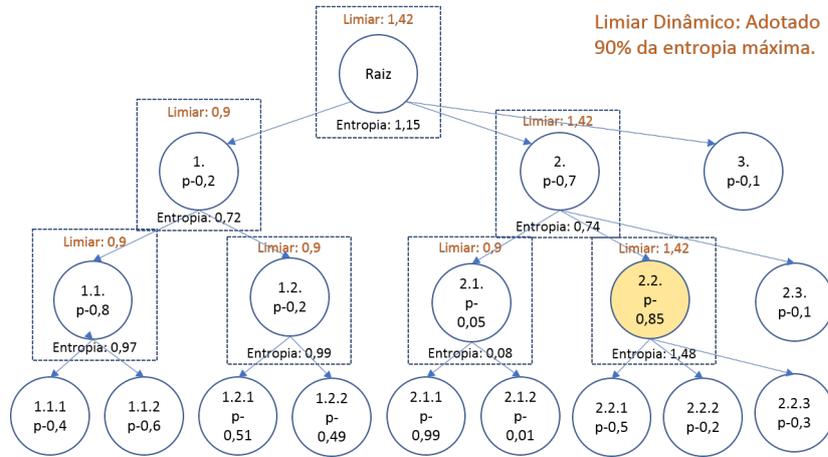
No exemplo da Figura 40, apresenta-se, passo a passo, os cálculos em cada Nó-pai e a avaliação realizada até o retorno do respectivo nó-predito:

$$H(\text{Raiz}) = -(0.2 * \log_2 0.2 + 0.7 * \log_2 0.7 + 0.1 * \log_2 0.1) \rightarrow H(\text{Raiz}) = 1.1568$$

$$H(\text{Raiz}) \leq 1.268(\text{True}) \rightarrow \text{Nó com maior probabilidade} \rightarrow \text{Nó : (2.)}$$

$$H(2) = -(0.05 * \log_2 0.05 + 0.85 * \log_2 0.85 + 0.1 * \log_2 0.1) \rightarrow H(2) = 0.74$$

Figura 40 – Exemplo com probabilidades e entropia



$H(2) \leq 1.268(True) \rightarrow$ Nó com maior probabilidade \rightarrow Nó : (2.2)

$H(2.2) = -(0.5 * \log_2 0.5 + 0.2 * \log_2 0.2 + 0.3 * \log_2 0.3) \rightarrow H(2.2) = 1.4855$

$H(2.2) \leq 1.268(False) \rightarrow$ Falso \rightarrow Retorna Nó: (2.2)

6.2.1 Resultados computacionais do método LCPNwVOE

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNwVOE, além disso, análises estatísticas das métricas hF e hP são realizadas para a verificação se há diferença estatística entre os métodos avaliados.

Os métodos bases que foram utilizados para comparação deste método proposto foram os mesmos utilizados no método LCPNwFOE, detalhados na seção 6.1.1. Ou seja, NMLNP Top-Down (CECI; MALERBA, 2007) e Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014).

A Tabela 12 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre o método proposto LCPNwVOE e os métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas; e CNN.

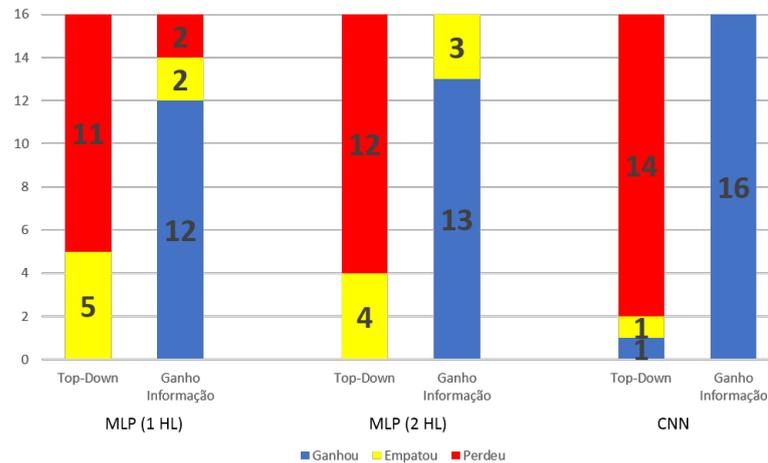
A abordagem proposta LCPNwVOE foi comparada de modo pareado com as outras abordagens da literatura mencionadas. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hF, para cada base de dados. Na Figura 41 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e

Tabela 12 – Resultados da métrica HF dos métodos LCPNwVOE, Top-Down e Ganho de Informação

Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCNPwVOE	NMLNP Top-Down	NMLNP Ganho In-formação	LCNPwVOE	NMLNP Top-Down	NMLNP Ganho In-formação	LCNPwVOE	NMLNP Top-Down	NMLNP Ganho In-formação
ECinterpro	93,97	94,01	86,85	93,53	93,84	84,38	92,86	93,59	70,16
ECpfam	93,22	93,12	85,70	92,53	92,50	86,21	92,10	92,37	85,85
ECprints	91,35	91,60	79,97	91,03	91,26	82,01	90,08	91,37	84,09
ECprosite	93,56	93,58	74,03	93,26	93,22	85,05	92,74	93,14	84,14
GPCRinterpro	80,84	81,15	73,08	80,95	81,32	70,09	80,50	80,82	66,35
GPCRpfam	66,37	66,52	51,52	66,08	66,41	50,70	66,09	65,76	32,00
GPCRprints	78,08	78,66	70,51	77,71	78,28	68,60	78,12	78,65	67,84
GPCRprosite	60,66	60,97	48,01	61,22	61,20	43,16	61,29	61,17	37,47
SM-LD-ECinterpro	49,03	49,51	49,96	48,00	48,32	47,65	59,11	59,49	57,08
SM-LD-ECpfam	49,64	49,78	49,30	49,46	49,70	48,11	59,96	60,52	58,03
SM-LD-ECprints	55,67	56,32	56,15	54,30	55,06	52,92	64,12	64,94	61,41
SM-LD-ECprosite	50,22	50,53	52,09	48,60	49,11	48,57	60,59	61,14	58,39
SM-LD-GPCRinterpro	73,63	74,35	69,74	68,90	70,19	64,49	79,70	80,17	71,51
SM-LD-GPCRpfam	74,60	75,52	70,51	70,67	71,92	64,81	81,44	81,94	70,21
SM-LD-GPCRprints	74,62	75,53	70,98	70,15	71,75	65,91	80,61	81,47	71,02
SM-LD-GPCRprosite	73,91	74,82	69,72	68,29	69,73	63,25	80,43	81,13	68,36
Média Geral	72,46	72,87	66,22	70,92	71,49	64,22	76,23	76,73	65,24

2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Figura 41 – Comparação pareada do método proposto LCPNwVOE com métodos da literatura



Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 41, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwVOE com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

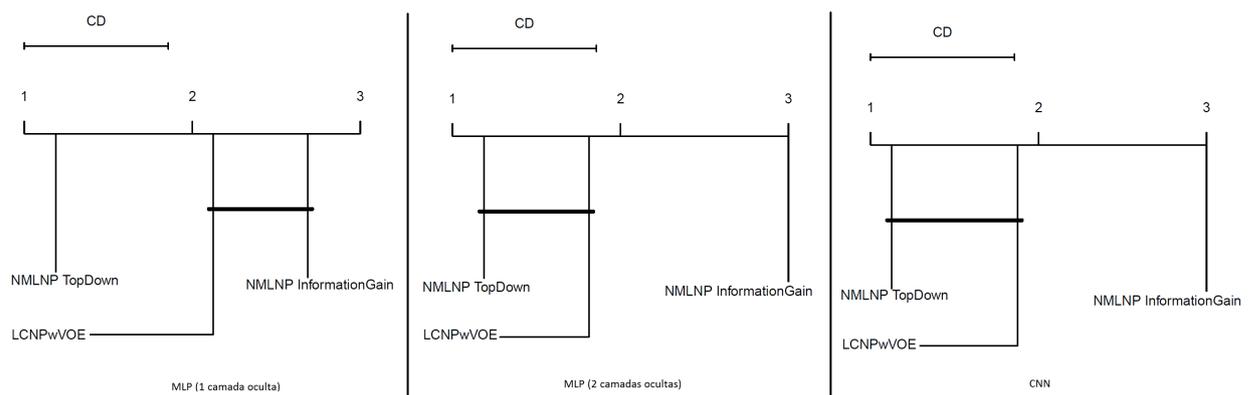
- Com classificador MLP com 1 camada oculta, o método proposto LCPNwVOE foi melhor em 12 e empatou em 2 bases de dados em comparação com o método Ganho

de Informação que ganhou apenas em 2 bases. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVOE obteve 5 empates perante ao método Top-Down que foi melhor nas outras 11 bases.

- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwVOE foi melhor em 13 e empatou em 3 bases de dados em comparação com o método Ganho de Informação. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVOE foi obteve 4 empates e perdeu nas outras 12 bases de dados em comparação com o método Top-Down.
- Com classificador CNN, o método proposto LCPNwVOE foi melhor em todas as bases de dados em comparação com o método Ganho de Informação. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVOE foi melhor em 1 e empatou em 1 base comparado ao método Top-Down que foi melhor nas outras 14 bases.

A Figura 42 apresenta o teste de Nemenyi dos resultados da métrica hF para avaliação estatística do método proposto LCPNwVOE em relação aos métodos da literatura NMLNP Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Figura 42 – Teste Nemenyi da Métrica hF com classificadores MLP (1 e 2 camadas ocultas) e CNN



Pelos testes de Nemenyi realizados nos resultados dos métodos pela métrica de avaliação hF, Figura 42, o método proposto LCPNwVOE ficou ranqueado em segundo lugar em todas as abordagens com algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN. Além disso, não obteve diferença estatística com o método Top-Down nas abordagens com algoritmos de classificação MLP com 2 camadas ocultas e CNN. Ainda nestas duas abordagens o método proposto LCPNwVOE apresentou diferença estatística

perante o método Ganho de Informação. O método base Top-Down ficou ranqueado em primeiro nas três abordagens de classificação avaliadas.

A Tabela 13 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwVOE e dos métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

Tabela 13 – Resultados da métrica hP dos métodos LCPNwVOE, Top-Down e Ganho de Informação

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCPNwVOE	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwVOE	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwVOE	NMLNP Top-Down	NMLNP Ganho Informação
ECinterpro	91,74	91,93	93,46	90,81	91,57	93,51	90,14	92,05	76,05
ECpfam	91,19	90,99	92,32	90,20	90,13	92,43	89,48	90,42	90,10
ECprints	89,36	89,95	87,56	89,01	89,61	91,07	87,91	91,56	90,77
ECprosite	91,36	91,63	79,68	91,23	91,30	92,74	90,51	91,96	90,83
GPCRinterpro	87,04	87,67	84,71	87,17	87,60	83,19	86,70	87,91	78,77
GPCRpfam	80,81	87,22	67,03	83,68	87,48	68,71	84,53	89,42	44,67
GPCRprints	84,02	86,31	83,02	83,71	86,06	80,91	84,72	88,09	80,07
GPCRprosite	73,20	76,61	62,70	74,60	77,85	58,05	76,10	81,00	50,74
SM-LD-ECinterpro	52,22	52,73	50,91	52,72	52,69	49,09	59,15	61,28	60,29
SM-LD-ECpfam	51,69	52,30	50,51	54,22	54,71	49,52	59,68	62,19	60,80
SM-LD-ECprints	55,14	57,35	57,62	54,44	57,09	54,72	63,12	66,39	65,07
SM-LD-ECprosite	51,48	52,42	53,61	51,33	52,25	50,19	60,32	62,71	61,34
SM-LD-GPCRinterpro	73,60	76,33	78,63	70,01	74,47	73,73	79,38	81,96	83,98
SM-LD-GPCRpfam	75,34	77,59	80,06	72,35	76,18	75,86	81,34	83,74	85,25
SM-LD-GPCRprints	73,72	77,91	79,47	71,23	77,78	75,40	79,81	83,11	84,39
SM-LD-GPCRprosite	73,54	77,20	78,97	69,97	75,55	73,82	79,82	82,43	84,88
Média Geral	74,72	76,63	73,89	74,17	76,39	72,83	78,29	81,01	74,25

A abordagem proposta LCPNwVOE foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 43 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 43, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwVOE com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwVOE foi melhor em 4 e empatou em 5 bases de dados em comparação com o método Ganho

Figura 43 – Comparação pareada do método proposto LCPNwVOE com métodos da literatura por meio da Métrica hP



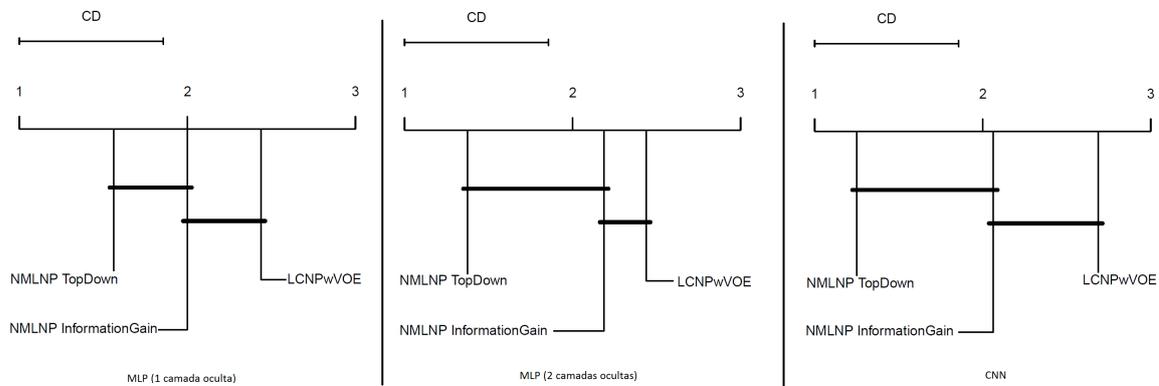
de Informação que foi melhor em 7 bases de dados. Já na comparação do método proposto com o método base Top-down, o método LCPNwVOE empatou em 4 bases perante ao método Top-Down que ganhou em 12 bases.

- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwVOE foi melhor em 4 e empatou em 5 bases de dados em comparação com o método Ganho de Informação que foi melhor em 7 bases de dados. Já na comparação do método proposto com o método base Top-down, o método LCPNwVOE obteve empate em 4 bases de dados perante ao método Top-Down que foi melhor nas outras 12 bases.
- Com classificador CNN, o método proposto LCPNwVOE foi melhor em 5 e empatou em 3 bases de dados em comparação com o método Ganho de Informação que foi melhor em 8 bases. Já na comparação do método proposto com o método base Top-down, o método LCPNwVOE foi inferior estatisticamente em todas as 16 bases de dados comparado com o método Top-Down.

A Figura 44 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwVOE em relação aos métodos da literatura Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Analisando os testes de Nemenyi pela métrica de avaliação hP, Figura 44, o método proposto LCPNwVOE não apresentou diferença estatística comparado ao método Ganho de Informação nas três abordagens utilizando respectivamente os algoritmos MLP (1 e

Figura 44 – Teste Nemenyi da Métrica hP com classificadores MLP (1 e 2 camadas ocultas) e CNN



2 camadas ocultas) e CNN. O método Top-Down ficou ranqueado em primeiro, seguido pelos métodos Ganho de Informação e LCPNwVOE nas três abordagens.

6.3 Método LCPNwFME

Nesta seção é apresentado o método proposto denominado LCPNwFME (Local Classifier per Parent Node with Fixed Modified Entropy), bem como a motivação para o ajuste no cálculo da Entropia para obtenção de melhores resultados, em comparação com a Entropia Original, aplicados em problemas de Classificação Hierárquica. Além disso, são apresentados os resultados computacionais comparados com os métodos da literatura NMLNP Top-Down (CECI; MALERBA, 2007) e Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014).

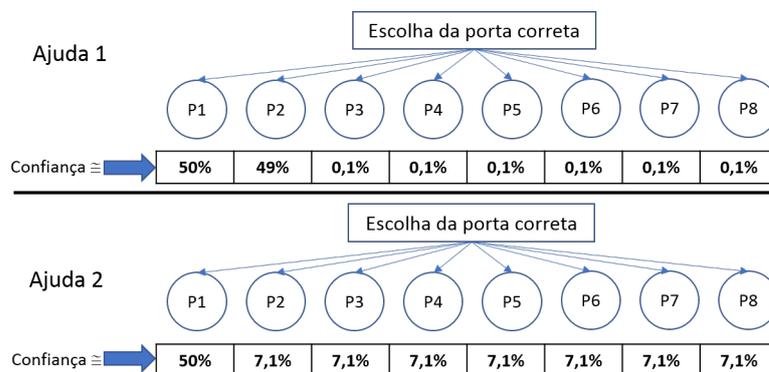
6.3.1 Motivação para ajuste na Fórmula da Entropia

De acordo com a definição de Epstein (1986) que sintetiza o pensamento de Shannon de uma forma objetiva, e parece corresponder melhor à essência da Teoria da Informação, a Informação é uma redução de incerteza, oferecida quando se obtém resposta a uma pergunta.

Para o entendimento da motivação para a alteração da fórmula da Entropia, aplicados neste trabalho para Classificação Hierárquica, faz-se uma analogia de uma forma simples baseada na escolha de uma porta de acordo com suas probabilidades de acerto. Conforme mostrado na Figura 45, deve-se escolher a porta correta observando as probabilidades de ajuda (acerto) para cada porta.

Qual porta será escolhida? Provavelmente a porta 1, pois apresenta uma probabilidade maior em relação as demais portas. Porém, em um contexto mais abrangente, onde

Figura 45 – Exemplo para escolha da porta correta



pretendemos avaliar a ajuda que foi fornecida levando-se em consideração as probabilidades das outras portas. Podemos fazer o seguinte questionamento: Qual a ajuda mais interessante para determinar a escolha da porta correta? A ajuda 1 ou ajuda 2?

Intuitivamente, é provável que escolheremos a ajuda 2. Pois nesta, a porta P1 tem uma probabilidade bem maior do que as outras portas, em comparação com as probabilidades das portas da ajuda 1.

Portanto, baseado nestes questionamentos vamos analisar a Entropia, que mede o grau de incerteza de variáveis aleatórias discretas, para nos auxiliar a determinar a melhor ajuda para a escolha da porta correta.

Calculando-se as Entropias das duas ajudas temos, $H(\text{ajuda 1}) = 1.09$ e $H(\text{ajuda 2}) = 2.40$. Conforme resultados apresentados no cálculo da Entropia das duas ajudas, a melhor escolha será a Ajuda 1, pois esta apresenta uma Entropia (incerteza) menor. No entanto, no aspecto da confiança para a escolha em um cenário computacional, onde pretendemos obter uma probabilidade mais alta para a Classe Verdadeira e probabilidades mais baixas para as demais classes, o auxílio da Entropia para a determinação de uma resposta pode apresentar problemas. Nota-se que nesse exemplo apresentado, a Ajuda 2 apresenta uma maior confiança para a escolha da porta em relação a Ajuda 1.

Ainda analisando o cálculo da Entropia com variáveis aleatórias distintas e fictícias, conforme mostrado na Figura 46, observa-se que, mesmo com uma variável com probabilidade alta e com as demais probabilidades similares, a Entropia tende a ser maior. Por exemplo, a instância 5 apresenta uma Entropia maior do que a instância 6, sendo que ao realizar uma escolha simples entre estas baseado na confiança, a escolha geralmente seria da instância 5. Nesse caso, utilizar um limiar T (Threshold) com a Entropia pode apresentar problemas de inconsistências de acordo com o exemplo apresentado.

Com base na definição complementar de [Shannon e Weaver \(1963\)](#), a entropia nada mais é do que a auto-informação média para cada símbolo no repertório ($\log p_i$) ponderada

Figura 46 – Exemplo cálculos da Entropia para determinadas probabilidades

Instância	Probabilidades Classes Filhas								H
1	0,9	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,741
2	0,9	0,09	0,001	0,002	0,002	0,002	0,002	0,001	0,541
3	0,8	0,0286	0,0286	0,0286	0,0286	0,0286	0,0286	0,0286	1,283
4	0,8	0,19	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	0,805
5	0,7	0,05	0,05	0,04	0,04	0,04	0,04	0,04	1,721
6	0,7	0,29	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	0,970
7	0,6	0,0571	0,0571	0,0571	0,0571	0,0571	0,0571	0,0571	2,094
8	0,6	0,394	0,001	0,001	0,001	0,001	0,001	0,001	1,031
9	0,5	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	2,404
10	0,5	0,49	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	1,097
11	0,4	0,0857	0,0857	0,0857	0,0857	0,0857	0,0857	0,0857	2,655
12	0,4	0,39	0,035	0,035	0,035	0,035	0,035	0,035	2,074
13	0,3	0,1	0,1	0,1	0,1	0,1	0,1	0,1	2,846
14	0,3	0,2	0,0833	0,0833	0,0833	0,0833	0,0833	0,0833	2,778
15	0,2	0,19	0,1017	0,1017	0,1017	0,1017	0,1017	0,1017	2,931
16	0,2	0,1143	0,1143	0,1143	0,1143	0,1143	0,1143	0,1143	2,968
17	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	3,000

por sua frequência (pi). Nesse caso, para uma melhor avaliação aplicada no contexto da classificação hierárquica, neste trabalho, propomos uma nova fórmula baseada na entropia somando a média de autoinformação $-\sum_{i=1}^m \frac{\log_2 p_i}{m}$, conforme mostrado na Equação 6.4. A coluna H' da Figura 47 mostra o valor da entropia usando a nova fórmula proposta. Além disso, podemos observar que com um limiar baseado neste novo valor de entropia dessas instâncias fictícias, o bloqueio (critério de parada) pode ser melhor executado em um determinado nó da hierarquia.

$$H'(X) = -\sum_{i=0}^m p_i \log_2 p_i * -\sum_{i=0}^m \frac{\log_2 p_i}{m} \tag{6.4}$$

Figura 47 – Exemplo ajuste de Entropia H'

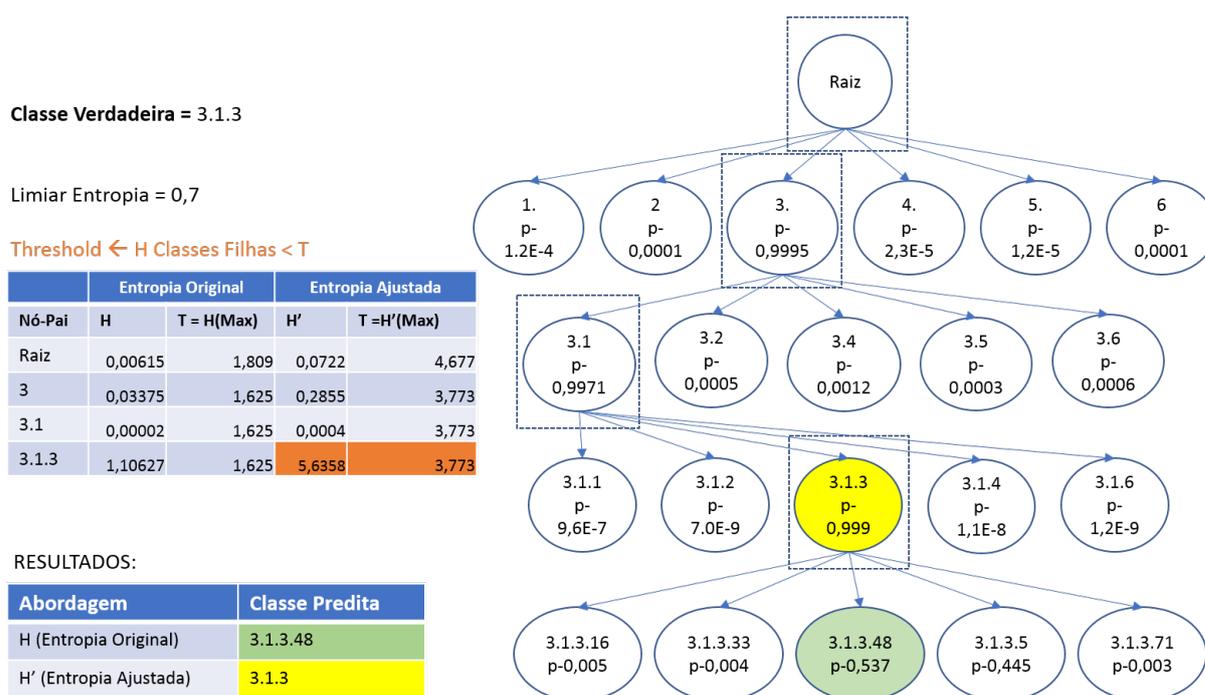
Instância	Probabilidades Classes Filhas								H	H'
1	0,9	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,741	4,045
2	0,9	0,09	0,001	0,002	0,002	0,002	0,002	0,001	0,541	4,019
3	0,8	0,0286	0,0286	0,0286	0,0286	0,0286	0,0286	0,0286	1,283	5,812
4	0,8	0,19	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	0,805	5,846
5	0,7	0,05	0,05	0,04	0,04	0,04	0,04	0,04	1,721	6,966
6	0,7	0,29	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	0,970	6,996
7	0,6	0,0571	0,0571	0,0571	0,0571	0,0571	0,0571	0,0571	2,094	7,758
8	0,6	0,394	0,001	0,001	0,001	0,001	0,001	0,001	1,031	7,977
9	0,5	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	0,0714	2,404	8,308
10	0,5	0,49	0,0017	0,0017	0,0017	0,0017	0,0017	0,0017	1,097	7,868
11	0,4	0,0857	0,0857	0,0857	0,0857	0,0857	0,0857	0,0857	2,655	8,674
12	0,4	0,39	0,035	0,035	0,035	0,035	0,035	0,035	2,074	8,219
13	0,3	0,1	0,1	0,1	0,1	0,1	0,1	0,1	2,846	8,892
14	0,3	0,2	0,0833	0,0833	0,0833	0,0833	0,0833	0,0833	2,778	8,879
15	0,2	0,19	0,1017	0,1017	0,1017	0,1017	0,1017	0,1017	2,931	8,980
16	0,2	0,1143	0,1143	0,1143	0,1143	0,1143	0,1143	0,1143	2,968	8,988
17	0,125	0,125	0,125	0,125	0,125	0,125	0,125	0,125	3,000	9,000

Pode-se observar também que a entropia máxima, baseada na igualdade total de todas as variáveis ($\frac{1}{n}$), é apresentada na instância 17.

Dessa forma avaliamos a Entropia com seu cálculo original como critério de parada nos métodos LCPNwFOE e LCPNwVOE, apresentados respectivamente nas Seções 6.1 e 6.2, bem como alteramos a fórmula da entropia a fim de melhorar os resultados, principalmente no quesito de precisão, em relação aos outros métodos comparados nesta pesquisa.

A Figura 48 apresenta um comparativo entre a Entropia Original e com o cálculo da Entropia modificado aplicado a uma instância de teste da base de dados ECinterpro, no domínio da Bioinformática. Neste exemplo, a classe verdadeira é a classe 3.1.3 e adota-se a porcentagem de 70% da Entropia Máxima como critério de parada para os cálculos das duas Entropias: Original e Ajustada.

Figura 48 – Comparativo Entropia Original x Modificada



Conforme exemplo mostrado, a classe verdadeira é 3.1.3 e aplicando o ajuste no cálculo da Entropia Original foi possível delimitar melhor o critério de parada, analisando não só a probabilidade individual da classe, porém o conjunto de todas as probabilidades das classes para a tomada de decisão.

Baseado neste contexto, neste terceiro método proposto LCPNwFME, da mesma forma que no método LCPNwFOE, aplica-se a Entropia como critério de parada na abordagem NMLNP. No entanto, foi utilizado a nova fórmula da Entropia proposta neste trabalho, conforme apresentado na Equação 6.4.

Na etapa de treinamento, da mesma forma que nos métodos anteriores, cria-se um classificador para cada classe nó-pai, no qual utiliza-se o mesmo classificador base (algoritmo), bem como os mesmos parâmetros para todos os classificadores da hierarquia. Os classificadores que foram avaliados, bem como suas configurações, foram discutidos na Seção 4.2.3.

Já na etapa de classificação deste método proposto, calcula-se a entropia das probabilidades geradas pelo classificador em cada nó-pai da hierarquia, iniciando pelo nó-pai Raiz. Caso a entropia calculada deste determinado nó-pai seja maior que o Limiar da Entropia Fixa T (*Threshold*), é retornado o nó-atual como resultado. Caso contrário, percorre-se o caminho para o próximo nó-filho de maior probabilidade. Recursivamente, calcula-se a entropia deste novo Nó-Pai com a finalidade de verificar se a condição de parada é atendida ou até que um Nó-Folha seja predito, conforme Algoritmo 4.

Algoritmo 4 Função de predição do método LCPNwFME

```

1: procedure LCPNwFME_RECURSIVO(noAtual)
2:   if noAtual  $\neq$  noFolha then
3:     probs  $\leftarrow$  retornarProbabilidadesdasClassesFilhas(noAtual)
4:     h'  $\leftarrow$  calcularEntropia(probs)
5:     if h'  $\leq$  limiarEntropiaFixa then
6:       noAtual  $\leftarrow$  retornaNoFilhoComMaiorProbabilidade(Max(Probs))
7:       noAtual  $\leftarrow$  LCPNwFME_Recursivo(noAtual)
8:     end if
9:     return noAtual
10:  else
11:    return noAtual
12:  end if
13: end procedure

```

Neste método adota-se o Limiar T Fixo, como critério de parada da Entropia, e este foi definido a partir de um protocolo empírico realizado no conjunto de validação, no qual foi avaliado e adotado o Limiar com a maior média geral da Métrica hF entre todas as bases de dados avaliadas nesta pesquisa.

A Figura 49 apresenta o exemplo de uma hierarquia com as probabilidades de cada classe e o valor da Entropia Modificada (H') de cada respectivo Nó-Pai, calculadas pela Equação 6.4.

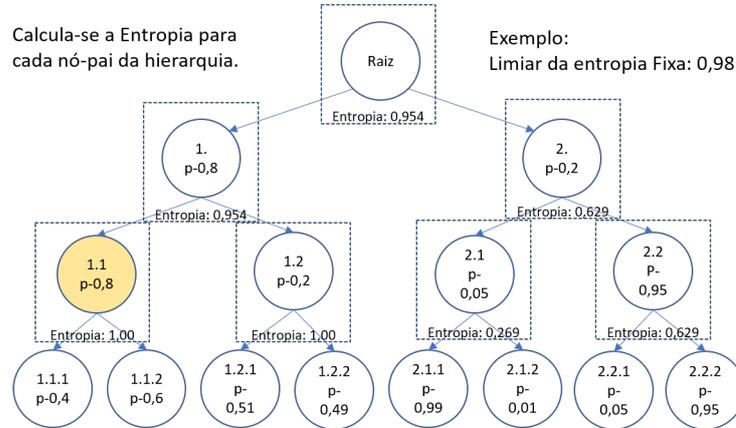
De acordo com o exemplo da Figura 49, foram detalhados os cálculos da entropia de cada Nó-Pai: H' (Raiz), H' (1) e H' (1.1), conforme Equação 6.4.

$$H'(Raiz) = (0.8 * \log_2 0.8 + 0.2 * \log_2 0.2) * \frac{\log_2 0.8 + \log_2 0.2}{2} \rightarrow H'(Raiz) = 0.954$$

$$H'(Raiz) \leq 0.98(\text{True}) \rightarrow \text{Probabilidade maior} \rightarrow \text{Nó} : (1.)$$

$$H'(1) = (0.8 * \log_2 0.8 + 0.2 * \log_2 0.2) * \frac{\log_2 0.8 + \log_2 0.2}{2} \rightarrow H'(1) = 0.954$$

Figura 49 – Exemplo com probabilidades e entropia



$H'(1) \leq 0.98(\text{True}) \rightarrow$ Probabilidade maior \rightarrow Nó : (1.1)

$$H'(1.1) = -(0.4 * \log_2 0.4 + 0.6 * \log_2 0.6) * \frac{\log_2 0.4 + \log_2 0.6}{2} \rightarrow H'(1.1) = 1.00$$

$H'(1.1) \leq 0.98(\text{False}) \rightarrow$ Falso \rightarrow Retorna Nó: (1.1)

Verifica-se neste exemplo que, adotando como limiar de entropia fixo o valor de 0.98, o resultado deste método de critério de parada é a classe: 1.1, pois a entropia das classes filhas do nó-pai 1.1 é de 1.00. Ou seja, a entropia calculada do respectivo nó é maior que o limiar adotado. Neste caso, é interrompida a etapa de avaliação e retorna-se como resultado da predição o respectivo Nó-Pai.

6.3.2 Resultados Computacionais do método LCPNwFME

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNwFME, além disso, análises estatísticas das métricas hF e hP são realizadas para a verificação se há diferença estatística do método proposto em relação aos métodos da literatura.

Os métodos bases que foram utilizados para comparação deste método proposto LCPNwFME foram os mesmos NMLNP Top-Down (CECI; MALERBA, 2007) e Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014) utilizados para comparação com os métodos LCPNwFOE e LCPNwVOE detalhados nos capítulos anteriores.

A Tabela 14 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre o método proposto LCPNwFME e os métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas; e CNN.

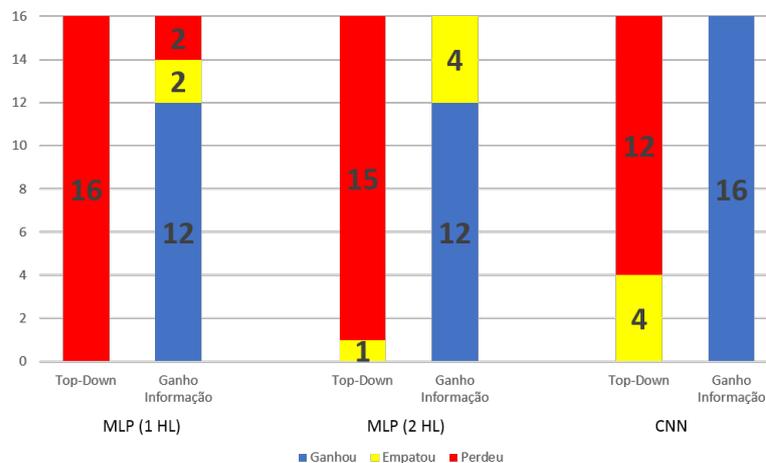
A abordagem proposta LCPNwFME foi comparada de modo pareado com as outras abordagens da literatura mencionadas. Foi utilizado o teste estatístico de Wilcoxon

Tabela 14 – Resultados da métrica hF dos métodos LCPNwFME, Top-Down e Ganho de Informação

Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCNPw FME	NMLNP Top-Down	NMLNP Ganho In-formação	LCNPw FME	NMLNP Top-Down	NMLNP Ganho In-formação	LCNPw FME	NMLNP Top-Down	NMLNP Ganho In-formação
ECinterpro	93,64	94,01	86,85	93,32	93,84	84,38	92,77	93,59	70,16
ECpfam	92,46	93,12	85,70	92,09	92,50	86,21	91,97	92,37	85,85
ECprints	91,31	91,60	79,97	90,92	91,26	82,01	90,32	91,37	84,09
ECprosite	93,12	93,58	74,03	92,80	93,22	85,05	92,59	93,14	84,14
GPCRinterpro	80,91	81,15	73,08	81,07	81,32	70,09	80,66	80,82	66,35
GPCRpfam	65,73	66,52	51,52	65,50	66,41	50,70	65,82	65,76	32,00
GPCRprints	78,10	78,66	70,51	77,79	78,28	68,60	78,15	78,65	67,84
GPCRprosite	60,35	60,97	48,01	61,00	61,20	43,16	61,36	61,17	37,47
SM-LD-ECinterpro	48,43	49,51	49,96	46,97	48,32	47,65	58,62	59,49	57,08
SM-LD-ECpfam	48,67	49,78	49,30	48,08	49,70	48,11	59,44	60,52	58,03
SM-LD-ECprints	55,67	56,32	56,15	54,07	55,06	52,92	64,24	64,94	61,41
SM-LD-ECprosite	49,60	50,53	52,09	47,64	49,11	48,57	60,28	61,14	58,39
SM-LD-GPCRinterpro	73,67	74,35	69,74	69,37	70,19	64,49	80,01	80,17	71,51
SM-LD-GPCRpfam	74,59	75,52	70,51	71,05	71,92	64,81	81,70	81,94	70,21
SM-LD-GPCRprints	74,64	75,53	70,98	71,00	71,75	65,91	81,11	81,47	71,02
SM-LD-GPCRprosite	73,95	74,82	69,72	68,61	69,73	63,25	80,97	81,13	68,36
Média Geral	72,18	72,87	66,22	70,71	71,49	64,22	76,25	76,73	65,24

para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 50 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Figura 50 – Comparação pareada do método proposto LCPNwFME com métodos da literatura



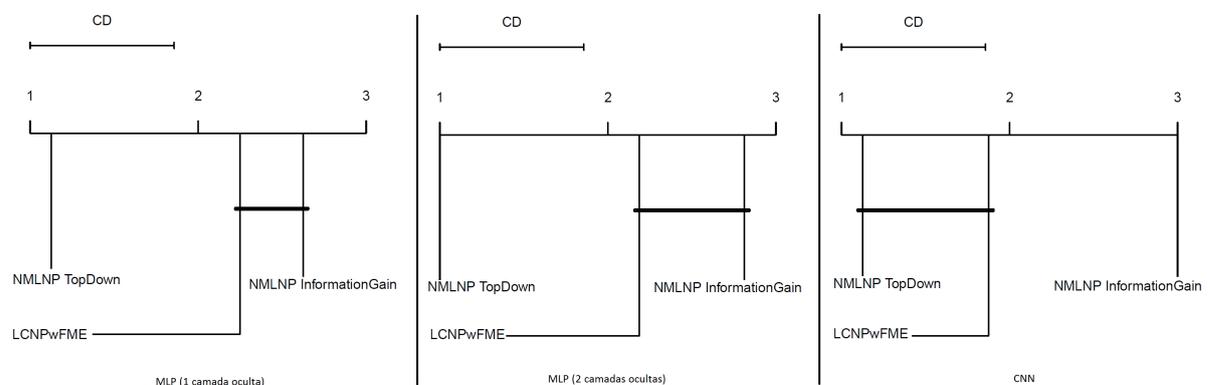
Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 50, para todas as bases de dados avaliadas, bem como comparando o método

proposto LCPNwFME com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwFME foi melhor estatisticamente em 12 bases de dados e empatou em 2 em comparação com o método Ganho de Informação que ganhou apenas em 2 bases. Já o método Top-Down obteve uma melhora estatística em todas as bases de dados em comparação com método LCPNwFME.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwFME foi melhor em 12 bases de dados e empatou em 4 comparado com o método Ganho de Informação. Já o método Top-Down foi melhor em 15 bases de dados e 1 empate comparado com o método LCPNwFME.
- Com classificador CNN, o método proposto LCPNwFME foi melhor estatisticamente em todas as bases de dados em comparação com o método Ganho de Informação. Já na comparação do método proposto com o método base Top-down, o método LCPNwFME empatou em 4 bases de dados perante ao método Top-Down que obteve superioridade estatística nas outras 12 bases.

A Figura 51 apresenta o teste de Nemenyi dos resultados da métrica hF para avaliação estatística do método proposto LCPNwFME em relação aos métodos da literatura NMLNP Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Figura 51 – Teste Nemenyi para comparação dos resultados da Métrica hF do método LCPNwFME



Pelos testes de Nemenyi realizados nos resultados dos métodos pela métrica de avaliação hF , Figura 42, o método proposto LCPNwFME ficou ranqueado em segundo

lugar em todas as abordagens com algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN e o método base Top-Down ficou ranqueado em primeiro. Além disso, o método proposto LCPNwFME apresentou diferença estatística perante o método Ganho de Informação na abordagem com algoritmo de classificação CNN.

A Tabela 15 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwFME e dos métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

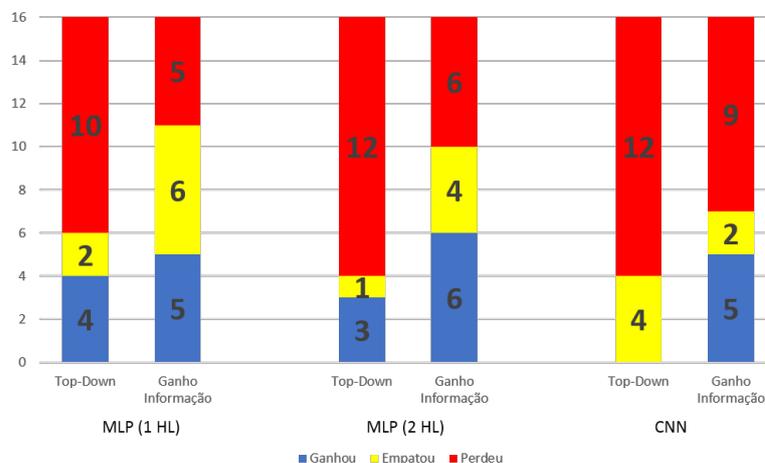
Tabela 15 – Resultados da métrica hP dos métodos LCPNwFME, Top-Down e Ganho de Informação

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCNPw FME	NMLNP Top-Down	NMLNP Ganho Informação	LCNPw FME	NMLNP Top-Down	NMLNP Ganho Informação	LCNPw FME	NMLNP Top-Down	NMLNP Ganho Informação
ECinterpro	91,07	91,93	93,46	90,32	91,57	93,51	89,98	92,05	76,05
ECpfam	89,88	90,99	92,32	89,18	90,13	92,43	89,26	90,42	90,10
ECprints	89,86	89,95	87,56	88,81	89,61	91,07	88,56	91,56	90,77
ECprosite	90,77	91,63	79,68	90,27	91,30	92,74	90,28	91,96	90,83
GPCRinterpro	86,08	87,67	84,71	86,03	87,60	83,19	86,17	87,91	78,77
GPCRpfam	83,26	87,22	67,03	81,79	87,48	68,71	86,39	89,42	44,67
GPCRprints	86,12	86,31	83,02	85,22	86,06	80,91	85,79	88,09	80,07
GPCRprosite	73,50	76,61	62,70	72,07	77,85	58,05	74,89	81,00	50,74
SM-LD-ECinterpro	48,20	52,73	50,91	46,71	52,69	49,09	57,41	61,28	60,29
SM-LD-ECpfam	48,63	52,30	50,51	47,86	54,71	49,52	58,23	62,19	60,80
SM-LD-ECprints	56,48	57,35	57,62	54,98	57,09	54,72	63,90	66,39	65,07
SM-LD-ECprosite	49,37	52,42	53,61	47,28	52,25	50,19	59,30	62,71	61,34
SM-LD-GPCRinterpro	78,99	76,33	78,63	76,23	74,47	73,73	81,96	81,96	83,98
SM-LD-GPCRpfam	80,06	77,59	80,06	77,56	76,18	75,86	83,47	83,74	85,25
SM-LD-GPCRprints	82,09	77,91	79,47	79,87	77,78	75,40	83,27	83,11	84,39
SM-LD-GPCRprosite	80,11	77,20	78,97	76,13	75,55	73,82	82,62	82,43	84,88
Média Geral	75,90	76,63	73,89	74,39	76,39	72,83	78,84	81,01	74,25

A abordagem proposta LCPNwFME foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 52 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 52, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwFME com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

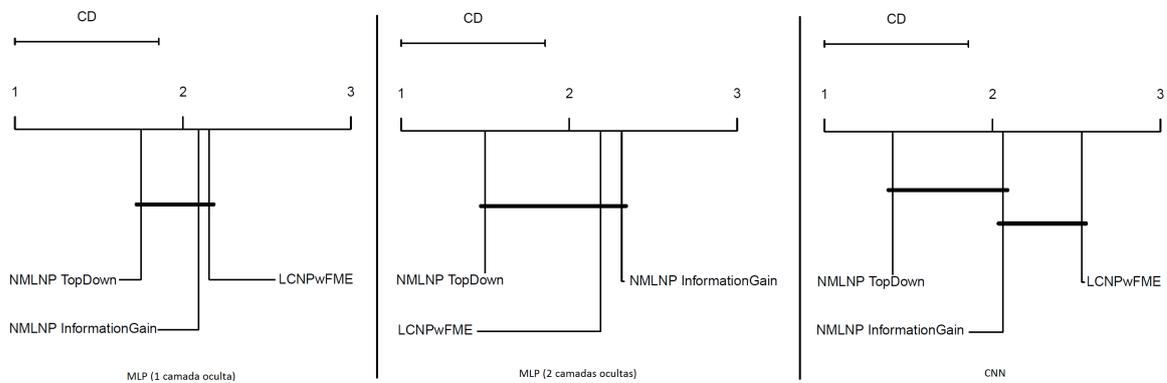
Figura 52 – Comparação pareada do método proposto LCPNwFME com métodos da literatura por meio da Métrica hP



- Com classificador MLP com 1 camada oculta, o método proposto LCPNwFME foi melhor estatisticamente em 5 bases de dados e empatou em 6 em comparação com o método Ganho de Informação que foi melhor em 5 bases de dados. Já na comparação do método proposto com o método base Top-down, o método LCPNwFME foi melhor em 4 e empatou em 2 bases perante ao método Top-Down que foi melhor em 10 bases de dados.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwFME foi melhor em 6 bases de dados e empatou em 4 em comparação com o método Ganho de Informação que foi melhor em 6 bases de dados. Já na comparação do método proposto com o método base Top-down, o método LCPNwFME foi melhor em 3 e empatou em 1 base de dados perante ao método Top-Down que foi melhor nas outras 12 bases.
- Com classificador CNN, o método proposto LCPNwFME obteve superioridade estatística em 5 bases de dados e empatou em 2 comparado com o método Ganho de Informação que foi melhor em 9 bases. Já na comparação do método proposto com o método base Top-down, o método LCPNwFME obteve 4 empates e perdeu estatisticamente nas outras 12 bases de dados em relação ao método Top-Down.

A Figura 53 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwFME em relação aos métodos da literatura Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Figura 53 – Teste Nemenyi para comparação dos resultados da Métrica hP do método LCPNwFME



Analisando os testes de Nemenyi pela métrica de avaliação hP, Figura 53, o método proposto LCPNwFME não apresentou diferença estatística comparado com os métodos da literatura Top-Down e Ganho de Informação nas abordagens com algoritmos de classificação MLP de 1 e 2 camadas ocultas. O método Top-Down ficou ranqueado em primeiro, sendo que este apresentou superioridade estatística em relação ao método proposto na abordagem com algoritmo de classificação CNN.

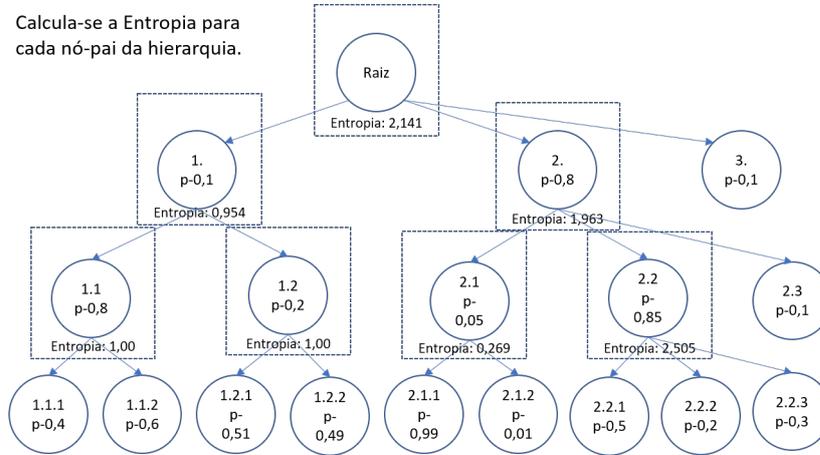
6.4 Método LCPNwVME

Nesta seção é apresentado o método proposto denominado LCPNwVME (Local Classifier per Parent Node with Variable Modified Entropy). Além disso, são apresentados os resultados computacionais comparados com os métodos da literatura NMLNP Top-Down (CECI; MALERBA, 2007) e Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014).

Conforme já dito anteriormente, com um número maior de probabilidades das classes-filhas de um determinado nó-pai, tende a entropia ser maior. De acordo com a Figura 54, podemos verificar nos nós-pai (Raiz, 2. e 2.2) que, com a inserção de mais um filho destes Nós-Pai, em relação ao exemplo anterior, aumentou o valor da entropia. Nesse caso, adotar um limiar de entropia fixa pode não apresentar a melhor solução para uma taxonomia desbalanceada.

Desta forma, neste método proposto, utiliza-se um limiar (T) baseado na entropia máxima de acordo com a quantidade de classes filhas do nó em questão. Ou seja, devido à oscilação dos valores da entropia em relação à quantidade de filhos em taxonomias desbalanceadas, deve-se avaliar a entropia calculada do Nó e aferir com o limiar T ajustado por meio da entropia máxima, ao contrário do método de entropia Fixa, que define um limiar (T) fixo para todos os nós-Pai da hierarquia.

Figura 54 – Exemplo com probabilidades e entropia



A diferença do método LCPNwVOE para este método proposto LCPNwVME é que no primeiro método foi utilizado o cálculo da Entropia Original, conforme Equação 2.11. Enquanto neste segundo método utiliza-se do ajuste na fórmula da Entropia proposto neste trabalho, conforme apresentado na Equação 6.4.

Para determinar o limiar de entropia variável (T), as seguintes premissas foram adotadas:

- Para cada nó pai na hierarquia, a entropia máxima (H') é calculada, com base no número total de classes filhas (CQ) e para eventos equiprováveis $\frac{1}{CQ}$ conforme definido em Equação 6.5.

$$\begin{aligned}
 H'(X) &= - \sum_{i=1}^m p_i \log_2 p_i * - \sum_{i=1}^m \frac{\log_2 p_i}{m} \\
 H'(X) &= -CQ * \frac{1}{CQ} * \log_2 \frac{1}{CQ} * -CQ * \frac{\log_2 \frac{1}{CQ}}{CQ} \\
 H'(X) &= - \log_2 \frac{1}{CQ} * - \log_2 \frac{1}{CQ} \\
 H'(X) &= - \log_2 \frac{1}{CQ} * - \log_2 \frac{1}{CQ} \\
 H'(X) &= - \log_2 CQ^{-1} * - \log_2 CQ^{-1} \\
 H'(X) &= \log_2 CQ * \log_2 CQ
 \end{aligned} \tag{6.5}$$

- Para o cálculo do limiar de entropia variável, é aplicada uma porcentagem sobre a entropia máxima (PER), determinando, nesta etapa, um limiar dinâmico (T) para cada nó pai da hierarquia, conforme definido na Equação 6.6.

$$T = \log_2 qtdeFilhos * (\log_2 qtdeFilhos) * porcentagemAjuste \tag{6.6}$$

O algoritmo 5 mostra o cálculo do limiar de ajuste baseado na entropia máxima.

Algoritmo 5 Cálculo para definição do Limite para Entropia Variável

```

1: procedure CALCULARLIMIARLCPNWVME(qtdeFilhos, porcentagemAjuste)
2:    $h' \leftarrow \log_2 qtdeFilhos * \log_2 qtdeFilhos$ 
3:    $T \leftarrow h' \times porcentagemAjuste$ 
4:   return  $T$  ▷ Retorna o limiar baseado na nova fórmula da entropia
5: end procedure

```

O algoritmo 6 apresenta a etapa de predição de uma determinada instância de teste.

Algoritmo 6 Predição recursiva baseado no método LCPNWVME

```

1: procedure RETORNARNO_LCPNWVME_RECURSIVO(noAtual, porcAjuste)
2:   if noAtual = noFolha then
3:     return noAtual
4:   end if
5:   probasFilhos  $\leftarrow$  retornarProbabilidadesClassesFilhas(noAtual)
6:    $H' \leftarrow$  calcularEntropia(probasFilhos)
7:    $Q \leftarrow$  quantidadeFilhos(noAtual)
8:    $T \leftarrow$  calcularLimiarLCPNWVME( $Q$ , porcAjuste)
9:   if  $H' \leq T$  then
10:    noAtual  $\leftarrow$  noFilho(max(probasFilhos))
11:    noAtual  $\leftarrow$  retornarNoLCPNWVME_RecurSivo(noAtual, porcAjuste)
12:   end if
13:   return noAtual
14: end procedure

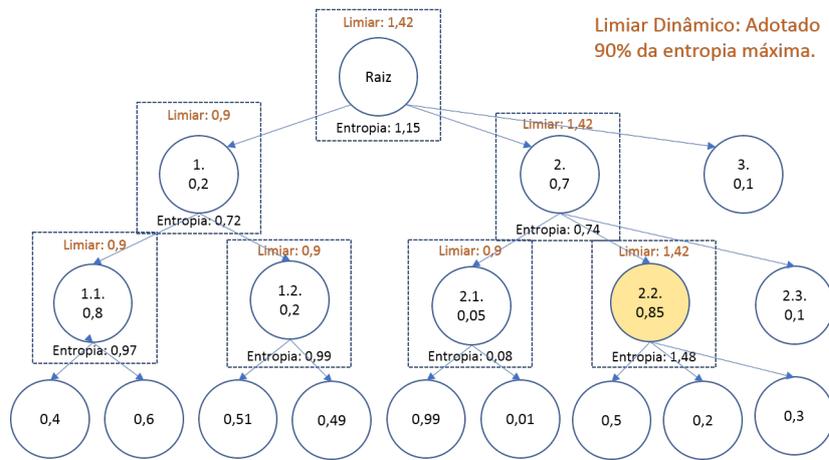
```

Na etapa de classificação de uma nova instância de teste é realizada a predição (opcional) em nós-folha, baseado na nova fórmula da entropia proposta, conforme descrito no algoritmo 6. Neste algoritmo adotam-se as seguintes premissas:

- Caso a entropia calculada do classificador do nó-pai atual, referente às probabilidades dos seus nós-filhos, seja menor que o limiar ajustado (da porcentagem da entropia máxima daquele respectivo nó-pai), então caminha para o nó-filho com o maior valor de probabilidade; e
- Caso a entropia calculada do classificador do nó-pai atual, referente às probabilidades dos seus nós-filhos, seja maior que o limiar ajustado (da porcentagem da entropia máxima daquele respectivo nó-pai), então não caminha na hierarquia e retorna o nó-pai atual, realizando o critério de parada pelo limiar calculado baseado na nova fórmula de entropia proposta e do percentual (porcentagem de ajuste) da entropia máxima.

A Figura 55 apresenta uma hierarquia com suas respectivas probabilidades e seus limiares de Entropia Variável. O cálculo da entropia máxima nos nós: Raiz, 2. e 2.2 tem o valor de: 2,512, pois apresentam 3 classes filhas em cada nó-pai. Já nos demais nós, onde contém apenas duas classes filhas, a entropia máxima é 1. Na sequência, estipula-se uma porcentagem desse valor máximo da entropia para cada nó-pai. Para a porcentagem de 90% do valor máximo, o limiar (T) dos nós RAIZ, 2 e 2.2 é de 2,260; e nos demais nós é de 0,9.

Figura 55 – Exemplo com probabilidades e entropia



Neste exemplo, apresentamos, passo a passo, os cálculos da Entropia Modificada em cada Nó-pai e a avaliação realizada até o retorno do respectivo nó-predito:

$$H(Raiz) = (0.1 * \log_2 0.1 + 0.8 * \log_2 0.8 + 0.1 * \log_2 0.1) * \frac{\log_2 0.1 + \log_2 0.8 + \log_2 0.1}{3}$$

$$H(Raiz) = 2.141$$

$$H(Raiz) \leq T(2.26)(True) \rightarrow \text{Nó com maior probabilidade} \rightarrow \text{Nó : (2.)}$$

$$H(2) = (0.05 * \log_2 0.05 + 0.85 * \log_2 0.85 + 0.1 * \log_2 0.1) * \frac{\log_2 0.05 + \log_2 0.85 + \log_2 0.1}{3}$$

$$H(2) = 1.963$$

$$H(2) \leq T(2.26)(True) \rightarrow \text{Nó com maior probabilidade} \rightarrow \text{Nó : (2.2)}$$

$$H(2.2) = (0.5 * \log_2 0.5 + 0.2 * \log_2 0.2 + 0.3 * \log_2 0.3) * \frac{\log_2 0.5 + \log_2 0.2 + \log_2 0.3}{3}$$

$$H(2.2) = 2.505$$

$$H(2.2) \leq T(2.26)(False) \rightarrow \text{Falso} \rightarrow \text{Retorna Nó: (2.2)}$$

6.4.1 Resultados Computacionais do método LCNPwVME

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNwVME em comparação com os métodos da literatura NMLNP Top-Down e Ganho de Informação.

Além disso, análises estatísticas das métricas hF e hP são realizadas para a verificação se há diferença estatística entre os métodos avaliados.

A Tabela 16 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre o método proposto LCPNwVME e os métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas e CNN.

Tabela 16 – Resultados da métrica hF dos métodos LCPNwVME, Top-Down e Ganho de Informação

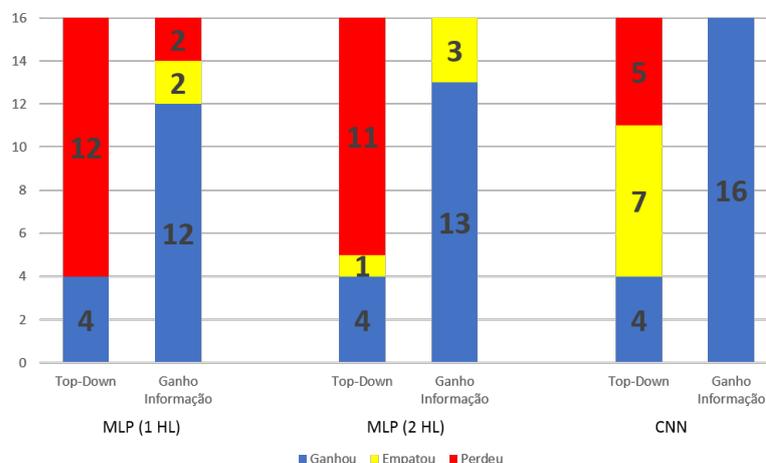
Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCPNwVME	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwVME	NMLNP Top-Down	NMLNP Ganho Informação	LCPNwVME	NMLNP Top-Down	NMLNP Ganho Informação
ECinterpro	94,39	94,01	86,85	94,34	93,84	84,38	93,76	93,59	70,16
ECpfam	93,63	93,12	85,70	93,14	92,50	86,21	92,62	92,37	85,85
ECprints	91,92	91,60	79,97	91,74	91,26	82,01	90,91	91,37	84,09
ECprosite	93,85	93,58	74,03	93,60	93,22	85,05	93,21	93,14	84,14
GPCRinterpro	80,67	81,15	73,08	81,07	81,32	70,09	80,47	80,82	66,35
GPCRpfam	65,65	66,52	51,52	65,74	66,41	50,70	66,13	65,76	32,00
GPCRprints	77,95	78,66	70,51	77,79	78,28	68,60	78,28	78,65	67,84
GPCRprosite	59,84	60,97	48,01	60,89	61,20	43,16	61,91	61,17	37,47
SM-LD-ECinterpro	49,07	49,51	49,96	48,03	48,32	47,65	59,50	59,49	57,08
SM-LD-ECpfam	49,05	49,78	49,30	49,21	49,70	48,11	60,41	60,52	58,03
SM-LD-ECprints	55,64	56,32	56,15	54,55	55,06	52,92	64,87	64,94	61,41
SM-LD-ECprosite	49,68	50,53	52,09	48,53	49,11	48,57	60,91	61,14	58,39
SM-LD-GPCRinterpro	73,45	74,35	69,74	69,36	70,19	64,49	80,05	80,17	71,51
SM-LD-GPCRpfam	74,78	75,52	70,51	71,12	71,92	64,81	81,82	81,94	70,21
SM-LD-GPCRprints	74,62	75,53	70,98	70,79	71,75	65,91	81,29	81,47	71,02
SM-LD-GPCRprosite	74,06	74,82	69,72	68,86	69,73	63,25	80,96	81,13	68,36
Média Geral	72,39	72,87	66,22	71,17	71,49	64,22	76,70	76,73	65,24

A abordagem proposta LCPNwVME foi comparada de modo pareado com as outras abordagens da literatura mencionadas. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hF, para cada base de dados. Na Figura 56 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 56, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwVME com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwVME foi melhor estatisticamente em 12 bases de dados e 2 empates em comparação com o

Figura 56 – Comparação pareada do método proposto LCPNwVME com métodos da literatura



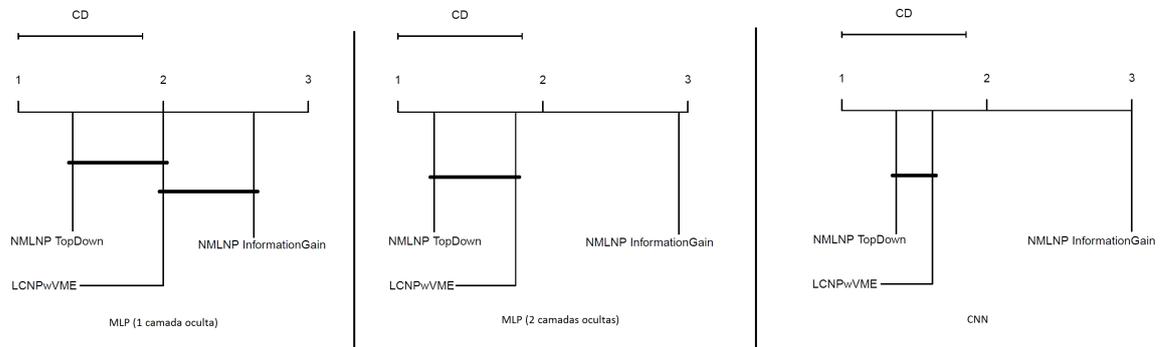
método Ganho de Informação que ganhou apenas em 2 bases. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVME foi melhor em 4 bases de dados perante ao método Top-Down que foi melhor em 12 bases.

- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwVME obteve superioridade estatística em 13 bases de dados e houve 3 empates comparado com o método Ganho de Informação. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVME foi melhor em 4 bases de dados e obteve 1 empate perante ao método Top-Down que foi melhor em 11 bases.
- Com classificador CNN, o método proposto LCPNwVME foi melhor estatisticamente em todas as bases de dados em comparação com o método Ganho de Informação. Já na comparação do método proposto com o método base Top-Down, o método LCPNwVME foi melhor em 4 bases de dados e obteve 7 empates perante ao método Top-Down que foi melhor nas outras 5 bases.

A Figura 57 apresenta o teste de Nemenyi dos resultados da métrica hF para avaliação estatística do método proposto LCPNwVME em relação aos métodos da literatura NMLNP Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Pelos testes de Nemenyi realizados nos resultados dos métodos pela métrica de avaliação hF, Figura 42, o método proposto LCPNwVME ficou ranqueado em segundo lugar e não apresentou diferença estatística em relação ao método Top-Down em todas as abordagens com algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN. Além

Figura 57 – Teste Nemenyi da Métrica hF dos métodos proposto LCPNwVME e da literatura



disso, o método proposto LCPNwVME apresentou diferença estatística para o método Ganho de Informação nas abordagens com algoritmos de classificação MLP com 2 camadas ocultas e CNN.

Para uma melhor análise com relação à precisão das predições, a Tabela 17 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwVME e dos métodos da literatura Top-Down e Ganho de Informação. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

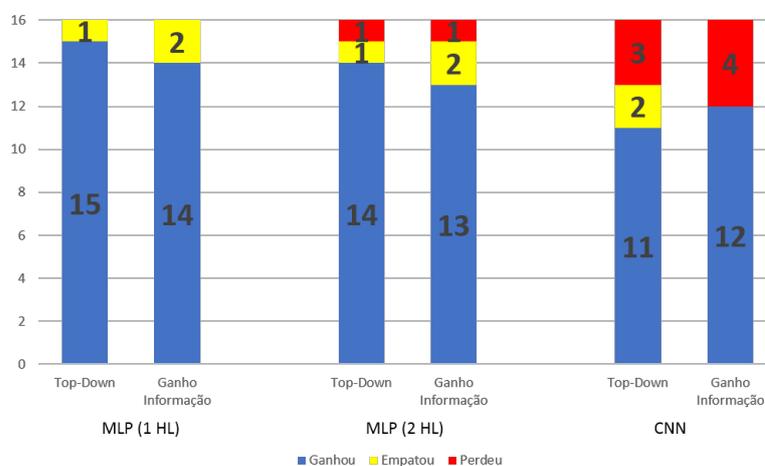
Tabela 17 – Resultados da métrica hP dos métodos LCPNwVME, Top-Down e Ganho de Informação

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCPNwVME	NMLNP Top-Down	NMLNP Ganho In-formação	LCPNwVME	NMLNP Top-Down	NMLNP Ganho In-formação	LCPNwVME	NMLNP Top-Down	NMLNP Ganho In-formação
ECinterpro	93,16	91,93	93,46	92,87	91,57	93,51	92,59	92,05	76,05
ECpfam	93,30	90,99	92,32	92,55	90,13	92,43	91,93	90,42	90,10
ECprints	93,15	89,95	87,56	92,52	89,61	91,07	91,44	91,56	90,77
ECprosite	94,27	91,63	79,68	94,01	91,30	92,74	93,41	91,96	90,83
GPCRinterpro	88,81	87,67	84,71	88,77	87,60	83,19	88,41	87,91	78,77
GPCRpfam	86,39	87,22	67,03	85,70	87,48	68,71	84,39	89,42	44,67
GPCRprints	88,70	86,31	83,02	87,96	86,06	80,91	87,31	88,09	80,07
GPCRprosite	78,14	76,61	62,70	76,62	77,85	58,05	75,72	81,00	50,74
SM-LD-ECinterpro	55,27	52,73	50,91	54,16	52,69	49,09	62,49	61,28	60,29
SM-LD-ECpfam	55,46	52,30	50,51	56,44	54,71	49,52	63,09	62,19	60,80
SM-LD-ECprints	61,20	57,35	57,62	60,55	57,09	54,72	67,31	66,39	65,07
SM-LD-ECprosite	55,35	52,42	53,61	54,58	52,25	50,19	63,54	62,71	61,34
SM-LD-GPCRinterpro	80,16	76,33	78,63	77,96	74,47	73,73	82,41	81,96	83,98
SM-LD-GPCRpfam	81,75	77,59	80,06	80,23	76,18	75,86	84,15	83,74	85,25
SM-LD-GPCRprints	82,69	77,91	79,47	81,71	77,78	75,40	83,27	83,11	84,39
SM-LD-GPCRprosite	81,29	77,20	78,97	79,02	75,55	73,82	82,85	82,43	84,88
Média Geral	79,32	76,63	73,89	78,48	76,39	72,83	80,89	81,01	74,25

A abordagem proposta LCPNwVME foi comparada de modo pareado com as

outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 58 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Figura 58 – Comparação pareada do método proposto LCPNwVME com métodos da literatura por meio da Métrica hP



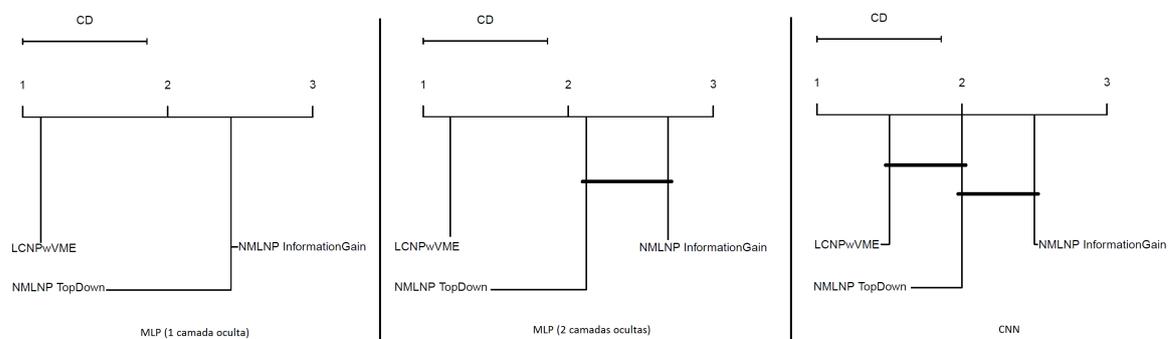
De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 58, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwVME com os métodos da literatura Top-Down e Ganho de Informação, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwVME foi melhor estatisticamente em 14 bases de dados e obteve 2 empates em comparação com o métodos Ganho de Informação. Já na comparação com o método Top-Down, o método LCPNwVME obteve superioridade estatística em 15 bases de dados e houve 1 empate.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwVME foi melhor em 13 e obteve 2 empates das bases de dados em comparação com o método Ganho de Informação que foi melhor em apenas 1 base. Já na comparação do método proposto com o método base Top-down, o método LCPNwVME foi melhor em 14 bases de dados e 1 empate perante ao método Top-Down que foi melhor em apenas 1 base.

- Com classificador CNN, o método proposto LCPNwVME foi melhor estatisticamente em 12 bases de dados e perdeu em 4 comparado com o método Ganho de Informação. Já com o método Top-Down, o LCPNwVME obteve superioridade em 11 bases de dados e houve 2 empates, bem como perdeu estatisticamente em 3 bases dados.

A Figura 59 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwVME em relação aos métodos da literatura Top-Down e Ganho de Informação. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Figura 59 – Teste Nemenyi da Métrica hP dos métodos LCPNwVME e da literatura



Analisando os resultados apresentados pelo teste Nemenyi, este método proposto obteve melhores resultados comparados com os métodos da literatura NMLNP Top-Down e Ganho de Informação, pois ficou ranqueado em primeiro lugar nas três abordagens com algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN. Além disso, apresentou superioridade estatística para o método Ganho de Informação nas três abordagens avaliadas e superioridade estatística em relação ao método Top-Down nas abordagens com algoritmos de classificação MLP com 1 e 2 camadas ocultas.

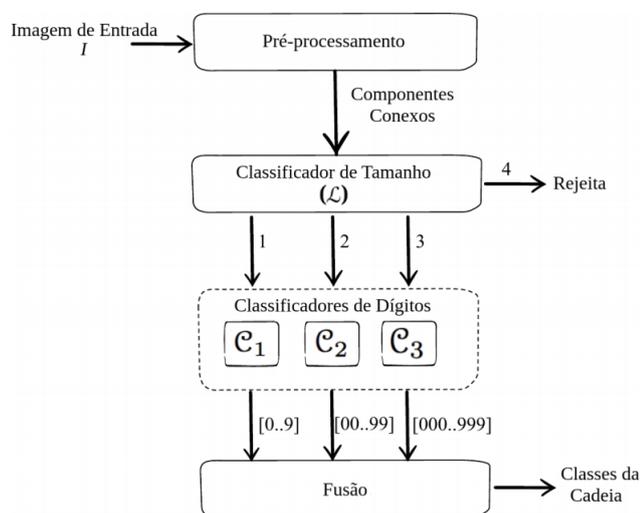
6.5 Método LCPNwLC

Nesta seção é apresentado o método proposto denominado LCPNwLC (Local Classifier per Parent Node with Level Classifier). Este método apresenta um modelo de Classificador de Nível como critério de parada, o qual indica em que nível da hierarquia a predição por meio da abordagem LCPN deve ser interrompida, realizando neste caso uma predição não obrigatória de nós-folha (NMLNP).

Esta estratégia foi baseada no trabalho de [Hochuli et al. \(2018\)](#) que propôs um método para reconhecimento de dígitos manuscritos adotando uma abordagem livre de

segmentação baseada em três módulos principais: Pré-processamento, Classificador de tamanho e Classificadores de Dígitos. Inicialmente, uma imagem I passa por um módulo de pré-processamento que identifica todos os componentes conexos (CC 's) presentes. Cada CC é classificado pelo classificador de tamanho \mathcal{L} o qual determinará a probabilidade deste ser composto por 1, 2, 3 ou 4 dígitos conectados. O módulo de classificação compreende três classificadores de dígitos (C_1, C_2, C_3), utilizados para discriminar respectivamente, dígitos de 0 a 9, 00 a 99, e 000 a 999. O classificador (C_1, C_2 ou C_3) que será utilizado para um dado CC é determinado pelo resultado do classificador de tamanho (\mathcal{L}), conforme mostrado na Figura 60.

Figura 60 – Método para reconhecimento de componentes conexos de dígitos manuscritos



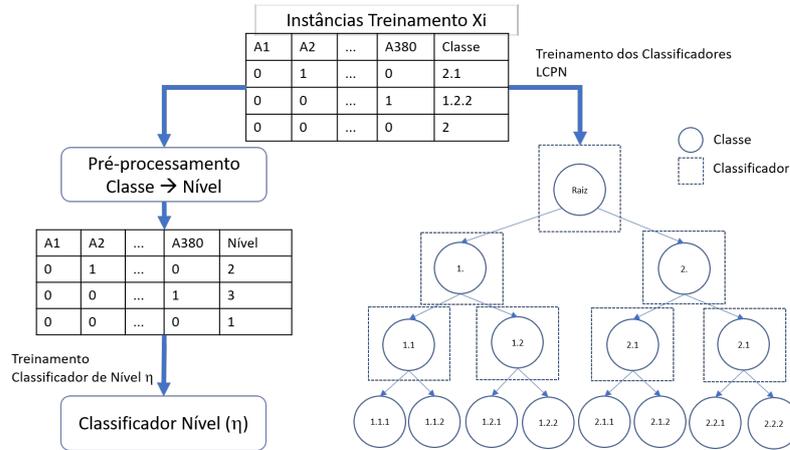
Fonte: (HOCHULI et al., 2018)

No método proposto LCPNwLC o modelo de Classificador de Nível η é análogo ao modelo de Classificador de tamanho \mathcal{L} . Entretanto, neste trabalho η foi aplicado em problemas de classificação hierárquica, no qual por meio da informação contextual do nível esta é utilizada como critério de parada para um determinado nível na hierarquia.

Na fase de treinamento deste método LCPNwLC, o conjunto de treinamento é utilizado tanto para os classificadores da abordagem LCPN, quanto para o classificador de nível η , conforme ilustrado na Figura 61. Porém, para o classificador de nível η é realizado um pré-processamento para a conversão das classes para os níveis de profundidade respectivos na hierarquia. Por exemplo a classe 2.1 é alterada para nível 2, classe 1.2.2 para nível 3 e a classe 2 para nível 1. Por fim, o treinamento deste classificador de nível η é feito baseando-se nos atributos de entrada e o atributo de saída é determinado pelo nível da hierarquia.

Na fase de teste deste método aplica-se uma instância de teste x no Classificador

Figura 61 – Treinamento dos Classificadores LCPN e de Nível η



de Nível para determinar o nível que a predição será interrompida, por meio do método Top-Down. Porém, caso a confiança deste modelo seja inferior ao Limiar T , outro método de critério de parada é utilizado para realizar a predição, sendo este o método LCPNwVME.

Este método proposto é definido pela Equação 6.7, tendo que $\eta^i(x) = p^i(x)$ é a probabilidade de uma instância x pertencer a um determinado nível i , tal que no domínio aplicado temos $i = (1, 2, 3, 4)$ níveis, bem como l representa o nível retornado da maior probabilidade $\max(\eta^i(x))$.

Dessa forma, o método LCPNwLC atribuirá uma classe φ para uma determinada instância x por meio da abordagem Top-Down e interrompendo a classificação em um determinado nível advindo pelo Classificador de Nível η caso a confiança seja superior ao Limiar T . Caso contrário, a predição será realizada por meio do método LCPNwVME, ignorando neste caso o Classificador de Nível.

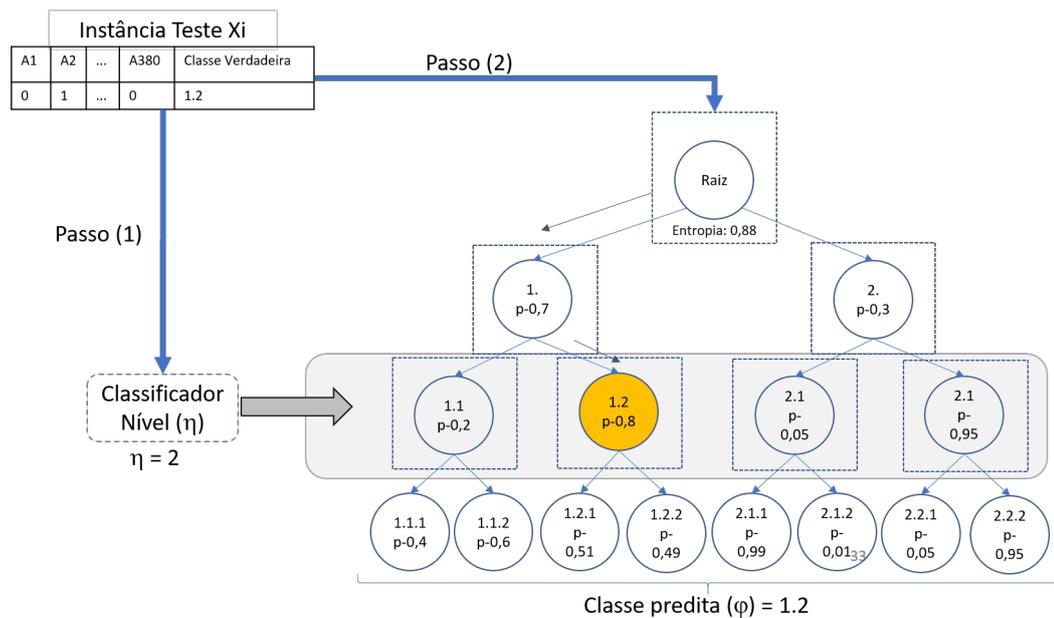
$$LCPNwLC(\varphi|x) = \begin{cases} \text{Se } \max(\eta^i(x)) > T, & \text{TopDown}(x) & \diamond \text{ até atingir nível } l \\ \text{Senão,} & LCPNwVME(x) \end{cases} \quad (6.7)$$

Para avaliar a confiança do Classificador de Nível, (η) é adotado um limiar (T) para verificar a utilização ou não deste para determinar em qual nível será interrompida a predição de uma determinada instância. Tanto o limiar de confiança T quanto o classificador base utilizados para este Classificador de Nível (η) foram delimitados pelo Conjunto de Validação.

Conforme exemplo na Figura 62, uma determinada instância de teste é aplicada primeiramente no Classificador de Nível (η). Este classificador de nível gerou um contexto de informação de que esta instância deve ser interrompida no segundo nível. Diante desta

informação, a instância é inferida na abordagem LCPN, seguindo o caminho da maior probabilidade até que a rotulação final seja efetuada de acordo com a informação de parada atribuída pelo Classificador de Nível. Desta forma, é realizada uma Predição Não Obrigatória de Nó-Folha e retorna-se, neste caso, à classe predita 1.1.

Figura 62 – Rotulação nó-interno baseado no Classificador de Nível



Neste método LCPNwLC, foi adotada a estratégia para percorrer a árvore pelo nó com maior probabilidade a partir da Raiz (Top-Down), pelo fato desta ter apresentado melhores resultados na avaliação extensiva dos métodos do estado da arte. Porém, outras estratégias poderiam ser utilizadas para a escolha do caminho a ser percorrido, como, por exemplo: Soma, Multiplicação ou Média das probabilidades dos Ramos discutidas na seção 3.

6.5.1 Resultados computacionais

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNwLC em comparação com os métodos da literatura NMLNP Top-Down (CECI; MALERBA, 2007) e do método baseado na Entropia LCPNwVME (Seção 6.4) pelo fato destes terem apresentados melhores resultados em comparação com os métodos do estado da arte. Observa-se nos resultados anteriores que o método LCPNwVME ficou ranqueado estatisticamente em primeiro lugar com a métrica de avaliação hP e por meio da métrica de avaliação hF os métodos NMLNP Top-Down e LCPNwVME não apresentaram diferenças estatísticas.

A Tabela 18 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre este método proposto LCPNwLC, LCPNwVME e o método da literatura Top-Down. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas; e CNN.

Tabela 18 – Resultados da métrica de avaliação hF dos métodos LCPNwLC, Top-Down e LCPNwVME

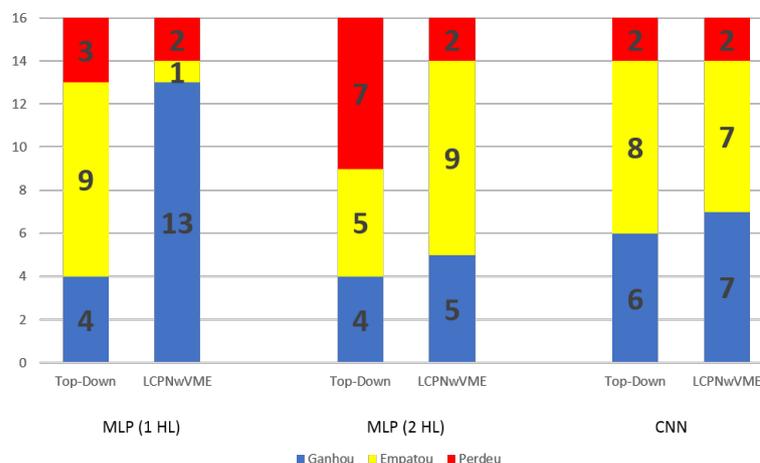
Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCPNwLC	NMLNP Top-Down	LCPNwVME	LCPNwLC	NMLNP Top-Down	LCPNwVME	LCPNwLC	NMLNP Top-Down	LCPNwVME
ECinterpro	95,67	94,01	94,39	95,78	93,84	94,34	95,23	93,59	93,76
ECpfam	94,54	93,12	93,63	94,17	92,50	93,14	94,03	92,37	92,62
ECprints	92,63	91,60	91,92	92,27	91,26	91,74	91,73	91,37	90,91
ECprosite	94,88	93,58	93,85	94,56	93,22	93,60	94,21	93,14	93,21
GPCRinterpro	79,21	81,15	80,67	79,54	81,32	81,07	79,01	80,82	80,47
GPCRpfam	66,22	66,52	65,65	66,23	66,41	65,74	66,55	65,76	66,13
GPCRprints	76,05	78,66	77,95	75,68	78,28	77,79	76,20	78,65	78,28
GPCRprosite	59,89	60,97	59,84	60,90	61,20	60,89	61,81	61,17	61,91
SM-LD-ECinterpro	49,44	49,51	49,07	48,17	48,32	48,03	59,53	59,49	59,50
SM-LD-ECpfam	49,68	49,78	49,05	49,51	49,70	49,21	60,62	60,52	60,41
SM-LD-ECprints	56,34	56,32	55,64	54,85	55,06	54,55	64,86	64,94	64,87
SM-LD-ECprosite	50,33	50,53	49,68	48,65	49,11	48,53	60,90	61,14	60,91
SM-LD-GPCRinterpro	74,18	74,35	73,45	69,61	70,19	69,36	80,33	80,17	80,05
SM-LD-GPCRpfam	75,38	75,52	74,78	71,33	71,92	71,12	82,03	81,94	81,82
SM-LD-GPCRprints	75,79	75,53	74,62	71,06	71,75	70,79	81,43	81,47	81,29
SM-LD-GPCRprosite	74,64	74,82	74,06	68,67	69,73	68,86	81,24	81,13	80,96
Média Geral	72,80	72,87	72,39	71,31	71,49	71,17	76,86	76,73	76,70

A abordagem proposta LCPNwLC foi comparada de modo pareado com as outras abordagens da literatura mencionadas. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hF, para cada base de dados. Na Figura 63 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 63, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwLC com os métodos LCPNwVME e da literatura Top-Down, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwLC foi melhor estatisticamente em 13 bases de dados, 1 empate e perdeu apenas em 2 bases em comparação com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwLC foi melhor em 4 bases de dados e 9 empates perante ao método Top-Down que foi melhor em 3 bases.

Figura 63 – Comparação pareada do método proposto LCPNwLC com métodos da literatura



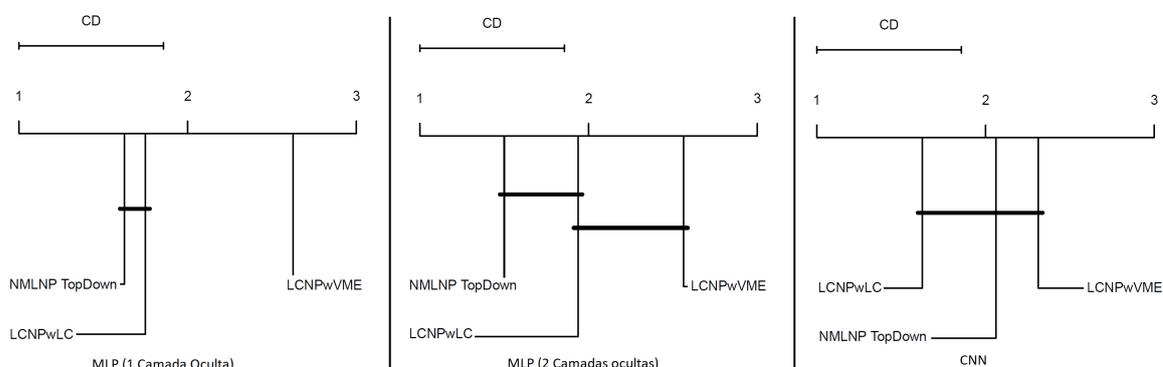
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwLC obteve superioridade estatística em 5 bases de dados, 9 empates e perdeu em 2 bases comparado com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwLC foi melhor em 4 bases de dados e obteve 5 empates perante ao método Top-Down que foi melhor em 7 bases.
- Com classificador CNN, o método proposto LCPNwLC foi melhor estatisticamente em 7 bases, 7 empates e perdeu em 2 bases em comparação com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwLC foi melhor em 6 bases de dados e obteve 8 empates perante ao método Top-Down que foi melhor nas outras 2 bases.

A Figura 64 apresenta o teste de Nemenyi dos resultados da métrica hF para avaliação estatística do método proposto LCPNwLC em relação aos métodos da literatura NMLNP Top-Down e LCPNwVME. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Pelos testes de Nemenyi realizados nos resultados dos métodos pela métrica de avaliação hF, Figura 64, o método proposto LCPNwLC apresentou superioridade estatística perante ao método LCPNwVME na abordagem com algoritmo de classificação MLP com 1 camada oculta. Além disso, este método proposto não apresentou diferença estatística em relação ao método Top-Down nas três abordagens avaliadas.

Para uma melhor análise com relação à precisão das predições, a Tabela 19 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwLC e dos métodos da literatura

Figura 64 – Teste Nemenyi da Métrica hF para comparação do método proposto LCPNwLC



Top-Down e LCPNwVME. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

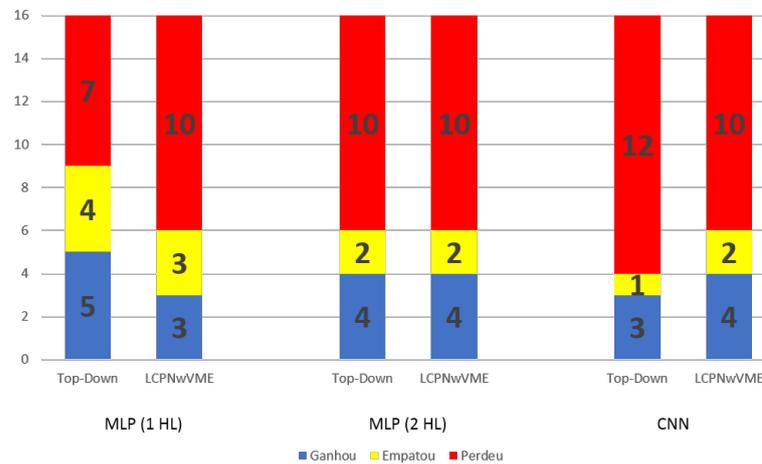
Tabela 19 – Resultados da métrica de avaliação hP dos métodos LCPNwLC, Top-Down e LCPNwVME

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCPNwLC	NMLNP Top-Down	LCPNwVME	LCPNwLC	NMLNP Top-Down	LCPNwVME	LCPNwLC	NMLNP Top-Down	LCPNwVME
ECinterpro	95,58	91,93	93,16	95,61	91,57	92,87	95,28	92,05	92,59
ECpfam	95,01	90,99	93,30	94,47	90,13	92,55	94,55	90,42	91,93
ECprints	93,17	89,95	93,15	93,16	89,61	92,52	91,90	91,56	91,44
ECprosite	95,87	91,63	94,27	95,61	91,30	94,01	94,80	91,96	93,41
GPCRinterpro	81,88	87,67	88,81	81,99	87,60	88,77	81,38	87,91	88,41
GPCRpfam	85,87	87,22	86,39	85,72	87,48	85,70	84,33	89,42	84,39
GPCRprints	80,21	86,31	88,70	79,48	86,06	87,96	79,29	88,09	87,31
GPCRprosite	78,34	76,61	78,14	76,84	77,85	76,62	76,08	81,00	75,72
SM-LD-ECinterpro	51,25	52,73	55,27	50,78	52,69	54,16	60,34	61,28	62,49
SM-LD-ECpfam	51,48	52,30	55,46	52,48	54,71	56,44	61,32	62,19	63,09
SM-LD-ECprints	57,76	57,35	61,20	56,98	57,09	60,55	65,26	66,39	67,31
SM-LD-ECprosite	51,34	52,42	55,35	50,43	52,25	54,58	61,17	62,71	63,54
SM-LD-GPCRinterpro	75,75	76,33	80,16	72,05	74,47	77,96	80,94	81,96	82,41
SM-LD-GPCRpfam	76,87	77,59	81,75	73,53	76,18	80,23	82,52	83,74	84,15
SM-LD-GPCRprints	77,49	77,91	82,69	73,75	77,78	81,71	81,74	83,11	83,27
SM-LD-GPCRprosite	76,14	77,20	81,29	70,89	75,55	79,02	81,53	82,43	82,85
Média Geral	76,50	76,63	79,32	75,23	76,39	78,48	79,53	81,01	80,89

A abordagem proposta LCPNwLC foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 65 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas

(respectivamente, 1 HL e 2 HL) e CNN.

Figura 65 – Comparação pareada do método proposto LCPNwLC com métodos da literatura por meio da Métrica hP



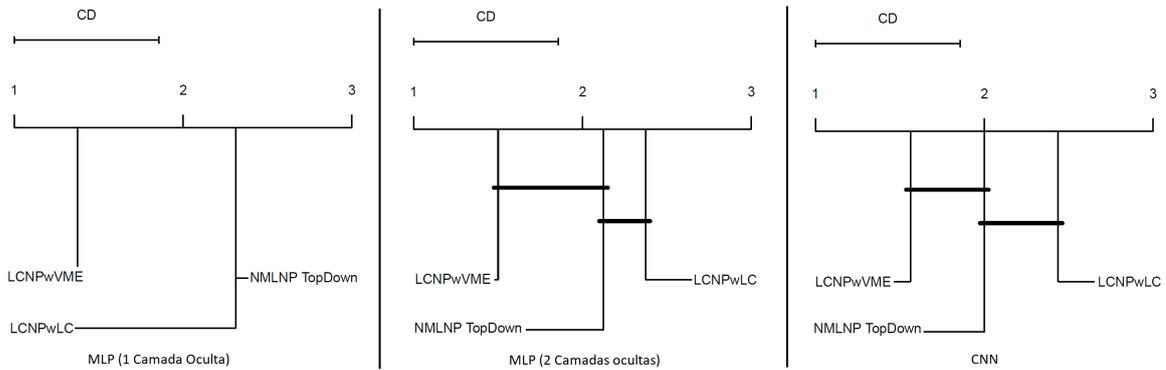
De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 65, para todas as bases de dados avaliadas, bem como comparando o método proposto LCPNwLC com os métodos da literatura Top-Down e LCPNwVME, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwLC foi melhor estatisticamente em 3 bases de dados, 3 empares e perdeu nas outras 10 bases em comparação com o método LCPNwVME. Já na comparação com o método Top-Down, o método LCPNwVME obteve superioridade estatística em 5 bases de dados, 4 empares e perdeu nas outras 7 bases.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwLC foi melhor em 4 e obteve 2 empates das bases de dados em comparação com o método LCPNwVME que foi melhor nas outras 10 bases. Já na comparação do método proposto com o método base Top-down, o método LCPNwLC foi melhor em 4 bases e 2 empates perante ao método Top-Down que foi melhor em 10 bases.
- Com classificador CNN, o método proposto LCPNwLC foi melhor estatisticamente em 4 bases de dados, 2 empares e perdeu em 10 comparado com o método LCPNwVME. Já com o método Top-Down, o LCPNwLC obteve superioridade em 3 bases de dados e houve 1 empate, bem como perdeu estatisticamente em 12 bases de dados.

A Figura 66 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwLC em relação aos métodos da literatura

Top-Down e LCPNwVME. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Figura 66 – Teste Nemenyi da Métrica hP para comparação estatística do método LCPNwLC



Analisando os resultados apresentados pelo teste Nemenyi pela métrica hP, conforme mostrado na Figura 66, este método proposto não apresentou diferença estatística em relação ao método da literatura Top-Down nas três abordagens de classificação analisadas. Além disso, o método LCPNwVME apresentou superioridade estatística em relação a este método proposto LCPNwLC nas três abordagens.

6.6 Método LCPNwINT

Nesta seção é apresentado o método proposto denominado LCPNwINT (Local Classifier per Parent Node with Intersection). Este método propõe a interseção de diferentes critérios de parada e tem por objetivo principal melhorar os resultados apresentados por estas abordagens principalmente no quesito da precisão dos resultados. A combinação (interseção) pode ser feita por dois ou mais critérios de parada. Além disso, podem ser combinados tanto critérios MLNP, quanto NMLNP.

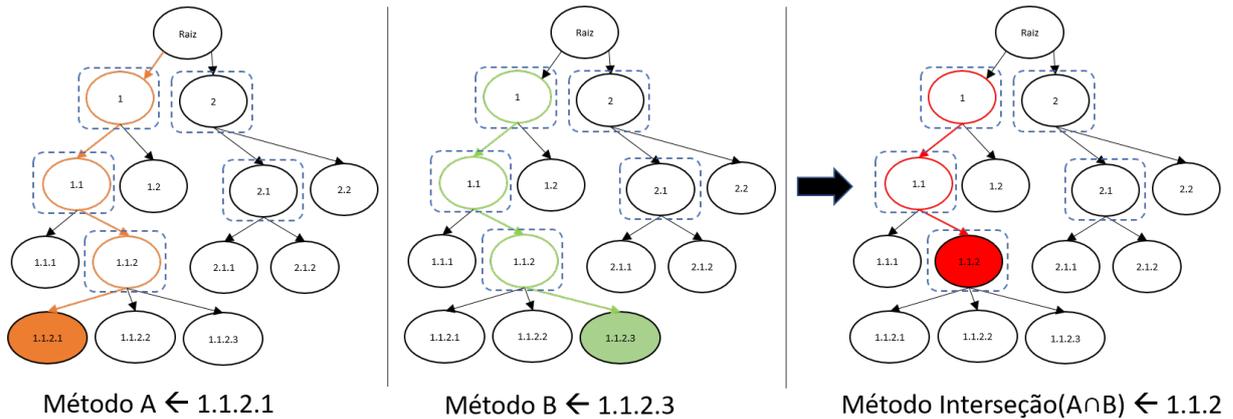
Dado que ρ_i contém os caminhos (paths) de 2 ou mais critérios de parada e φ representa o resultado da interseção dos caminhos retornados pelo método LCPNwINT. Este método é definido pela Equação 6.8.

$$\text{LCPNwINT}(\varphi|\rho_i) = \bigcap_{n=1}^i \rho_n \tag{6.8}$$

O exemplo da Figura 67 mostra a combinação de dois métodos MLNP's fictícios e o resultado apresentado por esse método de combinação, supondo que o primeiro método retornou o resultado da Classe Predita 1.1.2.1 e o segundo método retornou a Classe

Predita 1.1.2.3. Após a obtenção dos resultados destes dois métodos, aplica-se o método de critério de parada baseado na interseção. Nesta instância de teste, por exemplo, este método obteve retorno da classe predita 1.1.2.

Figura 67 – Exemplo do método LCPNwINT para Critério de Parada



O Algoritmo 7 apresenta o método proposto baseado na interseção entre dois ou mais métodos de critérios de parada.

Algoritmo 7 Interseção de Métodos Critérios de Parada

Inputs: N : Quantidade de Paths, $path_N$: conjuntos de caminhos dos métodos

Output: $pathInterseção$: path com a Interseção entre os métodos (1, 2,.. N)

```

1: procedure INTERSEÇÃO( $N, path_N$ )
2:    $pathInterseção \leftarrow path_1 \cap path_2 \cap \dots path_N$    ▷ Recebe a interseção dos métodos
   para cada instância de teste
3:   if  $pathIntersecao \neq \emptyset$  then
4:     return  $pathInterseção$ 
5:   else
6:      $noFilho \leftarrow retornarNoFilhocomMaiorProbabilidade(C_{Raiz}(probasFilhos))$ 
7:     return  $retornarpath(RaizaténoFilho)$ 
8:   end if
9: end procedure

```

Desta forma, mesmo tendo duas abordagens MLNP's, o método de interseção pode realizar a predição em um nó interno da hierarquia, ou seja, realizar uma predição não obrigatória de nó-folha (NMLNP).

Caso o conjunto da interseção de dois ou mais métodos retorne um conjunto vazio, ou seja, nenhum resultado seja igual entre os métodos analisados, nesta pesquisa optou-se por retornar o Nó-Filho com a maior probabilidade do primeiro classificador Nó-Pai (Raiz).

6.6.1 Resultados computacionais

Nesta seção são apresentados os resultados obtidos pelo método proposto LCPNwINT em comparação com os métodos da literatura NMLNP Top-Down (CECI; MALERBA, 2007) e do método baseado na Entropia LCPNwVME (seção 6.4) pelo fato destes apresentarem melhores resultados em comparação com Estado da Arte. Além disso, análises estatísticas das métricas hF e hP são realizadas para a verificação se há diferença estatística entre estes métodos avaliados.

A Tabela 20 apresenta os resultados da métrica hF para cada base de dados, destacando em negrito o melhor resultado entre este método proposto LCPNwINT, LCPNwVME e o método da literatura Top-Down. Tais resultados foram separados pelas três abordagens de classificação com a utilização dos algoritmos: MLP com 1 e 2 camadas ocultas; e CNN.

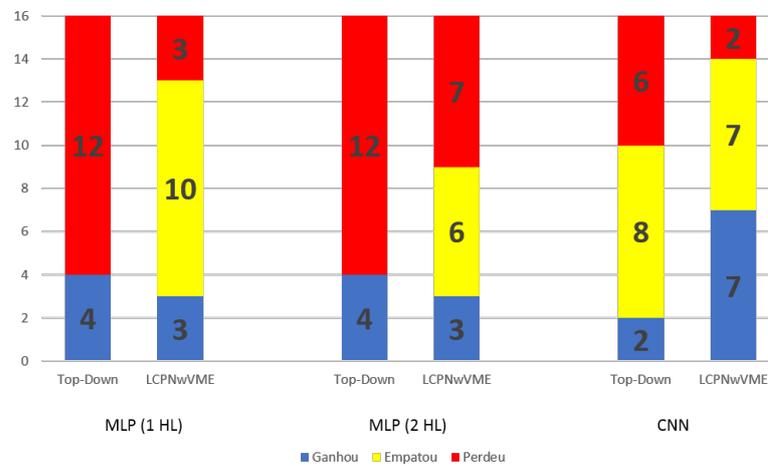
Tabela 20 – Resultados da métrica hF dos métodos LCPNwINT, Top-Down e LCPNwVME

Classificador	hF - MLP (1 Camada Oculta)			hF - MLP (2 Camadas Ocultas)			hF - CNN		
	LCPNw INT	NMLNP Top-Down	LCPNw VME	LCPNw INT	NMLNP Top-Down	LCPNw VME	LCPNw INT	NMLNP Top-Down	LCPNw VME
ECinterpro	94,57	94,01	94,39	94,45	93,84	94,34	93,79	93,59	93,76
ECpfam	93,80	93,12	93,63	93,21	92,50	93,14	92,68	92,37	92,62
ECprints	91,98	91,60	91,92	91,78	91,26	91,74	91,19	91,37	90,91
ECprosite	94,07	93,58	93,85	93,71	93,22	93,60	93,33	93,14	93,21
GPCRinterpro	80,61	81,15	80,67	81,00	81,32	81,07	80,51	80,82	80,47
GPCRpfam	65,45	66,52	65,65	65,36	66,41	65,74	65,59	65,76	66,13
GPCRprints	77,86	78,66	77,95	77,60	78,28	77,79	78,29	78,65	78,28
GPCRprosite	59,85	60,97	59,84	60,49	61,20	60,89	60,94	61,17	61,91
SM-LD-ECinterpro	48,96	49,51	49,07	47,89	48,32	48,03	59,55	59,49	59,50
SM-LD-ECpfam	48,94	49,78	49,05	49,10	49,70	49,21	60,49	60,52	60,41
SM-LD-ECprints	55,60	56,32	55,64	54,57	55,06	54,55	65,00	64,94	64,87
SM-LD-ECprosite	49,62	50,53	49,68	48,57	49,11	48,53	61,04	61,14	60,91
SM-LD-GPCRinterpro	73,39	74,35	73,45	69,29	70,19	69,36	80,08	80,17	80,05
SM-LD-GPCRpfam	74,73	75,52	74,78	71,02	71,92	71,12	81,89	81,94	81,82
SM-LD-GPCRprints	74,58	75,53	74,62	70,72	71,75	70,79	81,41	81,47	81,29
SM-LD-GPCRprosite	74,01	74,82	74,06	68,81	69,73	68,86	81,10	81,13	80,96
Média Geral	72,38	72,87	72,39	71,10	71,49	71,17	76,68	76,73	76,70

A abordagem proposta LCPNwINT foi comparada de modo pareado com as outras abordagens da literatura mencionadas. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hF, para cada base de dados. Na Figura 68 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

Por meio da análise pareada dos resultados da métrica hF, conforme apresentado na Figura 68, para todas as bases de dados avaliadas, bem como comparando o método

Figura 68 – Comparação pareada do método proposto LCPNwINT com métodos da literatura

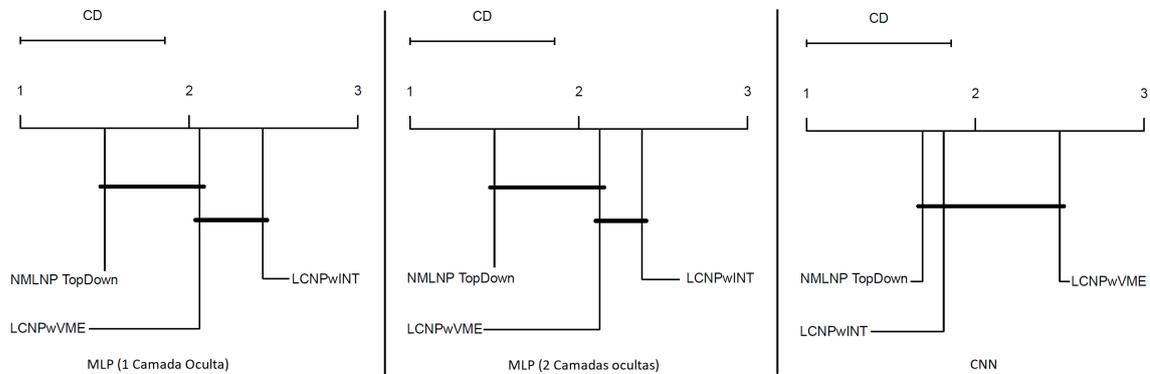


proposto LCPNwINT com os métodos LCPNwVME e da literatura Top-Down, obteve-se:

- Com classificador MLP com 1 camada oculta, o método proposto LCPNwINT foi melhor estatisticamente em 3 bases de dados, 10 empates e perdeu apenas em 3 bases em comparação com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwINT foi melhor em 4 bases de dados e perdeu em 12 bases perante ao método Top-Down.
- Com classificador MLP com 2 camadas ocultas, o método proposto LCPNwINT obteve superioridade estatística em 3 bases de dados, 6 empates e perdeu em 7 bases comparado com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwINT foi melhor em 4 bases de dados e perdeu em 12 bases perante ao método Top-Down.
- Com classificador CNN, o método proposto LCPNwINT foi melhor estatisticamente em 7 bases, 7 empates e perdeu em 2 bases em comparação com o método LCPNwVME. Já na comparação do método proposto com o método base Top-Down, o método LCPNwINT foi melhor em 2 bases de dados e obteve 8 empates perante ao método Top-Down que foi melhor nas outras 6 bases.

A Figura 69 apresenta o teste de Nemenyi dos resultados da métrica hF para avaliação estatística do método proposto LCPNwINT em relação aos métodos da literatura NMLNP Top-Down e LCPNwVME. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Figura 69 – Teste Nemenyi da Métrica hF para comparação do método proposto LCPNwINT



Pelos testes de Nemenyi realizados nos resultados dos métodos pela métrica de avaliação hF, Figura 69, o método proposto LCPNwINT não apresentou diferença estatística em relação ao método LCPNwVME nas três abordagens com algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN. Além disso, o método da literatura Top-Down apresentou superioridade estatística perante este método proposto LCPNwINT nas abordagens com algoritmos de classificação MLP com 1 e 2 camadas ocultas. Na abordagem com algoritmo classificador CNN os três métodos avaliados não apresentaram diferença estatística conforme teste de Friedman com confiança de 95%.

Para uma melhor análise com relação à precisão das predições, a Tabela 21 mostra os resultados da métrica hP para todas as bases de dados avaliadas, destacando-se em negrito o melhor resultado entre o método proposto LCPNwINT e dos métodos da literatura Top-Down e LCPNwVME. Tais resultados foram separadas por três abordagens com utilização dos algoritmos de classificação MLP (com 1 e 2 camadas ocultas) e CNN.

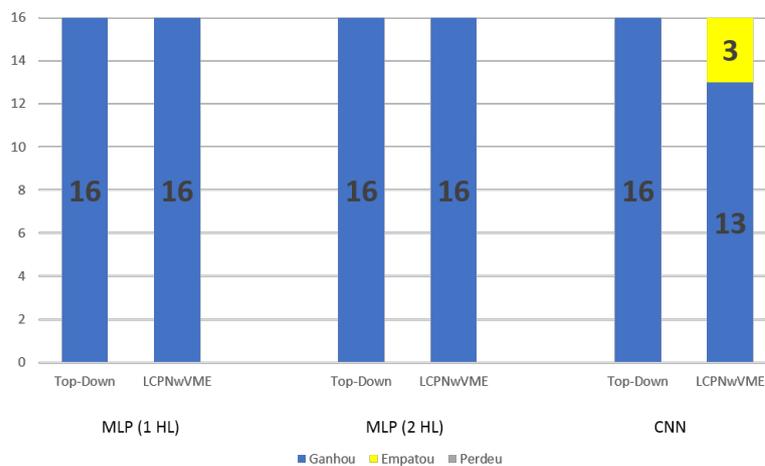
A abordagem proposta LCPNwINT foi comparada de modo pareado com as outras abordagens da literatura mencionadas pelos resultados apresentados pela métrica de avaliação hP. Foi utilizado o teste estatístico de Wilcoxon para a comparação par a par, utilizando os resultados dos 10 Folds da métrica hP, para cada base de dados. Na Figura 70 está presente o número de vitórias, empates e derrotas agrupados por colunas que refletem a comparação da proposta com os métodos da literatura nomeados logo abaixo da respectiva coluna, bem como foram apresentadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas (respectivamente, 1 HL e 2 HL) e CNN.

De acordo com a comparação pareada dos resultados da métrica hP, conforme apresentado pela Figura 70, para todas as bases de dados avaliadas, o método proposto LCPNwINT obteve resultados melhores em todas as bases de dados em comparação com os métodos da literatura Top-Down nos três algoritmos de classificação avaliados. Em

Tabela 21 – Resultados da métrica hP dos métodos LCPNwINT, Top-Down e LCPNwVME

Classificador	hP - MLP (1 Camada Oculta)			hP - MLP (2 Camadas Ocultas)			hP - CNN		
	LCPNwINT	NMLNP Top-Down	LCPNwVME	LCPNwINT	NMLNP Top-Down	LCPNwVME	LCPNwINT	NMLNP Top-Down	LCPNwVME
ECinterpro	93,51	91,93	93,16	93,12	91,57	92,87	92,70	92,05	92,59
ECpfam	93,71	90,99	93,30	92,75	90,13	92,55	92,12	90,42	91,93
ECprints	93,35	89,95	93,15	92,72	89,61	92,52	92,47	91,56	91,44
ECprosite	94,71	91,63	94,27	94,31	91,30	94,01	93,64	91,96	93,41
GPCRinterpro	88,93	87,67	88,81	88,93	87,60	88,77	88,42	87,91	88,41
GPCRpfam	89,38	87,22	86,39	89,03	87,48	85,70	89,66	89,42	84,39
GPCRprints	88,89	86,31	88,70	88,30	86,06	87,96	88,51	88,09	87,31
GPCRprosite	79,60	76,61	78,14	79,04	77,85	76,62	81,21	81,00	75,72
SM-LD-ECinterpro	56,06	52,73	55,27	55,18	52,69	54,16	62,77	61,28	62,49
SM-LD-ECpfam	56,09	52,30	55,46	57,31	54,71	56,44	63,42	62,19	63,09
SM-LD-ECprints	61,49	57,35	61,20	61,07	57,09	60,55	67,96	66,39	67,31
SM-LD-ECprosite	56,00	52,42	55,35	55,51	52,25	54,58	64,10	62,71	63,54
SM-LD-GPCRinterpro	80,35	76,33	80,16	78,40	74,47	77,96	82,85	81,96	82,41
SM-LD-GPCRpfam	82,01	77,59	81,75	80,50	76,18	80,23	84,69	83,74	84,15
SM-LD-GPCRprints	82,92	77,91	82,69	82,28	77,78	81,71	84,00	83,11	83,27
SM-LD-GPCRprosite	81,54	77,20	81,29	79,51	75,55	79,02	83,38	82,43	82,85
Média Geral	79,91	76,63	79,32	79,25	76,39	78,48	81,99	81,01	80,89

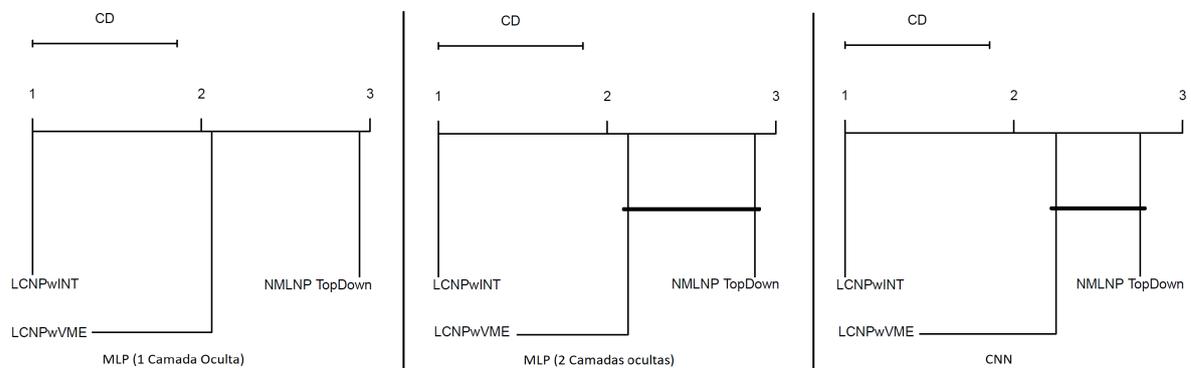
Figura 70 – Comparação pareada do método proposto LCPNwINT com métodos da literatura por meio da Métrica hP



comparação com o método LCPNwVME, o método LCPNwINT foi melhor estatisticamente em todas as bases de dados com os algoritmos de classificação MLP com 1 e 2 camadas ocultas. Com o algoritmo CNN o método LCPNwINT obteve superioridade estatística em 13 bases de dados e 3 empates em comparação com o método LCPNwVME.

A Figura 71 apresenta o teste de Nemenyi dos resultados da métrica hP para avaliação estatística do método proposto LCPNwINT em relação aos métodos da literatura Top-Down e LCPNwVME. Três avaliações estatísticas foram realizadas separadamente para comparação das abordagens com utilização dos algoritmos de classificação MLP de 1 e 2 camadas e CNN.

Figura 71 – Teste Nemenyi da Métrica hP para comparação estatística do método LCPNwINT



Analisando os resultados apresentados pelo teste Nemenyi pela métrica hP, conforme mostrado na Figura 71, este método proposto apresentou superioridade estatística em comparação aos métodos LCPNwVME e Top-Down nas três abordagens com algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

6.7 Avaliação de erros por nível

Nesta seção são mostrados a quantidade de erros por nível dos métodos propostos LCPNwFOE, LCPNwVOE, LCPNwFME, LCPNwVME, LCPNwLC e LCPNwINT em comparação com os métodos da literatura NMLNP Top-Down (CECI; MALERBA, 2007) e Ganho de Informação (HERNÁNDEZ; SUCAR; MORALES, 2014) utilizados para comparação nesta pesquisa.

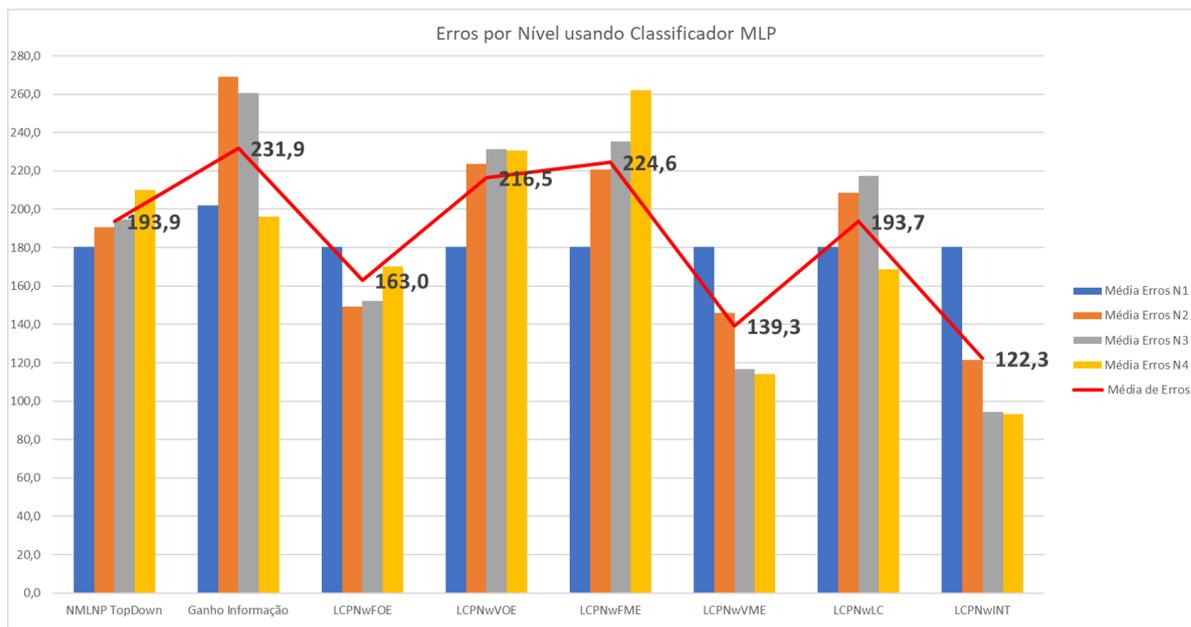
Para realizar a avaliação dos resultados por nível, visando analisar quais métodos obtêm uma quantidade menor de erros em suas predições neste trabalho adota-se a estratégia que prioriza os métodos de acordo com uma melhor precisão das classificações, cujos erros não são contabilizados para os nós-filhos caso a predição tenha parado antecipadamente, conforme mostrado nas instâncias "4" e "5" do exemplo mostrado na Figura 72.

Figura 72 – Exemplo da Avaliação de Erros por Nível

Instância	Nós Verdadeiros					Nós Preditos					Avaliação Erros por Nível				
	N1	N2	N3	N4		N1	N2	N3	N4		N1	N2	N3	N4	
1	'4'					'4'					C				
2	'4'					'3'	3	1'			E	E			
3	'4'					'4'	'4	1'			C	E			
4	'4'	'4	1'			'4'					C				
5	'4'	'4	1'	'4	1	1'	4	1	1	1'	C	C			
6	'4'					'4'	'4	1'	'4	1	1'	4	1	1	1'

A Figura 73 apresenta a quantidade de erros por nível dos métodos avaliados utilizando algoritmos de classificação MLP com 1 camada oculta, bem como a média de erros dos níveis (1,2,3,4) destacado pela linha vermelha para cada método avaliado.

Figura 73 – Avaliação da Média de Erros por Nível com classificador MLP



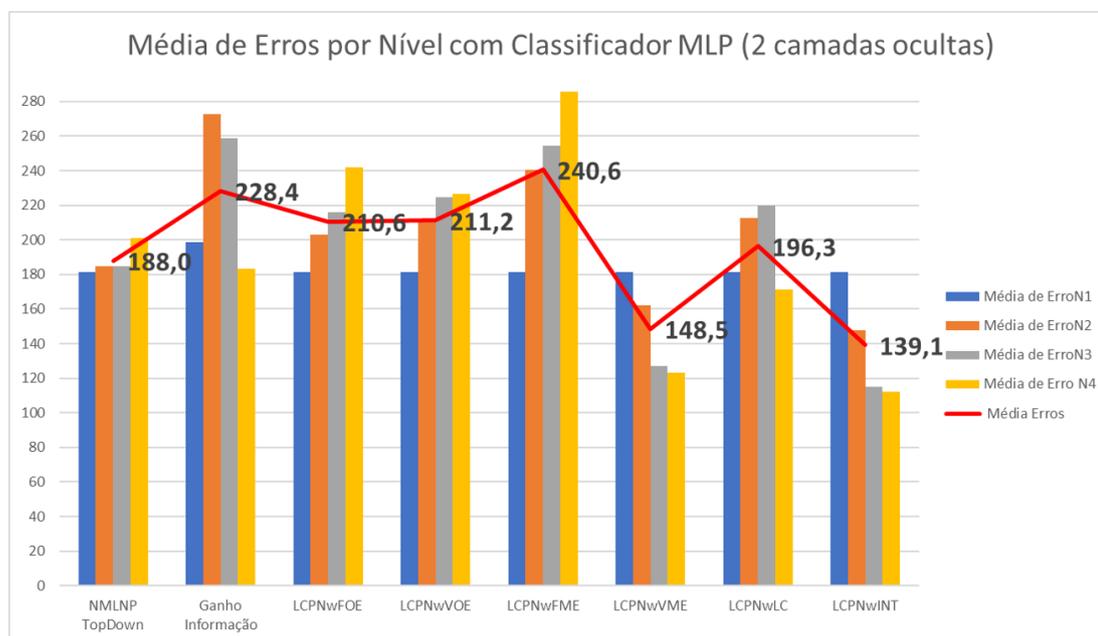
Observa-se nesta Figura 73 que os métodos propostos, LCPNwINT, LCPNwVME e LCPNwFOE obtiveram melhores resultados, na respectiva ordem decrescente de ranqueamento conforme a menor média de erros geral. Ou seja, propagaram menos erros para os níveis inferiores da hierarquia, bem como apresentaram uma menor quantidade de erros em comparação com os demais métodos avaliados.

Esta mesma avaliação de erros foi aplicado nos métodos utilizando os algoritmos de classificação MLP com 2 camadas ocultas e CNN. Tais avaliações foram analisadas individualmente, pois como já dito anteriormente, esta pesquisa propõe método de critério de parada independente do algoritmo classificador utilizado para indução.

A Figura 74 apresenta a quantidade de erros por nível para cada método utilizando algoritmo de classificação MLP com 2 camadas ocultas, bem como apresenta a média de erros geral dos níveis entre os mesmos, destacado pela linha vermelha para cada método de critério de parada avaliado.

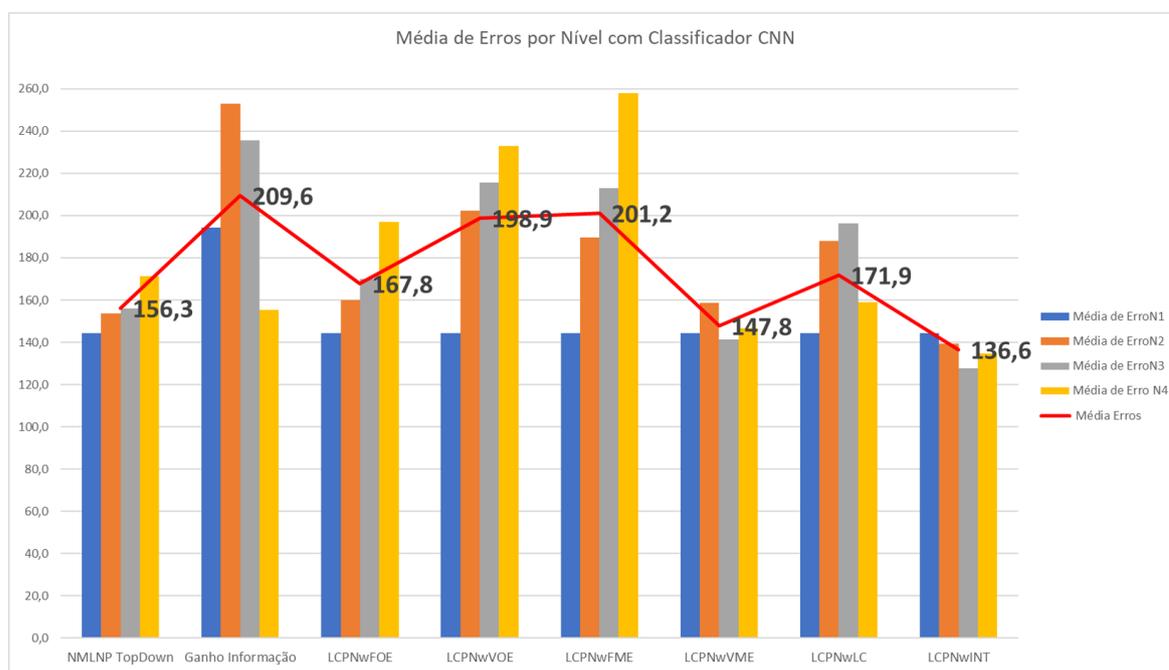
Observa-se neste Figura 74 que os métodos propostos, LCPNwINT e LCPNwVME obtiveram melhores resultados, na respectiva ordem decrescente de ranqueamento conforme a menor média de erros geral. Ou seja, propagaram menos erros para os níveis inferiores da hierarquia, bem como apresentaram uma menor quantidade de erros em comparação com o método da literatura Top-Down e os demais métodos avaliados.

Figura 74 – Avaliação da Média de Erros por Nível dos métodos com classificador MLP (2 camadas ocultas)



Por fim, a Figura 75 apresenta a quantidade de erros por nível para cada método utilizando algoritmo de classificação CNN, bem como apresenta a média de erros geral dos níveis entre os mesmos, destacado pela linha vermelha para cada método de critério de parada avaliado.

Figura 75 – Avaliação da Média de Erros por Nível com classificador CNN



Observa-se neste Figura 75 que os métodos propostos, LCPNwINT e LCPNwVME obtiveram melhores resultados, na respectiva ordem decrescente de ranqueamento conforme a menor média de erros geral. Ou seja, propagaram menos erros para os níveis inferiores da hierarquia, bem como apresentaram uma menor quantidade de erros em comparação com o método da literatura Top-Down e os demais métodos avaliados.

Para as bases de dados avaliadas nesta pesquisa que apresentam taxonomias desbalanceadas e quantidades menores de instâncias de treinamento, geralmente para classes que se encontram em níveis inferiores da hierarquia, a escolha do tipo de avaliação e do método de critério de parada (NMLNP) são importantes para a determinação do resultado de predição. Ou seja, tais abordagens devem analisar a profundidade e a qualidade da rotulação. Com a priorização da precisão, realiza-se a predição de nós mais internos, evitando-se a propagação de erros pela hierarquia. Todavia, uma rotulação mais profunda na hierarquia sujeita-se à uma propagação de erros maior em abordagens NMLNP. Estas questões devem ser analisadas para a escolha do método de acordo com o contexto ou domínio em que serão aplicadas tais abordagens de avaliação de classificação hierárquica.

6.8 Avaliação dos resultados com otimização do Limiar (Threshold) por Base de Dados

Nesta seção são apresentados os resultados obtidos pela otimização (*Tuning*) dos limiares delimitados individualmente por base de dados. Além disso, foram realizados testes estatísticos para comparação entre os métodos avaliados.

Para comparação destes resultados os métodos propostos LCPNwFOE, LCPNwVOE, LCPNwFME, LCPNwVME, LCPNwLC e LCPNwINT foram comparados com o método da literatura NMLNP Top-Down (CECI; MALERBA, 2007), pelo fato deste último ter apresentado melhores resultados em comparação com os demais trabalhos da literatura, avaliados no Capítulo 5.

Tais resultados são apresentados por meio das métricas de avaliação hF e hP, além da comparação separada pela utilização dos algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN nos respectivos métodos avaliados.

6.8.1 Resultados da otimização de Limiares com classificador MLP com 1 camada oculta

Nesta seção, são apresentados os resultados das métricas hF e hP obtidos nos experimentos pela otimização dos limiares por base de dados, bem como foram utilizados classificadores MLP com 1 camada oculta em todos os nós-pai da hierarquia.

A Tabela 22 apresenta os resultados da métrica hF aplicados aos métodos avaliados com algoritmo de classificação MLP com 1 camada oculta, bem como a média geral obtida para cada método.

Tabela 22 – Resultados hF com otimização de limiares - Classificador MLP com 1 camada oculta

Resultados HF com Otimização do Limiar por Dataset - Classificador MLP (1 camada oculta)							
Dataset	NMLNP TopDown	LCPNw FOE	LCPNw VOE	LCPNw FME	LCPNw VME	LCPNw LC	LCPNw INT
ECinterpro	94,95	94,85	93,97	94,56	95,04	95,79	95,00
ECpfam	93,95	93,94	93,22	92,86	93,91	94,70	93,94
ECprints	92,33	92,16	91,35	91,58	91,97	92,90	92,02
ECprosite	94,17	94,16	93,56	93,77	94,05	95,04	94,13
GPCRinterpro	81,10	81,13	80,84	80,85	80,67	79,22	80,63
GPCRpfam	66,83	66,37	66,37	65,73	65,68	66,24	65,72
GPCRprints	78,66	78,50	78,08	78,10	77,95	76,07	77,98
GPCRprosite	60,84	60,47	60,66	60,35	59,80	59,85	59,83
SM-LD-ECinterpro	49,51	49,11	49,03	48,43	49,07	49,60	49,05
SM-LD-ECpfam	49,78	49,34	49,64	48,67	49,05	49,78	49,02
SM-LD-ECprints	56,40	56,17	55,67	55,65	55,64	56,40	55,53
SM-LD-ECprosite	50,53	50,18	50,22	49,57	49,68	50,34	49,73
SM-LD-GPCRinterpro	74,35	74,17	73,63	73,99	73,45	74,10	73,43
SM-LD-GPCRpfam	75,52	75,31	74,63	74,95	74,78	75,24	74,80
SM-LD-GPCRprints	75,53	75,30	74,62	75,24	74,62	75,84	74,63
SM-LD-GPCRprosite	74,85	74,64	73,91	74,33	74,06	74,62	74,01
Média Geral	73,08	72,86	72,46	72,41	72,46	72,86	72,47

Conforme Tabela 22, avaliando os resultados da métrica hF de cada método e analisando os resultados individualmente para cada base de dados, o método LCPNwLC foi melhor em 8 bases de dados, seguido por NMLNP Top-Down que foi melhor em 7 e LCPNwFOE que foi melhor em apenas 1 base. Avaliando a média geral o método Top-Dow apresentou melhor média geral, embora as médias dos outros métodos apresentem-se similares.

Com relação a métrica de avaliação hP, a Tabela 23 apresenta os resultados dos métodos avaliados com algoritmo de classificação MLP, bem como a média geral obtida para cada método.

De acordo com a Tabela 23 e avaliando os resultados da métrica hF de cada método e analisando os resultados individualmente para cada base de dados, o método LCPNwINT foi melhor em 13 bases de dados, seguido por LCPNwLC que foi melhor em 3 bases de dados. Avaliando apenas a média geral o melhor resultado foi alcançado pelo método LCPNwINT. Além disso, somente os métodos LCPNwINT, LCPNwVME e LCPNwFOE, em ordem decrescente da média, obtiveram médias melhores que o método do estado da arte NMLNP Top-Down.

Tabela 23 – Resultados hP com otimização de limiares - Classificador MLP com 1 camada oculta

Resultados hP com otimização do limiar por base de dados - Classificador MLP (1 camada oculta)							
Base de Dados	NMLNP TopDown	LCPNw FOE	LCPNw VOE	LCPNw FME	LCPNw VME	LCPNwLC	LCPNw INT
ECinterpro	95,40	94,21	91,74	94,48	95,08	95,83	95,51
ECpfam	94,07	94,11	91,19	94,12	94,55	95,08	94,76
ECprints	92,27	92,42	89,36	91,50	93,47	93,06	93,75
ECprosite	94,27	94,11	91,36	94,35	95,18	95,45	95,42
GPCRinterpro	88,44	87,62	87,04	86,56	88,81	82,11	88,97
GPCRpfam	81,73	86,71	80,81	83,26	87,18	86,02	87,44
GPCRprints	86,31	87,49	84,02	86,12	88,70	80,32	88,79
GPCRprosite	72,63	78,73	73,20	73,50	78,63	78,03	78,76
SM-LD-ECinterpro	52,73	50,45	52,22	48,20	55,27	52,66	55,82
SM-LD-ECpfam	52,30	49,85	51,69	48,63	55,46	52,82	55,91
SM-LD-ECprints	59,76	59,03	55,14	55,79	61,20	58,59	61,77
SM-LD-ECprosite	52,42	54,02	51,48	49,99	55,35	51,74	55,84
SM-LD-GPCRinterpro	76,33	75,56	73,60	75,94	80,16	74,98	80,25
SM-LD-GPCRpfam	77,59	76,74	74,89	76,90	81,75	76,08	81,84
SM-LD-GPCRprints	77,91	76,86	73,72	77,90	82,69	77,65	82,80
SM-LD-GPCRprosite	79,48	79,31	73,54	76,06	81,29	76,74	81,65
Média Geral	77,10	77,33	74,69	75,83	79,67	76,70	79,95

6.8.2 Resultados da otimização de Limiares com classificador MLP com 2 camadas ocultas

Nesta seção, são apresentados os resultados das métricas hF e hP obtidos nos experimentos pela otimização dos limiares por base de dados, bem como foram utilizados classificadores MLP com 2 camadas ocultas em todos os Nós-Pai da hierarquia.

A Tabela 24 apresenta os resultados da métrica hF aplicados aos métodos avaliados com algoritmo de classificação MLP com 2 camadas ocultas, bem como a média geral obtida para cada método.

Conforme Tabela 24, avaliando-se os resultados da métrica hF de cada método e analisando os resultados individualmente para cada base de dados, o método NMLNP Top-Down foi melhor em 11 bases de dados, seguido pelo método LCPNwLC que apresentou resultados melhores em 4 bases de dados e LCPNwFOE com resultado melhor em apenas 1 base. Por fim, analisando a média geral, o método Top-Dow apresentou melhor resultado com a média geral em comparação com os outros métodos.

Com relação a métrica de avaliação hP, a Tabela 25 apresenta os resultados dos métodos avaliados com algoritmo de classificação MLP com 2 camadas ocultas, bem como a média geral obtida para cada método.

De acordo com a Tabela 25, avaliando-se os resultados da métrica hP de cada método e analisando os resultados individualmente para cada base de dados, o método LCPNwINT foi melhor em 12 bases de dados, seguido por LCPNwLC que foi melhor em

Tabela 24 – Resultados hF com otimização de limiares - Classificador MLP com 2 camadas ocultas

Resultados HF com Otimização do Limiar por Dataset - Classificador MLP (2 camadas ocultas)							
Dataset	NMLNP TopDown	LCPNwFOE	LCPNwVOE	LCPNwFME	LCPNwVME	LCPNwLC	LCPNwINT
ECinterpro	94,86	94,69	93,53	94,48	94,97	95,90	94,86
ECpfam	93,39	93,45	92,53	92,74	93,24	94,33	93,37
ECprints	92,05	91,99	91,03	91,33	91,77	92,65	91,82
ECprosite	93,85	93,80	93,26	93,62	93,68	94,73	93,72
GPCRinterpro	81,32	81,36	81,02	81,09	81,07	79,59	81,07
GPCRpfam	66,69	66,27	66,08	65,50	65,74	66,21	65,76
GPCRprints	78,28	78,25	77,71	77,79	77,79	75,76	77,76
GPCRprosite	61,41	60,92	61,22	61,00	60,89	60,98	61,03
SM-LD-ECinterpro	48,32	48,02	48,00	46,91	48,03	48,23	47,92
SM-LD-ECpfam	49,70	49,30	49,46	48,10	49,21	49,59	49,15
SM-LD-ECprints	55,08	54,94	54,30	54,07	54,55	54,83	54,58
SM-LD-ECprosite	49,11	48,60	48,60	47,81	48,53	48,72	48,62
SM-LD-GPCRinterpro	70,19	69,99	68,90	69,60	69,36	69,62	69,40
SM-LD-GPCRpfam	71,92	71,84	70,67	71,11	71,12	71,29	71,12
SM-LD-GPCRprints	71,75	71,52	70,15	71,07	70,79	71,06	70,74
SM-LD-GPCRprosite	69,73	69,54	68,29	68,76	68,86	68,76	68,89
Média Geral	71,73	71,53	70,92	70,94	71,23	71,39	71,24

Tabela 25 – Resultados hP com otimização de Limiares - Classificador MLP com 2 camadas ocultas

Resultados HP com Otimização do Limiar por Dataset - Classificador MLP (2 camadas ocultas)							
Dataset	NMLNP TopDown	LCPNwFOE	LCPNwVOE	LCPNwFME	LCPNwVME	LCPNwLC	LCPNwINT
ECinterpro	95,00	93,90	90,81	94,18	94,56	95,95	95,04
ECpfam	93,89	93,45	90,20	93,57	93,19	94,68	93,95
ECprints	92,09	92,40	89,01	92,00	92,93	93,12	93,35
ECprosite	94,01	93,49	91,23	93,57	95,35	95,19	95,46
GPCRinterpro	87,60	87,67	86,12	86,58	88,77	82,12	88,83
GPCRpfam	84,19	87,07	83,68	81,79	85,70	85,57	86,83
GPCRprints	86,06	87,19	83,71	85,22	87,96	79,85	88,14
GPCRprosite	73,51	79,17	74,60	72,07	76,62	77,78	77,30
SM-LD-ECinterpro	52,69	50,86	52,72	47,19	54,16	51,43	54,95
SM-LD-ECpfam	54,71	52,26	54,22	50,06	56,44	53,08	57,14
SM-LD-ECprints	59,43	59,07	54,44	54,98	60,55	57,46	61,40
SM-LD-ECprosite	52,25	49,96	51,33	49,34	54,58	51,48	55,31
SM-LD-GPCRinterpro	74,47	73,90	70,01	74,40	77,96	72,48	78,26
SM-LD-GPCRpfam	76,18	75,49	72,35	75,85	80,23	73,61	80,40
SM-LD-GPCRprints	77,78	77,01	71,23	77,16	81,71	73,97	82,11
SM-LD-GPCRprosite	75,55	74,96	69,97	73,99	79,02	71,56	79,32
Média Geral	76,84	76,74	74,10	75,12	78,73	75,58	79,24

2 bases de dados e LCPNwFOE com resultado melhor em 2 bases. Avaliando apenas a média geral o melhor resultado foi alcançado pelo método LCPNwINT. Além disso, os métodos LCPNwINT e LCPNwVME, em ordem decrescente da média, obtiveram médias melhores que o método do estado da arte NMLNP Top-Down.

6.8.3 Resultados da otimização de Limiares com classificador CNN

Nesta seção, são apresentados os resultados das métricas hF e hP obtidos nos experimentos pela otimização dos limiares por base de dados, bem como foram utilizados classificadores CNN em todos os nós-pai da hierarquia.

A Tabela 26 apresenta os resultados da métrica hF aplicados aos métodos avaliados com algoritmo de classificação CNN, bem como a média geral obtida para cada método.

Tabela 26 – Resultados hF com otimização de limiares - Classificador CNN

Resultados hF com Otimização do Limiar por Dataset - Classificador CNN							
Dataset	NMLNP Top-Down	LCPNwFOE	LCPNwVOE	LCPNwFME	LCPNwVME	LCPNwLC	LCPNwINT
ECinterpro	93,60	93,60	92,86	93,34	93,93	95,28	93,66
ECpfam	92,73	92,69	92,10	92,38	92,69	94,19	92,77
ECprints	91,37	91,24	90,08	90,77	91,08	91,92	91,17
ECprosite	93,49	93,32	92,74	93,13	93,30	94,32	93,39
GPCRinterpro	80,94	80,98	80,61	80,64	80,57	79,11	80,56
GPCRpfam	66,81	66,48	66,09	65,82	66,13	66,50	66,24
GPCRprints	78,72	78,66	78,12	78,46	78,28	76,29	78,48
GPCRprosite	62,07	61,55	61,34	61,36	61,91	61,95	61,64
SM-LD-ECinterpro	59,49	59,55	59,11	59,19	59,50	59,56	59,55
SM-LD-ECpfam	60,52	60,46	59,96	60,17	60,34	60,64	60,49
SM-LD-ECprints	65,03	64,91	64,12	64,72	64,90	64,91	64,93
SM-LD-ECprosite	61,14	61,07	60,59	60,71	60,91	60,92	61,01
SM-LD-GPCRinterpro	80,17	80,12	79,70	79,99	80,09	80,38	80,08
SM-LD-GPCRpfam	81,94	81,85	81,44	81,76	81,87	82,04	81,89
SM-LD-GPCRprints	81,47	81,43	80,61	81,08	81,27	81,46	81,35
SM-LD-GPCRprosite	81,13	81,09	80,38	80,95	80,99	81,29	81,10
Média Geral	76,91	76,81	76,24	76,53	76,73	76,92	76,77

Conforme Tabela 26, avaliando-se os resultados da métrica hF de cada método e analisando os resultados individualmente para cada base de dados, o método LCPNwLC apresentou resultados melhores em 9 bases de dados, seguido pelo método NMLNP Top-Down que foi melhor em 6 bases e LCPNwFOE com resultado melhor em 1 base. Por fim, analisando a média geral o método proposto LCPNwLC apresentou o melhor resultado em comparação com os demais métodos.

Com relação a métrica de avaliação hP, a Tabela 27 apresenta os resultados dos métodos avaliados com algoritmo de classificação CNN, bem como a média geral obtida para cada método.

Tabela 27 – Resultados hP com otimização de limiares - Classificador CNN

Resultados HP com Otimização do Limiar por Dataset - Classificador CNN							
Dataset	NMLNP Top-Down	LCPNwFOE	LCPNwVOE	LCPNwFME	LCPNwVME	LCPNwLC	LCPNwINT
ECinterpro	93,41	92,25	90,14	92,95	93,33	95,36	93,61
ECpfam	92,00	92,19	89,48	91,72	92,23	94,61	92,68
ECprints	91,56	91,85	87,91	91,16	92,53	92,25	92,92
ECprosite	94,74	93,41	90,51	93,78	94,43	94,89	94,92
GPCRinterpro	87,00	86,64	85,68	87,55	88,20	81,59	88,25
GPCRpfam	84,21	87,57	84,53	86,39	84,39	81,53	86,16
GPCRprints	86,71	87,62	84,72	87,94	86,41	80,00	87,67
GPCRprosite	77,95	79,13	71,88	74,89	75,72	74,34	78,72
SM-LD-ECinterpro	61,28	61,78	59,15	61,00	62,49	60,67	62,90
SM-LD-ECpfam	62,19	62,43	59,68	61,95	62,76	61,41	63,42
SM-LD-ECprints	68,04	66,65	63,12	68,42	67,52	65,63	68,75
SM-LD-ECprosite	62,71	63,02	60,32	61,53	63,54	61,52	64,17
SM-LD-GPCRinterpro	81,96	81,43	79,38	81,07	82,04	81,04	82,76
SM-LD-GPCRpfam	83,74	83,14	81,34	82,69	84,01	82,65	84,69
SM-LD-GPCRprints	83,11	83,95	79,81	82,10	83,58	81,96	84,23
SM-LD-GPCRprosite	82,43	82,12	79,48	81,69	82,56	81,98	83,32
Média Geral	80,81	80,95	77,95	80,43	80,98	79,46	81,82

De acordo com a Tabela 25, avaliando-se os resultados da métrica hP de cada método e analisando os resultados individualmente para cada base de dados, o método LCPNwINT foi melhor em 11 bases de dados, seguido por LCPNwLC que foi melhor em 2 bases de dados, LCPNwFOE com resultados melhores em 2 bases e LCPNwFME com resultado melhor em 1 base. Avaliando apenas a média geral o melhor resultado foi alcançado pelo método LCPNwINT. Além disso, os métodos LCPNwINT, LCPNwVME e LCPNwFOE, em ordem decrescente, obtiveram médias melhores que o método do estado da arte NMLNP Top-Down.

6.8.4 Avaliação estatística com otimização de Limiares

Nesta seção foram realizados testes estatísticos para avaliar se há diferença estatística entre os resultados apresentados por meio da otimização de Limiares para cada base de dados. Todos os testes foram aplicados com confiança de 95% e pelo fato do teste de Friedman ter mostrado que há diferença estatística (rejeitando-se a hipótese nula) entre os resultados dos métodos, aplicou-se o teste de Nemenyi.

As Figuras 76 e 77 apresentam os testes de Nemenyi, de acordo com a métrica hF, para avaliação estatística e comparação de desempenho dos métodos propostos e da literatura. Três avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Analisando os testes de Nemenyi de acordo com a métrica de avaliação hF, apresentados nas Figuras 76 e 77, o método base Top-Down ficou ranqueado em primeiro, porém

Figura 76 – Teste de Nemenyi com métrica hF com MLP 1 e 2 camadas

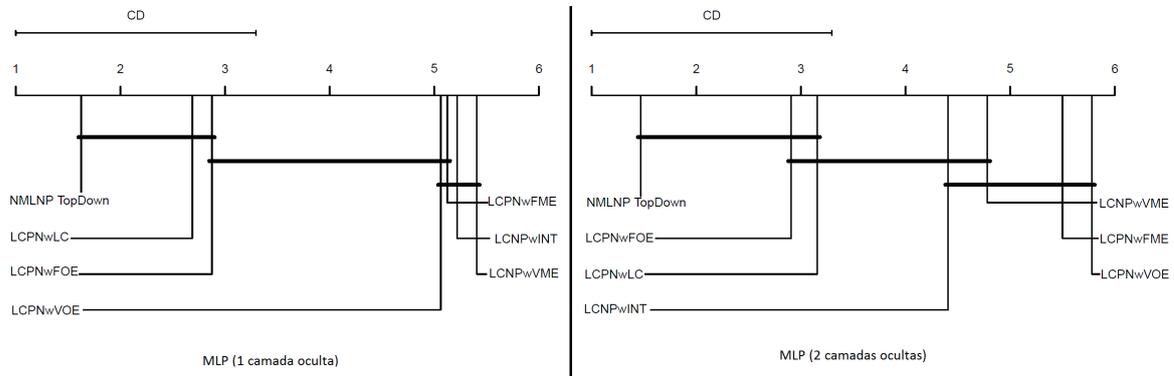
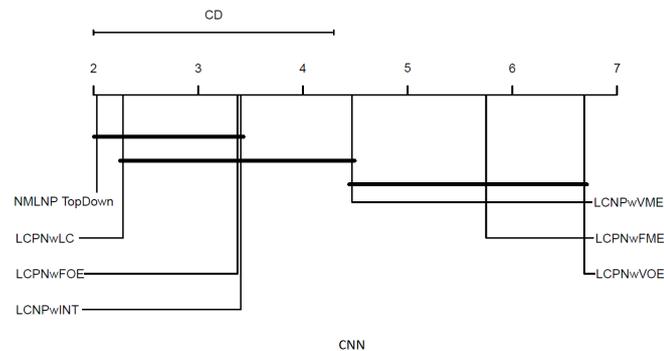


Figura 77 – Teste de Nemenyi com métrica hF com CNN



não apresentou diferença estatística para os métodos propostos LCPNwFOE e LCPNwLC nas abordagens com os três algoritmos de classificação utilizados MLP (1 e 2 camadas) e CNN.

As Figuras 78 e 79 apresentam os testes de Nemenyi, de acordo com a métrica hP, para avaliação estatística e comparação de desempenho dos métodos propostos e da literatura. Conforme já explicado anteriormente, tais avaliações estatísticas foram realizadas separadamente para comparação dos métodos com utilização dos algoritmos de classificação MLP com 1 e 2 camadas ocultas e CNN.

Analisando os testes de Nemenyi de acordo com a métrica de avaliação hP, mostrados nas Figuras 78 e 79, os métodos propostos LCPNwINT e LCPNwVME ficaram, respectivamente, nas primeiras posições e entre eles não apresentaram diferenças estatísticas. Além disso, o método LCPNwINT apresentou superioridade estatística em comparação com o método base NMLNP Top-Down com utilização dos algoritmos de classificação MLP (1 e 2 camadas ocultas) e CNN.

Figura 78 – Teste de Nemenyi com métrica hP com MLP 1 e 2 camadas

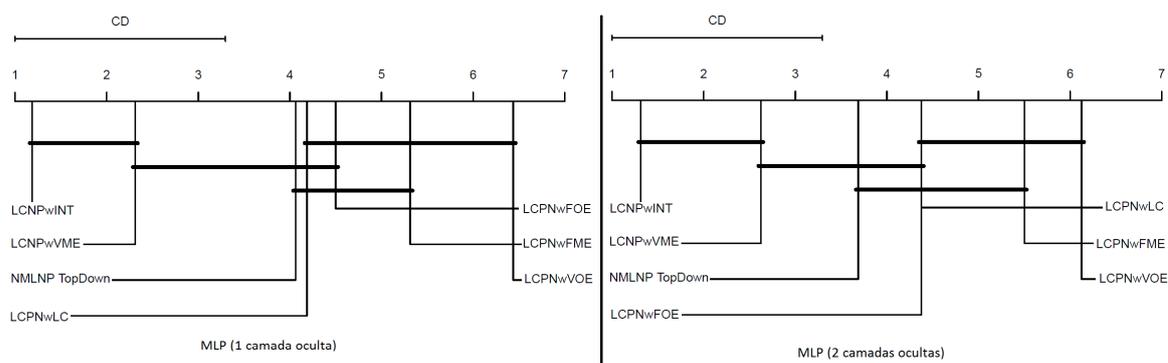
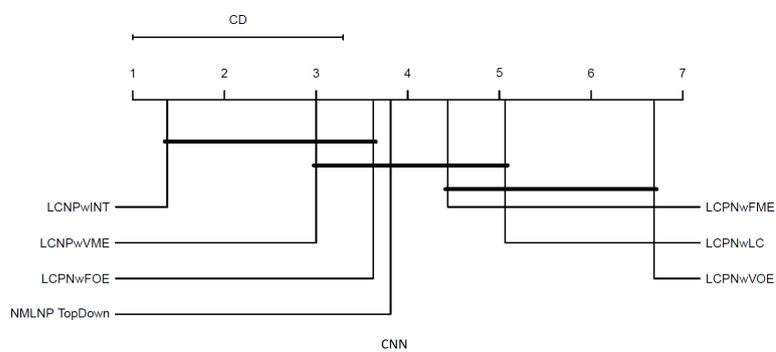


Figura 79 – Teste de Nemenyi com métrica hP com CNN



7 Conclusão

Este trabalho avaliou diversas abordagens de métodos de critérios de parada da literatura por meio de uma análise extensiva, bem como foram apreciados tais métodos no domínio de predição da função de proteínas pelas abordagens MLNP e NMLNP. Além disso, tais bases de dados apresentam os seguintes problemas hierárquicos: taxonomias em formato de árvore, desbalanceadas, quantidade pequena de instâncias de treinamento para determinadas classes, rotulação em caminho único e possuem características de rotulações parciais. Este último foi o foco principal deste trabalho.

A partir da análise extensiva dos métodos de critério de parada do estado da arte, sendo estes com abordagens diversas: Top-Down, Bottom-up e Global, identificou-se que o método NMLNP Top-Down (CECI; MALERBA, 2007) apresentou melhores resultados em comparação com os principais métodos de critérios de parada da literatura. Dessa forma, adotamos este método como baseline para comparação com os métodos propostos, bem como serviu de motivação para desenvolvimento dos métodos nesta tese, utilizando abordagem de predição top-down. O segundo método da literatura utilizado para comparação foi o Ganho de Informação por usar implicitamente a entropia.

Baseado neste contexto, foram propostos diversos métodos de critérios de parada para predição não obrigatória (opcional) em nós-folha para aplicação em problemas de Classificação Hierárquica Local.

Os quatro primeiros métodos propostos denominados LCPNwFOE (Local Classifier per Parent Node with Fixed Original Entropy), LCPNwVOE (Local Classifier per Parent Node with Variable Original Entropy), LCPNwFME (Local Classifier per Parent Node with Fixed Modified Entropy) e LCPNwVME (Local Classifier per Parent Node with Variable Modified Entropy), utilizam a Entropia como critério de parada. Nestes dois últimos citados, foram aplicados os mesmos métodos de classificação hierárquica e avaliação dos anteriores. Porém foi proposto um ajuste no cálculo da entropia, denotando-se esta nova fórmula baseada na entropia como uma outra contribuição importante desta tese.

Para taxonomias balanceadas, por exemplo, de árvores binárias ou que a quantidade de classes filhas para todos os nós-pai são iguais, pode-se utilizar os métodos de critérios de parada com limiar de entropia fixa (LCPNwFOE e LCPNwFME). Porém, em casos que a quantidade de classes filhas seja diferente para determinados nós-pai da hierarquia, os métodos de critérios de parada com limiar de entropia variável (LCPNwVOE e LCPNwVME) podem apresentar melhores resultados, pelo fato deste método ser mais flexível e verificar a entropia máxima de cada nó-pai em relação à quantidade de suas respectivas classes-filhas. Além disso, observa-se nos resultados que a nova fórmula baseada

na entropia proposta nesta tese, por meio do método LCPNwVME, apresentou melhores resultados, principalmente pela métrica de avaliação hP , em comparação com os principais métodos da literatura.

O quinto método LCPNwLC (Local Classifier per Parent Node with Level Classifier) define o critério de parada por meio de um Classificador de Nível, no qual identifica-se em que nível da árvore deve ser realizada a predição e, por consequência, a parada. Caso a confiança deste Classificador de Nível não seja suficiente, desconsidera-se este e um outro método critério de parada é aplicado.

Mediante a utilização de um Classificador de Nível, com a finalidade de avaliar inicialmente o contexto do nível em que a instância deverá ser rotulada, bem como da utilização do método proposto LCPNwVME, caso a confiança do classificador de nível não seja suficiente para realizar a rotulação, este método apresentou resultados satisfatórios.

Por fim, o método denominado LCPNwINT (Local Classifier per Parent Node with Intersection) é proposto a interseção de dois ou mais diferentes métodos de critérios de parada de classificação hierárquica.

Utilizar a interseção entre dois ou mais métodos de avaliação hierárquicos pode prover bons resultados, principalmente com relação à métrica de avaliação de precisão hierárquica. Ou seja, por meio dos resultados apresentados pela Interseção dos métodos de critérios de parada "NMLPN Top-Down" e a LCPNwVME, melhorou-se ainda mais os resultados da métrica hP em comparação com os outros métodos avaliados. Este método pode ser útil para domínios que necessitam de menos erros de especialização na classificação hierárquica.

Pode-se concluir que, percorrer a hierarquia de forma Top-Down e usar a Entropia como critério de parada NMLNP, pode apresentar bons resultados, visto que, além de analisar a probabilidade de um nó em questão para tomar uma decisão, avalia-se também a qualidade ou a incerteza (entropia) referente aos seus respectivos nós-irmãos de um determinado classificador nó-pai.

7.1 Trabalhos futuros

Os métodos propostos neste trabalho podem ser avaliados em outros domínios, com a finalidade de avaliar tais métodos com outras características de dados para a solução de um determinado problema hierárquico.

Como esta tese trabalhou no seguinte cenário da classificação hierárquica (LCPN, Tree, SPL e NMLNP), pode-se estender a abordagem para a solução de outros problemas hierárquicos. Por exemplo: a predição de vários caminhos da hierarquia (MPL), bem como para a classificação hierárquica com a taxonomia de DAG. Além disso, pode ser avaliado

tais métodos propostos utilizando outras abordagens de soluções hierárquicas.

Pode-se ainda ser realizada uma análise mais completa relacionada ao tipo de estratégia da solução avaliada em diferentes cenários de problemas hierárquicos por meio da comparação e avaliação dos impactos das estratégias para percorrer caminhos da hierarquia (Top-Down/Global), bem como os principais métodos de critérios de parada em todos os cenários de um problema hierárquico, sejam estes: bases de dados com taxonomias balanceadas e desbalanceadas; problemas relacionados a profundidade de rotulação, MLNP e NMLNP; taxonomias de árvore ou DAG; e rotulação em único caminho (Single Path) e múltiplos caminhos (Multi Path).

Referências

- ACKERMANN, Nils. *Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences*. 2019. <<https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf>>. Accessed: 2019-03-10. Citado na página 32.
- ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: IEEE. *2017 International Conference on Engineering and Technology (ICET)*. [S.l.], 2017. p. 1–6. Citado na página 31.
- ALVES, Roberto Teixeira; DELGADO, MR; FREITAS, Alex Alves. Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In: IEEE. *International Conference on Fuzzy Systems*. [S.l.], 2010. p. 1–8. Citado na página 42.
- AREL, Itamar; ROSE, Derek C; KARNOWSKI, Thomas P et al. Deep machine learning—a new frontier in artificial intelligence research. *IEEE computational intelligence magazine*, The University of Tennessee, v. 5, n. 4, p. 13–18, 2010. Citado na página 29.
- BABBAR, Rohit; PARTALAS, Ioannis; GAUSSIER, Eric; AMINI, Massih R. On flat versus hierarchical classification in large-scale taxonomies. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 1824–1832. Citado na página 35.
- BARUTCUOGLU, Zafer; DECORO, Christopher. Hierarchical shape classification using bayesian aggregation. In: IEEE. *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. [S.l.], 2006. p. 44–44. Citado 2 vezes nas páginas 19 e 37.
- BENGIO, Yoshua et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, Now Publishers, Inc., v. 2, n. 1, p. 1–127, 2009. Citado na página 30.
- BINDER, Alexander; KAWANABE, Motoaki; BREFELD, Ulf. Efficient classification of images with taxonomies. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2009. p. 351–362. Citado na página 19.
- BITTENCOURT, Valnaide Gomes. *Aplicação de técnicas de aprendizado de máquina no reconhecimento de classes estruturais de proteínas*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2005. Citado na página 60.
- BLOCKEEL, Hendrik; BRUYNOOGHE, Maurice; DŽEROSKI, Sašo; RAMON, Jan; STRUYF, Jan. Hierarchical multi-classification. In: *Workshop Notes of the KDD'02 Workshop on Multi-Relational Data Mining*. [S.l.: s.n.], 2002. p. 21–35. Citado 3 vezes nas páginas 20, 44 e 45.
- BORGES, Helyane Bronoski. *Classificador hierárquico multirrótulo usando uma rede neural competitiva*. Tese (Doutorado) — Pontifical Catholic University of Paraná, Curitiba, Brazil, 2012. Citado na página 25.
- BORGES, Helyane Bronoski; NIEVOLA, Julio Cesar. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In: IEEE. *The 2012*

International Joint Conference on Neural Networks (IJCNN). [S.l.], 2012. p. 1–8. Citado 3 vezes nas páginas 19, 20 e 42.

CARVALHO, Rafael V; BRUNORO, Gustavo; PAPPA, Gisele L. Hega: A genetic algorithm for hierarchical classification. In: IEEE. *2011 IEEE Congress of Evolutionary Computation (CEC)*. [S.l.], 2011. p. 933–940. Citado na página 21.

CECI, Michelangelo; MALERBA, Donato. Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, Springer, v. 28, n. 1, p. 37–78, 2007. Citado 14 vezes nas páginas 20, 21, 54, 64, 78, 87, 92, 97, 102, 113, 120, 124, 127 e 135.

CERRI, Ricardo; BARROS, Rodrigo C; CARVALHO, André CPLF De. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, Elsevier, v. 80, n. 1, p. 39–56, 2014. Citado 2 vezes nas páginas 19 e 40.

CERRI, Ricardo; PAPPA, Gisele L; CARVALHO, André Carlos PLF; FREITAS, Alex A. An extensive evaluation of decision tree-based hierarchical multilabel classification methods and performance measures. *Computational Intelligence*, Wiley Online Library, v. 31, n. 1, p. 1–46, 2015. Citado na página 43.

COSTA, Eduardo; LORENA, Ana; CARVALHO, ACPLF; FREITAS, Alex. A review of performance evaluation measures for hierarchical classifiers. In: *Evaluation Methods for machine Learning II: papers from the AAAI-2007 Workshop*. [S.l.: s.n.], 2007. p. 1–6. Citado na página 25.

COVER, Thomas M; THOMAS, Joy A. *Elements of information theory*. [S.l.]: John Wiley & Sons, 2012. Citado na página 49.

CUNHA, Anderson; VELHO, Luiz. *Métodos Probabilísticos para Reconhecimento de Voz*. [S.l.], 2003. Citado na página 48.

DAVIES, Matthew N; SECKER, Andrew; FREITAS, Alex A; CLARK, Edward; TIMMIS, Jon; FLOWER, Darren R. Optimizing amino acid groupings for gpcr classification. *Bioinformatics*, Oxford University Press, v. 24, n. 18, p. 1980–1986, 2008. Citado 3 vezes nas páginas 8, 62 e 63.

DECORO, Christopher; BARUTCUOGLU, Zafer; FIEBRINK, Rebecca. Bayesian aggregation for hierarchical genre classification. In: VIENNA. *ISMIR*. [S.l.], 2007. p. 77–80. Citado na página 19.

DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006. Citado na página 66.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado na página 32.

DUMAIS, Susan; CHEN, Hao. Hierarchical classification of web content. In: ACM. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. [S.l.], 2000. p. 256–263. Citado 3 vezes nas páginas 19, 37 e 52.

- EPSTEIN, Isaac. *Teoria da informação*. [S.l.]: Editora Atica, 1986. v. 35. Citado na página 92.
- FABRIS, Fabio; FREITAS, Alex A. An efficient algorithm for hierarchical classification of protein and gene functions. In: IEEE. *2014 25th International Workshop on Database and Expert Systems Applications*. [S.l.], 2014. p. 64–68. Citado 3 vezes nas páginas 20, 37 e 53.
- FAGNI, Tiziano; SEBASTIANI, Fabrizio. On the selection of negative examples for hierarchical text categorization. In: CITESEER. *Proceedings of the 3rd Language & Technology Conference (LTC'07)*. [S.l.], 2007. p. 24–28. Citado 2 vezes nas páginas 37 e 64.
- FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996. Citado na página 24.
- FERRANDIN, Mauri; ROMAO, Luiz Melo. Hierarchical classification with jumping emerging patterns. *IEEE Latin America Transactions*, IEEE, v. 14, n. 9, p. 4143–4149, 2016. Citado na página 20.
- FILMORE, David. It's a gpcr world. *Modern drug discovery*, American Chemical Society, v. 7, p. 24–28, 2004. Citado na página 61.
- FREITAS, Alex; CARVALHO, André. A tutorial on hierarchical classification with applications in bioinformatics. In: *Research and trends in data mining technologies and applications*. [S.l.]: IGI Global, 2007. p. 175–208. Citado 3 vezes nas páginas 19, 32 e 39.
- FREITAS, Alex A. *Data mining and knowledge discovery with evolutionary algorithms*. [S.l.]: Springer Science & Business Media, 2013. Citado na página 24.
- FRIEDMAN, Milton. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937. Citado na página 66.
- GRUBER, Thomas R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Elsevier, v. 5, n. 2, p. 199–220, 1993. Citado na página 60.
- HAYKIN, Simon. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2007. Citado 4 vezes nas páginas 25, 26, 27 e 28.
- HERNÁNDEZ, Julio; SUCAR, L Enrique; MORALES, Eduardo F. Multidimensional hierarchical classification. *Expert Systems with Applications*, Elsevier, v. 41, n. 17, p. 7671–7677, 2014. Citado 9 vezes nas páginas 20, 59, 75, 78, 87, 92, 97, 102 e 124.
- HERNANDEZ, Julio Noe; SUCAR, Luis Enrique; MORALES, Eduardo F. A hybrid global-local approach for hierarchical classification. In: *The Twenty-Sixth International FLAIRS Conference*. [S.l.: s.n.], 2013. Citado 4 vezes nas páginas 20, 36, 51 e 52.
- HIGGS, Paul G; ATTWOOD, Teresa K. *Bioinformatics and molecular evolution*. [S.l.]: John Wiley & Sons, 2013. Citado na página 61.
- HOCHULI, Andre G; OLIVEIRA, Luiz S; JR, AS Britto; SABOURIN, Robert. Handwritten digit segmentation: Is it still necessary? *Pattern Recognition*, Elsevier, v. 78, p. 1–11, 2018. Citado 2 vezes nas páginas 110 e 111.

HOCHULI, André Gustavo et al. Abordagens livres de segmentação para reconhecimento automático de cadeias numéricas manuscritas utilizando aprendizado profundo. 2019. Citado na página 30.

HOLDEN, Nicholas; FREITAS, Alex A. Improving the performance of hierarchical classification with swarm intelligence. In: SPRINGER. *European conference on Evolutionary computation, machine learning and data mining in bioinformatics*. [S.l.], 2008. p. 48–60. Citado na página 39.

HOLDEN, Nicholas; FREITAS, Alex A. Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation. *Soft Computing*, Springer, v. 13, n. 3, p. 259–272, 2009. Citado 2 vezes nas páginas 45 e 62.

HOPFIELD, John J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982. Citado na página 25.

IPEIROTIS, Panagiotis G; GRAVANO, Luis; SAHAMI, Mehran. Probe, count, and classify: categorizing hidden web databases. In: ACM. *ACM SIGMOD Record*. [S.l.], 2001. v. 30, n. 2, p. 67–78. Citado na página 45.

KIRITCHENKO, Svetlana; MATWIN, Stan; NOCK, Richard; FAMILI, A Fazel. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In: SPRINGER. *Conference of the Canadian Society for Computational Studies of Intelligence*. [S.l.], 2006. p. 395–406. Citado 4 vezes nas páginas 42, 43, 46 e 47.

KOLLER, Daphne; SAHAMI, Mehran. *Hierarchically classifying documents using very few words*. [S.l.], 1997. Citado na página 37.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado 3 vezes nas páginas 29, 30 e 32.

LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 29.

LESK, Arthur M; SANGAR, Vineet; PARKINSON, Helen; WHISSTOCK, James C. Bioinformatics of protein function. In: *Structural Proteomics And Its Impact On The Life Sciences*. [S.l.]: World Scientific, 2008. p. 79–119. Citado na página 20.

LORD, Phillip W.; STEVENS, Robert D.; BRASS, Andy; GOBLE, Carole A. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, Oxford University Press, v. 19, n. 10, p. 1275–1283, 2003. Citado na página 45.

MERSCHMANN, Luiz Henrique de Campos; FREITAS, Alex Alves. An extended local hierarchical classifier for prediction of protein and gene functions. In: SPRINGER. *International Conference on Data Warehousing and Knowledge Discovery*. [S.l.], 2013. p. 159–171. Citado 2 vezes nas páginas 37 e 53.

- MOSKOVITCH, Robert; COHEN-KASHI, Shiva; DROR, Uzi; LEVY, Iftah; MAIMON, Amit; SHAHAR, Yuval. Multiple hierarchical classification of free-text clinical guidelines. *Artificial Intelligence in Medicine*, Elsevier, v. 37, n. 3, p. 177–190, 2006. Citado 2 vezes nas páginas 54 e 55.
- NAIK, Azad; RANGWALA, Huzefa. Inconsistent node flattening for improving top-down hierarchical classification. In: IEEE. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. [S.l.], 2016. p. 379–388. Citado 2 vezes nas páginas 32 e 35.
- NAKANO, Felipe Kenji; PINTO, Walter José; PAPPA, Gisele Lobo; CERRI, Ricardo. Top-down strategies for hierarchical classification of transposable elements with neural networks. In: IEEE. *2017 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2017. p. 2539–2546. Citado 3 vezes nas páginas 56, 57 e 58.
- OLSON, David L; DELEN, Dursun. *Advanced data mining techniques*. [S.l.]: Springer Science & Business Media, 2008. Citado na página 42.
- PAES, Bruno C; PLASTINO, Alexandre; FREITAS, Alex A. Improving local per level hierarchical classification. *Journal of Information and Data Management*, v. 3, n. 3, p. 394, 2012. Citado na página 40.
- ROMAO, Luiz Melo; NIEVOLA, Julio César. Hierarchical classification of gene ontology with learning classifier systems. In: SPRINGER. *Ibero-American Conference on Artificial Intelligence*. [S.l.], 2012. p. 120–129. Citado na página 20.
- RUEPP, Andreas; ZOLLNER, Alfred; MAIER, Dieter; ALBERMANN, Kaj; HANI, Jean; MOKREJS, Martin; TETKO, Igor; GÜLDENER, Ulrich; MANNHAUPT, Gertrud; MÜNSTERKÖTTER, Martin et al. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic acids research*, Oxford University Press, v. 32, n. 18, p. 5539–5545, 2004. Citado 3 vezes nas páginas 32, 60 e 61.
- SAM, Kin Meng; CHATWIN, Chris. Ontology-based sentiment analysis model of customer reviews for electronic products. In: *Encyclopedia of Information Science and Technology, Third Edition*. [S.l.]: IGI Global, 2015. p. 892–904. Citado na página 60.
- SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015. Citado na página 27.
- SECKER, Andrew D; DAVIES, Matthew N; FREITAS, Alex A; TIMMIS, Jon; MENDAO, Miguel; FLOWER, Darren R. An experimental comparison of classification algorithms for hierarchical prediction of protein function. *Expert Update (Magazine of the British Computer Society's Specialist Group on AI)*, v. 9, n. 3, p. 17–22, 2007. Citado 2 vezes nas páginas 19 e 38.
- SERRE, Thomas; KREIMAN, Gabriel; KOUH, Minjoon; CADIEU, Charles; KNOBLICH, Ulf; POGGIO, Tomaso. A quantitative theory of immediate visual recognition. *Progress in brain research*, Elsevier, v. 165, p. 33–56, 2007. Citado na página 29.
- SHANNON, CE; WEAVER, Warren. The mathematical theory of communication, (first published in 1949). *Urbana University of Illinois Press*, 1963. Citado na página 93.

- SHANNON, Claude Elwood. A mathematical theory of communication. *Bell system technical journal*, Wiley Online Library, v. 27, n. 3, p. 379–423, 1948. Citado 4 vezes nas páginas 48, 49, 76 e 84.
- SILLA, Carlos N; FREITAS, Alex A. A global-model naive bayes approach to the hierarchical prediction of protein functions. In: IEEE. *2009 Ninth IEEE International Conference on Data Mining*. [S.l.], 2009. p. 992–997. Citado 3 vezes nas páginas 20, 41 e 56.
- SILLA, Carlos N; FREITAS, Alex A. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In: IEEE. *2009 IEEE International Conference on Systems, Man and Cybernetics*. [S.l.], 2009. p. 3499–3504. Citado 2 vezes nas páginas 19 e 39.
- SILLA, Carlos N; FREITAS, Alex A. Selecting different protein representations and classification algorithms in hierarchical protein function prediction. *Intelligent Data Analysis*, IOS Press, v. 15, n. 6, p. 979–999, 2011. Citado 2 vezes nas páginas 62 e 63.
- SILLA, Carlos N; FREITAS, Alex A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, Springer, v. 22, n. 1-2, p. 31–72, 2011. Citado 14 vezes nas páginas 19, 20, 33, 34, 35, 36, 37, 38, 39, 40, 41, 48, 60 e 66.
- SUN, Aixin; LIM, Ee-Peng. Hierarchical text classification and evaluation. In: IEEE. *Proceedings 2001 IEEE International Conference on Data Mining*. [S.l.], 2001. p. 521–528. Citado 5 vezes nas páginas 19, 39, 41, 42 e 43.
- SUN, Aixin; LIM, E-P; NG, W-K; SRIVASTAVA, Jaideep. Blocking reduction strategies in hierarchical text classification. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 16, n. 10, p. 1305–1308, 2004. Citado 2 vezes nas páginas 20 e 55.
- SUN, Zhengya; ZHAO, Yangyang; CAO, Dong; HAO, Hongwei. Hierarchical multilabel classification with optimal path prediction. *Neural Processing Letters*, Springer, v. 45, n. 1, p. 263–277, 2017. Citado na página 56.
- TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. *Introdução ao datamining: mineração de dados*. [S.l.]: Ciência Moderna, 2009. Citado na página 25.
- TONG, Joo Chuan; TAMMI, Martti T. Prediction of protein allergenicity using local description of amino acid sequence. *Frontiers in Bioscience*, v. 13, n. 16, p. 6072–6078, 2008. Citado 3 vezes nas páginas 8, 62 e 63.
- VENS, Celine; STRUYF, Jan; SCHIETGAT, Leander; DŽEROSKI, Sašo; BLOCKEEL, Hendrik. Decision trees for hierarchical multi-label classification. *Machine learning*, Springer, v. 73, n. 2, p. 185, 2008. Citado na página 41.
- WANG, Xiao-Lin; LU, Bao-Liang. Flatten hierarchies for large-scale hierarchical text categorization. In: IEEE. *2010 Fifth International Conference on Digital Information Management (ICDIM)*. [S.l.], 2010. p. 139–144. Citado na página 35.
- ZHU, Xiaojin Jerry. *Semi-supervised learning literature survey*. [S.l.], 2005. Citado na página 24.