



FERRAMENTA DE APOIO AO DIAGNÓSTICO DE ENFERMAGEM EM UTIs

Trabalho de Conclusão de Curso

Engenharia da Computação

Raphael Farias Carneiro

Orientador: Prof. Fernando Buarque de Lima Neto



UNIVERSIDADE
DE PERNAMBUCO

**Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

RAPHAEL FARIAS CARNEIRO

**FERRAMENTA DE APOIO AO
DIAGNÓSTICO DE ENFERMAGEM EM
UTIs**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho de 2014.

De acordo

Recife

____/____/____



Fernando Buarque de Lima Neto

*À Deus, que é a minha fortaleza
À minha noiva
Aos meus pais.*

Agradecimentos

Primeiramente, agradeço à **Deus** por toda a força que Ele tem me dado para sempre querer continuar e nunca desistir e por tudo de bom que tem acontecido em minha vida: passar no vestibular para Engenharia de Computação (e hoje estar aqui, concluindo o curso!), emprego desde o início da graduação... Enfim... Por tudo!

Agradeço à pessoa mais especial desse mundo, ao amor da minha vida, à pessoa mais maravilhosa, mais perfeita, mais linda que já conheci, à minha amada, à minha noiva, à minha futura (ainda bem que falta tão pouco!! *-*) esposa, **Emanuelle Varão Vasconcelos**, por tudo de bom que ela tem feito por/pra mim, por existir, por ser essa pessoa tão dedicada ao nosso amor e sempre preocupada com meus estudos (sempre dizia: “vá estudar!!!” e ainda diz! :P). Obrigado por todo companheirismo, por todo o carinho, por toda a força, por todo o apoio, por todos os momentos maravilhosos que eu tenho ao seu lado... OBRIGADO POR TUDO! EU TE AMO MUITO, MEU AMOR!!!

Agradeço à minha amiga **Sabrina Andrade**, por toda ajuda que ela tem me dado neste trabalho: corrigindo, sugerindo... e não só neste trabalho, mas como em todas as disciplinas (complicadas ou não) durante a graduação :P Valeu Sabs! ☺

Ao pessoal da **FAST Aplicativos e Soluções Tecnológicas**, por terem me dado a oportunidade de trabalhar em uma empresa maravilhosa, com tanta gente legal (e viciadas em café hehehe), por estar em contato com o cliente e colocar em prática todo o conhecimento adquirido durante a graduação. Além das vezes em que precisei faltar para estudar, concluir este trabalho, e ainda ter tido a possibilidade de usar a sala de reunião para estudar! Obrigado por tudo, gente! ☺

Aos amigos da graduação **Itúrbide**, **Thiago Brito** (Thiago Taiada), **Thiago Ferreira** (Thiago Barca), **Vinícius Souza**, **Clodomir** (CJ!), **Dierson**, **Jefferson Diniz**, **Victor Hugo** (Papa), por todos os momentos de descontração proporcionados durante a graduação (paint ball e o bom e velho dominó kkkkkkkkkkk). Obrigado pessoal!

Ao professor **Meuser Valença**, pela pequena e rápida aula sobre Redes Neurais sem a qual eu não conseguiria realizar os meus experimentos. Obrigado professor!

À **Bruna Galle, Lara Dantas, Paulo Roger e Denis Martins** (Denizinho), sem a grande ajuda deles eu não teria concluído este trabalho e iria continuar achando que estava tudo errado hehe. Obrigado gente!

Aos **meus pais**, por toda força, dedicação, cuidado, sem os quais eu não estaria aqui! Por sempre investir em meus estudos e nunca me deixar parar! Obrigado paiinho **Reginaldo** e mainha **Severina**! Amo MUITO vocês!

À **Paulo Epifânio**, que mesmo com 2 empregos, 2 meninos pequenos em casa, ainda conseguia tempo para termos algumas pequenas reuniões nas sextas feiras no almoço, além de todas as dicas dadas durante o trabalho e preocupação com a conclusão. Obrigado Paulo!

Ao meu **despertador**, apesar das tentativas frustradas de tentar me acordar de madrugada para continuar a escrita desta monografia e para estudar o conteúdo das outras disciplinas :P.

À **Angélica**, pela ajuda neste trabalho na parte de enfermagem. Obrigado, Angélica! ☺

Ao meu orientador **Fernando Buarque**, pela oportunidade do contato com a área de Inteligência Artificial (que era o que eu queria desde o início da graduação) e ainda poder aplicar isso na área de saúde (era o que eu mais queria!!!). Obrigado professor!!

Ao professor **Joabe Bezerra** por ter feito parte da banca avaliadora e por todo o conhecimento compartilhado durante as disciplinas da graduação! Obrigado, professor!

A toda a minha família, por tudo!

Aos amigos **Arnold e Egryson**, desde a 5ª série proporcionando momentos de descontração! Obrigado meninos!

A todos que contribuíram direta ou indiretamente com este trabalho.

Muito obrigado!

Resumo

Unidades de terapia intensiva (UTIs) são ambientes críticos em que os pacientes são submetidos a 24h de cuidados intensivos de um profissional chamado intensivista. O fato de estes profissionais estarem envolvidos com muitas tarefas burocráticas gera cansaço e estresse, o que pode prejudicar a qualidade do atendimento aos pacientes internados nas UTIs. Diante deste cenário tão complexo, sistemas de apoio à decisão tornam-se essenciais para tentar reduzir a quantidade de erros clínicos/humanos e auxiliar – mas não substituir – médicos e enfermeiros no seu processo de decisão. Um Sistema de Apoio à Decisão Inteligente, utilizando Redes Neurais Artificiais do tipo *Multilayer perceptron*, foi proposto para auxiliar no processo de decisão de diagnósticos de enfermagem em pacientes de UTI.

Palavras-chave: Unidade de Terapia Intensiva, diagnósticos de enfermagem, Redes Neurais Artificiais, *Multilayer perceptron*, Sistema de apoio à decisão inteligente.

Abstract

Intensive care units (ICUs) are critical environments in which patients are subjected to 24h by 24h care of a professional called intensivist. The fact that these professionals are involved with many bureaucratic tasks generates fatigue and stress, which may undermine the quality of care for patients admitted to ICUs. Given this complex scenario, decision support systems become essential to reduce the amount of clinical / human errors and help – but not to replace - doctors and nurses in their decision process. An Intelligent Decision Support System, using Multilayer Perceptron Artificial Neural Networks, has been proposed to assist in the decision process of nursing diagnoses in ICU patients.

Key-words: Intensive Care Unit, nursing diagnoses, Artificial Neural Networks, Multilayer perceptron, Intelligent Decision Support System.

Sumário

Índice de Figuras	xi
Índice de Tabelas	xii
Tabela de Símbolos e Siglas	xiii
Capítulo 1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.2.1 Objetivo principal	2
1.2.2 Objetivos secundários	3
1.3 Estrutura do Documento	4
Capítulo 2 Revisão Bibliográfica	5
2.1 Unidades de Terapia Intensiva	5
2.1.1 História	6
2.1.2 Sinais Vitais	7
2.2 Sistema de Apoio à Decisão	11
2.3 Redes Neurais Artificiais	13
2.3.1 Neurônio Biológico	15
2.3.2 Neurônio Artificial	16
2.3.3 Multilayer Perceptron	18
2.3.4 Algoritmos de treinamento	20
2.3.5 Validação Cruzada	23
Capítulo 3 Ferramenta Proposta	24
3.1 Modelo proposto	24
3.2 Ferramenta Proposta	26
Capítulo 4 Experimentos e Resultados	30
4.1 Metodologia	30
4.1.1 Análises estatísticas	32
4.2 Experimentos	34
4.2.1 Função de ativação TANH e método de treinamento <i>Resilient Propagation</i>	35

4.2.2	Função de ativação sigmoide logística e método de treinamento <i>Resilient propagation</i>	35
4.2.3	Função de ativação TANH e método de treinamento <i>Back propagation</i>	36
4.2.4	Função de ativação sigmoide logística e método de treinamento <i>Back propagation</i>	36
4.3	Resultados	36
4.3.1	Taxa de acerto	36
4.3.2	Análises Estatísticas	38
4.3.3	Comparação com outros resultados disponíveis na literatura	40
	Capítulo 5 Conclusões e Trabalhos Futuros	42
5.1	Conclusões	42
5.2	Trabalhos Futuros	43
	Bibliografia	44
	Apêndice A Regras para a definição dos diagnósticos	49
	Apêndice B Exemplo de arquivo CSV	51
	Apêndice C Detalhes dos experimentos	52
	Apêndice D Resultado dos testes estatísticos	53
	Apêndice E <i>Script</i> utilizado para a execução das análises estatísticas no software R	57

Índice de Figuras

Figura 1.	Monitor multiparamétrico	7
Figura 2.	Esfigmomanômetro	9
Figura 3.	Aferidor eletrônico de pressão.....	9
Figura 4.	Transdutor de pressão	10
Figura 5.	Oxímetro de pulso	11
Figura 6.	Aprendizado supervisionado (ou com professor)	14
Figura 7.	Aprendizado não-supervisionado	14
Figura 8.	Aprendizado por reforço	15
Figura 9.	Estrutura do neurônio	15
Figura 10.	Sinapses químicas	16
Figura 11.	Neurônio Artificial de McCulloch e Pits	17
Figura 12.	Exemplo de um <i>Perceptron</i>	18
Figura 13.	Exemplo de uma MLP	18
Figura 14.	Exemplo de problema linearmente separável.....	19
Figura 15.	Exemplo de problema não separável linearmente.....	19
Figura 16.	Arquitetura da ferramenta proposta	26
Figura 17.	Diagrama de classes da rede neural <i>Multilayer perceptron</i>	28
Figura 18.	Exemplo de arquivo CSV contendo informações sobre os sinais vitais do paciente.....	51
Figura 19.	Resultados do teste de Shapiro-Wilk.....	53
Figura 20.	Resultados do Teste F - Experimentos 1 à 7.....	54
Figura 21.	Resultados do teste F - Experimentos 8 à 12.....	55
Figura 22.	Resultados do teste T-Student - Experimentos 1 à 7	56
Figura 23.	Resultados do teste T-Student - Experimento 8 à 12	56

Índice de Tabelas

Tabela 1.	Classes e pacotes da ferramenta proposta	27
Tabela 2.	Hipóteses nula e alternativa para o teste de Shapiro-Wilk	32
Tabela 3.	Hipóteses nula e alternativa para o teste F	33
Tabela 4.	Hipóteses nula e alternativa para o teste T-Student.....	33
Tabela 5.	Hipóteses nula e alternativa para o teste da Soma dos postos de Wilcoxon	34
Tabela 6.	Taxas de acerto – Experimentos 1 a 12	37
Tabela 7.	Melhor configuração da MLP	37
Tabela 8.	Teste Shapiro-Wilk – Hipóteses nula e alternativa – experimentos 1 a 12	38
Tabela 9.	Teste F – Hipóteses nula e alternativa – experimentos 1 a 12.....	39
Tabela 10.	T-Student – Hipóteses nula e alternativa – experimentos 1 a 12	39
Tabela 11.	Comparação com resultados encontrados na literatura	41
Tabela 12.	Regras para definição dos diagnósticos	49
Tabela 13.	Detalhamento dos experimentos	52

Tabela de Símbolos e Siglas

ADALINE – *ADaptive LInear NEuron*

CREMESP – Conselho Regional de Medicina do Estado de São Paulo

CSV – *Comma Separated Values*

MIMIC II – *Multiparameter Intelligent Monitoring in Intensive Care II*

MLP – *Multilayer Perceptron*

PAD – Pressão Arterial Diastólica

PAM – Pressão Arterial Média

PAS – Pressão Arterial Sistólica

RNA – Rede Neural Artificial

RPROP – *Resilient Propagation*

SAD – Sistema de Apoio à Decisão

TANH – Tangente Hiperbólica

TLU – *Threshold Logic Unit*

UML – *Unified Modeling Language*

UTI – Unidade de Terapia Intensiva

Capítulo 1

Introdução

Este capítulo apresenta os fatores que motivaram a proposta deste trabalho, que culmina com um protótipo de uma ferramenta de apoio ao diagnóstico de enfermagem em pacientes em tratamento intensivo, e finalmente a organização deste documento.

1.1 Motivação

De acordo com Donna L. Hudson et. al [16], há um século, médicos e enfermeiros não possuíam muitas informações para diagnosticar de forma precisa. Com o passar dos anos e com o avanço da tecnologia, surgiram novos equipamentos, consequentemente revelando mais detalhes sobre a saúde dos pacientes, os quais impactavam no diagnóstico e tratamento das doenças.

A Unidade de Terapia Intensiva (UTI) foi trazida para o Brasil na década de 70 e é um ambiente diferenciado onde se preza a manutenção da vida [42]. É, também, o local onde os pacientes que precisam de um acompanhamento mais intensivo estão ligados a vários aparelhos sofisticados [9], que permanecem ininterruptamente monitorando seus sinais vitais, de forma a manter a equipe responsável pelo leito ciente de seu estado de saúde, permitindo uma intervenção clínica no momento apropriado.

Nas UTIs, diversos aparelhos estão ligados ao paciente, monitorando seus sinais vitais, o que torna complexo o processo de decisão clínica profissional da saúde, considerando os diversos parâmetros que são analisados simultaneamente. Assim, aumentam-se as chances de um diagnóstico incerto e, consequentemente, um tratamento e/ou assistência incorretos. Além disso, o nível de segurança de um paciente está diretamente relacionado à ocorrência de erros médicos [20][21]. Segundo o 1984 Harvard Medical Practice Study [2], danos iatrogênicos [17] ocorrem em 3,7% das hospitalizações, dos quais 27,6% são causados por negligência médica e 69% por erro humano.

Diante deste cenário tão complexo, um sistema de apoio à decisão (SAD) clínica se faz necessário para auxiliar – não substituir – os médicos e os enfermeiros no processo de decisão [16] e minimizar a ocorrência de erros, visto que os enfermeiros, principalmente, estão cada vez mais sobrecarregados com tarefas burocráticas [22], gerando cansaço, estresse e prejudicando a prestação de uma assistência de qualidade [12]. Os SADs são ferramentas computacionais que já aparecem em alguns cenários da enfermagem [19], mas nas UTIs, onde os pacientes necessitam de cuidados cada vez mais qualificados, o processo de tomada de decisão se torna mais eficiente, pois estas ferramentas permitem uma visão geral do estado de saúde imediato do paciente (i.e. antes mesmo da chegada de um médico) [14].

Dentro do contexto da interpretação dos sinais, a Semiótica, ciência dos signos [37] e de todas as linguagens [38], surge como uma grande possibilidade de ser incorporada a ferramentas que possam individualizar diagnósticos a partir de sinais oriundos de equipamentos conectados a pacientes de UTIs.

Diante disso, faz-se necessário a elaboração de uma ferramenta de apoio à decisão para diagnósticos de enfermagem, especificamente, capaz de se adaptar aos pacientes dos leitos das UTIs, individualizando os diagnósticos, tendo como referência os sinais vitais do próprio paciente obtidos a partir de seu histórico no hospital. Caso este histórico ainda não exista, serão considerados os valores padrão já definidos na literatura [3].

Posteriormente, uma análise semiótica dos dados obtidos poderá oportunamente ser aplicada à esta ferramenta para a implantação de um novo protocolo de enfermagem.

1.2 Objetivos

1.2.1 Objetivo principal

Este trabalho tem como objetivo principal o desenvolvimento de uma ferramenta inteligente para o apoio ao diagnóstico de enfermagem em pacientes de UTIs. A ferramenta deve analisar os sinais vitais oriundos dos monitores multiparamétricos, encontrando alguma relação entre estes sinais a ser integrado ao histórico do paciente para a elaboração de pré-diagnósticos, que serão posteriormente

analisados pelos enfermeiros a fim de se obter um diagnóstico final para aquele paciente.

Além disso, a ferramenta também deve:

- Adaptar-se ao paciente que está sendo monitorado, gerando pré-diagnósticos específicos (i.e. individualizados), tendo como valores de referência as informações vitais do próprio paciente, e não um valor padrão já estabelecido [3].
- Indicar a prioridade de atendimento dos pacientes em leitos de UTIs de acordo com a gravidade do estado de saúde do paciente. Esta gravidade será determinada pelo resultado do monitoramento de seus próprios sinais vitais, e será indicada com um score que, quanto maior quando comparado ao score dos demais pacientes da unidade, maior a gravidade no estado de saúde e maior a prioridade do atendimento prestado pelo enfermeiro.
- Auxiliar, indiretamente, a gerenciar melhor o tempo do(s) enfermeiro(s) que esteja(m) responsável(eis) pelo leito, para que eles possam dar a devida atenção aos pacientes, evitando deslocamentos desnecessários e o conseqüente cansaço físico.

1.2.2 Objetivos secundários

Para atingir o objetivo principal deste trabalho, os seguintes objetivos secundários foram determinados:

- Realizar a filtragem da base de dados MIMIC II, a fim de se obter apenas os sinais vitais que serão utilizados no processo de aprendizagem da ferramenta. Esta base de dados possui informações reais de dezenas de milhares de pacientes de UTIs que ficaram internados no Beth Israel Deacones Medical Center [22] e foi utilizada neste trabalho pois, na literatura, outros trabalhos também já a utilizaram [23] [35].
- Com os sinais vitais filtrados, realizar o treinamento da rede neural artificial, para que a ferramenta aprenda as idiossincrasias e possa exarar sugestões para os diagnósticos de acordo com o que está

definido no Apêndice A
Regras para a definição dos diagnósticos.

- Com a rede neural treinada, a ferramenta está finalmente pronta para iniciar o funcionamento preliminar, recebendo informações dos monitores multiparamétricos e fornecendo diagnósticos em tempo real para o paciente, ajudando, após calibração e validações futuras, no processo de análise destas informações e de tomada de decisão pela equipe médica responsável.

1.3 Estrutura do Documento

A seguir, um breve resumo de cada capítulo deste documento, que estará organizado da seguinte forma:

- **Capítulo 1: Introdução** - Neste capítulo estão contidos a motivação para a construção da ferramenta proposta, os objetivos e como este documento está organizado.
- **Capítulo 2: Revisão Bibliográfica** – Neste capítulo constam os textos necessários para o embasamento teórico, descrevendo o que são unidades de terapia intensiva (UTIs), sistemas de apoio à decisão (SADs) e redes neurais artificiais (RNAs).
- **Capítulo 3: Ferramenta Proposta** - Este capítulo engloba toda a definição da ferramenta de apoio à decisão para diagnósticos de enfermagem proposta neste trabalho, desde o diagrama UML até a descrição de um SAD inteligente e adaptativo.
- **Capítulo 4: Resultados** - Neste capítulo estão os resultados obtidos após a execução dos experimentos e simulações realizadas.
- **Capítulo 5: Conclusões e Trabalhos Futuros** - Neste capítulo estão uma síntese dos resultados obtidos e discussões de como a ferramenta proposta pode contribuir para a solução do problema abordado nesta monografia, além da identificação de trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Este capítulo apresenta uma visão geral sobre Unidades de Terapia Intensiva (UTI): o que são, como é o ambiente, como os enfermeiros estão inseridos neste ambiente e como são obtidos os dados a partir dos equipamentos ligados aos pacientes nos leitos. Além disso, conceitos sobre Sistemas de Apoio à Decisão (SAD) serão abordados, bem como um embasamento teórico sobre Redes Neurais Artificiais (RNA).

2.1 Unidades de Terapia Intensiva

Segundo o CREMESP [5], unidades de terapia intensiva, ou simplesmente UTIs, são estruturas hospitalares que se caracterizam como uma "unidade complexa dotada de sistema de monitoração contínua que admite pacientes potencialmente graves ou com descompensação de um ou mais sistemas orgânicos e que, com o suporte e tratamento intensivos, tenham possibilidade de se recuperar". Tratamento intensivo não diz respeito à intensidade de verificação de sinais e medidas para sustentação da vida, mas sim à proximidade intensa do profissional de saúde [1], estando presente para prestar qualquer assistência médica a qualquer momento.

Com o passar do tempo, foi notado que pacientes com doenças agudas, com risco de vida ou lesão, que precisavam de um cuidado mais intensivo e de mais atenção, poderiam ser mais bem tratados se fossem agrupados em áreas específicas dos hospitais [26]. A UTI nasceu dessa necessidade de se oferecer suporte avançado de vida a pacientes que estejam gravemente doentes e que possuam alguma chance de sobreviver, e é um ambiente de alta complexidade, dotado de equipamentos com tecnologia avançada. Sua principal função é fornecer monitoramento e vigilância vinte e quatro horas à um paciente.

Os recursos tecnológicos em uma UTI representam a possibilidade de uma assistência com qualidade de resolutividade, pois o uso da tecnologia, cada vez mais sofisticada, pode fazer a diferença a favor da vida [39]. Mas, atrelado à disponibilidade

constante de tantos aparelhos, o paciente que esteja internado pode começar a ficar preocupado com seu estado de saúde, gerando algum estado de ansiedade, pelo fato de ter diversos aparelhos monitorando-o e ele poder associar essa grande quantidade como uma piora em seu quadro de saúde. Isso é reforçado pela afirmação de que “os pacientes nas unidades de cuidados intensivos atuais são cercados por tecnologia avançada que, embora essencial para salvar vidas, pode criar um ambiente estranho e, até mesmo, ameaçador para eles” [15].

O profissional que se dedica a trabalhos intensivos em ambientes de UTIs são chamados de intensivistas, e a equipe é sempre formada por diversos profissionais, como fisioterapeutas, assistentes sociais, psicólogos, nutricionistas, enfermeiros e outros, além dos médicos. Outra função destes profissionais, principalmente dos enfermeiros e dos assistentes sociais, é dar conforto e suporte aos familiares do paciente internado, uma vez que a situação de internação de algum parente, ainda mais em UTIs, é sempre delicada para os familiares.

Nas próximas seções, constarão um breve histórico do surgimento das UTIs, bem como uma descrição dos sinais vitais que foram levados em consideração nesta monografia.

2.1.1 História

As UTIs não são recentes. Na guerra da Criméia (1854-1856), a enfermeira Florence Nightingale separou os feridos da guerra em estado grave daqueles menos graves, e com isso conseguiu reduzir os índices de mortalidade e infecções, que era de 40%, para 2%. Com isso, Florence mostrou a importância da monitoração ao paciente grave, o que depois viria a idealizar as UTIs [33].

Com o passar dos anos, foram surgindo novos recursos (novos equipamentos, por exemplo), e o modelo inicial de UTI foi se estabelecendo nos Estados Unidos pelo neurocirurgião Walter Dandy em 1930, e era composto por três leitos para pós-operatório neurocirúrgico [33]. Em seguida, em 1950, surgiu o primeiro médico intensivista – que é o profissional que se dedica ao atendimento em UTIs –, Peter Safar. As UTIs tiveram sua origem nas salas de recuperação, para onde os pacientes que se submeteram a procedimentos anestésico-cirúrgicos eram levados e tinham seus sinais vitais monitorados [33].

2.1.2 Sinais Vitais

As alterações das funções corporais geralmente se refletem na temperatura do corpo, na pressão arterial, na pulsação (ou frequência cardíaca) e na respiração, podendo indicar enfermidades. Por causa disso, são chamados de sinais vitais [40].

A avaliação dos sinais vitais auxilia a equipe de saúde na tomada de decisão sobre as intervenções. Essas medidas fornecem informações muito importantes sobre o estado de saúde atual do paciente, sendo assim um método eficiente de monitoramento. Estes sinais são obtidos através de um equipamento chamado de monitor multiparamétrico (Figura 1), que está ligado ao paciente, fornecendo informações importantes sobre seu atual estado de saúde. Através de sensores, aparelhos de pressão automatizados e eletrodos ligados ao paciente, é possível acompanhar a saturação de oxigênio do sangue, a pressão arterial e a frequência cardíaca e respiratória [28]. Este equipamento é programado para alarmar quando ocorrem as seguintes situações:

- Acelerações ou desacelerações da frequência cardíaca.
- Picos hipertensivos ou hipotensão.
- Queda da saturação de oxigênio no sangue.



Figura 1. Monitor multiparamétrico

[Fonte: Retirada do site: <http://goo.gl/PDE0xc>. Acesso em 20 de junho de 2014]

Nas próximas seções, estarão descritos o significado de cada um dos sinais vitais e como é realizada a mensuração.

Frequência Cardíaca

A frequência cardíaca é representada pelo número de vezes que o coração bate por minuto. É um bom indicador do trabalho cardíaco e é controlado pelo nódulo sinoatrial, também conhecido como marca-passo natural do coração [10].

A unidade de medida é o BPM (batimentos por minuto). O valor de referência da frequência cardíaca pode variar de paciente para paciente, de acordo com sua idade. Sendo assim:

- Recém-nascidos até 23 meses – 120 a 140 batimentos por minuto.
- Crianças de 2 a 6 anos – de 100 a 110 batimentos por minuto.
- Crianças de 8 anos até a adolescência – 80 a 100 batimentos por minuto.
- Adulto sedentário – 70 a 80 batimentos por minuto.
- Atletas – cerca de 50 batimentos por minuto.

A frequência cardíaca é mensurada, em geral, pela palpação manual da artéria radial, que fica um pouco acima do pulso, e deve ser realizada apenas com os dedos indicador e médio. Em ambientes de UTIs, esta frequência é mensurada através dos monitores multiparamétricos, cujos eletrodos estão ligados ao paciente internado.

Pressão Arterial

A pressão arterial é a pressão exercida pelo sangue no interior das artérias e depende da força de contração do coração, da quantidade de sangue circulante e da resistência oferecida pelas paredes das artérias. Durante a contração do ventrículo esquerdo (fenômeno chamado de sístole), a pressão sanguínea está em seu valor máximo, sendo chamada de pressão sistólica. Durante o relaxamento deste ventrículo (fenômeno chamado de diástole), a pressão está em seu valor mínimo, sendo chamada de pressão diastólica.

A unidade de medida de pressão arterial é o mmHg (milímetros de mercúrio). O primeiro número (de maior valor) é o da pressão sistólica, enquanto o segundo (de menor valor) é o da pressão diastólica. Não há um valor preciso de pressão arterial

normal. Em geral, 120/80 mmHg é considerado ideal para um jovem adulto. Acima deste valor é considerado hipertensão e, abaixo, hipotensão.

A pressão arterial pode ser aferida de forma intrusiva ou não. A forma não intrusiva é realizada por meio de equipamentos externos, como o esfigmomanômetro (ou aparelho de pressão), que pode ser visualizado na Figura 2, ou com aferidores eletrônicos de pressão, que pode ser visualizado na Figura 3. Já na forma intrusiva, a pressão é medida através de um cateter introduzido na artéria, o qual é conectado a uma coluna líquida. O valor da pressão é obtido através do transdutor de pressão (Figura 4).



Figura 2. Esfigmomanômetro

[Fonte: Retirada do site: <http://goo.gl/rz4vmP>. Acesso em 20 de junho de 2014]



Figura 3. Aferidor eletrônico de pressão

[Fonte: Retirada do site: <http://goo.gl/BDHjH8>. Acesso em 20 de junho de 2014]



Figura 4. Transdutor de pressão

[Fonte: Retirada do site: <http://goo.gl/xEiheC>. Acesso em 20 de junho de 2014]

A pressão arterial média (PAM) é calculada de acordo com a Equação 2.1:

$$PAM = \frac{PAS + (PAD * 2)}{3} \quad (2.1)$$

Onde:

- *PAM* é a pressão arterial média;
- *PAS* é a pressão arterial sistólica; e
- *PAD* é a pressão arterial diastólica.

Para a pressão arterial média, o valor de referência é em torno de 70 a 110 mmHg.

Frequência Respiratória

A frequência respiratória é, em geral, mensurada através da observação da expansão da caixa torácica do paciente, contando o número de inspirações por minuto. A unidade de medida é RPM (respirações por minuto).

Um adulto normal respira confortavelmente 12 a 18 vezes por minuto. Já em recém-nascidos, o valor normal é de 30 a 40 respirações por minuto. Em crianças, esse valor fica em torno de 25 a 30 respirações por minuto [11].

Temperatura

A temperatura reflete o funcionamento do sistema termorregulador do corpo humano, o equilíbrio entre a produção e a perda de calor pelo organismo. O equipamento utilizado para verificação da temperatura corporal é o termômetro e a

região que é mais comumente utilizada para se realizar esta verificação é a axila, mas pode ser verificada também em outras regiões, como reto e boca.

A unidade de medida de temperatura é o °C (grau Celsius). Em um adulto, a temperatura axilar normal varia entre 36 e 37,2°C.

Saturação de Oxigênio

A saturação de oxigênio é uma medida da quantidade de oxigênio no sangue, sendo o valor máximo 100%. Os níveis normais, medidos por um oxímetro de pulso (Figura 5), variam de 95% à 100%, com 90% já sendo considerado um valor baixo.



Figura 5. Oxímetro de pulso

[Fonte: Retirada do site: <http://goo.gl/HO8Zk4>. Acesso em 20 de junho de 2014]

2.2 Sistema de Apoio à Decisão

Sistemas de Apoio à Decisão (SAD) são ferramentas desenvolvidas para dar suporte na resolução de problemas e aperfeiçoar o processo de tomada de decisão. Tais sistemas utilizam geralmente uma grande carga de dados, fornecem uma interface agradável ao usuário e permitem à ele ter sua própria percepção do problema [41]. Usando técnicas computacionais apropriadas e modelos interativos humano-máquina, estes sistemas auxiliam e aumentam a efetividade da tomada de decisão.

Os SADs têm sido muito aplicados em situações onde a quantidade de informações é muito grande ou em problemas que requerem muita precisão, tornando inviável a análise destas situações apenas por meio de um profissional humano. Com isso, a implementação de um sistema de apoio à decisão serviria para aumentar a

eficiência no processo decisório. Mas, se uma ferramenta de apoio não for considerada enriquecedora ao processo de decisão, ela poderá ser abandonada com a justificativa de poupar esforço cognitivo do tomador de decisão [27].

Na enfermagem, os sistemas de apoio à decisão já aparecem em alguns cenários [19], onde facilitam e realçam a capacidade clínica de tomar decisões para a efetividade do cuidado com os pacientes. Em UTIs, onde os pacientes precisam de cuidados cada vez mais qualificados e intensos, o processo de tomada de decisão pode se tornar mais eficiente com o uso de SADs, uma vez que a quantidade de informações simultâneas providas pelos equipamentos ligados ao paciente torna inviável o processo de análise e tomada de decisão pela natureza humana, o que pode resultar num diagnóstico incerto, implicando num tratamento incorreto.

O nível de segurança de um paciente está diretamente relacionado à ocorrência de erros médicos [20][21] e, para evitar a ocorrência destes erros, um sistema de apoio à decisão clínica se faz necessário para auxiliar – não substituir – os médicos e enfermeiros na tomada de decisão e melhorar ainda mais o atendimento aos pacientes dos leitos das UTIs, mantendo-os sempre confortáveis. Os enfermeiros, principalmente, estão cada vez mais sobrecarregados com tarefas burocráticas [22] além de sua própria função, que é evoluir pacientes, e acabam exercendo funções que não deveriam, como organizar documentos, procurar exames, entre outras funções administrativas, o que gera cansaço, estresse e prejudica a prestação de uma assistência médica de qualidade [12] por conta da sobrecarga de funções.

Neste contexto, o emprego de técnicas de inteligência artificial para a construção destas ferramentas traz alguns benefícios, como: adaptabilidade, capacidade de generalização e aprendizagem. SADs que utilizam estas técnicas inteligentes são chamados, de acordo com a literatura, de Sistemas de Apoio à Decisão Inteligentes – SADIs [27].

A capacidade de aprendizagem está relacionada com a possibilidade de um sistema aprender como responder a determinadas situações, de acordo com o conjunto de entradas de exemplo. Já a capacidade de generalização diz respeito a capacidade de responder a situações inesperadas utilizando o aprendizado já adquirido [27]. E a adaptabilidade é a capacidade que o sistema tem de modificar o seu comportamento de maneira automática, de acordo com os estímulos que recebe.

Neste trabalho, a técnica inteligente utilizada no sistema de apoio à decisão foi a rede neural artificial (RNA), que estará descrita na seção seguinte.

2.3 Redes Neurais Artificiais

Redes Neurais Artificiais (RNA), ou simplesmente Redes Neurais, são modelos de processamento de informações baseados no funcionamento do cérebro humano com suas redes neurais biológicas [43], cuja unidade básica é o neurônio. Este órgão tão poderoso e tão complexo é composto por cerca de 100 bilhões de neurônios interconectados [45] formando uma intensa rede que é capaz de processar milhões de informações e de realizar algo em torno de 60 trilhões de ligações sinápticas. Apesar de ser constituído de neurônios individualmente lentos, o cérebro é capaz de processar trilhões de informações simultaneamente.

Genericamente, uma rede neural pode ser definida como uma estrutura composta por um conjunto de unidades de processamento (neurônios) interconectadas [43]. Com a capacidade de aprendizado, feita através do treinamento – que é um processo de ajuste aplicado aos pesos das interconexões dos neurônios –, a rede neural consegue encontrar uma solução generalizada para um conjunto de problemas [30]. O treinamento das RNA é feito através de épocas, em que uma época é concluída após o último registro do conjunto de dados utilizado para o treinamento ser apresentado à rede. As principais técnicas de aprendizado de uma RNA são:

- Aprendizado supervisionado (ou com professor), quando são apresentadas à rede as respostas esperadas para os respectivos valores de entrada, de modo que ela possa verificar o quão errada foi sua resposta e, com isso, ajustar os pesos das interconexões entre os neurônios (Figura 6).

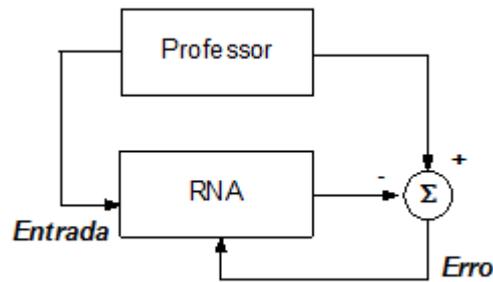


Figura 6. Aprendizado supervisionado (ou com professor)

[Fonte: Retirada do site: <http://goo.gl/74bCfx>. Acesso em 10 de junho de 2014]

- Aprendizado não supervisionado (ou auto-organização), quando não são apresentadas à rede as saídas esperadas e ela é capaz de aprender apenas com os valores passados como entrada (Figura 7).

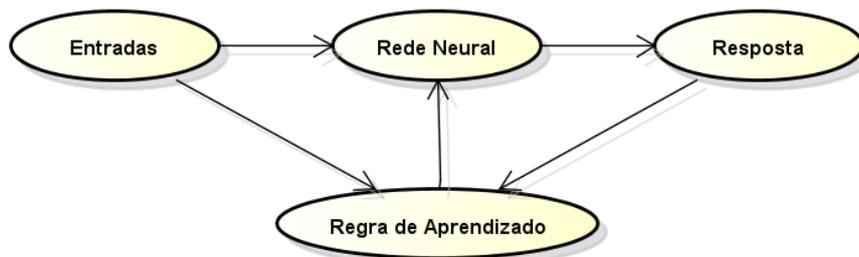


Figura 7. Aprendizado não-supervisionado

- Aprendizado por reforço, quando há a avaliação de um especialista (um enfermeiro, por exemplo), informando se a resposta fornecida pela rede está correta ou não e, em caso negativo, a RNA irá reajustar os pesos, fornecendo uma nova resposta. O especialista não informa à rede a saída esperada para aqueles valores de entrada (pois isso seria um aprendizado supervisionado). Ele apenas informa o quanto errada está a resposta (se muito ou pouco) (Figura 8).

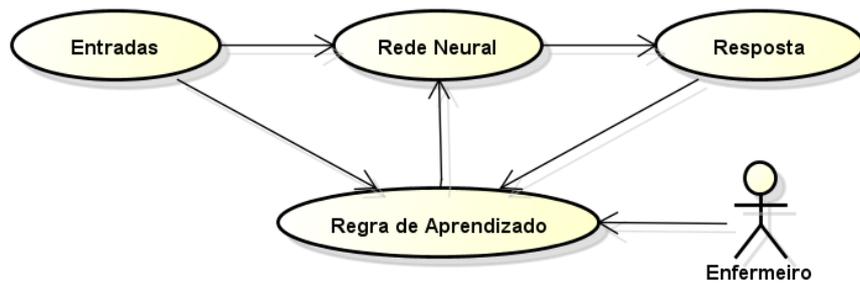


Figura 8. Aprendizado por reforço

Pode-se dizer que, assim como o cérebro, as Redes Neurais Artificiais são compostas por elementos de processamento interconectados – os neurônios – que estão dispostos em camadas (uma de entrada, uma ou mais intermediárias e uma de saída) e que são responsáveis pela não linearidade e pela memória da rede [43].

2.3.1 Neurônio Biológico

O neurônio é a unidade básica do sistema nervoso humano e é responsável pela condução dos impulsos elétricos. Sua estrutura é composta pelos dendritos, que são os terminais de entrada e é por eles que são recebidas informações dos outros neurônios; pelo corpo celular, onde essas informações são processadas; e pelo axônio, por onde a informação é propagada até os terminais, que são também chamados de telodendros. Esta estrutura pode ser visualizada na Figura 9.

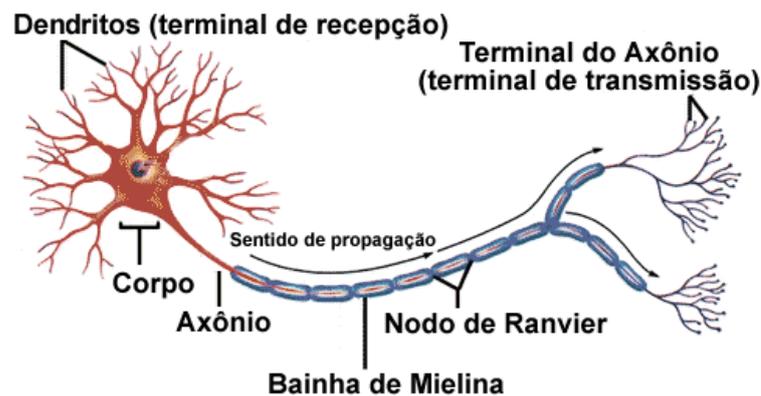


Figura 9. Estrutura do neurônio

[Fonte: Retirada do site: <http://goo.gl/DIhhA9l>. Acesso em 10 de junho de 2014]

A transmissão dos impulsos nervosos se dá do axônio do neurônio anterior para os dendritos dos próximos, nunca no sentido contrário, e é realizada nas sinapses, que são regiões de comunicação entre os neurônios (Figura 10). Esta transmissão é

feita por meio de mediadores químicos, os neurotransmissores, através da fenda sináptica.

Um dos aspectos importantes para o entendimento do neurônio biológico, segundo Valença [43], está relacionado com a “Lei do Tudo ou Nada”, em que se o estímulo nervoso for muito pequeno, implicando que a intensidade do estímulo nervoso será menor que o limiar excitatório daquele neurônio, o impulso nervoso não ocorrerá e a informação não será transmitida para os outros neurônios. Por outro lado, se este limiar for alcançado, o potencial de ação do neurônio será sempre o mesmo, não importando a intensidade do estímulo nervoso.

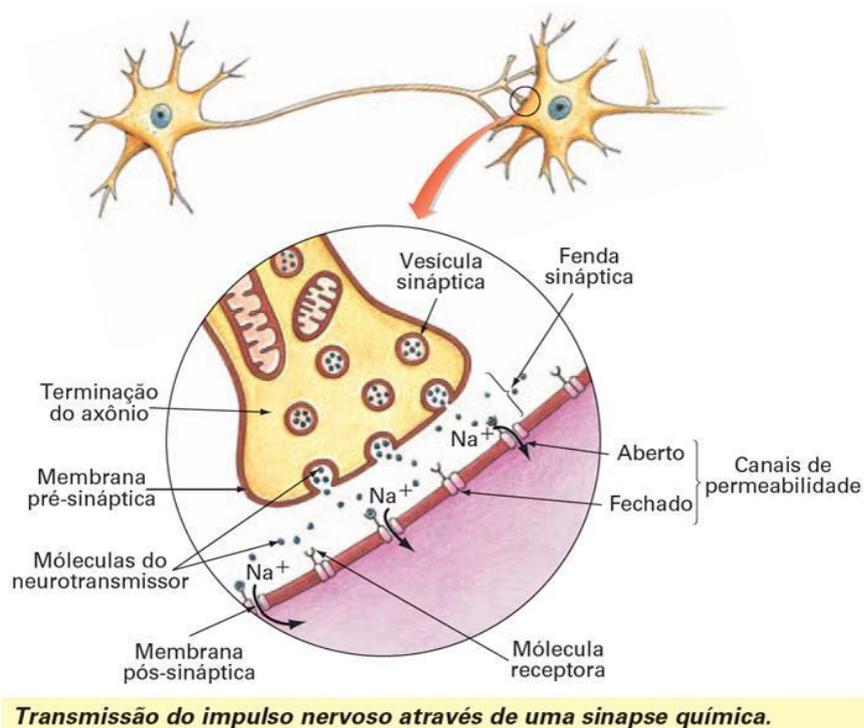


Figura 10. Sinapses químicas

[Fonte: Retirada do site: <http://goo.gl/MiyC8L>. Acesso em 10 de junho de 2014]

2.3.2 Neurônio Artificial

A primeira proposta de representação para o neurônio biológico foi o *Threshold Logic Unit* (TLU), feita por Warren McCulloch (neurofisiologista e psiquiatra) e Walter Pits (lógico matemático) em 1943 [25]. Este modelo, de forma bastante simplificada, consegue representar o neurônio biológico utilizando uma regra de propagação e uma função de ativação. Sua estrutura pode ser visualizada na Figura 11.

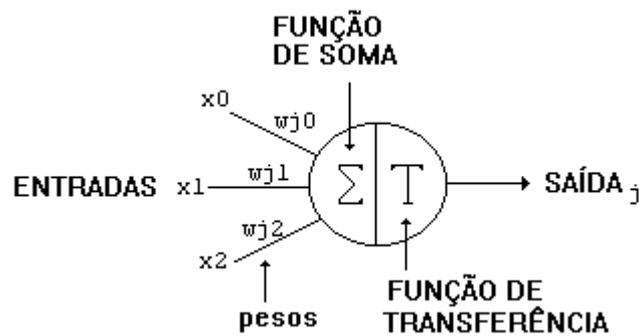


Figura 11. Neurônio Artificial de McCulloch e Pitts

[Fonte: Retirada do site: <http://goo.gl/1d36oT>. Acesso em 13 de junho de 2014]

O neurônio artificial possui uma notável semelhança com os neurônios biológicos, a começar pelos dendritos que recebem o vetor de entrada $[x_0, x_1, x_2, \dots, x_n]$ – que são os estímulos – e passam esse valor para a unidade de processamento – o corpo celular –, que realiza uma operação de soma ponderada (dada pela equação 2.2) e, caso o estímulo (o resultado da equação 2.2) seja maior que o limiar excitatório do neurônio (que depende da função de transferência utilizada), a saída a_j é propagada pelo axônio do neurônio para o dendrito do próximo neurônio artificial da rede (que representa uma sinapse química entre dois neurônios).

$$\text{neurônio}_j = \sum_{i=1}^n w_{ji} * x_i - \theta \quad (2.2)$$

Os dendritos do neurônio artificial possuem pesos, que indicam o quão importante é aquele estímulo para o aprendizado da rede, ou seja, quanto maior o peso, mais importante é aquele estímulo. Estes pesos devem possuir valores que, ao serem multiplicados pelas entradas, tornem a saída a_j próxima do valor desejado. Esta saída, como dito anteriormente, é dependente da função de transferência utilizada. As mais utilizadas na prática são a sigmoide logística e a tangente hiperbólica (TANH), que podem ser visualizadas, respectivamente, nas equações 2.3 e 2.4.

$$f(\text{neurônio}_j) = \frac{1}{1 + e^{-\text{neurônio}_j}} \quad (2.3)$$

$$f(\text{neurônio}_j) = \frac{e^{\text{neurônio}_j} - e^{-\text{neurônio}_j}}{e^{\text{neurônio}_j} + e^{-\text{neurônio}_j}} \quad (2.4)$$

Com a evolução das RNA, surgiram novas implementações, entre elas o *Perceptron* (Figura 12) e o ADALINE (*ADaptive LInear NEuron*). Ambas possuem uma estrutura com uma camada de entrada e uma camada de saída. A principal diferença entre elas estava apenas na função de transferência utilizada: enquanto na primeira era a função linear, na segunda era a função sigmoide. Ambas implementações têm como base o neurônio matemático de McCulloch e Pits.

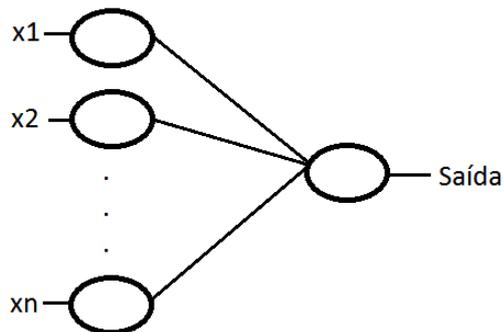


Figura 12. Exemplo de um *Perceptron*

Com base nas redes mencionadas anteriormente, surgiu uma das mais importantes classes da RNA: o *Perceptron* de múltiplas camadas (do inglês, *Multilayer Perceptron*).

2.3.3 Multilayer Perceptron

Na rede *Multilayer Perceptron* (MLP) (Figura 13), é feita a adição de pelo menos uma camada intermediária (também chamada de camada escondida) ao *Perceptron* simples. Além disso, em geral, é adicionado mais um neurônio, chamado de *bias*, que é utilizado para aumentar os graus de liberdade e permitir uma melhor adaptação da rede neural ao conhecimento fornecido à ela.

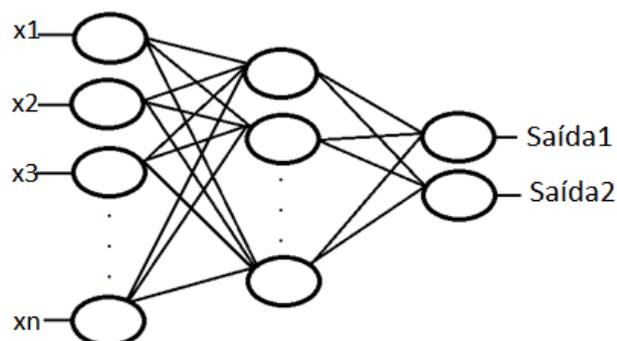


Figura 13. Exemplo de uma MLP

O fato de se adicionar uma camada intermediária ao *Perceptron* simples faz com que a MLP seja capaz de resolver problemas que não são linearmente separáveis, possibilitando a aproximação de qualquer função contínua [6]. Já ao adicionar mais de uma camada, a rede é capaz de resolver qualquer função matemática [7].

Problemas linearmente separáveis são aqueles que podem ser separados por uma reta em um hiperplano (Figura 14). Já problemas não separáveis linearmente não podem ser separados por apenas uma reta, sendo necessário a utilização de mais de uma reta ou uma região (Figura 15).

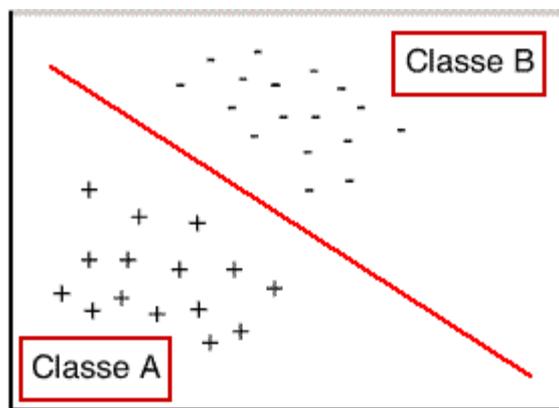


Figura 14. Exemplo de problema linearmente separável.

[Fonte: Retirada do site: <http://goo.gl/48RJLM>. Acesso em 14 de junho de 2014]

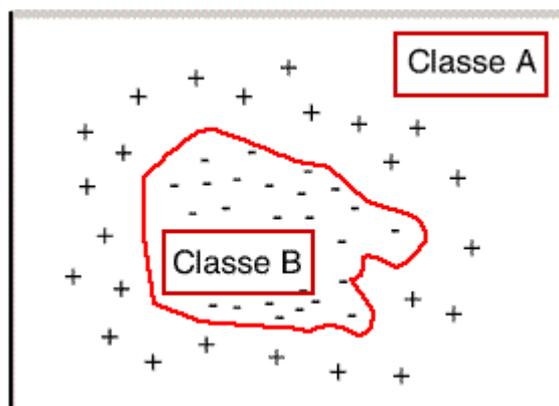


Figura 15. Exemplo de problema não separável linearmente.

[Fonte: Retirada do site: <http://goo.gl/yNQD2S>. Acesso em 14 de junho de 2014]

A estrutura da MLP pode ser resumida em: uma camada de entrada, onde cada neurônio representa uma variável para o problema; pelo menos uma camada

intermediária, onde é feito todo o processamento da rede e é responsável pela não linearidade, fazendo com que a rede seja capaz de resolver problemas reais (não linearmente separáveis, como dito anteriormente) e possuem função de ativação sigmoideal (como a sigmoide logística ou a TANH); e uma camada de saída, que é responsável pela resposta da rede àquele padrão de entrada e seus neurônios também possuem função de ativação sigmoideal ou linear.

Assim como o *Perceptron* e o ADALINE, as redes neurais MLP também precisam de um algoritmo de treinamento que, em geral, são de aprendizado supervisionado, e tem como objetivo encontrar os pesos ótimos para cada uma das conexões entre os neurônios da rede. O algoritmo mais utilizado é o *Back propagation*, desenvolvido por Paul Werbos em 1974 [44] – que é uma generalização da regra delta¹, que é também conhecida como técnica do gradiente descendente –, mas existem outros, como o *Resilient propagation*, desenvolvido por Martin Riedmiller e Heinrich Braun em 1992 [34]. Estes algoritmos serão melhor detalhados na próxima seção.

2.3.4 Algoritmos de treinamento

Nesta seção estarão descritos com mais detalhes os algoritmos mencionados na seção anterior: o *Back propagation*, desenvolvido por Paul Werbos, e o *Resilient propagation* (ou Rprop), desenvolvido por Riedmiller e Braun.

Back propagation

O algoritmo mais tradicionalmente utilizado consiste numa generalização da regra delta e é conhecido como *Back propagation*. Este algoritmo foi desenvolvido por Paul Werbos em 1974 em sua tese de doutoramento, mas a popularização desta técnica ocorreu apenas na década de 80 com os trabalhos de Geoffrey Hinton, David Rumelhart e Ronald Williams [36].

As redes MLP são de treinamento supervisionado [43] e um dos maiores problemas encontrados durante a fase de treinamento destas RNAs é o fato de elas

¹ Consiste em realizar ajustes apropriados nos pesos de modo a reduzir o erro entre a saída esperada e a saída obtida pela rede neural.

possuírem camadas intermediárias, não sendo possível calcular o erro destas camadas necessário para realizar o reajuste dos pesos. Mas o algoritmo *Back propagation* consegue resolver este problema pois nele é realizada uma propagação recursiva dos erros, começando com a camada de saída até a camada de entrada.

A aprendizagem da MLP com o uso deste algoritmo ocorre em duas fases: na primeira, conhecida como *forward*, o sinal é propagado no sentido progressivo, ou seja, da camada de entrada para a de saída da rede, onde o sinal de saída e o erro são calculados. O erro é calculado com a diferença entre a saída esperada e a saída obtida e os pesos da rede permanecem constantes, ou seja, não são alterados nessa primeira fase. Já na segunda fase, conhecida como *backward*, os erros são propagados recursivamente da camada de saída para a camada de entrada e é nesta etapa que os pesos sinápticos são ajustados. O ajuste dos pesos se dá de acordo com a equação 2.5:

$$w_{ij}^m(\text{novo}) = w_{ij}^m(\text{antigo}) + \alpha \delta_i^m f^{m-1}(\text{neurônio}_j^{m-1}) + \beta \Delta w_{ij}^m \quad (2.5)$$

Onde:

- $w_{ij}^m(\text{novo})$ é o novo valor atribuído ao i -ésimo peso do neurônio j da camada m ;
- $w_{ij}^m(\text{antigo})$ é o valor atual do i -ésimo peso do neurônio j da camada m ;
- α é a taxa de aprendizagem;
- δ_i^m é a sensibilidade do neurônio i da camada m e é calculada segundo as equações 2.6 – para os neurônios da camada de saída – e 2.7 – para os neurônios das camadas escondidas;
- $f^{m-1}(\text{neurônio}_j^{m-1})$ é o cálculo da função de ativação do neurônio j da camada $m - 1$;
- β é o valor do momento, uma constante que determina a influência da variação passada do i -ésimo peso do neurônio j da camada m ;
- Δw_{ij}^m é a variação do i -ésimo peso do neurônio j da camada m .

$$\delta_i^m = (d_i - y_i) f'(neurônio_i) \quad (2.6)$$

$$\delta_j^{m-1} = f'^{m-1}(neurônio_j^{m-1}) \sum_{i=1}^N w_{ij}^m \delta_i^m \quad (2.7)$$

Mais informações detalhadas sobre a execução deste algoritmo podem ser encontradas na literatura [36][44][36].

Resilient propagation

Resilient propagation, ou simplesmente RPROP, não é o melhor método de treinamento em todos os casos, mas na grande maioria sim. É uma heurística para aprendizado supervisionado que pode ser utilizada em redes neurais do tipo *feedforward* (a MLP, por exemplo) e foi criado por Martin Riedmiller e Heinrich Braun em 1992 [34].

Este algoritmo tipicamente supera consideravelmente o *Back propagation*. Ele não possui parâmetros que precisam ser determinados [32], ao contrário do *Back propagation*, em que é necessário especificar um valor para a taxa de aprendizado e para o momento, e encontrar valores ótimos para estes dois parâmetros não é uma tarefa fácil.

Diferentemente do algoritmo *Back propagation*, que leva em conta quanto errada está a saída obtida com relação à saída desejada e a atualização dos pesos é feita a cada padrão apresentado à rede, o algoritmo *Resilient propagation* apenas leva em conta o sinal da derivada parcial sobre todos os padrões (e não a magnitude), atua independentemente em cada peso e a atualização é feita em *batch* – ou seja, após serem apresentados todos os padrões à rede, os pesos são atualizados.

Se para cada peso houver uma mudança do sinal da derivada parcial da função de erro total comparada com a última iteração, o valor de atualização para aquele peso é multiplicado por um fator n^- , que é menor que 1. Se a última iteração o sinal foi o mesmo, o valor de atualização é multiplicado por um fator n^+ , que é maior que 1. Os valores de atualização são calculados para cada peso da maneira descrita, e só então cada peso é alterado de acordo com o seu próprio valor de atualização [31]. n^+ e n^- são empiricamente definidos como 1,2 e 0,5, respectivamente.

2.3.5 Validação Cruzada

A validação cruzada é uma técnica que tem como objetivo evitar o treinamento em excesso (*overfitting*) da rede neural [43]. Isso evita que a RNA perca sua capacidade de generalização decorando os exemplos que foram utilizados no momento do treinamento. Esta técnica utiliza um conjunto de dados (chamado de conjunto de validação cruzada) diferente do conjunto de treinamento. A cada ciclo de treinamento é calculada a taxa de erro dos conjuntos de treinamento e validação cruzada. No momento em que a taxa de erro do conjunto de validação cruzada começar a aumentar com o tempo, o treino é interrompido para que a capacidade de generalização da RNA não seja comprometida.

Capítulo 3

Ferramenta Proposta

Este capítulo tem como objetivo brevemente apresentar uma visão do modelo proposto, baseado nos conceitos abordados no capítulo anterior.

3.1 Modelo proposto

Unidades de terapia intensiva são ambientes onde os pacientes precisam de cuidados intensivos e ter sempre os seus sinais vitais monitorados. Conectados por diversos equipamentos de monitoramento, estes pacientes permanecem sob os cuidados de profissionais, conhecidos como intensivistas.

Monitores multiparamétricos são utilizados para monitorar os sinais vitais do paciente submetido aos cuidados de uma UTI. Diversas informações são exibidas nestes equipamentos e o processo de análise destas informações pode se tornar uma tarefa complicada para a natureza humana, visto que o intensivista teria que processar várias informações simultaneamente para poder estabelecer o diagnóstico correto para, só assim, prestar assistência corretamente. Como os enfermeiros, principalmente, estão muito envolvidos com tarefas burocráticas, o processo de decisão clínica pode se tornar ainda mais difícil, por conta das condições inerentes à pessoa humana: cansaço, estresse e falta de atenção, por exemplo.

Para reduzir a ocorrência destes possíveis erros provenientes do cansaço físico/mental e estresse (causados tanto pelas tarefas burocráticas, como por diversos plantões seguidos realizados pelos intensivistas e por outros fatores) e melhorar o conforto e a qualidade do atendimento aos pacientes dos leitos das UTIs, propomos um Sistema Inteligente de Apoio à Decisão ao Diagnóstico de Enfermagem em Pacientes de UTI, que tem como objetivo auxiliar a tomada de decisão da equipe de saúde que esteja responsável por aqueles pacientes nas unidades de terapia intensiva. Como dito anteriormente, a ferramenta não pretende substituir a equipe de saúde, mas sim auxiliar a análise dos dados provenientes dos equipamentos de

monitoramento do paciente e fornecer um diagnóstico mais preciso para uma melhor intervenção médica.

A ferramenta proposta por este trabalho utiliza uma rede neural MLP (*Multilayer Perceptron*) para a aprendizagem dos diagnósticos de enfermagem, tendo como valores de entrada os sinais vitais obtidos pelo monitor multiparamétrico ligado ao paciente monitorado. Foi escolhido utilizar redes neurais no sistema de apoio à decisão inteligente (SADI) por conta da capacidade de generalização e aprendizado, fazendo com que a ferramenta seja capaz de individualizar os diagnósticos.

A rede neural artificial (RNA) utilizada possui a seguinte configuração:

- Todas as saídas da MLP estão restritas ao intervalo [0, 1]. Essas saídas são arredondadas para 0, caso sejam menores que 0,5, e para 1, caso sejam maiores.
- 6 neurônios na camada de entrada, sendo 5 para os sinais vitais utilizados (frequência cardíaca, frequência respiratória, temperatura, pressão arterial e saturação de oxigênio) e 1 para o *bias*.
- 7 neurônios na camada de saída, sendo os dois primeiros utilizados para definir o estado de saúde do paciente (11 – Estado Crítico [Vermelho]; 10 e 01 – Atendimento não emergencial [Amarelo]; 00 – Estado normal [Verde]), e os outros 5 responsáveis por fornecer os diagnósticos.
- $N_{escondida}$ neurônios na camada escondida. Esta quantidade de neurônios será definida experimentalmente mais à frente.

O processo de adaptação da ferramenta se inicia no momento do treinamento. Com o histórico do paciente, é possível saber quais são os valores de normalidade para cada sinal vital, que são aqueles em que não é apresentado risco de saúde. Com isso, a rede adaptar-se-á à cada paciente, gerando pré-diagnósticos individualmente. Os valores considerados normais para cada paciente serão aqueles que mais se repetem em seu histórico. Caso o paciente ainda não possua histórico no hospital, a ferramenta irá considerar, como valores normais, os definidos na literatura [3].

Após o treinamento e consequente adaptação da ferramenta, a sua utilização pode ser iniciada. Os dados de entrada da ferramenta serão os valores dos sinais vitais provenientes do monitor multiparamétrico, que está conectado ao paciente. A

resposta, que será o estado de saúde imediato do paciente (indicado por cores, como dito anteriormente) e o pré-diagnóstico, será exibida em uma pequena tela apenas para simples conferência do enfermeiro e correta intervenção clínica.

A arquitetura proposta para a ferramenta pode ser visualizado na Figura 16.

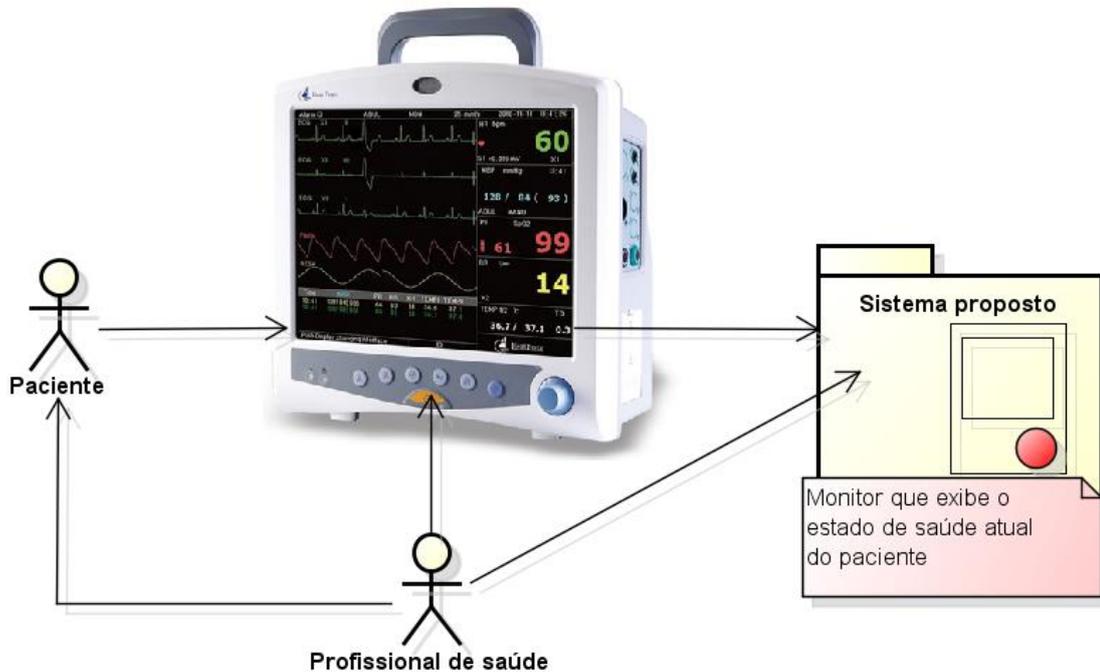


Figura 16. Arquitetura da ferramenta proposta

3.2 Ferramenta Proposta

O sistema implementado baseado na arquitetura proposta foi desenvolvido na linguagem de programação Java, devido a flexibilidade e modularidade da linguagem, além de possuir bibliotecas utilitárias que facilitaram o processo de desenvolvimento da ferramenta. As classes implementadas estão organizadas em pacotes, de acordo com a Tabela 1 abaixo:

Tabela 1. Classes e pacotes da ferramenta proposta

Pacote	Classes
databaseOperations	Reader Filter Transformator
constants	Constants
neuralNetwork	MLP TrainNetwork

A classe Constants, do pacote constants, possui todas as constantes que foram utilizadas por toda a ferramenta. Esta classe foi criada apenas para melhorar a leitura do código, criando nomes para as constantes.

As classes Reader, Filter e Transformator, do pacote databaseOperations, realizam operações na base de dados MIMIC-II. A classe Reader realiza a leitura dos arquivos da base de dados. A classe Filter filtra os dados lidos pela classe anterior, gerando um novo arquivo apenas com as linhas que contém os sinais vitais que serão utilizados pela ferramenta. Já a classe Transformator converte os dados filtrados para o formato CSV (*Comma Separated Values*), para facilitar a utilização dos dados no treinamento da MLP. Esta classe também ignora, após ter gerado o arquivo CSV, as linhas que não possuem os valores dos 5 sinais vitais, ou seja, as linhas que possuem algum valor igual a 0.

A classe TrainNetwork, do pacote neuralNetwork, é utilizada para realizar o treinamento da rede neural MLP, e nela são passados como parâmetros o algoritmo de treinamento, a rede neural artificial que será treinada e a condição de parada, podendo ser: o menor erro médio quadrático; a quantidade de épocas máxima; e o erro do conjunto de validação cruzada aumentar com o tempo. Todas estas condições estão definidas na classe Constants.

A classe TrainNetwork também é responsável pela separação – pseudo-aleatória – dos conjuntos utilizados no treinamento da rede neural: o conjunto de treino (50% da base de dados), o de validação cruzada (25% da base de dados) e o de

testes (os outros 25% restantes). Após a separação destes dados, o treinamento é iniciado utilizando os conjuntos de treino e validação cruzada, sempre verificando o erro no conjunto de validação cruzada. Quando o treinamento é encerrado (condição de parada é estabelecida), o conjunto de testes é utilizado para verificar, de fato, qual a taxa de acerto da rede neural.

O treinamento da MLP só é realizado quando a ferramenta é iniciada pela primeira vez, pois os pesos das conexões entre os neurônios são armazenados em um arquivo de texto (chamado de “Weights_UTI.txt”) para que, caso a ferramenta seja reiniciada, os pesos possam ser recuperados a partir deste arquivo sem a necessidade de um novo treinamento.

A classe MLP, também do pacote neuralNetwork, é a rede neural em si. Para a construção da MLP, foi utilizada a biblioteca Encog² [8], pois de acordo com os testes de *benchmarking* realizados em [4], comparando o *framework* Encog com o JOONE e o Neuroph (também *frameworks* para Inteligência Artificial), o Encog obteve o melhor desempenho, sendo o mais rápido a convergir, obtendo a melhor taxa de erro com poucas épocas. O diagrama de classes da MLP está mostrado na Figura 17.

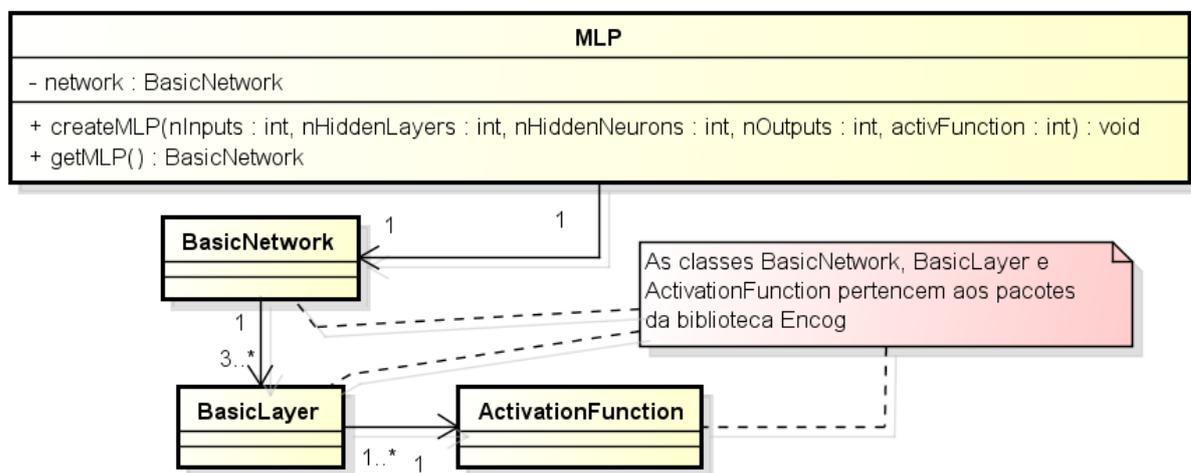


Figura 17. Diagrama de classes da rede neural *Multilayer perceptron*

² *Framework* avançado de aprendizagem de máquina, e possui diversos algoritmos, como Programação Genética, Redes Neurais Artificiais, Algoritmos Genéticos, Hidden Markov Models, entre outros. Está disponível em Java, .Net e C/C++.

A MLP possui um atributo `network`, do tipo `BasicNetwork`, que será a MLP propriamente dita. Esta `BasicNetwork` possui pelo menos 3 camadas (classe `BasicLayer`), pois como dito anteriormente, a principal diferença entre o *Perceptron* simples e a MLP é a adição de pelo menos uma camada intermediária (escondida). Cada camada da MLP possui uma função de ativação (`ActivationFunction`), que pode ser a sigmoide logística ou a tangente hiperbólica (TANH).

A classe MLP possui dois métodos. O método `createMLP` cria a MLP com a quantidade de neurônios de entrada e saída, a quantidade de camadas escondidas e de neurônios nestas camadas e com a função de ativação passadas como parâmetros. Já o método `getMLP()` retorna o atributo `network`.

Com isso, a ferramenta está pronta para receber os sinais vitais do monitor multiparamétrico conectado ao paciente e inferir o diagnóstico de acordo com os valores recebidos. Evidentemente, calibrações e testes clínicos exaustivos devem ser realizados para qualquer aplicação em ambiente real futuro.

Capítulo 4

Experimentos e Resultados

Este capítulo descreve os experimentos realizados com a ferramenta desenvolvida e os resultados encontrados após a execução destes. O objetivo é encontrar quais os melhores parâmetros de configuração que a RNA deve possuir, de forma que os pré-diagnósticos encontrados pela ferramenta sejam os mais coerentes possíveis e que tenham a menor taxa de erro. Os parâmetros investigados foram:

- quantidade de neurônios na camada escondida;
- algoritmo de treinamento; e,
- função de ativação dos neurônios na camada escondida e na camada de saída.

4.1 Metodologia

A base de dados MIMIC II foi pré-processada, realizando uma filtragem a fim de se obter apenas as linhas que contivessem os sinais vitais que foram considerados para o treinamento da rede (pressão arterial média não invasiva, saturação do oxigênio, temperatura, frequência cardíaca e frequência respiratória). Após a filtragem, foi feita uma transformação da nova base de dados resultante a fim de deixá-la no formato CSV (*Comma Separated Values*) com os valores de cada sinal vital separado por ponto e vírgula. Para maiores detalhes sobre o conteúdo do arquivo, ver o Apêndice B Exemplo de arquivo CSV.

Além do arquivo com os sinais vitais, outro arquivo foi gerado com a saída esperada (diagnóstico) para cada linha dessa nova base de dados. Esses diagnósticos são dados de acordo com o estado (valores dos sinais vitais) de normalidade de cada paciente, sendo diferente para cada um. Os diagnósticos são definidos de acordo com as regras estabelecidas no Apêndice A Regras para a definição dos diagnósticos.

Após a transformação da base de dados, foram gerados os conjuntos utilizados no treinamento da rede: o conjunto de treinamento, o de validação cruzada e o de testes, obedecendo a proporção 50%, 25% e 25%, respectivamente (ou seja, 50% da base de dados é utilizada para treinamento, 25% para validação cruzada e os outros 25% restantes para testar o treinamento da rede neural). Estes conjuntos foram gerados pseudo-aleatoriamente.

Após a divisão da base de dados, os valores de cada sinal vital foram normalizados pois, para o treinamento da rede neural, é necessário que os dados de entrada estejam em um mesmo intervalo numérico. Essa normalização é importante para que a RNA não perca sua capacidade de generalização, visto que valores muito diferentes podem influenciar no treinamento e causar a perda dessa capacidade [43].

A normalização é feita segundo a equação (4.1), que é conhecida como transformação linear:

$$y = \frac{(b-a)(x_i - x_{min})}{(x_{max} - x_{min})} + a \quad (4.1)$$

Onde:

- y é o valor normalizado;
- a e b são os limites escolhidos: a é o valor mínimo e b é o valor máximo. Estes valores foram fixados em 0,15 e 0,85, respectivamente;
- x_i é o valor original que será normalizado;
- x_{min} é o valor mínimo de x ; e
- x_{max} é o valor máximo de x .

Terminada a normalização dos dados, o processo de treinamento pode ser iniciado.

Para todos os experimentos, foram consideradas para o treinamento 30.000 épocas, e essas épocas foram executadas 30 vezes com as mesmas bases de dados para treino e validação cruzada. Esta quantidade de execuções é considerada ideal para a realização de análises estatísticas mais significantes [18].

4.1.1 Análises estatísticas

Após terem sido executadas todas as simulações, as análises estatísticas dos resultados obtidos podem ser iniciadas. Para isso, foi utilizado o software matemático R, que utiliza, por padrão, um nível de significância (α) de 0,05. Nas próximas seções, estarão descritos o objetivo de cada um dos testes estatísticos e como analisar o resultado de cada um deles.

Teste de Shapiro-Wilk

Antes de realizar qualquer teste paramétrico em um conjunto de dados, é necessário verificar se as amostras são normalmente distribuídas e se possuem variâncias estatisticamente iguais. Com essas duas condições sendo obedecidas, é possível aplicar um teste paramétrico, caso contrário, deverá ser aplicado um teste não-paramétrico.

Um dos testes mais utilizados para se testar a normalidade das amostras é o de Shapiro-Wilk. Para isso, torna-se necessária a formulação das hipóteses nula e alternativa. A tabela abaixo mostra estas duas hipóteses.

Tabela 2. Hipóteses nula e alternativa para o teste de Shapiro-Wilk

Hipótese	Descrição
Nula	A amostra provém de uma população normal.
Alternativa	A amostra não provém de uma população normal.

Após definir as hipóteses nula e alternativa, o teste pode ser aplicado. Ao executar o teste, o *p-value* encontrado deve ser analisado e comparado com o valor de significância adotado. Caso este valor seja menor que o α estabelecido, ou seja, menor que o nível de significância, a hipótese nula deverá ser rejeitada e a amostra não poderá ser considerada como normalmente distribuída, caso contrário, a amostra provém de uma população normal.

Teste F

O teste F tem por objetivo verificar se as variâncias entre dois conjuntos são estatisticamente iguais, ou seja, se provém de uma mesma população. Assim como

no teste anterior, faz-se necessária a formulação das hipóteses nula e alternativa para a realização deste teste. A tabela abaixo mostra essas hipóteses.

Tabela 3. Hipóteses nula e alternativa para o teste F

Hipótese	Descrição
Nula	As amostras possuem variância estatisticamente iguais.
Alternativa	As amostras não possuem variância estatisticamente iguais.

Se o valor do *p-value* encontrado após a realização deste teste for menor que o α estabelecido, a hipótese nula deverá ser rejeitada e as amostras não podem ser consideradas estatisticamente iguais em suas variâncias. Com este resultado e com o do teste anterior, pode-se decidir a aplicação ou não de um teste paramétrico.

Teste T-Student

Este teste faz análise a partir da média de duas amostras independentes. Como este é um teste paramétrico, é necessário que as amostras sejam antes validadas com os outros dois testes descritos anteriormente, para verificar se seguem uma distribuição normal e se são estatisticamente iguais em suas variâncias.

Assim como os outros testes, para este também se faz necessário a elaboração das hipóteses nulas e alternativas. Este teste tem como objetivo verificar se as médias de cada uma das amostras são estatisticamente iguais ou se há alguma diferença entre elas. Se houver, a amostra que possuir a maior média pode ser considerada a melhor. As hipóteses estão descritas na tabela abaixo.

Tabela 4. Hipóteses nula e alternativa para o teste T-Student

Hipótese	Descrição
Nula	As médias das amostras são estatisticamente iguais.
Alternativa	As médias das amostras não são estatisticamente iguais.

Após a execução deste teste, o *p-value* encontrado é analisado. Caso este valor seja menor que o α definido, a hipótese nula deverá ser rejeitada e a amostra

com a maior média poderá ser considerada a melhor. Neste trabalho, as amostras analisadas serão as taxas de acerto médias das diferentes configurações da rede neural MLP. Caso o *p-value*, após realizar a análise sobre essas amostras, seja menor que o nível de significância adotado, fazendo com que a hipótese nula seja rejeitada, aquela configuração detentora da maior média será considerada a melhor.

Soma dos postos de Wilcoxon

Este teste é um teste não-paramétrico e possui um poder estatístico mais baixo que o teste anterior, mas é muito útil quando é necessário realizar testes estatísticos em amostras que não seguem uma distribuição normal e nem suas variâncias são iguais estatisticamente.

Assim como todos os outros testes anteriores, é necessário a formulação das hipóteses nula e alternativa. Estas hipóteses estão descritas na tabela abaixo. Diferentemente do teste T-Student, a soma dos postos de Wilcoxon realiza a análise baseado na mediana de cada uma das amostras.

Tabela 5. Hipóteses nula e alternativa para o teste da Soma dos postos de Wilcoxon

Hipótese	Descrição
Nula	As medianas das amostras são estatisticamente iguais.
Alternativa	As medianas das amostras não são estatisticamente iguais.

O *p-value*, após a execução deste teste, deve ser comparado com o nível de significância estabelecido. Caso seja menor, a hipótese nula deverá ser rejeitada, ou seja, há diferenças significativas entre as duas amostras. Caso seja maior, a hipótese nula não será rejeitada, o que indica que as duas amostras são estatisticamente iguais.

4.2 Experimentos

Esta seção mostra os experimentos realizados e a configuração da MLP em cada um deles. O objetivo destes experimentos é encontrar a melhor configuração da

RNA que obtenha a melhor taxa de acerto para os diagnósticos e para o status do paciente.

4.2.1 Função de ativação TANH e método de treinamento *Resilient Propagation*

Os experimentos 1, 2 e 3 foram realizados mantendo fixos a função de ativação dos neurônios da camada escondida e da camada de saída e o algoritmo de treinamento da RNA. O algoritmo de treinamento utilizado foi o RPROP e a função de ativação foi a TANH. O número de neurônios na camada escondida, para o experimento 1, foi dado de acordo a Equação 4.2:

$$N_{escondida} = (N_{entradas} * N_{saídas}) - (N_{entradas} + N_{saídas}) \quad (4.2)$$

Já para o experimento 2, o número de neurônios na camada escondida foi dado de acordo com a Equação 4.3:

$$N_{escondida} = (N_{entradas} * N_{saídas}) \quad (4.3)$$

Por fim, para o experimento 3, o número de neurônios na camada escondida foi dado de acordo com a Equação 4.4:

$$N_{escondida} = (N_{entradas} * N_{saídas}) + (N_{entradas} + N_{saídas}) \quad (4.4)$$

4.2.2 Função de ativação sigmoide logística e método de treinamento *Resilient propagation*

Assim como nos experimentos anteriores, nos experimentos 4, 5 e 6 manteve-se a função de ativação para os neurônios da camada escondida e da camada de saída e o algoritmo de treinamento da RNA. A função de ativação foi agora a sigmoide logística e o método de treinamento foi o RPROP. O número de neurônios na camada escondida, para o experimento 4, é dado pela Equação 4.2; para o experimento 5 é dado pela Equação 4.3; e para o experimento 6 é dado pela Equação 4.4.

4.2.3 Função de ativação TANH e método de treinamento *Back propagation*

Para os experimentos 7, 8 e 9, foi considerada a função de ativação TANH para os neurônios da camada escondida e da camada de saída, e o método de treinamento utilizado foi o *Back propagation*. O número de neurônios na camada escondida, para o experimento 7, é determinado pela Equação 4.2. Já para o experimento 8, é determinado pela Equação 4.3. E, para o experimento 9, pela Equação 4.4.

4.2.4 Função de ativação sigmoide logística e método de treinamento *Back propagation*

No último bloco de experimentos, experimentos 10, 11 e 12, foi considerada a função de ativação sigmoide logística para os neurônios da camada escondida e da camada de saída, e o método de treinamento foi o *Back propagation*. A quantidade de neurônios na camada escondida, no experimento 10, é dado pela Equação 4.2, no experimento 11 pela Equação 4.3, e no experimento 12 pela Equação 4.4.

4.3 Resultados

Nesta seção estarão descritos os resultados obtidos da execução dos experimentos, bem como os resultados dos testes estatísticos, indicando, com mais precisão, qual das configurações utilizadas da MLP obteve o melhor desempenho.

4.3.1 Taxa de acerto

Após a execução dos experimentos descritos na seção anterior, pôde-se verificar qual das configurações da rede neural MLP teve a melhor taxa de acerto média para os pré-diagnósticos e para o estado de saúde do paciente. Para mais detalhes sobre as configurações dos experimentos, consultar o Apêndice C Detalhes dos experimentos. As taxas de acerto médias para os pré-diagnósticos e para o estado de saúde do paciente estão mostradas na Tabela 6. A linha em destaque indica a configuração que obteve o melhor resultado. O melhor resultado é considerado aquele que obteve a maior taxa de acerto tanto para os pré-diagnósticos quanto para o estado de saúde do paciente.

Tabela 6. Taxas de acerto – Experimentos 1 a 12

Experimento	Taxa de acerto – Diagnóstico	Taxa de acerto – Estado de saúde
1	84,7%	95,07%
2	87,46%	98,08%
3	87,52%	98,1%
4	88,09%	98,37%
5	87,99%	98,14%
6	88,02%	97,87%
7	87,62%	98,01%
8	87,92%	98,18%
9	87,58%	98,19%
10	87,4%	98,19%
11	87,95%	97,89%
12	87,93%	98,04%

Como mostra a tabela acima, a configuração da MLP que obteve o melhor resultado foi a do experimento 4, que está descrita na Tabela 7 abaixo.

Tabela 7. Melhor configuração da MLP

Experimento	Função de Ativação	Algoritmo de Treinamento	Quantidade de neurônios na camada escondida
4	Sigmoide logística	<i>Resilient propagation</i>	23

A partir destes resultados, a sequência de testes estatísticos foi realizada.

4.3.2 Análises Estatísticas

Para se decidir entre a aplicação de um teste paramétrico ou não, primeiro é necessário saber se as amostras estão normalmente distribuídas ou não. Logo, o primeiro teste executado é o de Shapiro-Wilk, que tem como objetivo testar a normalidade das amostras. Para isso, é necessário a formulação das hipóteses nula e alternativa, que estão descritas na Tabela 8.

Tabela 8. Teste Shapiro-Wilk – Hipóteses nula e alternativa – experimentos 1 a 12

Hipótese	Descrição
Nula	A amostra com taxas de acerto do experimento i é normalmente distribuída
Alternativa	A amostra com taxas de acerto do experimento i não é normalmente distribuída

Onde $i = 1, 2, \dots, 12$.

Após a execução do teste Shapiro-Wilk para os 12 experimentos, observou-se que o p -value de todas as amostras era maior que o nível de significância $\alpha = 0,05$ definido pelo *software* R. Logo, a hipótese nula não deve ser rejeitada e todas as amostras são normalmente distribuídas. Com isso, o primeiro critério para a execução de um teste paramétrico é atendido. Os resultados deste teste podem ser encontrados no

Apêndice

D

Resultado dos testes estatísticos.

O segundo critério a ser analisado é se a variância das amostras pode ser considerada estatisticamente igual e, para isso, é executado o teste F. Este teste é realizado comparando duas amostras simultaneamente. Como o experimento 4 deteve o melhor resultado, ele foi comparado com os outros 11. Assim como no teste anterior, se faz necessário a formulação das hipóteses nula e alternativa, que estão descritas na Tabela 9.

Tabela 9. Teste F – Hipóteses nula e alternativa – experimentos 1 a 12

Hipótese	Descrição
Nula	A variância da amostra com taxas de acerto do experimento 4 é estatisticamente igual a variância da amostra com taxas de acerto do experimento i .
Alternativa	A variância da amostra com taxas de acerto do experimento 4 não é estatisticamente igual a variância da amostra com taxas de acerto do experimento i .

Onde $i = 1, 2, \dots, 12$ e $i \neq 4$.

Após a execução do teste F – comparando a variância das amostras do experimento 4 com as amostras dos outros 11 experimentos –, observou-se que o p -value de todas as amostras era maior que o nível de significância $\alpha = 0,05$ definido pelo software R. Logo, a hipótese nula não deve ser rejeitada e a variância das amostras deve ser considerada estatisticamente igual.

Como as amostras são consideradas normalmente distribuídas e estatisticamente iguais em suas variâncias – que foi constatado nos testes anteriores –, um teste paramétrico foi executado, tornando-se desnecessária a aplicação do teste soma dos postos de Wilcoxon (não-paramétrico). O teste aplicado foi o T-Student, que é um teste paramétrico. Para isso, assim como nos outros testes, é necessária a formulação das hipóteses nula e alternativa, e esta formulação está na Tabela 10.

Tabela 10. T-Student – Hipóteses nula e alternativa – experimentos 1 a 12

Hipótese	Descrição
Nula	O desempenho da configuração da MLP mostrada no experimento 4 e das configurações mostradas nos outros experimentos é estatisticamente igual.
Alternativa	O desempenho da configuração da MLP mostrada no experimento 4 é maior que o desempenho das outras configurações.

Feito isso, a execução do teste paramétrico foi iniciada. Mais detalhes sobre a execução destes testes podem ser visualizados no Apêndice D Resultado dos testes estatísticos. Apesar do experimento 4 ter a melhor taxa de acerto, o teste T-Student revelou que todas as configurações da MLP abordadas neste trabalho tiveram, estatisticamente, o mesmo desempenho. Mas mesmo o desempenho de todas as configurações sendo praticamente o mesmo, a que foi utilizada no experimento 4 continua sendo a melhor, pelo fato de utilizar menos neurônios na camada escondida (23), o que diminui o custo computacional quando comparada às outras. Os *scripts* para execução dos testes paramétricos podem ser encontrados no Apêndice E *Script* utilizado para a execução das análises estatísticas no *software* R.

4.3.3 Comparação com outros resultados disponíveis na literatura

Foram realizadas buscas nas principais bases de material científico existentes, como IEEEExplore, ACM Digital Library e Springer Link, utilizando as seguintes *strings*:

- MIMIC + II + Neural + Networks;
- MIMIC + II + Classification;
- Decision + Support + Systems + MIMIC + II;
- MIMIC + II + Diagnostic + Neural + Network; e,
- Decision + Support + Systems + Diagnostic + Neural + Network + Intensive + Care + Unit.

Não foram encontrados trabalhos que se propusessem a fazer o que foi proposto no presente trabalho. Dois, que foram considerados relevantes, estarão descritos abaixo.

O primeiro, realizado por LEE e MARK (2010) [23], utilizou redes neurais MLP para previsão de episódios hipotensivos em UTIs. Já o segundo, realizado por ROCHA *et al.* (2010) [35], também utilizou redes neurais MLP para previsão de episódios de hipotensão aguda. Ambos os trabalhos utilizaram a base de dados MIMIC-II. Os resultados dos dois trabalhos em comparação os resultados deste estão descritos na Tabela 11.

Tabela 11. Comparação com resultados encontrados na literatura

Presente trabalho	LEE e MARK	ROCHA <i>et al.</i>
88,09%	87,3%	94%

Vale ressaltar que os três trabalhos são aplicados a problemas diferentes, mas todos utilizam a RNA *Multilayer perceptron* (MLP).

Capítulo 5

Conclusões e Trabalhos Futuros

Este capítulo apresenta as conclusões obtidas após a realização deste trabalho, bem como uma lista de trabalhos futuros que tem como objetivo melhorar a ferramenta preliminar proposta.

5.1 Conclusões

O objetivo deste trabalho foi construir uma ferramenta, prova de conceito, que fosse capaz de fornecer diagnósticos em tempo real para pacientes internados em ambientes de UTIs, melhorando a qualidade do atendimento prestado pelo profissional de saúde que esteja responsável pelo leito, bem como o conforto do paciente neste ambiente.

Os resultados obtidos nos experimentos mostraram que a ferramenta pode contribuir de forma significativa no processo de tomada de decisão pela equipe de saúde que esteja responsável pelo leito de UTI, reduzindo o impacto que o cansaço e estresse podem ter no processo de tomada de decisão.

Além disso, a ferramenta é capaz de se adaptar a diferentes pacientes pois, para a geração dos diagnósticos, a ferramenta toma como base o histórico de saúde do paciente, encontrando quais os sinais vitais daquele paciente que indicam a normalidade do seu estado de saúde.

Outra importante contribuição desta ferramenta é a possibilidade de se realizar uma análise semiótica dos resultados, a fim de melhorar ainda mais o processo de tomada de decisão. Na ferramenta, um novo protocolo de enfermagem poderá ser implantado no que diz respeito à tomada de decisão nos diagnósticos de enfermagem e na interpretação das respostas obtidas pela ferramenta.

5.2 Trabalhos Futuros

- Implementação de outras abordagens de rede neural visando comparar e eventualmente substituir a que está sendo usada na ferramenta;
- Verificar a possibilidade de variar a quantidade de saídas da rede neural (de acordo com a quantidade de diagnósticos aprendidos pela rede);
- Melhorar o processo de adaptação da ferramenta aos pacientes que estejam sendo monitorados;
- Investigar a possibilidade de serem utilizadas outras Redes Neurais Artificiais em conjunto com a MLP;
- Investigar a possibilidade de serem utilizados mais de um algoritmo de treinamento;
- Investigar a possibilidade de serem utilizadas outras técnicas de inteligência computacional para melhorar as respostas obtidas pela rede;
- Aplicar a ferramenta proposta nos mesmos cenários analisados por [23] e [35], a fim de comparar os resultados obtidos pela ferramenta proposta e pelas análises realizadas pelos dois trabalhos;
- Proposição a partir dos resultados com o uso da ferramenta computacional de uma metodologia para uma assistência mais individualizada para os pacientes internados em Unidades de Terapia Intensiva;
- Melhorar os diagnósticos de enfermagem fornecidos pela ferramenta, o que irá gerar menor demanda para o profissional de saúde, ressaltando o papel de um sistema de apoio à decisão;
- Modelagem de uma sistematização de assistência de enfermagem menos hermética e pautada na individualização do paciente.

Bibliografia

- [1] ANCP – Academia Nacional de Cuidados Paleativos. Manual de cuidados paliativos. 1ª Edição. ed. Diagraphic, 2009. 340p.
- [2] BRENNAN, T. A., LEAPE, L. L., LAIRD, N. M., HEBERT, L., LOCALIO, A. R., LAWTHERS, A. G., NEWHOUSE, J. P., WEILER, P. C., HIATT, H. H. Incidence of adverse events and negligence in hospitalized patients. Qual Saf Health Care, Abril 2004. 145–152.
- [3] BRUNNER L. S., SUDDARTH D. S., SMELTZER S. C. O. C., BARE B. G., HINKLE, J. L., CHEEVER, K. H. Brunner & Suddarth's Textbook of Medical-Surgical Nursing. Lippincott Williams & Wilkins, 2010. 2240 p.
- [4] Comparing Neural Networks in Neuroph, Encog and JOONE. Disponível em: <<http://www.codeproject.com/Articles/85385/Comparing-Neural-Networks-in-Neuroph-Encog-and-JOO>>. Acesso em 22 de junho de 2014
- [5] CREMESP – Conselho Regional de Medicina do Estado de São Paulo. Disponível em: <<http://www.medicinaintensiva.com.br/cremesp.htm>>. Acesso em 13 de junho de 2014.
- [6] CYBENKO, George. Approximation by superpositions of a sigmoid function. Mathematics of Control. Signals and Systems, 2:303-314, 1989.
- [7] CYBENKO, George. Continuous valued neural networks with two hidden layers are sufficient. Technical report. Department of Computer Science, Tufts University, 1988.
- [8] Encog AI Framework. Disponível em: <<http://www.heatonresearch.com/encog>>. Acesso em 22 de junho de 2014.
- [9] Enfermagem em Unidade de Terapia Intensiva (UTI). Disponível em: <<http://www.pucsp.br/pos-graduacao/especializacao-e-mba/enfermagem-em-unidade-de-terapia-intensiva-uti>>. Acesso em 22 de Abril de 2014.
- [10] Frequência Cardíaca. Disponível em: <<http://www.portaleducacao.com.br/educacao/artigos/32331/frequencia-cardiaca-fc>>. Acesso em 21 de junho de 2014.

- [11] Frequência Respiratória. Disponível em: <<http://www.portaleducacao.com.br/educacao/artigos/32335/frequencia-respiratoria-fr>>. Acesso em 21 de junho de 2014.
- [12] GALVÃO, C. M., SAWADA, N. O., MENDES, I. A. C. A busca das melhores evidências. Rev Esc Enferm, São Paulo, v.37, n.4, p.43-50, 2003.
- [13] GUYTON, A. C., HALL, J. E. Tratado de Fisiologia Médica. 12ª ed. Elsevier, 2011. 1176 p.
- [14] HIRDES, J. P., POSS, J. W., CURTIN-TELEGDI, N. The Method for Assigning Priority Levels (MAPLe) A new decision-support system for allocating home care resources. BMC Medicine, Março 2008.
- [15] HUDAK, C., GALLO, B. M. Cuidados intensivos de enfermagem: uma abordagem holística. 6ª ed. Rio de Janeiro: Guanabara Koogan; 1997
- [16] HUDSON, D. L., COHEN, M. E. Multidimensional Medical Decision Making. 31st Annual International Conference of the IEEE EMBS, Minnesota, 2009. 3405-3408.
- [17] Iatrogenia. Disponível em: <<http://pt.wikipedia.org/wiki/Iatrogenia>>. Acesso em 22 de Abril de 2014.
- [18] JURISTO, N.; MORENO M., A. Basics of Software Engineering Experimentation. [S.l.]: Kluwer Academic Publishers, 2001
- [19] KADER, M. F. B., BARBOSA, S. F. Sistema de apoio à decisão para a realização do exame físico da gestante: um procedimento eletrônico de aprendizagem em enfermagem. 2009. Trabalho apresentado ao 15. Congresso Internacional ABED de Educação a Distância, Fortaleza, 2009.
- [20] KHAIRAT, S., GONG, Y. Enhancing Patient Safety through Clinical Communication Knowledge Representation. IEEE 13th International Conference on e-Health Networking, Applications and Services, 2011. 130-133.
- [21] KOHN, L. T., CORRIGAN, J. M., DONALDSON, M. S. To err is human: Building a safer health system. Committee on Quality of Health Care in America, Institute of Medicine, Washington D.C. National Academy Press, 2000.

- [22] LAUTERT, L. A sobrecarga de trabalho na percepção de enfermeiras que trabalham em hospital. *Revista Gaúcha de Enfermagem*, Porto Alegre, v.20, n.2, p.50-64, jul. 1999.
- [23] LEE, J., MARK, R. G. A Hypotensive Episode Predictor for Intensive Care based on Heart Rate and Blood Pressure Time Series. *Computing in Cardiology*. 2010. 37:81–84.
- [24] MIMIC II: Clinical Database Overview. Disponível em: <http://physionet.org/mimic2/mimic2_clinical_overview.shtml>. Acesso em 19 de Abril de 2014.
- [25] McCULLOCH, W., PITTS, W. A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. 1943. 5:115-133.
- [26] MENDES, T. N. C. UTI – Passado, presente e futuro. Atividade apresentada ao curso de enfermagem. CESIM – Centro de Estudos Superiores de Itapecuru-Mirim, Maranhão.
- [27] OLIVEIRA, F. R. S. Novas abordagens para diálogos flexíveis em sistemas de apoio à decisão inteligentes. Universidade de Pernambuco - [S.I.]. 2009.
- [28] Pacientes na UTI. Disponível em: <<http://www.mdsaude.com/2008/10/pacientes-na-uti.html>>. Acesso em 21 de junho de 2014.
- [29] Quais os perigos da baixa saturação do oxigênio. Disponível em: <http://www.ehow.com.br/quais-perigos-baixa-saturacao-oxigenio-info_41517/>. Acesso em 21 de junho de 2014.
- [30] Regras de Treinamento. Disponível em: <<http://pt.slideshare.net/jkbrandao/redes-neurais-artificiais-regras-de-aprendizado>>. Acesso em 14 de junho de 2014.
- [31] Resilient Propagation. Disponível em: <<http://en.wikipedia.org/wiki/Rprop>>. Acesso em 21 de junho de 2014.
- [32] Resilient Propagation. Disponível em: <http://www.heatonresearch.com/wiki/Resilient_Propagation>. Acesso em 21 de junho de 2014.

- [33] RIBEIRO, K. R. B. O sofrimento do paciente na UTI: escutando a sua vivência. Dissertação de mestrado. Universidade Federal do Rio Grande do Norte. 2009.
- [34] RIEDMILLER, M., BRAUN, H. Rprop - A Fast Adaptive Learning Algorithm. Proceedings of the International Symposium on Computer and Information Science VII, 1992.
- [35] ROCHA, T., PAREDES, S., CARVALHO, P., HENRIQUES, J., HARRIS, M. Wavelet based Time Series Forecast with Application to Acute Hypotensive Episodes Prediction. 32nd Annual International Conference of the IEEE EMBS, Buenos Aires, 2010. 2403-2406.
- [36] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations* Volume 1: Foundations, MIT Press, Cambridge, MA.
- [37] SANTAELLA, L. O que é Semiótica. São Paulo, Brasiliense, 2012.
- [38] SANTAELLA, L. *Semiótica Aplicada*. São Paulo. Cengage Learning, 2002.
- [39] SEVERO, G. C., GIRARDON-PERLINI, N., M., O. Estar internado em Unidade de Terapia Intensiva: percepção de pacientes. *Scientia Medica*, Porto Alegre: PUCRS, v. 15, n. 1, jan/mar 2005.
- [40] Sinais Vitais. Disponível em: <<http://aenfermagem.com.br/procedimentos/sinais-vitais/>>. Acesso em 21 de junho de 2014.
- [41] TURBAN, E. *Support Systems and Expert Systems*. 4ª Edição, Prentice-Hall International Editions. New Jersey, USA, 1995.
- [42] UTI – Unidade de Terapia Intensiva. Disponível em: <<http://www.hu.ufsc.br/uti/paciente.html>>. Acesso em 22 de Abril de 2014.
- [43] VALENÇA, M. J. S. *Fundamentos das Redes Neurais: Exemplos em java*. 2ª Edição. ed. Olinda: Livro Rápido, 2010. 310 p.
- [44] WERBOS, P. J. *Beyond Regression: New tools for prediction and analysis in behavioral sciences*. Tese (Doutorado) Harvard University, 1974.

- [45] ZORZETTO, R. Números em revisão. Disponível em: <<http://revistapesquisa.fapesp.br/en/2012/02/23/n%C3%BAmeros-em-revis%C3%A3o-3/>>. Acesso em 07 de junho de 2014.

Apêndice A

Regras para a definição dos diagnósticos

Neste apêndice, estão descritas as regras para a geração dos diagnósticos de acordo com os sinais vitais do paciente [13].

Tabela 12. Regras para definição dos diagnósticos

Diagnóstico	Condição
Taquicardia	frequência cardíaca > MAX_FREQ_CARD
Bradycardia	frequência cardíaca < MIN_FREQ_CARD
Hipertermia	temperatura > MAX_TEMP
Hipotermia	temperatura < MIN_TEMP
Bradipnéia	frequência respiratória < MIN_FREQ_RESP
Taquipnéia	frequência respiratória > MAX_FREQ_RESP
Infarto agudo do miocárdio	(Taquicardia) && (PAM > MAX_PAM)
Choque anafilático	(Bradycardia) && (PAM < MIN_PAM)
Insuficiência respiratória	(Bradycardia) && (saturação do oxigênio < MIN_SAT_O2)
Edema pulmonar	(Taquicardia) && (saturação do oxigênio < MIN_SAT_O2)
Insuficiência Cardíaca Congestiva	(Bradycardia) && (PAM > MAX_PAM)
DPOC	(saturação do oxigênio < MIN_SAT_O2) && (PAM < MIN_PAM)
Pico hipertensivo	(Taquipnéia) && (PAM > MAX_PAM)
Apnéia	(Bradipnéia) && (saturação do oxigênio < MIN_SAT_O2)
Sepse	(temperatura > MAX_TEMPERATURA) && (saturação do oxigênio < MIN_SAT_O2)
Hemorragia	(temperatura < MIN_TEMPERATURA) && (saturação do oxigênio < MIN_SAT_O2)

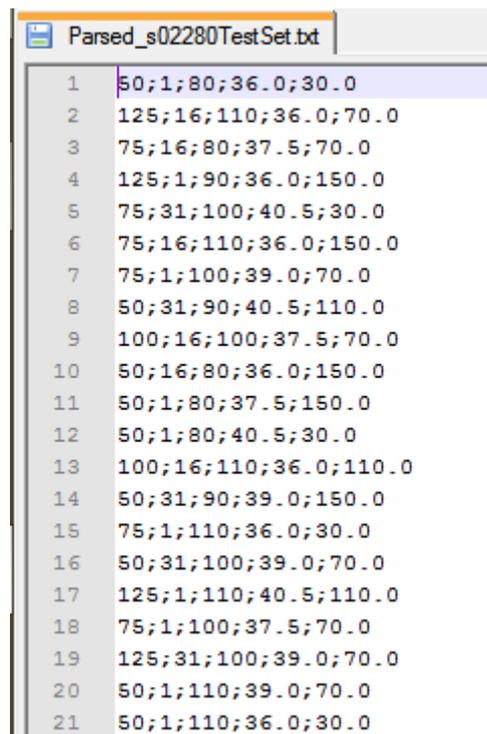
Onde:

- DPOC – Doença pulmonar obstrutiva crônica
- MAX_FREQ_CARD – Frequência cardíaca máxima do paciente monitorado
- MAX_FREQ_RESP – Frequência respiratória máxima do paciente monitorado
- MAX_PAM – Pressão arterial média máxima do paciente monitorado
- MAX_SAT_O2 – Saturação do oxigênio máxima do paciente monitorado
- MAX_TEMPERATURA – Temperatura máxima do paciente monitorado
- MIN_FREQ_CARD – Frequência cardíaca mínima do paciente monitorado
- MIN_FREQ_RESP – Frequência respiratória mínima do paciente monitorado
- MIN_PAM – Pressão arterial média mínima do paciente monitorado
- MIN_SAT_O2 – Saturação do oxigênio mínima do paciente monitorado
- MIN_TEMPERATURA – Temperatura mínima do paciente monitorado
- PAM – Pressão Arterial Média

Apêndice B

Exemplo de arquivo CSV

Neste apêndice está um exemplo de um arquivo CSV utilizado no decorrer do desenvolvimento da ferramenta.



The image shows a screenshot of a text editor window titled "Parsed_s02280TestSet.txt". The window contains a list of 21 rows, each representing a patient's vital signs. The data is formatted as a CSV file with five columns separated by semicolons. The first column contains row numbers from 1 to 21. The subsequent columns contain numerical values representing different vital signs.

Row	Value 1	Value 2	Value 3	Value 4	Value 5
1	50	1	80	36.0	30.0
2	125	16	110	36.0	70.0
3	75	16	80	37.5	70.0
4	125	1	90	36.0	150.0
5	75	31	100	40.5	30.0
6	75	16	110	36.0	150.0
7	75	1	100	39.0	70.0
8	50	31	90	40.5	110.0
9	100	16	100	37.5	70.0
10	50	16	80	36.0	150.0
11	50	1	80	37.5	150.0
12	50	1	80	40.5	30.0
13	100	16	110	36.0	110.0
14	50	31	90	39.0	150.0
15	75	1	110	36.0	30.0
16	50	31	100	39.0	70.0
17	125	1	110	40.5	110.0
18	75	1	100	37.5	70.0
19	125	31	100	39.0	70.0
20	50	1	110	39.0	70.0
21	50	1	110	36.0	30.0

Figura 18. Exemplo de arquivo CSV contendo informações sobre os sinais vitais do paciente

Apêndice C

Detalhes dos experimentos

Neste apêndice, estão descritos, em detalhes, a configuração de cada experimento realizado. A melhor configuração está em destaque.

Tabela 13. Detalhamento dos experimentos

Experimento	Função de Ativação	Algoritmo de Treinamento	Quantidade de Neurônios
1	TANH	<i>Resilient propagation</i>	23
2	TANH	<i>Resilient propagation</i>	35
3	TANH	<i>Resilient propagation</i>	47
4	Sigmoide Logística	<i>Resilient propagation</i>	23
5	Sigmoide Logística	<i>Resilient propagation</i>	35
6	Sigmoide Logística	<i>Resilient propagation</i>	47
7	TANH	<i>Back propagation</i>	23
8	TANH	<i>Back propagation</i>	35
9	TANH	<i>Back propagation</i>	47
10	Sigmoide Logística	<i>Back propagation</i>	23
11	Sigmoide Logística	<i>Back propagation</i>	35
12	Sigmoide Logística	<i>Back propagation</i>	47

Apêndice D

Resultado dos testes estatísticos

Neste apêndice, encontra-se os resultados das execuções dos testes estatísticos de Shapiro-Wilk, F e T-Student.

<pre>> shapiro.test(experimento1) Shapiro-Wilk normality test data: experimento1 W = 0.9774, p-value = 0.7537</pre>	<pre>> shapiro.test(experimento2) Shapiro-Wilk normality test data: experimento2 W = 0.9832, p-value = 0.9023</pre>	<pre>> shapiro.test(experimento3) Shapiro-Wilk normality test data: experimento3 W = 0.9632, p-value = 0.3738</pre>
<pre>> shapiro.test(experimento4) Shapiro-Wilk normality test data: experimento4 W = 0.9428, p-value = 0.1082</pre>	<pre>> shapiro.test(experimento5) Shapiro-Wilk normality test data: experimento5 W = 0.9687, p-value = 0.5049</pre>	<pre>> shapiro.test(experimento6) Shapiro-Wilk normality test data: experimento6 W = 0.9765, p-value = 0.7267</pre>
<pre>> shapiro.test(experimento7) Shapiro-Wilk normality test data: experimento7 W = 0.947, p-value = 0.1401</pre>	<pre>> shapiro.test(experimento8) Shapiro-Wilk normality test data: experimento8 W = 0.9101, p-value = 0.01498</pre>	<pre>> shapiro.test(experimento9) Shapiro-Wilk normality test data: experimento9 W = 0.9738, p-value = 0.6487</pre>
<pre>> shapiro.test(experimento10) Shapiro-Wilk normality test data: experimento10 W = 0.9662, p-value = 0.4403</pre>	<pre>> shapiro.test(experimento11) Shapiro-Wilk normality test data: experimento11 W = 0.9602, p-value = 0.3131</pre>	<pre>> shapiro.test(experimento12) Shapiro-Wilk normality test data: experimento12 W = 0.9494, p-value = 0.163</pre>

Figura 19. Resultados do teste de Shapiro-Wilk

<pre>> var.test(experimento4,experimento1) F test to compare two variances data: experimento4 and experimento1 F = 1.216, num df = 29, denom df = 29, p-value = 0.6019 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.5787793 2.5548379 sample estimates: ratio of variances 1.216013</pre>	<pre>> var.test(experimento4,experimento2) F test to compare two variances data: experimento4 and experimento2 F = 0.7493, num df = 29, denom df = 29, p-value = 0.4419 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.3566569 1.5743489 sample estimates: ratio of variances 0.7493346</pre>
<pre>> var.test(experimento4,experimento3) F test to compare two variances data: experimento4 and experimento3 F = 0.7864, num df = 29, denom df = 29, p-value = 0.5218 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.3743155 1.6522974 sample estimates: ratio of variances 0.7864354</pre>	<pre>> var.test(experimento4,experimento5) F test to compare two variances data: experimento4 and experimento5 F = 0.9817, num df = 29, denom df = 29, p-value = 0.9606 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.4672386 2.0624768 sample estimates: ratio of variances 0.9816663</pre>
<pre>> var.test(experimento4,experimento6) F test to compare two variances data: experimento4 and experimento6 F = 1.4194, num df = 29, denom df = 29, p-value = 0.351 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.6756011 2.9822271 sample estimates: ratio of variances 1.419435</pre>	<pre>> var.test(experimento4,experimento7) F test to compare two variances data: experimento4 and experimento7 F = 0.7656, num df = 29, denom df = 29, p-value = 0.4764 alternative hypothesis: true ratio of variances is not equal to 1 95 percent confidence interval: 0.3643856 1.6084649 sample estimates: ratio of variances 0.7655726</pre>

Figura 20. Resultados do Teste F - Experimentos 1 à 7

```

> var.test(experimento4,experimento8)

      F test to compare two variances

data:  experimento4 and experimento8
F = 0.9042, num df = 29, denom df = 29, p-value = 0.7882
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4303894 1.8998177
sample estimates:
ratio of variances
 0.9042463

> var.test(experimento4,experimento9)

      F test to compare two variances

data:  experimento4 and experimento9
F = 1.0341, num df = 29, denom df = 29, p-value = 0.9288
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.4921737 2.1725450
sample estimates:
ratio of variances
 1.034055

> var.test(experimento4,experimento10)

      F test to compare two variances

data:  experimento4 and experimento10
F = 0.7383, num df = 29, denom df = 29, p-value = 0.4189
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3514077 1.5511782
sample estimates:
ratio of variances
 0.7383062

> var.test(experimento4,experimento11)

      F test to compare two variances

data:  experimento4 and experimento11
F = 1.0812, num df = 29, denom df = 29, p-value = 0.8349
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5146219 2.2716353
sample estimates:
ratio of variances
 1.081218

> var.test(experimento4,experimento12)

      F test to compare two variances

data:  experimento4 and experimento12
F = 0.7684, num df = 29, denom df = 29, p-value = 0.4825
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3657289 1.6143943
sample estimates:
ratio of variances
 0.7683948

```

Figura 21. Resultados do teste F - Experimentos 8 à 12

<pre>> t.test(experimento4,experimento1) Welch Two Sample t-test data: experimento4 and experimento1 t = -1.4139, df = 57.454, p-value = 0.1628 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.002158119 0.000371583 sample estimates: mean of x mean of y 0.04383355 0.04472682</pre>	<pre>> t.test(experimento4,experimento2) Welch Two Sample t-test data: experimento4 and experimento2 t = -1.2921, df = 56.833, p-value = 0.2016 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0023558252 0.0005080419 sample estimates: mean of x mean of y 0.04383355 0.04475744</pre>
<pre>> t.test(experimento4,experimento3) Welch Two Sample t-test data: experimento4 and experimento3 t = -1.2202, df = 57.183, p-value = 0.2274 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0022729257 0.0005516859 sample estimates: mean of x mean of y 0.04383355 0.04469417</pre>	<pre>> t.test(experimento4,experimento5) Welch Two Sample t-test data: experimento4 and experimento5 t = 0.4807, df = 57.995, p-value = 0.6326 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.001011362 0.001650579 sample estimates: mean of x mean of y 0.04383355 0.04351394</pre>
<pre>> t.test(experimento4,experimento6) Welch Two Sample t-test data: experimento4 and experimento6 t = 0.2244, df = 56.308, p-value = 0.8233 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.001086695 0.001360912 sample estimates: mean of x mean of y 0.04383355 0.04369644</pre>	<pre>> t.test(experimento4,experimento7) Welch Two Sample t-test data: experimento4 and experimento7 t = -1.7922, df = 56.995, p-value = 0.07842 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0026968095 0.0001494669 sample estimates: mean of x mean of y 0.04383355 0.04510722</pre>

Figura 22. Resultados do teste T-Student - Experimentos 1 à 7

<pre>> t.test(experimento4,experimento8) Welch Two Sample t-test data: experimento4 and experimento8 t = -0.4108, df = 57.854, p-value = 0.6828 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.001638451 0.001080529 sample estimates: mean of x mean of y 0.04383355 0.04411251</pre>	<pre>> t.test(experimento4,experimento9) Welch Two Sample t-test data: experimento4 and experimento9 t = -1.7943, df = 57.984, p-value = 0.07797 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0024915753 0.0001361294 sample estimates: mean of x mean of y 0.04383355 0.04501127</pre>
<pre>> t.test(experimento4,experimento10) Welch Two Sample t-test data: experimento4 and experimento10 t = -1.8424, df = 56.715, p-value = 0.07065 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0027610885 0.0001151105 sample estimates: mean of x mean of y 0.04383355 0.04515654</pre>	<pre>> t.test(experimento4,experimento11) Welch Two Sample t-test data: experimento4 and experimento11 t = -0.5848, df = 57.912, p-value = 0.561 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.0016794072 0.0009200377 sample estimates: mean of x mean of y 0.04383355 0.04421323</pre>
<pre>> t.test(experimento4,experimento12) Welch Two Sample t-test data: experimento4 and experimento12 t = -0.566, df = 57.022, p-value = 0.5736 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.001823496 0.001019789 sample estimates: mean of x mean of y 0.04383355 0.04423540</pre>	

Figura 23. Resultados do teste T-Student - Experimento 8 à 12

Apêndice E

***Script* utilizado para a execução das análises estatísticas no software R**

Neste apêndice, está o *script* que foi utilizado para a execução das análises estatísticas no *software* R.

```
dados<-read.table("dados.csv", row.names=NULL, sep=";")
experimento1<-as.numeric(dados[1,])
experimento2<-as.numeric(dados[2,])
experimento3<-as.numeric(dados[3,])
experimento4<-as.numeric(dados[4,])
experimento5<-as.numeric(dados[5,])
experimento6<-as.numeric(dados[6,])
experimento7<-as.numeric(dados[7,])
experimento8<-as.numeric(dados[8,])
experimento9<-as.numeric(dados[9,])
experimento10<-as.numeric(dados[10,])
experimento11<-as.numeric(dados[11,])
experimento12<-as.numeric(dados[12,])
#Teste Shapiro-Wilk
shapiro.test(experimento1)
shapiro.test(experimento2)
shapiro.test(experimento3)
shapiro.test(experimento4)
shapiro.test(experimento5)
shapiro.test(experimento6)
shapiro.test(experimento7)
shapiro.test(experimento8)
shapiro.test(experimento9)
shapiro.test(experimento10)
shapiro.test(experimento11)
shapiro.test(experimento12)
```

```
#Teste F
var.test(experimento4, experimento1)
var.test(experimento4, experimento2)
var.test(experimento4, experimento3)
var.test(experimento4, experimento5)
var.test(experimento4, experimento6)
var.test(experimento4, experimento7)
var.test(experimento4, experimento8)
var.test(experimento4, experimento9)
var.test(experimento4, experimento10)
var.test(experimento4, experimento11)
var.test(experimento4, experimento12)

#Teste T-Student
t.test(experimento4, experimento1)
t.test(experimento4, experimento2)
t.test(experimento4, experimento3)
t.test(experimento4, experimento5)
t.test(experimento4, experimento6)
t.test(experimento4, experimento7)
t.test(experimento4, experimento8)
t.test(experimento4, experimento9)
t.test(experimento4, experimento10)
t.test(experimento4, experimento11)
t.test(experimento4, experimento12)
```