

UNIVERSIDADE FEDERAL DO PARANÁ

WALDEMAR VOLANSKI

**DESENVOLVIMENTO E VALIDAÇÃO DE SISTEMA DE APOIO À  
DECISÃO EM URINÁLISE COM INTELIGÊNCIA ARTIFICIAL  
UTILIZANDO REDES NEURONAIS**

**CURITIBA**

**2011**

WALDEMAR VOLANSKI

**DESENVOLVIMENTO E VALIDAÇÃO DE SISTEMA DE APOIO À  
DECISÃO EM URINÁLISE COM INTELIGÊNCIA ARTIFICIAL  
UTILIZANDO REDES NEURONAIIS**

Dissertação (Mestrado) - Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, Programa de Pós-Graduação em Bioinformática, Área de Concentração: Bioinformática.

Orientador: Prof. Dr. Geraldo Picheth  
Co-orientador: Prof. Dr. Roberto Tadeu Raittz

**CURITIBA**

**2011**

VOLANSKI, Waldemar  
DESENVOLVIMENTO E VALIDAÇÃO DE SISTEMA DE APOIO À DECISÃO EM URINÁLISE  
COM INTELIGÊNCIA ARTIFICIAL UTILIZANDO REDES NEURONAIS  
Waldemar Volanski- Curitiba, 2011.

96 f.: il.; 30 cm.

Orientador Prof. Dr. Geraldo Picheth  
Co-orientador Prof. Dr. Roberto Tadeu Raitz

Dissertação (Mestrado) - Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, Programa de Pós-Graduação em Bioinformática, Área de Concentração: Bioinformática.

Inclui bibliografia

1. Segunda opinião. 2. Redes neuronais. 3. Análise de urina. 4. Urinálise.  
I. Picheth, Geraldo. II. Raitz, Roberto Tadeu. III. Universidade Federal do Paraná. IV. Título.  
CDD 616.07566

TERMO DE APROVAÇÃO

WALDEMAR VOLANSKI

**“Desenvolvimento e validação de sistema de apoio à decisão em Urinálise com Inteligência Artificial utilizando redes neurais”**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Bioinformática, pelo Programa de Pós-graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná, pela seguinte banca examinadora:


Orientador:

  
Prof Dr Geraldo Picheth

Coorientador:

  
Prof Dr Roberto Tadeu Raittz

  
Prof. Dr. Paulo Afonso Bracarense Costa  
Universidade Federal do Paraná

  
Profª Drª Laura Lúcia Cogo  
Universidade Federal do Paraná

Curitiba, 03 de fevereiro de 2011

*Dedico a minha namorada Fabiola Borges da Silva,  
que suportou pacientemente meu exílio aceitando  
os poucos momentos que pudemos estar juntos.*

## **AGRADECIMENTOS**

teste Agradeço a todas as pessoas que de alguma forma me ajudaram na realização deste trabalho, não podendo deixar de reconhecer a ajuda dos Professores e amigos Guilherme Souza Cavalcanti de Albuquerque e Ives Sbalqueiro que me confiaram as cartas de apresentação.

Aos meus orientadores Prof. Dr. Geraldo Picheth e Prof. Dr. Roberto Tadeu Raittz, que com competência, paciência, determinação e sabedoria conduziram este trabalho.

Ao programa de mestrado em Bioinformática pelo acolhimento, representado pelas secretárias Mariana Marques Pereira, Lea Regina Welter e Suzana de Azevedo Gobetti.

Ao Dr. Francisco Antonio Marçalo pessoa com quem fiz iniciação científica, e que infelizmente não se encontra mais conosco.

Ao amigo Farmacêutico Bioquímico Claudio Henrique de Mello Gonçalves que com seu espírito científico aceitou participar na realização do trabalho como segundo analista.

Ao amigo Professor. Ivo Queiroz pela ajuda e incentivo dispensado em todo esse tempo.

À minha querida amiga Anilda Gomes da Silva pelo companheirismo e a torcida dedicada a mim.

A grande amiga Dra. Eliane Ribeiro Carmes pela doação incondicional de sua amizade e conhecimento.

Aos Professores Fabio de Oliveira Pedrosa, Maria Berenice Reynaud Steffens, Jeroniza Nunes Marchaukoski, Leda Satie Chubatsu, Emanuel Maltempi de Souza, Giselle Munhoz Alves, Liu Un Rigo, Leonardo Magalhães Cruz, Paulo Afonso Bracarense Costa, Luiz Antônio Pereira Neves e Lucas Ferrari de Oliveira pela contribuição.

Aos colegas Ademir Luiz Prado, Daniela Tapety, Dieval Guizelini, Guilherme

Willemann, Leandro Steim, Luiz Fernando Soares, Rosa Cordélli, Terumi Kamada, por terem contribuído com este trabalho.

Ao Raphael de Assumpção Jamielniak, pela colaboração e pelo convívio durante a realização do seu Trabalho de Conclusão de Curso de Tecnologia em Sistemas de Informação da Universidade Federal do Paraná pela grande contribuição dispensada.

Aos colegas do programa de Pós-graduação em Bioquímica clínica pela convivência agradável e pela troca de informações científicas.

À minha querida mãe Maria Eli Volanski, não por ter me concebido ou por dedicar todo o seu amor, mas sim por ter me alfabetizado e me ensinado as primeiras lições de ciência.

*“De cualquier modo, yo formo parte de los sujetos sintientes y pensantes, capaces de establecer una relación con los rayos solares, y de interpretar y valorar las percepciones y las ilusiones.”*

---

(CALVINO, 1986)Mr.Palomar



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>2</b>	<b>OBJETIVO E JUSTIFICATIVA</b>	<b>21</b>
2.1	OBJETIVO GERAL	21
2.2	OBJETIVOS ESPECÍFICOS	21
2.2.1	Selecionar ensaios	21
2.2.2	Desenvolver programa computacional	21
2.2.3	Compatibilizar o software a ser desenvolvido	21
2.2.4	Validar o software a ser desenvolvido	21
2.2.5	Analisar o desempenho das redes neuronais em estudo	21
2.3	JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO	21
<b>3</b>	<b>REVISÃO DE LITERATURA</b>	<b>23</b>
3.1	SISTEMA DE APOIO À DECISÃO	23
3.2	SEGUNDA OPINIÃO	23
3.3	AUTOVALIDAÇÃO	24
3.4	NEURÔNIO BIOLÓGICO	24
3.5	NEURÔNIO ARTIFICIAL	26
3.6	HISTÓRICO DAS REDES NEURONAIS	27
3.7	TECNOLOGIA REDES NEURONAIS	30
3.8	ESTRUTURA DAS REDES NEURONAIS	32
<b>4</b>	<b>MATERIAIS E MÉTODOS</b>	<b>34</b>
4.1	AMOSTRA	34
4.1.1	Classificação da amostra	34
4.2	ANÁLISE URINÁRIA	35
4.2.1	Análise Química da Urina	35
4.2.2	Citologia urinária	35
4.2.3	Parâmetros analisados na urina e valores de referência	35
4.3	PROGRAMAS COMPUTACIONAIS	37
4.3.1	Treinamento e validação das redes neuronais	37
4.3.2	Redes Neuronais	38
4.4	PROGRAMA COMPUTACIONAL DESENVOLVIDO	38
4.5	ANÁLISES ESTATÍSTICA	39
<b>5</b>	<b>RESULTADOS</b>	<b>41</b>
5.1	AMOSTRA	41
5.2	LIBERAÇÃO DE RESULTADOS POR PROFISSIONAIS	44
5.3	REDES NEURONAIS	46
5.4	DESEMPENHO DOS SISTEMAS DE LIBERAÇÃO DE RESULTADOS	49
<b>6</b>	<b>DISCUSSÃO</b>	<b>52</b>
6.1	AMOSTRA	53
6.2	LIBERAÇÃO DE RESULTADOS POR PROFISSIONAIS MANUALMENTE	56
6.3	REDES NEURONAIS	57
6.4	DESEMPENHO DOS SISTEMAS DE LIBERAÇÃO DE RESULTADOS	59

<b>7 CONCLUSÕES .....</b>	<b>62</b>
<b>APÊNDICE A – INTERFACE GRÁFICA .....</b>	<b>70</b>
<b>APÊNDICE B – PASSO A PASSO PARA O SISTEMA DE APOIO À DECISÃO.....</b>	<b>71</b>
<b>ANEXO A – SAD.....</b>	<b>76</b>

## LISTA DE FIGURAS

FIGURA 1 – PINTURAS ANTIGAS MOSTRANDO ANÁLISE DE URINA	.....	17
FIGURA 2 – EXEMPLO DE TIRA REAGENTE PARA ANÁLISES QUÍMICAS		19
FIGURA 3 – DESENHO ESQUEMÁTICO DO NEURÔNIO BIOLÓGICO	.....	25
FIGURA 4 – FOTOGRAFIA MICROSCÓPICA DE NEURÔNIO BIOLÓGICO	.	26
FIGURA 5 – NEURÔNIO DE 5 CAMADAS	.....	32
FIGURA 6 – INTERFACE GRÁFICA DO SOFTWARE BRAILE	.....	39
FIGURA 7 – HISTOGRAMAS DA VARIÁVEL IDADE	.....	43
FIGURA 8 – TESTES COM 34.292 AMOSTRAS	.....	46
FIGURA 9 – TESTES COM 17.200 AMOSTRAS	.....	47
FIGURA 10– DESEMPENHO DA REDE MLP COM N=34.292 E N=17.200	...	48
FIGURA 11– INTERFACE GRÁFICA	.....	70

## LISTA DE TABELAS

TABELA 1	– ENSAIOS REALIZADOS E VALORES DE REFERÊNCIA	.....	36
TABELA 2	– PARÂMETROS UTILIZADOS NA AVALIAÇÃO	.....	39
TABELA 3	– DISTRIBUIÇÃO NORMAL DOS PARÂMETROS EM ESTUDO		40
TABELA 4	– CARACTERÍSTICAS DA AMOSTRA EM ESTUDO	.....	42
TABELA 5	– CARACTERÍSTICAS DEMOGRÁFICAS E LABORATORIAIS DAS AMOSTRAS	.....	45
TABELA 6	– DESEMPENHO DA VALIDAÇÃO DA REDE MLP COM DIFERENTES BASES	.....	49
TABELA 7	– DIVERGÊNCIAS ENTRE REDES MLP, FAN E AVALIADORES HUMANOS	.....	49
TABELA 8	– DESEMPENHO DE AVALIADORES HUMANOS E REDES NEURONAIS	.....	50
TABELA 9	– DESEMPENHO DOS PROCESSOS DE LIBERAÇÃO DE RESULTADOS	.....	50
TABELA 10	– COMPARAÇÃO ENTRE AS CARACTERÍSTICAS DAS REDES MLP E FAN	.....	51

## LISTA DE SIGLAS E ABREVIATURAS

**AC** Antes de Cristo

**ADALINE** Adaptive Linear Element

**ALE** Adaptive Linear Element

**ART** Adaptive Resonance Theory

**CAD** Computer-aided diagnosis

**CLSI** Clinical and Laboratory Standards Institute

**DARPA** Defense Advanced Research Projects Agency

**DSS** Decision Support System

**EAS** Elementos Anormais em Sedimento

**ECU** Exame Comum de Urina

**EQU** Exame Químico de Urina

**FAN** Free Associative Neurons

**IA** Inteligência Artificial

**IEEE** Institute of Electrical and Electronics Engineers

**INNS** International Neural Networks Society

**LMC** Laboratório Municipal de Curitiba

**MADALINE** Multiple Adaline

**MIT** Massachusetts Institute of Technology

**MLP** Multi Layer Perceptron

**NCCLS** National Committee for Clinical Laboratory Standards

**OCR** Optical Character Recognition

**PEAS** Pesquisa dos Elementos Anormais em Sedimento

**pH** Potencial hidrogenianico

**RBF** Radial basis functions

**RNA** Rede Neuronal Artificial

**RNAs** Redes Neurais Artificiais

**SAD** Sistema de Apoio à Decisão

**SVM** Support Vector Machine

**TAT** Turnaround Time

**UFPR** Universidade Federal do Paraná

## RESUMO

A urinálise contempla uma avaliação semi-quantitativa de analitos e contagem diferencial de células presentes na urina. A análise fornece informações diagnósticas relevantes principalmente sobre o sistema urinário e renal. Neste trabalho, foi estudada a aplicação de redes neurais aplicadas a tomada de decisão sobre a liberação do ensaio de urinálise, comparada ao processo de decisão realizados por profissionais do laboratório clínico. Registros de resultados de urinálise (n=140.482), anônimos, foram obtidos no Laboratório Municipal de Curitiba, após a aprovação da pesquisa pelo Comitê de Ética em Pesquisa com Seres Humanos da UFPR (CAE: 0106.0.091.000-09). Os pacientes em estudo apresentaram idade média de 38 anos, com amplitude de variação de 0 a 101 anos. A predominância do sexo feminino foi superior a 65% na amostra. Quando os elementos do ensaio foram analisados individualmente, alterações nas concentrações de esterase leucocitária e hemoglobina, bem como as contagens de leucócitos e eritrócitos estão presentes em cerca de 95% de todos os pacientes com alterações na urinálise. Aplicando-se classificação da amostra baseada em valores de referência, cerca de 65% dos registros mostraram no mínimo um parâmetro urinário fora dos valores de referência. Quando a mesma amostra foi classificada por profissionais, que consideram além dos valores de referência, a consistência entre ensaios relacionados (esterase leucocitária-contagem de leucócitos; hemoglobina-contagem de eritrócitos) para a liberação dos ensaios, cerca de 30% dos resultados ficaram “retidos” para análises posteriores de verificação devido a inconsistência nos resultados. Duas redes neurais (MLP, multilayer perceptron e FAN, free associative neurons) foram treinadas, testadas e validadas, utilizando como critérios de grupos os resultados dos profissionais designados “liberados” e “retidos”. O treinamento das redes mostrou que para a rede MLP o banco de dados deve ser “balanceado”, números equilibrados de registros “liberados” e “retidos” para a obtenção do melhor desempenho. Para permitir o acesso ao banco de dados e facilitar a obtenção de resultados um software de apoio foi desenvolvido e utilizado em todo o trabalho. As redes neurais em estudo foram avaliadas com uma amostra de 484 pacientes da rotina, avaliado por dois profissionais. A rede MLP apresentou uma divergência de cerca de 9 a 12% (sensibilidade 100%; especificidade 90,1%) e com os profissionais, significativamente menor que a rede FAN, 17 a 19% (sensibilidade 39,0% e especificidade 86,9%), respectivamente. A comparação entre o desempenho das redes neurais com o banco de dados, superior a 100.000 registros, mostrou que a rede MLP apresentou desempenho compatível ao dos profissionais (29,5% vs 21,6%, respectivamente) na classificação dos ensaios retidos. A rede FAN reteve menos amostras (18,2%) sugerindo a maior divergência com a resposta dos profissionais. Em síntese, é possível afirmar que a rede neuronal MLP (Multilayer Perceptrons) validada com uma base de dados “balanceada” permite identificar padrões liberação e retenção de resultados de forma a mimetizar os critérios de profissionais para o ensaio de urinálise. Também o programa computacional em linguagem Python, utilizando a rede MLP,

desenvolvido para acessar o banco de dados e classificar a amostra apresentou bom desempenho e segurança, sendo recomendado para uso no laboratório clínico.

**Palavras-chave:** segunda opinião, redes neuronais, análise de urina, urinálise, MLP, FAN.



## ABSTRACT

The urinalysis assay represents a combination of semi-quantitative analysis of the differential cells counts in urine. This analysis provides relevant information mainly on the renal and urinary systems. The aim of the work was to compare the study of neural networks associated with laboratory decision support systems, against the process of clinical laboratory professionals' decisions. A de-identified clinical database (140.482 records) was obtained from *Laboratório Municipal de Curitiba*. The research was approved by UFPR Ethical Committee for Research with Humans (CAE: 0106.0.091.000-09). The sample population showed mean age of 38 years with an age range of 0-101 years. Females predominated at more than 65%. When individual urinalysis components were quantified the concentration of leucocytes esterase and haemoglobin, as well as the count of leucocytes and erythrocytes occurred in about 95% of all pathologic cases. When reference value was applied to the sample classification, about 65% of all cases were outside the reference values by at least one parameter. Laboratory clinical professionals often consider beyond the reference values: the consistency of the associated parameters (leucocytes esterase concentration vs. leucocytes count; haemoglobin concentration vs. erythrocyte count) on the criteria for the "release" or "hold" of the result. When the sample was analyzed by professionals about 30% was "held" for additional investigation due to inconsistency in results. Two neural networks Multi Layer Perceptron (MLP) and Free Associative Neurons (FAN) were trained, tested and validated, applying the laboratory professional's criteria for result "release" or "hold". The MLP network showed that the training with a "balanced" data set (characterized by the some number of groups "release" and "hold") improved performance . To access the database and facilitate the work software was developed and utilized throughout the entire project. The two studied networks were evaluated with a sample of 484 routine patients, previously screened by the professionals. The MLP network showed a discrepancy of about 9-12% (sensitivity 100%; specificity 91.1%) when compared to professionals. These results were significantly lower that those obtained using the network FAN (17-19%) which showed a sensitivity of 39% and a specificity of 89.9%. The comparison of the performance of the studied neural networks with a large data set ( $n > 100.000$  registers) revealed that the MLP network produced a performance closer to those of professionals (21.6% vs. 29.5%, respectively) for the detection of "hold" samples. In summary we can highlight that MLP neural network, validated with a "balanced" data set, was able to identify the pattern of urinalysis results released, comparably with the laboratory professionals .Also, the software developed (written in Python) for the MLP network showed a good performance and safety and thus we recommended it's application. in the clinical laboratory.

**Keywords:** second opinion, neural network, urine analysis, urinalysis, MLP network, FAN network.

# 1 INTRODUÇÃO

O estudo das características dos componentes presentes na urina, urinálise, está presente ao longo da história da humanidade, a Figura 1 mostra pinturas e afrescos antigos retratando a urinálise. A análise da urina com o objetivo diagnóstico é um dos ensaios mais realizados e antigos associado ao laboratório clínico. A prática da análise de urina é um dos mais antigos exames realizados no homem. Já no ano 300 Antes de Cristo (AC), Hipócrates afirmava que não há sistema que ofereça mais informações que o sistema urinário.



**FIGURA 1:** PINTURAS ANTIGAS E AFRESCOS MOSTRANDO A ANÁLISE DE URINA FONTE: (STRASINGER *et al.*, 2008)

A análise da urina realizada tanto por processos automatizados, quanto manuais, cuja finalidade é obter diagnóstico ou prognóstico de estados fisiológicos ou patológicos, tornando-se um dos métodos mais comuns empregado pelas análises clínicas, por revelar, principalmente, as condições morfofisiológicas do sistema excretor, bem como as condições metabólicas dos pacientes submetidos a esta análise. O exame de urina é relevante no apoio ao diagnóstico de vários estados patológicos e sua coleta é simples, não invasiva e indolor, o que propicia ao setor de urinálise do laboratório clínico um grande fluxo de amostras. O exame de urina é conhecido com várias denominações, tais como, rotina de urina, sumário de urina, urina do tipo 1, urina parcial, parcial de urina, Elementos Anormais

em Sedimento (EAS), Exame Químico de Urina (EQU), Exame Comum de Urina (ECU) e Pesquisa dos Elementos Anormais em Sedimento (PEAS). O *Clinical and Laboratory Standards Institute* (CLSI), previamente designado *National Committee for Clinical Laboratory Standards* (NCCLS), recomenda a análise de urina (urinálise) no auxílio ao diagnóstico de doenças, rastreio de doença assintomática na população e acompanhamento da evolução da doença, bem como a eficácia da terapêutica. A urinálise compreende as análises qualitativa e quantitativa da urina, realizada nas amostras em três etapas, a saber:

- Exame Físico;
- Exame químico;
- Exame microscópico.

O exame físico-químico é a primeira etapa da urinálise, objetivando a obtenção de dados concernentes aos aspectos físicos da amostra analisada, como cor, odor, aspecto. Os aspectos químicos contemplam pH, densidade, presença de substâncias, como glicose, corpos cetônicos, proteínas, bilirrubina, urobilinogênio e nitrito. Esta análise é feita utilizando fitas reagentes ilustradas na Figura 2. Os resultados obtidos podem revelar informações importantes sobre o sistema urinário, metabolismo, presença de infecções e outras patologias sistêmicas que podem acometer o indivíduo. As tiras reagentes empregadas na realização deste exame contêm agentes químicos ou enzimáticos que reagem com os analitos urinários de forma específica. A análise físico-química pode ser realizada manualmente, ou através de equipamentos automatizados desenvolvidos para esta finalidade. A apreciação final deve ser acompanhada da análise das estruturas biológicas que podem estar presentes na urina, e que é realizada posteriormente.

A análise microscópica do sedimento urinário tem como objetivo identificar e quantificar os elementos presentes na amostra, como células epiteliais, eritrócitos, leucócitos, cristais, cilindros, bactérias. Durante muitas décadas esta etapa da análise urinária era realizada exclusivamente por profissionais analistas clínicos através de microscopia. No presente estão disponíveis equipamentos automatizados que identificam e quantificam os elementos urinários através de citometria de fluxo ou por exame de imagem. Estes analisadores automáticos dedicados a urinálise estão ainda restritos a laboratórios que apresentam grande demanda de ensaios, decorrente do alto custo dos equipamentos e insumos para as análises.



**FIGURA 2:** EXEMPLO DE TIRA REAGENTE PARA ANÁLISES QUÍMICAS SEMI-QUANTITATIVA DA URINA FABRICADO POR ROCHE DIAGNOSTICAS FONTE: (RENAL CANCER, 2010)

O laudo do exame de urinálise é composto pelos elementos físico-químicos, e biológicos (microscopia) e necessita ser validado e assinado pelo analista clínico antes da liberação.

O elevado volume de solicitações do exame de urinálise exige dos profissionais analistas clínicos que atuam na rotina laboratorial muita agilidade, eficiência e responsabilidade, o que pode ocasionar o estresse do profissional, principalmente quando atrelado ao sobrecarga de trabalho. A tarefa de validar cada exame individualmente em uma grande rotina laboratorial é cansativa, redundante e expõem o analista a possíveis equívocos durante o processo de validação.

O desenvolvimento e disponibilidade de tecnologias de interfaceamento que capturam os resultados dos analisadores automáticos após a identificação quantitativa e qualitativa dos exames proporcionou maior segurança na liberação dos resultados, e permitiu o de grande número de amostras de forma rápida.

Porém, ainda há carência de ferramentas eficientes que auxiliem os profissionais do laboratório na análise de consistência dos resultados obtidos e conseqüentemente facilitem e acelerem a liberação do laudo.

A proposta deste trabalho é construir ferramentas para apoio à decisão para liberação de exames de urinálise com o desenvolvimento de um programa computacional (*software*) utilizando redes neurais.

A abordagem da aplicação de redes neurais para auxílio à liberação de resultados, também conhecida como “segunda opinião”, original em sua proposta para urinálise, tem como fundamento, procurar emular na resposta eletrônica a ação

e decisão dos profissionais do laboratório clínico. Neste contexto, é esperado que os resultados do software proposto se aproximem dos critérios de decisão humana e com isto tenha aderência dos profissionais e uso extensivo na rotina.

## 2 OBJETIVO E JUSTIFICATIVA

### 2.1 OBJETIVO GERAL

Desenvolver e validar programa computacional com base em rede neuronal para apoio à tomada de decisão em procedimentos de urinálise.

### 2.2 OBJETIVOS ESPECÍFICOS

- 2.2.1 Selecionar ensaios de urinálise com controle de qualidade estabelecido a serem utilizados como modelo para estudo e validação de redes neuronais,
- 2.2.2 Desenvolver programa computacional, empregando rede neuronal para o apoio à decisão em programas de controle de qualidade, com processo de decisão automatizado e interface intuitiva
- 2.2.3 Compatibilizar o *software* a ser desenvolvido com os protocolos já em uso pelo laboratório e segurança na informação,
- 2.2.4 Validar o *software* a ser desenvolvido por comparação dos resultados obtidos com o critério de decisão manual de profissionais do laboratório clínico (padrão ouro) e decisão automática baseada em sistema de redes neuronais,
- 2.2.5 Analisar o desempenho das redes neuronais em estudo.

### 2.3 JUSTIFICATIVA E RELEVÂNCIA DO TRABALHO

A liberação dos resultados por profissionais, tradição do laboratório clínico, está associado à boa qualidade e confiabilidade do laudo. Este processo de liberação por profissionais, no entanto não esta livre de erros e inconsistências. É reconhecidamente difícil padronizar todos os critérios de liberação entre profissionais, em especial aqueles em que a interpretação requer múltiplas análises comparativas.

Outro elemento relevante é o cansaço e a perda de atenção dos profissionais após

analisarem e liberarem grande número de resultados, o que favorece o aparecimento de erros.

Este estudo propõe o desenvolvimento de um programa computacional com base em redes neuronais como apoio a decisão. A tecnologia proposta permitirá mimetizar os critérios de liberação e oferecerá aos analistas do laboratório uma decisão eletrônica ou “segunda opinião”.

Verificada e aceita a eficácia do sistema, os profissionais terão disponíveis na tela do computador no momento da liberação, em tempo real, a informação com o julgamento da “liberação eletrônica”.

Ensaio recomendado para “liberação” pelo sistema eletrônico, poderão ser liberados mais rapidamente pelo analista ampliando sua produtividade, reduzindo o estresse e eliminando os possíveis erros. Em contrapartida, ensaios, que por alguma razão foram retidos pelo programa poderão ter atenção particular dos analistas na procura de inconsistência, ou discrepâncias observadas nas amostras analisadas.

É possível prever que a evolução dos programas computacionais permitirá em futuro próximo uma interação mais intensa e decisiva na liberação dos resultados laboratoriais, e este projeto é o primeiro passo para o futuro das análises clínicas.

## 3 REVISÃO DE LITERATURA

### 3.1 SISTEMA DE APOIO À DECISÃO

O termo Sistema de Apoio à Decisão (SAD), abreviado em português, e em inglês *Decision Support System* (DSS), refere-se a um processo informatizado que reúne e apresenta dados de uma ampla variedade de fontes e normalmente são utilizados para fins comerciais onde, sistemas e subsistemas podem ajudar as pessoas a tomar decisões baseadas em dados que são obtidos a partir de uma ampla variedade de fontes (MELO *et al.*, 2006).

### 3.2 SEGUNDA OPINIÃO

O diagnóstico auxiliado por computador ou *Computer-aided diagnosis* (CAD) pode ser definido como um diagnóstico realizado por um profissional, por exemplo, um radiologista, que utiliza o resultado de análises quantitativas automatizadas de imagens radiográficas processadas por programas computacionais como uma segunda opinião para a tomada de decisões diagnósticas (HAWKER, 2007). É relevante ressaltar que a informação computacional tem sido utilizada como ferramenta de informação adicional, sendo diagnóstico final sempre feito pelo profissional. Este processo diferencia claramente o conceito básico de diagnóstico auxiliado por computador do conceito de diagnóstico automatizado, que foi um proposto e estudado nas décadas de 60 e 70 (LISBOA, 2002).

A finalidade do CAD é melhorar a acurácia do diagnóstico, assim como a consistência da interpretação da imagem radiológica, mediante o uso da resposta do computador como referência (QUINLAN, 1987). Esta resposta pode ser útil, uma vez que o diagnóstico do profissional é baseado em avaliações subjetivas, estando sujeito a variações intra e interpessoais, bem como a perda de informação devido à natureza sutil do achado diagnóstico, baixa qualidade da imagem, sobreposição de estruturas,



fadiga visual ou distração. Além disso, foi demonstrado que uma dupla leitura por dois profissionais pode aumentar a sensibilidade do diagnóstico. A proposta do CAD é funcionar como um segundo especialista (MARQUES; PAULO, 2001).

### 3.3 AUTOVALIDAÇÃO

A autoverificação ou autovalidação é um processo em que o software de computador avalia automaticamente os resultados do exame laboratorial, para após verificar a integridade do mesmo, liberar os resultados sem qualquer intervenção de humana (BATES; GAWANDE, 2003). Os critérios de liberação são definidos pelo usuário e podem ser personalizados.

O processo de desenvolvimento e implementação da autoverificação é viável para a maioria dos resultados de exames em laboratório clínico (GOLDSCHMIDT, 2002). O uso de Autoverificação melhora a qualidade do resultado, o aumento receitas e o *Turnaround Time* (TAT). Em conjunto, este processo promove uma melhor satisfação do cliente. Computadores e programas de autoverificação são a ferramentas para a sobrevivência no mercado competitivo em serviços de diagnósticos laboratoriais (JOHNSTONE, 2004).

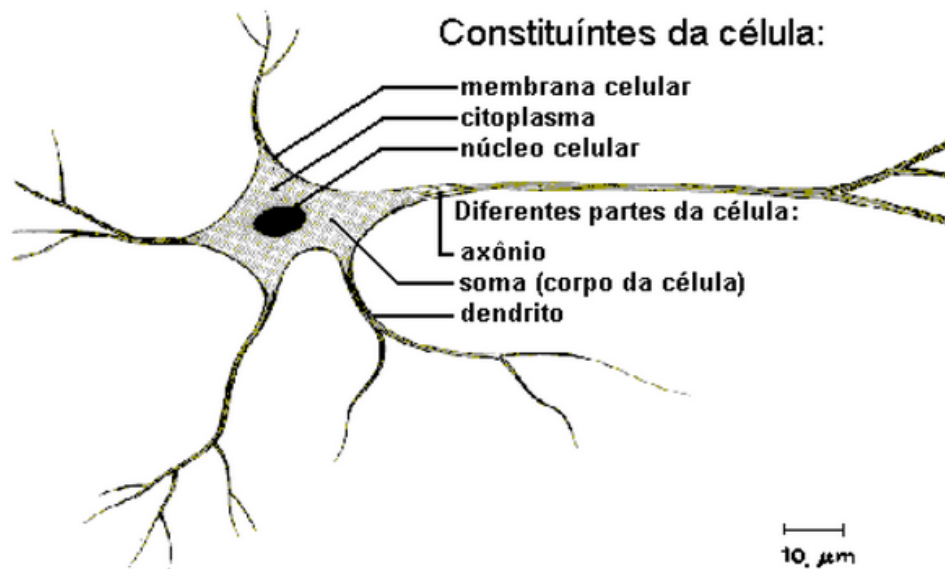
### 3.4 NEURÔNIO BIOLÓGICO

O cérebro humano, componente do sistema nervoso, é considerado o mais fascinante processador baseado em carbono existente, sendo composto por células nervosas, os neurônios, que estão diretamente relacionados a todas as funções nobres e movimentos do organismo e cuja estrutura está exemplificada nas Figuras 3 e 4. Os neurônios estão conectados uns aos outros através de sinapses, constituindo uma grande rede, a rede neuronal ou rede neural. As sinapses transmitem os estímulos a todo o corpo, e a rede neuronal proporciona uma fabulosa capacidade de processamento e armazenamento de informações. (GUYTON et al,2006)

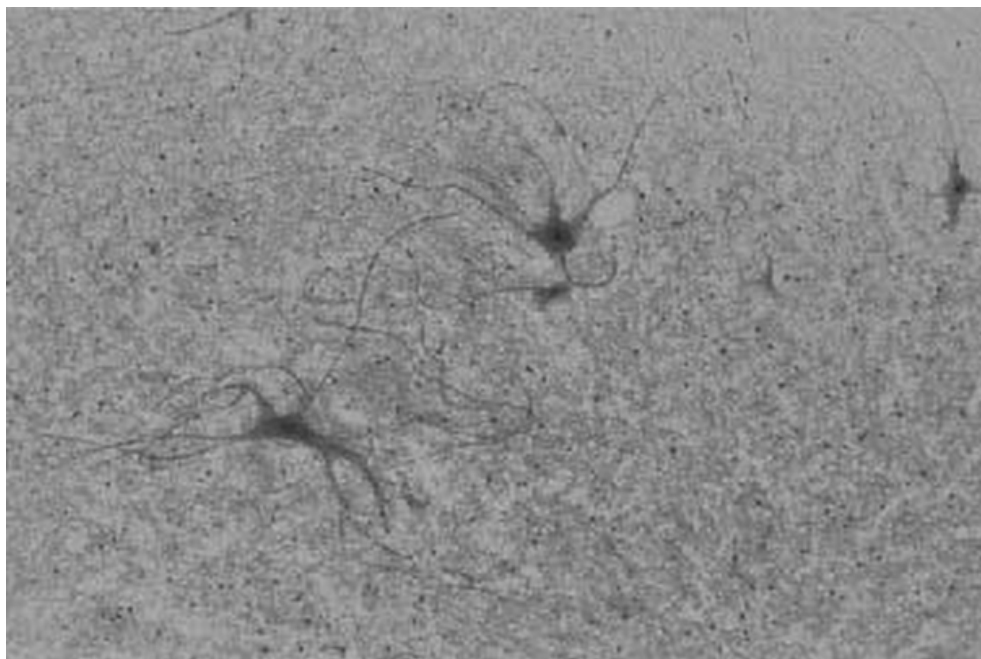
O conjunto extremamente complexo de neurônios forma o sistema nervoso, e sua fisiologia está baseada na comunicação realizada através de impulsos nervosos entre os neurônios. Quando um impulso nervoso é recebido, um neurônio o processa, e transmite um limite de ação, que dispara um segundo impulso, resultando na liberação de uma substância neurotransmissora, a acetilcolina, que flui do corpo celular para o axônio, que pode, ou não, estar conectado a um dendrito de outra

célula. O neurônio transmissor do pulso pode controlar tal frequência, aumentando ou diminuindo a polaridade na membrana pós-sináptica, desempenhando papel essencial na determinação do funcionamento, comportamento e do raciocínio do ser humano. O potencial de repouso de um neurônio pode ser influenciado pelas sinapses em volta dele. As redes neuronais naturais não transmitem sinais negativos, pois sua ativação é medida pela frequência com que emitem pulsos, e no geral, não são uniformes, apresentando uniformidade apenas em alguns pontos do organismo. Seus pulsos não são síncronos ou assíncronos, devido ao fato de não serem contínuos. (GRAUPE, 2007)

Os principais componentes dos neurônios são os dendritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios, mais o corpo do neurônio, também chamado de soma, que é responsável por coletar e combinar informações vindas de outros neurônios e o axônio, constituído de uma fibra tubular que pode alcançar até alguns metros, é responsável por transmitir os estímulos para outras células (FERNEDA, 2006).



**FIGURA 3:** DESENHO ESQUEMÁTICO DAS PARTES CONSTITUINTES DE UMA CÉLULA NEURONAL FONTE: (FERNEDA, 2006)



**FIGURA 4:** FOTOGRAFIA MICROSCÓPICA DE CORTE HISTOLÓGICO MOSTRANDO UM NEURÔNIO  
FONTE:(KÜHNEL, 2003)

### 3.5 NEURÔNIO ARTIFICIAL

O neurônio artificial que também é conhecido como *perceptron*, assim como o seu inspirador, o neurônio natural, possui várias ou conexões sinápticas. Cada conexão possui um valor associado a um peso, onde o conhecimento da rede é adquirido e armazenado pela atualização de cálculos (RUSSELL; NORVIG, 2009).

O *perceptron* conta também com ligações com o ambiente externo. Para a ligação com o ambiente externo existem as entradas, que são conhecidas como *inputs* e para a conexão de saída *output*, que pode ser ligado a outro *perceptron*. A rede neuronal artificial é semelhante à estrutura cerebral, sendo composta de um conjunto de vários neurônios que poderá ser simulado com a ajuda de um computador. A propriedade da rede neural artificial é o fato de mudar as conexões entre duas células conforme a aprendizagem, simulando a capacidade do cérebro de abstrair, e reagir às situações novas, permitindo diferentes aplicações, desde jogos, à tomada de decisão.

As redes neuronais artificiais transmitem sinais negativos, uniformes. Os pulsos são contínuos, síncronos e diferem das redes naturais onde o neurônio artificial possui várias entradas podendo receber sinais de outros neurônios, sinais de fora da rede ou de entradas fixas. A função de ativação *activation function* fornece então o estado de ativado, que além das entradas também é dependente do estado anterior armazenado

do neurônio. O estado de ativação *activation state* resultante determina qual saída o neurônio produzirá. A saída em si é definida pela função de saída *output function* fornecendo um valor exato (MCCULLOCH; PITTS, 1943).

### 3.6 HISTÓRICO DAS REDES NEURONAIS

O conceito de RNA foi desenvolvido em 1943, quando Warren McCulloch, neurofisiologista e o matemático Walter Pitts, da Universidade de Illinois, fizeram uma analogia entre as células nervosas e os processos eletrônicos num artigo publicado no *Bulletin of Mathematical Biophysics* com o título, *A Logical Calculus of the Ideas Immanent in Nervous Activity* (MCCULLOCH; PITTS, 1943)

Em 1949, o biólogo e psicólogo Donald Olding Hebb, que estudava o comportamento dos animais, autor do livro *The Organization of Behavior: A neuropsychological approach*, reforçaram a teoria de que o condicionamento psicológico estava presente em qualquer animal, por ser esta uma propriedade de neurônios individuais (HEBB, 1949).

Os conceitos de Hebb não foram pioneiros, mas ele propôs um princípio de aprendizado em sistemas nervosos complexos, ou seja, uma lei que descreve o funcionamento quantitativo da sinapse e do processo do treinamento humano. Desde então, outros pesquisadores voltaram-se para esta linha de pesquisa.

Na década de 1950, o conceito de filtros lineares adaptativos usada em processamento de sinais foi desenvolvida por Gabor e em 1956 no *Dartmouth College, Hanover, EUA*, os dois paradigmas da Inteligência Artificial, o simbólico e o conexionista foram estabelecidos. A Inteligência Artificial Simbólica tenta simular o comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal. Já a Inteligência Artificial Conexionista acredita que construindo um sistema que simule a estrutura do cérebro, este adquire inteligência, ou seja, será capaz de aprender, assimilar, errar e aprender com seus erros (ROSENBLATT, 1958).

Em 1951, Marvin Minsky, co-fundador do Laboratório de Inteligência Artificial do *Massachusetts Institute of Technology* (MIT), construiu o Snark, um simulador de cadeia neuronal (KELEMEN, 2007). O Snark era funcional e podia ajustar seus pesos sinápticos automaticamente. Este programa nunca chegou a executar alguma função de processamento de informação relevante, servindo somente de fator motivador para novas ideias que surgiram posteriormente.

Na conferência de 1956, na Dartmouth Conference de 1956, foi apresentado o primeiro modelo de rede neuronal artificial pelo pesquisador da IBM Nathaniel Rochester. O modelo consistia numa simulação de centenas de neurônios interconectados através de um sistema que verificaria como a rede responderia aos estímulos ambientais.

Em 1959 Frank Rosenblatt na Universidade de Cornell, Nova Iorque, EUA, criou uma rede de múltiplos neurônios do tipo discriminadores lineares e denominando-a MLP. Rosenblatt e colaboradores utilizaram os conceitos de McCulloch para desenvolverem o seu modelo matemático de sinapse humana. Estes trabalhos e as múltiplas contribuições em pesquisa e desenvolvimento técnico fizeram o nome de Rosenblatt ser considerado por muitos o fundador da ciência da neurocomputação (DAYHOFF; DELEO, 2001).

Em 1969 Marvin Minsky e Seymour Papert lançaram em uma obra chamada *Perceptron*, na qual demonstravam que o modelo apresentado por Rosenblatt não era promissor, e as limitações do perceptron em resolver certos problemas devido ao uso de técnicas empíricas, das grandes dificuldades da matemática envolvida e dos poucos recursos computacionais disponíveis na época. Onde Minsky e Papert, demonstraram matematicamente que estes modelos, na forma como estavam, não eram capazes de aprender determinadas funções fundamentais para a execução de tarefas que exigiam a tomada de decisão, como por exemplo, a função lógica do “XOR OU Exclusivo”, que possui padrões de valores de entrada e saída cuja associação não poderia ser aprendida pelos modelos baseados em Perceptrons (DAYHOFF; DELEO, 2001).

Tal constatação impactou negativamente as pesquisas que vinham sendo realizadas sobre redes neuronais nas décadas de 1960 e 1970 (LORITZ, 1991).

Nos anos seguintes, muitos artigos foram publicados, e diversas previsões exageradas e pouco confiáveis para a época foram anunciadas. A maioria destas suposições propunha a possibilidade de desenvolvimento de computadores com um poder de raciocínio e/ou processamento igual ou superior ao do cérebro humano.

No início da década de 1980, as pesquisas em redes neuronais foram retomadas e rapidamente expandidas. Foi quando o administrador de programas da DARPA Ira Skurnick resolveu dar atenção às proposições da neurocomputação, contrariando todos os preceitos, fundando em 1983 as pesquisas em neurocomputação da DARPA. Este fato acabou abrindo novos horizontes para a neurocomputação. (ZHANG;

FRIEDRICH, 2003)

Em 1982 John Hopfield, físico e biólogo de reputação mundial também se interessou pela neurocomputação e escreveu vários artigos que levaram cientistas a se unirem nesta nova área emergente. Hopfield reascendeu as pesquisas em neurocomputação, criticando fortemente as teorias apresentadas por Minsky e Papert na década de 60.

A grande evolução neste campo de pesquisa ocorreu em 1986, quando o professor de psicologia da Universidade de Stanford, David E. Rumelhart, e seu colega James L. McClelland, professor de psicologia da Universidade de CarnegieMellon, publicaram o livro *Parallel Distributed Processing: Explorations in the Microstructure of Cognition (vol.1: Foundations, vol.2: Psychological and Biological Models)*. Nesse livro, os autores apresentam um modelo matemático e computacional que propicia o treinamento supervisionado dos neurônios artificiais, descrevendo a *backpropagation*, um algoritmo de otimização global sem restrições (HECHT N, 1989).

Várias aplicações em redes neuronais em MLP vem sendo implementadas para aplicações como, classificação de padrões, reconhecimento óptico de caracteres *Optical Character Recognition* (OCR) e reconhecimento de fala, entre outros (HOPFIELD, 1982; OTTENBACHER *et al.*, 2004). Outros tipos de treinamento de redes têm sido apresentados, como o método de gradiente conjugado, proposto inicialmente por Roger Fletcher e C M Reeves em 1964, utilizado por (LEONARD; KRAMER, 1990), para acelerar o treinamento do *perceptron*, bem como o método de Newton (ROJAS, 1996).

Em 1987 ocorreu a primeira conferência específica para abordar Redes Neuronais. Também foi formada a Sociedade Internacional de Redes Neurais *International Neural Networks Society* (INNS) juntamente com o INNS Journal em 1989, do Neural Computation e do *Institute of Electrical and Electronics Engineers* (IEEE) e Transactions on Neural Networks em 1990.

Também em 1990 Bernard Widrow da Universidade de *Stanford* com a ajuda de estudantes desenvolveu um novo modelo de processamento de redes neuronais chamado de *Adaptive Linear Element* (ADALINE), a qual se destacava pela sua poderosa lei de aprendizado. O princípio de treinamento para as redes ADALINE ficou conhecido como a Regra Delta, que foi mais tarde generalizada para redes com modelos neuronais mais sofisticados. Posteriormente, Widrow criou a *Multiple Adaline* (MADALINE), que era uma generalização multidimensional do ADALINE (GRAUPE,

2007).

A Sociedade Brasileira de Redes Neurais promoveu em 1994 o primeiro congresso de Redes Neurais no Brasil. Atualmente, pelo menos dois congressos brasileiros específicos em redes neuronais são realizados regularmente Congresso Brasileiro de Redes Neurais e Simpósio Brasileiro de Redes Neurais, além de inúmeros eventos internacionais, entre eles o *International Joint Conference on Neural Networks* e o *International Conference on Artificial Neural Networks*, bem como eventos nacionais de abrangência mais geral que contam com seções específicas em redes neuronais.

### 3.7 TECNOLOGIA REDES NEURONAIAS

A tecnologia de Redes Neuronais Artificiais tem como finalidade solucionar problemas envolvendo reconhecimento de padrões com base em um conjunto de informações previamente conhecida (MUSEN *et al.*, 2006). Esse conjunto é dividido em dois outros conjuntos que são os conjuntos de teste e treinamento (eventualmente há um terceiro conjunto de dados utilizados para validação). A rede é então submetida ao treinamento através do primeiro conjunto e para a avaliação deste treinamento é submetido o segundo conjunto, que o classifica conforme seu treinamento. As respostas da rede são comparadas e o grau de acerto e divergência avaliado.

As pesquisas mais recentes em Redes Neuronais Artificiais (RNAs) buscam a compreensão das capacidades da natureza humana, que possibilitam às pessoas a formulação de soluções para problemas que não podem ser resolvidos através dos métodos tradicionais (FRIZE *et al.*, 2001). As RNA's, objetivam solucionar problemas de Inteligência Artificial (IA), modelando sistemas através de circuitos para que possam simular a capacidade de aprender e agir sistema do nervoso humano perante as mais adversas situações, bem como adquirir conhecimento através da experiência e da observação (LISBOA; TAKTAK, 2006).

A simulação da biologia do sistema nervoso é feita pelas redes neuronais artificiais, no que diz respeito ao processamento microscópico, realizado por uma célula do sistema nervoso de forma individual, como macroscopicamente, seja em termos de padrões de interconexão entre elementos de processamento, seja em termos dos comportamentos coletivos que emergem em uma rede biológica com vários neurônios (SAMUELSON; ZECKHAUSER, 1988).

Conforme o paradigma atual em neurofisiologia, a informação é representada

no sistema nervoso através de pulsos elétricos gerados pelos neurônios, que são conhecidos como potenciais de ação. No que se refere ao campo de redes de neurônios artificiais, a forma específica na qual a atividade neuronal é representada no modelo matemático pode assumir diversas simplificações (SIVASANKAR *et al.*, 2009). Isso depende do grau de refinamento na representação de fenômenos biológicos que se tem em mente, o que por sua vez varia grandemente de grupo de pesquisa a grupo de pesquisa, de acordo com a orientação e a ênfase dos seus trabalhos.

Em essência, todos os tipos de redes neuronais apresentam a mesma unidade de processamento em um neurônio artificial, o qual simula o comportamento do neurônio biológico (GOULART *et al.*, 2006).

O neurônio artificial possui várias entradas, que correspondem às conexões sinápticas com outras unidades similares a ele, e uma saída, cujo valor depende diretamente da somatória ponderada de todas as saídas dos outros neurônios a esse conectado. Neste cálculo de integração dos estímulos das várias entradas, a ponderação é dada pelos pesos que cada conexão possui. Assim, a somatória corresponde à saída do modelo neuronal, constantes de ponderação “ $w_i$ ” representam as conexões sinápticas biológicas que intermedeiam a interação entre neurônios vizinhos, “ $x_i$ ” é o estímulo proveniente do neurônio vizinho “ $i$ ”. O resultado “Soma” representa o estímulo global, com as devidas ponderações relativas, recebido pelo neurônio de todos os seus vizinhos.

O neurônio artificial descrito corresponde ao modelo matemático conhecido como *Adaptive Linear Element* (ADALINE). É importante notar que o modelo somente incorpora transformações lineares; como consequência, ele não permite a implementação de computações complexas decorrente da limitação do poder matemático que impedem a representação de relações entre estímulo e resposta não lineares que de fato são observadas com frequência no neurônio biológico. Assim, há limitação do poder de modelagem de sistemas biológicos, onde uma pequena alteração no modelo neuronal é necessária para viabilizar as funcionalidades não lineares, através da introdução de uma função de transferência. Na sua forma mais simples, a função de transferência não linear “ $f$ ” pode ser binária, como no caso do modelo conhecido como *Perceptron* de ROSENBLATT. Neste caso, o modelo tendo uma saída binária permite representar funcionalidades de natureza lógica realizadas pelos neurônios biológicos. É o caso da tomada de decisões que é uma funcionalidade digital por natureza, Por outro lado, para o caso de modelagem de funcionalidades do sistema nervoso em que não apenas grandezas digitais estão mostradas, mas



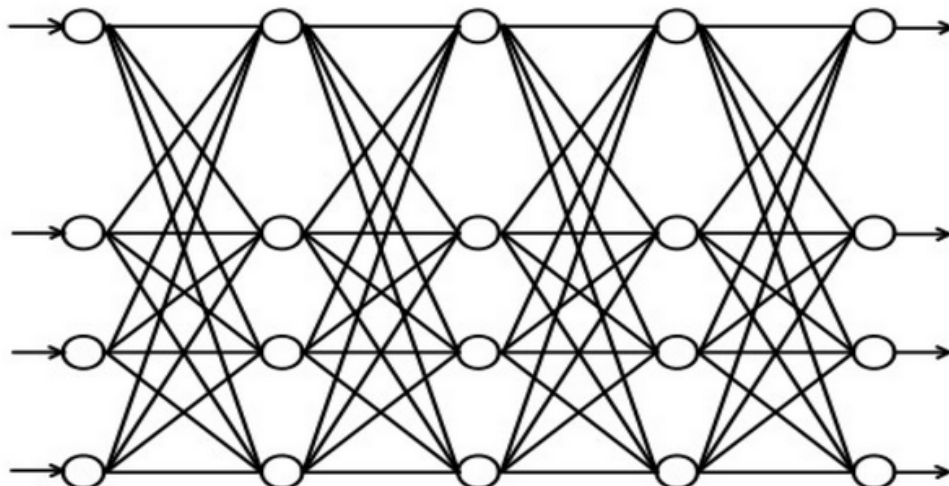
também uma gama onde valores analógicos são necessários para a representação da atividade neuronal, devemos empregar uma função de transferência “f” com saída contínua, como por exemplo a função tangente hiperbólica que neste caso o modelo pode ser chamado de sigmoidal. Em ambos os casos, a ideia é que a função de transferência possa imitar, em algum grau, a relação entre entrada e saída observada nos neurônios biológicos (HECHT N, 1989).

### 3.8 ESTRUTURA DAS REDES NEURONAIIS

As redes do tipo MLP com um único *perceptron* apresentam uma estrutura de cálculo simples, e não atendem a todas as necessidades de classificação de padrões, então a medida que as bases de dados são mais complexas, e que a classificação com separação linear se faz necessária, uma estrutura com diversos neurônios encadeados pode ser empregada (MAZUROWSKI *et al.*, 2008), formando a rede MLP, cujas características principais são:

- Possuem múltiplas entradas / múltiplas saídas / múltiplas camadas;
- Entradas e saídas podem ser tanto analógicas quanto digitais;
- Não ha mais a restrição de separação linear entre classes.

A Figura 5 representa um exemplo da arquitetura da rede tipo MLP com 20 neurônios organizados em 5 camadas. Cada nó (círculo) representa um neurônio e cada linha uma conexão entre saída de um neurônio e uma entrada de neurônio da camada seguinte.



**FIGURA 5:** DESENHO EQUEMÁTICO DE NEURÔNIO DE 5 CAMADAS FONTE:(HAYKIN, 1994)

Na rede representada na Figura 5, cada um dos 20 círculos representa um neurônio artificial, e cada linha representa uma conexão sináptica, com seu peso programável. O fluxo de informação ocorre em camadas de neurônios, sem que haja realimentação (retorno da informação da saída para a entrada de algum dos neurônios da rede). Esta organização em camadas e a ausência de realimentação é própria da arquitetura da rede MLP, não se observando tais características em muitas outras arquiteturas neuronais importantes como, por exemplo, a arquitetura de Hopfield, que é densa em realimentações e não possui qualquer organização em camadas (ROJAS, 1996).

No caso da rede MLP, a função de transferência adotada tipicamente é do tipo sigmoideal (tangente hiperbólica, por exemplo), de forma que com os escalamentos adequados do argumento da função de transferência (ou dos pesos que indiretamente escalam as entradas) é possível passar de mapeamentos entrada-saída aproximadamente lineares a mapeamentos essencialmente digital, sem mudar a função “f” propriamente dita, mas apenas com o recurso de escalamento do argumento. Esta transição entre comportamento analógico e digital é possível porque a função “sinal (x)” pode ser considerada como o caso limite da função tangente hiperbólica com constante escalamento tendendo a infinito (ZHANG; FRIEDRICH, 2003).

Entre as aplicações da rede MLP, podemos citar a classificação de padrões sem separação linear já mencionada, a aproximação de funções genéricas (inclusive analógicas) e a fusão não linear de grandezas analógicas multidimensionais, além da previsão de séries temporais não lineares. O aprendizado através de exemplos da rede MLP permite que as funções descritas sejam realizadas sem a necessidade de um modelo matemático conhecido ou confiável (TANGRI *et al.*, 2008).

## 4 MATERIAIS E MÉTODOS

O projeto, registrado no BANPESQ (2009024103), foi aprovado pelo Comitê de Ética em Pesquisa com Seres Humanos do Setor de Ciências da Saúde da Universidade Federal do Paraná com o número CAE: 0106.0.091.000-09.

### 4.1 AMOSTRA

Foram analisados 140.482 registros aleatórios de pacientes, obtidos no banco de dados do Laboratório Municipal de Curitiba (LMC) no período de março a outubro de 2009, contendo resultados laboratoriais de análise de urina (urinálise). A amostra estudada contempla todo o espectro de pacientes que procuram o serviço do LMC, sem distinção de idade, sexo, etnia ou patologias associadas. O tamanho amostral representa aproximadamente a rotina de urinálise de oito meses do LMC.

#### 4.1.1 Classificação da amostra

As amostras foram classificadas em dois grupos: “Liberado” e “Retidos” conforme critérios de liberação utilizados pelos profissionais do Laboratório Municipal de Curitiba.

O grupo designado “liberado” não apresentava alterações significativas quanto às análises químicas e citológicas realizadas de rotina, considerando os valores de referência ajustados para o sexo dos pacientes. Na presença de alterações dos parâmetros analisados, estes se mostraram coerentes, preservando as associações entre a concentração de esterase leucocitária e contagem de leucócitos, concentração de hemoglobina e contagem de eritrócitos.

O grupo designado “alterado” apresentava no mínimo um parâmetro analisado significativamente fora dos valores de referência estabelecidos, discordância entre as análises ou ambos. O grupo também inclui amostras que não apresentaram

associações entre esterase e leucócitos, hemoglobina e contagem de eritrócitos. Os pacientes deste grupo selecionados pelos profissionais foram analisados separadamente, para citologia microscópica e repetição das análises químicas. Depois de obtida consistência dos dados, estas amostras eram liberadas.

## 4.2 ANÁLISE URINÁRIA

### 4.2.1 Análise Química da Urina

As determinações químicas dos parâmetros urinários foram realizadas no laboratório Municipal de Curitiba, com o sistema automatizado Urisys 2400 (Sysmex, Roche) seguindo as instruções do fabricante do equipamento para calibração e controle do sistema. A metodologia emprega tiras contendo nove áreas reativas que em contato com a urina provem através de reações colorimétricas informações semi-quantitativas dos parâmetros bioquímicos.(HAUSS, 2008)

A Tabela 1 sumariza os ensaios realizados, as faixas de resultados, bem como os valores de referência.

### 4.2.2 Citologia urinária

A quantificação de eritrócitos, leucócitos, cilindros, células epiteliais e bactérias foi determinada por citometria de fluxo utilizando o sistema automatizado UF100 (Sysmex, Roche) seguindo as instruções do fabricante do equipamento para calibração e controle do sistema.

A Tabela 1 sumariza os valores de referência para estes parâmetros.

### 4.2.3 Parâmetros analisados na urina e valores de referência

A Tabela 1 contempla os parâmetros químicos e citológicos analisados, bem como os valores de referências utilizados e critérios reportados nos sistemas automatizados utilizados.

**TABELA 1:** ENSAIOS REALIZADOS E VALORES DE REFERÊNCIA DOS PARÂMETROS URINÁRIOS EM ESTUDO.

Parâmetros	Valores de referência	Intervalos reportados
Análise química (Urisys 2400)*		
pH	4,8-7,4	5 - 6.5 7 8 - 9
Esterase Leucocitária	<10 leucócitos/ $\mu L$	Negativo <10/ $\mu L$ 25 - 100 - 500 Leucócitos/ $\mu L$
Nitrito	0	NEGATIVO = 0 POSITIVO = 1
Proteínas	<10 mg/dL	NEGATIVO <10 mg/dL 25 - 75 - 150 500 mg/dL
Glucose	<30 mg/dL	NORMAL <30mg/dL 50 100 300 1000 mg/dL
Cetonas	<5 mg/dL	NEGATIVO <5mg/dL 5 - 15 - 50 - 150 mg/dL
Urobilinogênio	<1 mg/dL	NORMAL <1 mg/dL 1 - 4 - 8 - 12 mg/dL
Bilirrubina	<0,2 mg/dL	NEGATIVO <0,2 mg/dL 1 - 3 - 6 mg/dL
Hemoglobina	0 - 5 eritrócitos/ $\mu L$	NEGATIVO 0-5/ $\mu L$ 10 - 25 - 50 - 150 - 250 / $\mu L$
Citologia (UF100)**		
Eritrócitos	a: 0 - 16 células/ $\mu L$ b: 0 - 27 células/ $\mu L$	
Leucócitos	0 - 27 células/ $\mu L$	
Cilindros	0	
Células epiteliais	0 - 22 células/ $\mu L$	
Bactérias	0 - 3000 bactérias/ $\mu L$	

\* informares obtidas na bula do kit Urisys 2400 cassete (Roche Diagnostics, cat 3029514(61))

\*\* valores informados no laudo do LMC, estudo próprio.

(a) Masculino

(b) Feminino

### 4.3 PROGRAMAS COMPUTACIONAIS

Foram estudadas as características e desempenho de duas redes neuronais. A rede MLP e FAN. O software MATLAB (MATHWORKS, 2009) que contém as redes neuronais em estudo e promove suporte para os cálculos necessários, foi utilizado na pesquisa.

A configuração da rede MLP está descrita em detalhes no Apêndice B. Para aperfeiçoar a classificação das análises da rede MLP, um software (SAD) com programação em Python, foi desenvolvido pelo mestrando, em conjunto com o Tecnólogo em Sistema de Informação Raphael de Assunção Jamielniak. O software designado SAD (sistema de apoio à decisão) compila as informações da base de dados e promove a classificação dos registros em “normal”, codificado como zero (“0”) e “alterado”, codificado como “um” (“1”), após a rede MLP estar apropriadamente treinada, testada e validada. O script do software está listado no Anexo A.

Um programa com interface gráfica intuitiva que apresenta os dados dos ensaios de urinalise para os profissionais do laboratório do LMC foi desenvolvido para proporcionar a classificação das amostras. Neste *software* os profissionais, após a análise dos vários analitos, informam a classificação do resultado em Liberado (código=0) ou retido (código=1). Os analitos que não se encontram na faixa dos valores de referência estão destacados em vermelho para facilitar a visualização. Em amostras masculinas o nome do paciente está grafado em letra de cor azul e crianças (0-10 anos), o campo referente a idade, está destacado em amarelo. A máscara desta interface é mostrada no Apêndice A.

#### 4.3.1 Treinamento e validação das redes neuronais

O treinamento, validação e testes das redes MLP e FAN foram realizados em duas etapas experimentais.

Em uma primeira etapa, uma sub amostra de 102.876 registros, escolhidas aleatoriamente, foi dividida em três grupos iguais ( $n=34.292$ ) e utilizadas para treinamento, validação e testes (base de dados 1).

No segundo experimento, uma sub amostra de 51.600 registros, contendo o mesmo número de registros liberados e retidos, designada “balanceada” foi dividida em três grupos iguais ( $n=17.200$ ) e utilizada para treinamento, teste e validação (base

de dados 2).

#### 4.3.2 Redes Neurais

Para trabalhar com Redes Neurais MLP, FAN, RBF, SVM foram estabelecidas as condições:

- Validação do sistema: A base de dados foi dividida em três partes iguais, formando a base de treinamento, a base de teste e a base de validação;
- Softwares utilizados para as Redes Neurais: biblioteca ffnet e Python para rede neuronal do tipo MLP, o software EasyFAN (EASYFAN, ) para a rede neuronal do tipo FAN e para as redes neuronais do tipo RBF e SVM foi utilizado o software MATLAB® R2009a.

#### 4.4 PROGRAMA COMPUTACIONAL DESENVOLVIDO

A biblioteca ffnet 0.6.2 (WOJCIECHOWSKI, 2007), foi desenvolvida em Python com funções de portabilidade para FORTRAN, possuindo diferentes algoritmos de otimização de treinamento, e algumas funções auxiliares, como a de algoritmos genéticos. Por ser desenvolvida para fins de pesquisa, não possui muitas facilidades para visualização de resultados ou automatização de processos, sendo então utilizada como parte desta solução.

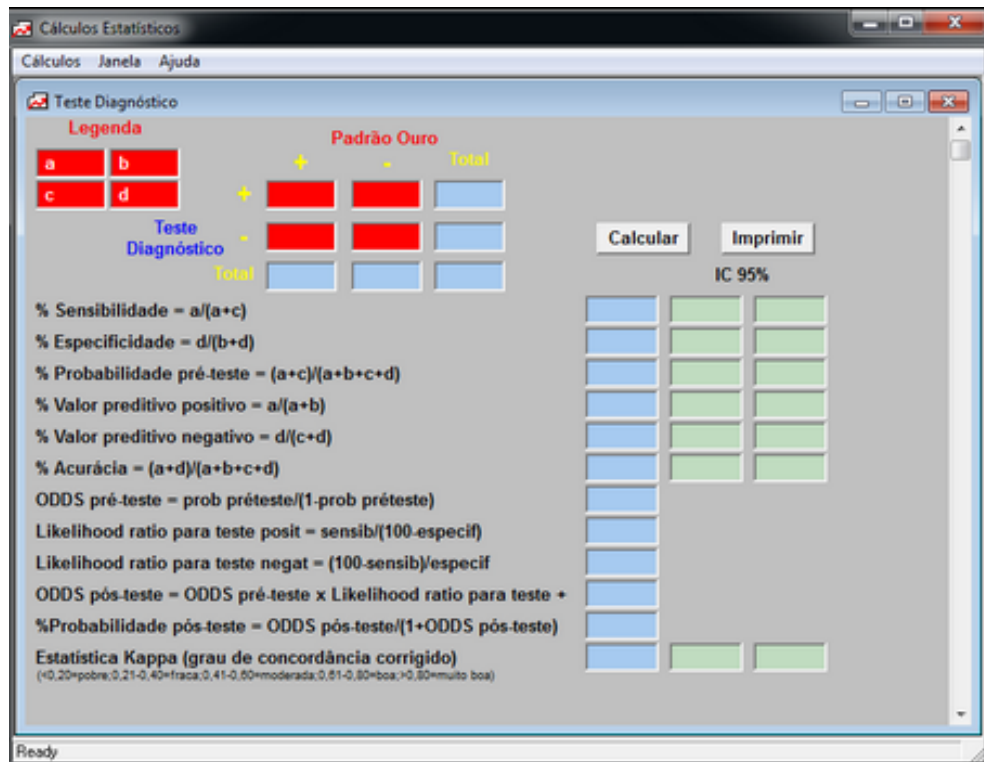
Posteriormente, foi desenvolvido o software, empregado na realização deste trabalho, que aplicou a biblioteca desenvolvida, portadora de várias funcionalidades, para estudo e apresentação de resultados, e o mais importante, para a automatização da atividade de busca dos resultados dos exames, classificação e armazenamento do resultado conforme desejado.

O software foi desenvolvido em linguagem Python (DRAKE, ) que é uma linguagem de programação de alto nível, interpretada, imperativa, orientada a objetos, de tipagem dinâmica e forte. O Python possui várias bibliotecas disponíveis, principalmente no campo científico sendo disponibilizada sob a licença GPL (General Public License; Licença Geral Pública). A Biblioteca ffnet 0.6.2o é descrita como uma solução de rede neuronal com retroalimentação rápida e fácil de usar. Possui várias facilidades como interface com a biblioteca “matplotlib”, a qual facilita a plotagem de gráficos e resultados, além de vários esquemas de algoritmos de otimização de

treinamento. Para preparar o ambiente de programação foi também instalados python 2.6 (PYTHON, 2008), python networkx (NETWORKX, ), python numpy (NUMPY, ), pywin32 214 (PYTHON, ) e python 2.6 scipy (SCIPY, ) que são software auxiliares, para a execução dos programas.

#### 4.5 ANÁLISES ESTATÍSTICA

Os valores que compõem a tabela 2 x 2 (quadrantes a, b, c, d) para os cálculos de sensibilidade e outros indicadores com o software Braile (BIOMEDICA, ) mostrado na figura 6 foram obtidos diretamente da base de dados.



**FIGURA 6:** INTERFACE GRÁFICA DO SOFTWARE BRAILE, UTILIZADO NA REALIZAÇÃO DE CÁLCULOS

**TABELA 2:** PARÂMETROS UTILIZADOS DAS REDES NEURONAIS (FAN E MLP) E AVALIADORES PROFISSIONAIS (A1 E B)

Resultados	Positivo	Negativo
Positivo	Verdadeiro Positivo (a)	Falso Positivo (b)
Negativo	Falso Negativo (c)	Verdadeiro Negativo (d)

a = verdadeiros positivos, resultados “retidos” das análises que concordam com os profissionais.

b = falso-positivos, resultados “retidos” das análises que discordam dos profissionais.

c = falso-negativo, resultados “liberados” das análises que discordam dos profissionais.

d = verdadeiro negativo, resultados “liberados” das análises que concordam com os profissionais



As variáveis em estudo foram analisadas quanto a normalidade utilizando os testes de Kolmogorov-Smirnoff e Lilliefors. A Tabela 3 mostra os valores de probabilidade para estes testes evidenciando que todos os analitos não apresentam distribuição normal.

**TABELA 3: VERIFICAÇÃO DA DISTRIBUIÇÃO NORMAL DOS PARÂMETROS EM ESTUDO.**

	<b>n</b>	<b>max D</b>	<b>Kolmogorov-Smirnoff</b> <b>p</b>	<b>Lilliefors</b> <b>p</b>
<b>pH</b>	140482	0,248524	p <0,01	p <0,01
<b>Esterase</b>	140482	0,426764	p < 0,01	p < 0,01
<b>Nitrito</b>	140482	0,50799	p <0,01	p <0,01
<b>Proteína</b>	140482	0,498882	p <0,01	p <0,01
<b>Glucose</b>	140482	0,518813	p < 0,01	p < 0,01
<b>Cetonas</b>	140482	0,437853	p < 0,01	p < 0,01
<b>Urobilinogênio</b>	140482	0,520866	p <0,01	p <0,01
<b>Bilirrubina</b>	140482	0,505653	p < 0,01	p < 0,01
<b>Hemoglobina</b>	140482	0,379651	p <0,01	p <0,01
<b>Eritrócitos</b>	140482	0,481514	p <0,01	p <0,01
<b>Leucócitos</b>	140482	0,455577	p <0,01	p <0,01
<b>Células</b>	140482	0,35636	p <0,01	p <0,01
<b>Cilindros</b>	140482	0,389757	p <0,01	p <0,01
<b>Bactérias</b>	140482	0,373965	p <0,01	p <0,01
<b>Idade</b>	140482	0,065571	p <0,01	p <0,01

Para a comparação entre as variáveis foram utilizados o teste não paramétrico U de Mann-Whitney. Para a análise de correlação foi utilizado o modelo de correlação linear descrito por Pearson, sendo considerado significativo aquelas que apresentaram  $p < 0,001$ , após a correção de Bonferroni.

As frequências entre sexo (masculino=1 e feminino=2) foram comparadas entre os grupos utilizando teste de chi-quadrado com 1 grau de liberdade.

O programa Statistica para Windows 8.0 (STATSOFT, ) foi utilizado em todas as análises.

Uma probabilidade ( $p$ ) inferior a 5% ( $p < 0.05$ ) foi considerada significativa em todas as análises, exceto quando indicado.

## **5 RESULTADOS**

### **5.1 AMOSTRA**

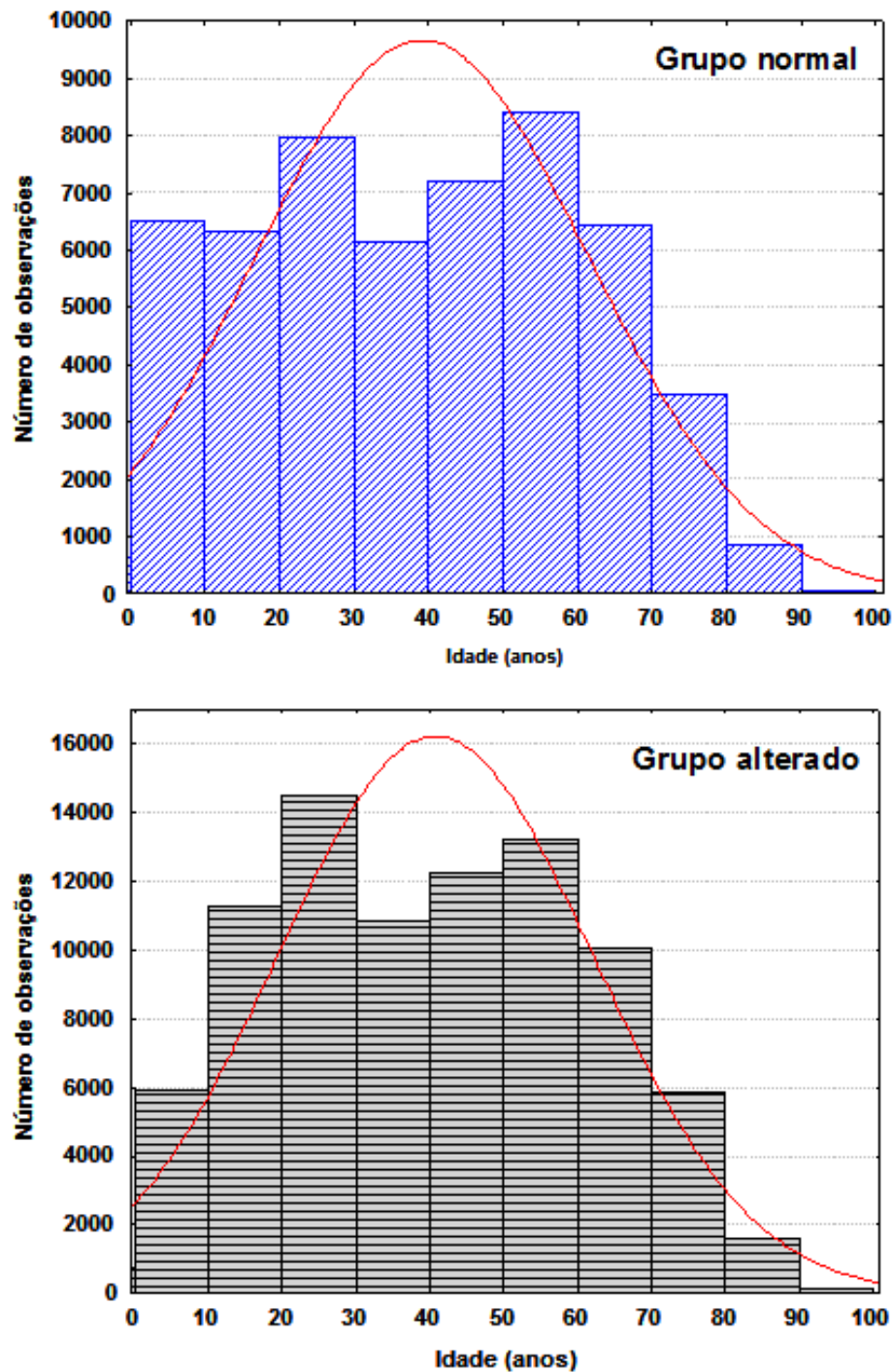
Registros caracterizando pacientes sem alterações nos parâmetros urinários estudados (“normal”), bem como aqueles “alterados”, caracterizados por apresentarem valores de referência para um ou mais analitos fora dos valores de referência estão inseridos na Tabela 4. Também é apresentado de forma hierárquica os analitos conforme sua contribuição para a caracterização das amostras alteradas.

**TABELA 4:** CARACTERÍSTICAS DAS AMOSTRAS EM ESTUDO E FREQUÊNCIA DE ANALITOS ALTERADOS.

Grupos* (n=140.482)	Frequências	Características
Normal	54.079 (38,5%)	Idade: 38,9a22,3 anos 40,0 [0-98] Sexo (M/F;%): 20,7/79,3
Alterado	86.403 (61,5%)	Idade: 40,5a21,3 anos 40,0 [0-101] Sexo (M/F;%): 35,4/64,6
Alterados (n=86.403)		Acima da referência (%)
Esterase Leucocitária		37,8
Hemoglobina		30,5
Leucócitos		24,4
Bactérias		19,2
Células epiteliais		15,3
Eritrócitos		12,5
Glucose		5,6
Proteínas		5,5
Nitrito		5,1
Cetonas		4,9
Cilindros		4,1
Urobilinogênio		2,4
Bilirrubina		0,1

Os grupos foram caracterizados como “sem alteração” pela ausência de qualquer valor dos analitos fora dos valores de referência ou “alterado” quando um analito ou mais se apresentaram alterado. Em destaque os analitos de maior contribuição para detecção de alteração no ensaio.

A Figura 7 representa a distribuição dos valores da idade das amostras em estudo classificadas segundo a presença (grupo alterado) ou ausência (grupo normal) de alterações nos parâmetros urinários em estudo.



**FIGURA 7:** HISTOGRAMAS DA VARIÁVEL IDADE NAS AMOSTRAS EM ESTUDO CLASSIFICADAS SEGUNDO A PRESENÇA OU AUSÊNCIA DE ALTERAÇÕES NOS PARÂMETROS URINÁRIOS.

A figura superior contempla as amostras sem alterações nos parâmetros laboratoriais em estudo (Grupo normal, n=54.079) e a figura inferior o grupo que apresentou alterações (Grupo alterado, n=86.403).

A linha curva em ambos os gráficos representa a distribuição normal dos dados.

## 5.2 LIBERAÇÃO DE RESULTADOS POR PROFISSIONAIS

A Tabela 5 mostra a caracterização demográfica e laboratorial da amostra classificada segundo o processo de liberação manual do LMC. O grupo caracterizado como “Liberado” representa resultados liberados pelos profissionais do setor de urinálise e, que apresentaram coerência e consistência nos resultados. O grupo “retido” foi liberado após análise individualizada da amostra e a critério do profissional apresentava na avaliação inicial inconsistente ou falta de correlação entre alguns parâmetros.

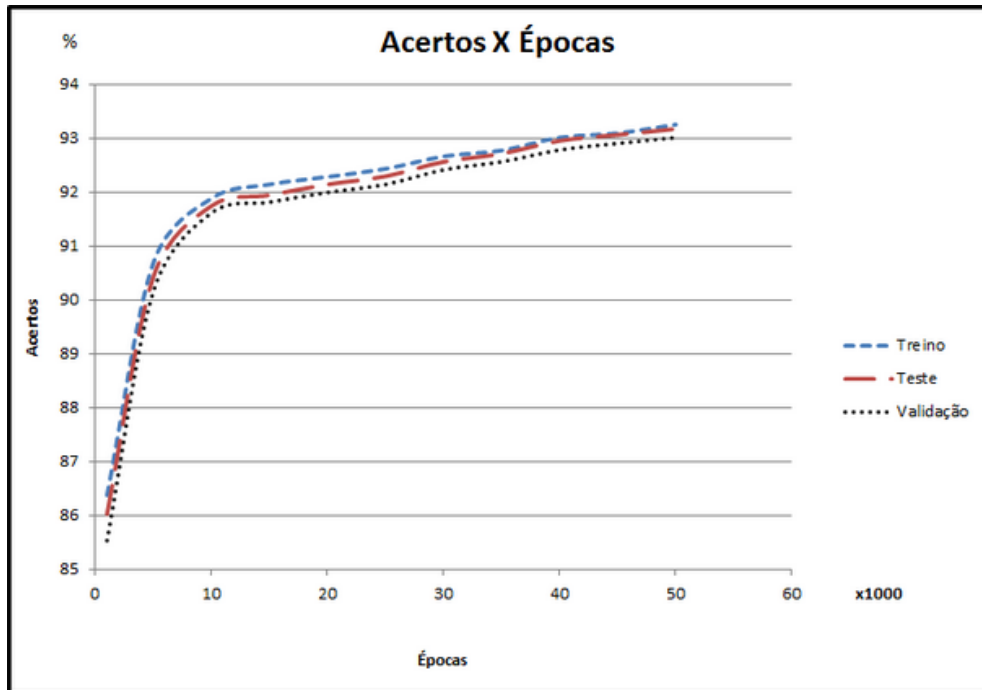
**TABELA 5:** CARACTERÍSTICAS DEMOGRÁFICAS E LABORATORIAIS DAS AMOSTRAS VALIDADAS POR PROFISSIONAIS EM ESTUDO.

Parâmetros (n=140.482)	Liberado	Revisado
n (%)	99.086 (70,5)	41.396 (29,5)
Idade (anos)	38,8 a 22,1 38,0 [0-100]	42,6,7 a 20,4 43,0 [0-101]
Sexo (M/F;%)	29,7/70,3	18,4/81,6
Esterase leucocitária (células/ $\mu$ L)	19,5 a 75,8 M [0-500]	64,5 a 181,4 25,0 [0-500]
Hemoglobina (mg/dL)	4,9 a 20,9 M [0-250]	25,7 a 55,5 10 [0-250]
Leucócitos contagem (células/ $\mu$ L)*	23,9 a 106,2 6,2 [0-9.039]	231,6 a 1384,7 14,3 [0-69.547]
Bactérias (células/ $\mu$ L)	3.108 a 8.985 957,6 [0-235.687]	7.682,2 a 15.935 1.674 [0-284.755]
Eritrócitos contagem (células/ $\mu$ L)*	12,2 a 539,7 4,6 [0-168901]	67,7 a 770,5 8.800 [0-70.950,4]
Glucose (mg/dL)	0 M [0]	98,6 a 276,4 M[0-1000]
Proteína (mg/dL)	0 M [0]	11,1 a 48,1 M[0-500]
Nitrito (mg/dL)	0 M [0]	0,09a0,58 M[0-1]
Cetonas (mg/dL)	0,87 a 6,3 M [0-150]	2,1 a 10,7 M[0-150]
Cilindros (elementos/ $\mu$ L)*	0,24 a 0,7 0,1 [0-58]	0,6 a 1,9 7,7 [0-91,8]
Urobilinogênio (mg/dL)	0,03 a 0,36 M [0-12]	0,1 a 0,7 M[0-12]
Bilirrubina (mg/dL)	0 M [0]	0,002 a 0,1 M[0-6]

Os valores mostrados são n (%) ou média 1-desvio padrão e mediana [mínimo-máximo]. Os valores de mediana marcados com "M" não foram calculados. A comparação entre homens e mulheres nos dois grupos foi significativa ( $p < 0,001$ ) utilizando o teste de chi-quadrado. Os analitos comparados pelo teste não paramétrico de Mann-Whitney apresentaram diferença significativa ( $p < 0,001$ ) entre os grupos, bem como as comparações com médias de idade. \* valores obtidos por citometria de fluxo (UF100, Sysmex). \*\* valores obtidos por interpretação de tira reagente (Urisys 2400).

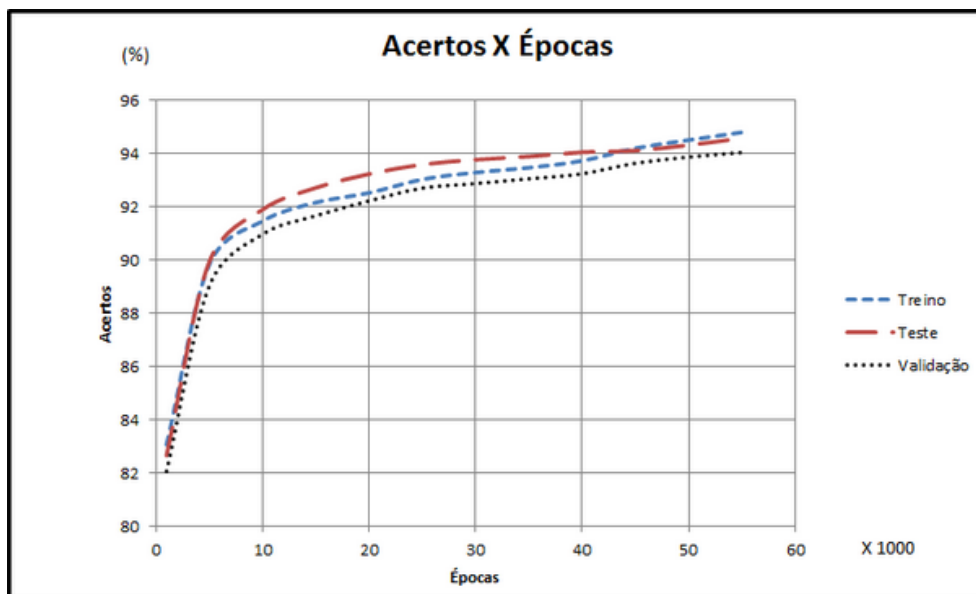
### 5.3 REDES NEURONAIS

As Figuras 8 e 9 evidenciam a evolução do acerto em número de épocas (leitura completa do arquivo de dados) para a rede MLP com bases de dados de diferentes tamanhos amostrais.



**FIGURA 8:** ANÁLISE DE TREINAMENTO E VALIDAÇÃO E TESTE DA REDE MLP UTILIZANDO BASE DE DADOS DE 34.292 AMOSTRAS.

A linha tracejada curto representa a evolução do treinamento, a tracejada longa o teste e a linha pontilhada a validação.



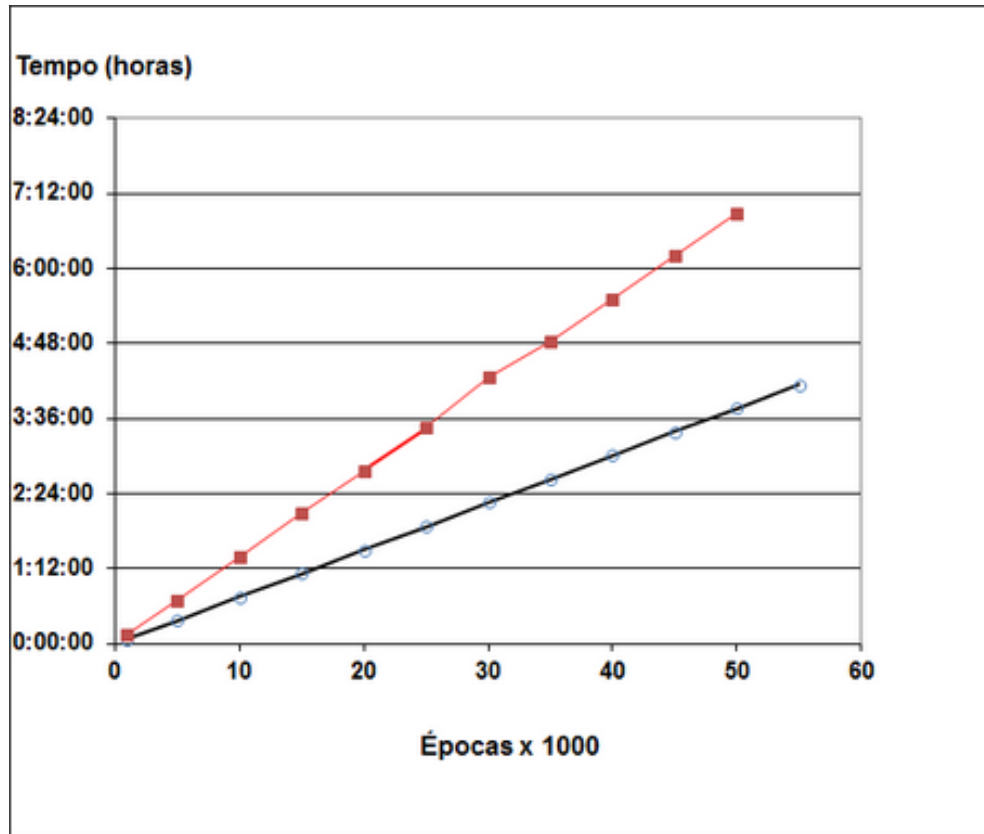
**FIGURA 9:** ANÁLISE DE TREINAMENTO, VALIDAÇÃO E TESTE DA REDE MLP UTILIZANDO BASE DE DADOS DE 17.200 AMOSTRAS.

A linha tracejada curta representa a evolução do treinamento, a linha tracejada longa o teste e a linha pontilhada a validação.

A base de dados de 17.200 tem seus dados de liberados e retidos em quantidades iguais (“balanceados”).

A Figura 10 mostra a comparação entre o desempenho dos dois processos de treinamento para a rede MLP em relação ao tempo de aprendizagem.





**FIGURA 10:** COMPARAÇÃO ENTRE O DESEMPENHO DO TREINAMENTO DA REDE MLP COM ARQUIVOS DE DIFERENTES TAMANHO AMOSTRAL.

A linha com quadrados representa a base de dados com  $n=34.292$  (Base 1) e aquela com círculos abertos representa a base de dados com  $n=17.200$  (Base 2, "balanceada"), O tempo está reportado em horas.

A análise do desempenho da rede MLP considerando o treinamento com bases de dados de diferentes tamanhos amostrais esta mostrada na Tabela 6.

**TABELA 6:** COMPARAÇÃO DE DESEMPENHO DA VALIDAÇÃO DA REDE MLP COM TREINAMENTO COM BASE DE DADOS COM 34.292 E 17.200 AMOSTRAS.

Indicadores (%)	Validação (MLP)		p*
	Base de dados n=34.292	Base de dados n=17.200	
Verdadeiro-negativo	72,4	45,5	< 0,001
Falso-positivo	3,7	1,3	< 0,001
Verdadeiro-positivo	21,4	48,7	< 0,001
Falso-negativo	2,5	4,7	< 0,001
Sensibilidade	89,6	91,2	< 0,001
Especificidade	95,1	97,3	< 0,001
Acurácia	93,8	94	0,385

\* Probabilidade (p) diferença entre duas proporções (STATSOFT, ).

#### 5.4 DESEMPENHO DOS SISTEMAS DE LIBERAÇÃO DE RESULTADOS

As redes MPL e FAN, bem como a análise de avaliadores humanos (profissionais do laboratório) foram comparadas com o critério de liberação de um profissional experiente na área, considerando sua análise no início da jornada de trabalho como referência para as demais comparações. Amostras de 484 indivíduos foram analisadas, representando aproximadamente a rotina de um dia do LMC. Os resultados estão apresentados na Tabela 7

**TABELA 7:** COMPARAÇÃO ENTRE AS DIVERGÊNCIAS DAS REDES MLP E FAN E AVALIADORES HUMANOS EM AMOSTRA DE 484 INDIVÍDUOS.

	Avaliadores humanos			Redes Neurais	
	A (%)	A1 (%)	B (%)	FAN (%)	MLP (%)
Avaliador A	—	5,2	8,1	17,1	9,1
Avaliador A1	5,2	—	11,2	19,4	10,5
Avaliador B	8,1	11,2	—	16,9	12,2
FAN	17,1	19,4	16,9	—	15,1
MLP	9,1	10,5	12,2	15,1	—
Média divergências	9,9	11,6	12,1	17,1	11,7

Os valores representam a divergência porcentual entre os elementos em estudo.

O avaliador A e A1 representam as análises do mesmo profissional em dois momentos distintos, respectivamente manhã e tarde, do mesmo dia de trabalho.

Os resultados do avaliador A foram considerados como referência para as demais comparações.

O desempenho das redes neurais e avaliadores humanos em estudo para a

amostra de 484 indivíduos estão apresentados na Tabela 8

**TABELA 8:** DESEMPENHO DE AVALIADORES HUMANOS E REDES NEURONAIS EM ESTUDO PARA AMOSTRAGEM DE 484 INDIVÍDUOS

Parâmetros	Avaliadores humanos			Redes Neurais		
	A1 (%)	B (%)	P*	FAN (%)	MLP (%)	p*
Sensibilidade	92,7	51,2	<0,001	39	100	<0,001
Especificidade	95	95,7	0,605	86,9	90,1	0,119
Valor preditivo Positivo	63,3	52,5	0,001	21,6	48,2	<0,001
Valor preditivo Negativo	99,3	95,5	0,002	93,9	100	<0,001
Acurácia	93,8	91,9	0,251	82,9	90,9	<0,001

Valores dos parâmetros em estudo foram calculados com o software Braille.

O avaliador A foi considerado como referência para comparação com os demais processos.

\* Probabilidade (p) diferença entre duas proporções (STATSOFT, ).

O banco de dados com mais de 100.000 registros foi analisado com as redes MLP e FAN, comparados com o desempenho de profissionais. Os resultados da comparação estão mostrados na Tabela 9.

**TABELA 9:** DESEMPENHO DOS PROCESSOS DE LIBERAÇÃO DE RESULTADOS EM ESTUDO.

Sistema de Liberação de Resultados	Registros analisados (n)	Amostras Retidas (%)
Valores de Referências 1	140482	61,5
Profissional 2	140482	29,5
Rede Neuronal MLP 3	102876	21,6
Rede Neuronal FAN 3	102876	18,2

1 Liberação utilizando os valores de referência. Dados obtidos da Tabela 4.

2 Dados obtidos da Tabela 5.

3 Resultados obtidos aplicando as redes MLP e FAN após treinamento e validação. A rede MLP foi treinada com base de dados "balanceada".

**TABELA 10:** COMPARAÇÃO ENTRE AS CARACTERÍSTICAS DAS REDES MLP E FAN PARA DETERMINAR PADRÕES DE RECONHECIMENTO DE CRITÉRIOS DE LIBERAÇÃO DE RESULTADOS DE URINÁLISE.

Características	Rede MLP	Rede FAN
Acurácia, sensibilidade, especificidade	+	-
Concordância com avaliador humano	+	-
Tempo de treinamento	+	-
Portabilidade	+	+
Facilidade de programação	+	-

(+) característica positiva da rede;

(-) característica negativa da rede

## 6 DISCUSSÃO

Este trabalho pretendeu estudar redes neuronais e suas aplicações no auxílio à tomada de decisão para liberação de ensaios laboratoriais de análise de urina.

A urinálise foi escolhida como modelo para estudar redes neuronais no apoio à decisão por ser um ensaio não invasivo que facilita a aderência do paciente ao teste, é de elevada solicitação e capacidade de interpretação complexa, sendo associada a múltiplas patologias (ADDIS, 1926; COSTAVAL *et al.*, 2001).

Outro elemento decisivo na escolha da urinálise como modelo para estudo das redes neuronais está relacionado à homogeneidade em que o procedimento é realizado. Todos os pacientes, independente de idade, sexo ou processos patológicos são analisados com os mesmos procedimentos (BOTTINI; GARLIPP, 2006; MACHADO *et al.*, 2003). Também foi relevante para esta escolha, a disponibilidade no Laboratório Municipal de Curitiba (LMC) de sistemas automatizados para a realização dos ensaios químicos (Urisys 2400; Sysmex, Roche) e citológicos (UF100; Sysmex, Roche), o que padroniza a fase analítica.

Poucos estudos estão disponíveis na literatura empregando redes neuronais para apoio à decisão e liberação de resultados. Nenhum foi encontrado relacionado à urinálise.

É relevante mencionar que o processo que envolve a liberação de resultados em laboratório clínico, elemento da fase pós-analítica, aparentemente tem um desenvolvimento mais lento quanto ao processo de automação, quando comparado aos analisadores automáticos que atualmente dominam a fase analítica de muitos laboratórios (CHILDS; EGAN, 1996).

A liberação de exames laboratoriais está centrada na ação e decisão de profissionais que analisam os resultados com base no valor de referência, associações com sexo, idade e condições clínicas dos pacientes (FERREIRA; ANDRIOLO, 2008).

Algumas instituições disponibilizam aos profissionais que liberam resultados

sistemas informatizados simplistas, que utilizam a baseados em valores de referência para facilitar a liberação. Neste caso, resultados que foram triados pelo programa e encontram-se dentro da referência são liberados automaticamente. Aqueles retidos são avaliados por profissionais.

A principal crítica aos softwares com base em valores de referências recai no elevado número de amostras retidas, uma vez que qualquer alteração da referência gera um critério de retenção da liberação (GEIGER; ASCP, 1995). Deve ser ressaltado, no entanto as vantagens do emprego de valores de referência na análise de resultados esta associadas à simplicidade de desenvolvimento e implantação do programa.

(MELO *et al.*, 2006) desenvolveram um software com base em regras Booleanas para liberação de ensaios hormonais. Embora os autores ressaltem a eficácia do software, reportam a necessidade da criação de mais de 150 regras para desenvolver o sistema e sugerem que para trabalhos complexos a lógica fuzzy e as redes neuronais, podem ser abordagens mais atrativas.

A liberação por profissionais, proposta como padrão, não é isenta de críticas e problemas. A homogeneidade e padronização de critérios de liberação entre profissionais é com frequência um processo difícil de ser alcançado. A liberação por humanos está intimamente relacionada a erros associados ao cansaço e ao estresse que estes profissionais estão submetidos nas rotinas de suas instituições de atividades laborais, como reportado por (UETTWILLER-GEIGER, 2005).

Entre as principais vantagens associadas à utilização de redes neuronais para a detecção de padrões e fornecer informação para o apoio à decisão de liberação de exames laboratoriais está centrado na capacidade do software em aprender com as regras estabelecidas no local de trabalho e apresentar flexibilidade na interpretação de múltiplas variáveis correlacionadas ((TU, 1996; AGATONOVIC; BERESFORD, 2000; MELO *et al.*, 2006)).

## 6.1 AMOSTRA

As características dos pacientes que procuraram atendimento no LMC para análise de urina apresentaram idade média de 38 anos, variando de dias de vida até 101 anos de idade, e predominância de mulheres, na proporção de 2:1 com homens Tabela 4. O predomínio feminino pode ser explicado pela maior frequência de

infecção urinária neste grupo, e nesta situação, a urinálise é dos principais exames requisitados. (HÖRNER; , Aline Mastella , Adenilde Salla , Roselene Alves Righi, 2006), reportaram para uma população do Hospital Universitário de Santa Maria, RS, Brasil, um predomínio de 76% de mulheres com idade de 13 a 35 anos, com infecção urinária, corroborando com o observado no presente estudo, além disso, o elevado número de gestantes provenientes do “Programa Mãe Curitibana” contribui para elevar o percentual feminino da amostra.

Considerando os valores de referência (Materiais e Métodos, Tabela 1) para o exame de urinálise, as amostras em estudo avaliadas por valores de referência mostrou que 61,5% dos resultados analisados apresentavam algum tipo de alteração Tabela 4. Portanto, se empregado um software com esta característica no apoio à decisão, os profissionais deveriam reter cerca de 2 em cada 3 amostras, para análises mais detalhadas antes da liberação, o que representa uma carga de trabalho substancial na rotina laboratorial.

(KROENKE *et al.*, 1986) analisando ensaios de rotina e aqueles com indicação clínica, mostraram que resultados anormais eram maiores no segundo grupo (18,1% vs. 39,6%). Os ensaios do LMC apresentaram alteração significativamente superior a este trabalho (61,5%), podendo ser justificado pelas diferentes características das populações em estudo e metodologias empregadas.

Quando foi considerado o efeito da participação individual dos analitos que compõem o ensaio de urinálise, seis destes ensaios (esterase leucocitária, hemoglobina, contagem de leucócitos, contagem de bactérias, contagem de células epiteliais e contagem de eritrócitos) estão presentes em cerca 95% de todas as amostras alteradas Tabela 4. Outros estudos confirmam a relevância e prevalência dos parâmetros citados (LOHR *et al.*, 1993; SIMERVILLE *et al.*, 2005; SANGRADOR *et al.*, 2007). Os demais parâmetros, embora possam ser relevantes no diagnóstico, como glicose e proteínas, estão associados em apenas 5% ou menos das amostras alteradas.

(CAREL *et al.*, 1987) estudando uma grande número de análises químicas da urina (n=21.000) reportaram alterações em cerca de 10% de todas as amostras, sendo mais frequentes em homens e mulheres, respectivamente, a proteinúria (4,9%) e a hematúria (8,1%), sendo que somente 1% das amostras apresentava dois ou mais parâmetros alterados. Os resultados obtidos neste trabalho são compatíveis com a relevância da hematúria, presente em 30,5% das amostras estudadas. No entanto, a

proteinúria representa uma alteração presente em apenas 5,3% das amostras (Tabela 4). Uma possível explicação para esta divergência está na prevalência de mulheres na amostragem deste trabalho, cuja alteração principal possa estar relacionada à hematúria.

(ROGGEMAN; ZAMAN, 2001) demonstraram que analisando em conjunto os resultados da tira reagente com aqueles obtidos por citometria de fluxo, com o UF100, foi possível eliminar a confirmação dos exames realizada através de exame microscópico pelos profissionais em cerca de 40%. A imprecisão analítica (coeficiente de variação intraensaio) do sistema automatizado UF100 para elementos citológicos na faixa normal foram reportados como eritrócitos 31%, leucócitos 14%, células epiteliais 18%, cilindros 45% e bactérias 2-12% (BEN-EZRA; BORK, 1998).

Entre os seis elementos mais representativos associados às amostras alteradas, as contagens de bactérias e células epiteliais são pouco informativas (DELANGHE, 2007). A contagem de células epiteliais aumentada, por exemplo, tem baixo valor diagnóstico, não sendo associada especificamente com patologias de elevada frequência populacional (MACHADO *et al.*, 2003).

(CHRISTENSON *et al.*, 1985) reforçam esta afirmativa e sugerem que a contagem elevada de células epiteliais em mulheres é sugestivo de contaminação da amostra.

A contagem de bactérias realizada pelo sistema automatizado UF100 pode apresentar inconsistências, em especial resultados falso-positivos pela contagem de elementos não-orgânicos que sinalizam na citometria de fluxo como bactérias (HAUSS, 2008; MANONI *et al.*, 2009). O novo modelo da Sysmex, o UF1000, uma evolução do UF100 empregado neste estudo, contempla um canal exclusivo na citometria de fluxo para a contagem de bactérias aumentando a especificidade em cerca de 10 vezes (OKADA *et al.*, 2007).

Segundo (EVANS *et al.*, 2006) as contagens de bactérias na urina em valores superiores a 3.000/L são significativas apenas quando associadas às contagens de leucócitos superiores a 11/L, o que reforça o baixo valor diagnóstico da contagem bacteriana isolado.

As determinações das concentrações da esterase leucocitária, e hemoglobina, bem como as contagens de leucócitos e eritrócitos foram consideradas os principais elementos discriminantes no ensaio de urinálise e alvos primários no estudo com redes neurais.



A idade dos pacientes em estudo apresentou distribuição bimodal nos grupos sem alteração (“normal”; Figura 5) e naquele que apresentou no mínimo um parâmetro fora dos valores de referência (“alterado”, Figura 5). Predominaram na amostra em estudo as faixas etárias de 20 a 30 anos e 50 a 60 anos com frequências de 14,8; 15,7% e 16,5; 15,0% para os grupos normal e alterado, respectivamente. A predominância de pacientes entre 20-30 anos pode ser explicada pelo elevado número de gestantes atendidas no LMC. A frequência de crianças com idade de 0 a 10 anos foi maior no grupo normal (12%) quando comparado ao grupo alterado (7,1%), sem uma explicação para o achado no presente (Figura 5).

## 6.2 LIBERAÇÃO DE RESULTADOS POR PROFISSIONAIS MANUALMENTE

A análise de consistência de resultados laboratoriais por profissionais considera outros elementos além dos valores de referência ((BECK, 1986). Entre estes, o principal destaque está na procura de correlações entre os parâmetros químicos e citométricos da análise. Amostras que apresentem reação para esterase leucocitária positiva e elevada contagem de leucócitos são consistentes e, portanto o resultado pode ser liberado. A mesma lógica associativa se aplica para outros elementos como hemoglobina e hematúria. A referida associação é relevante, a ponto de (LANGLOIS *et al.*, 1999) proporem que havendo concordância entre as contagens de leucócitos e eritrócitos com esterase leucocitária e hemoglobina, respectivamente, a análise microscópica seria desnecessária.

Quando os resultados das amostras em estudo foram analisados por profissionais, o número de amostras que foram retidas para “revisão” foi 29,5% (Tabela 5). Comparada com a retenção proposta com valores de referência (61,5%) representa 30 pontos percentuais a menos, refletindo agora que cerca de 1 amostra em 3 serão analisadas e tratadas de forma diferenciada, o que representa substancial redução de trabalho aos profissionais.

Outra observação relevante mostra que a frequência de mulheres entre as amostras encaminhadas para “revisão” por profissionais é superior àquela “alterada”, obtida quando se aplica apenas os valores de referência (81,6% vs 64,5%) como observado nas Tabelas 4 e 5. Este achado pode sugerir que a análise de consistência em pacientes do sexo feminino é mais difícil para os profissionais, considerando a maior probabilidade nas mulheres para a contaminação com fluídos vaginal e/ou menstrual, estes profissionais tendem a serem mais críticos e segregar mais estas

pacientes (AMORIM *et al.*, 2009; BRILHA *et al.*, 2010).

As concentrações entre os analitos nos grupos “liberado” e “retido” foi significativa ( $p < 0,001$ ) para todos os parâmetros, como esperado (Tabela 4). Este achado realça que entre os principais critérios para profissionais reterem uma amostra para análise posterior está na presença de concentrações ou contagens elevadas dos analitos.

Na liberação por profissionais, as determinações de glicose, proteínas, nitrito e bilirrubina foram elementos relevantes na tomada de decisão. Valores para estes analitos acima da referência foram critérios impeditivos para a liberação direta, visto que somente valores dentro da referência são encontrados no grupo liberado (Tabela 5).

Quando se compara as médias das contagens de células, os leucócitos apresentaram a maior diferença (cerca de 10 vezes) entre os grupos “liberado” e “retido” reforçando a relevância deste parâmetro como elemento de decisão para a liberação de resultados. Em estudo como uma grande população Boliviana supostamente saudável ( $n=14.082$ ), 30% apresentaram alteração no ensaio de urinálise, com predominância de hematúria (47%), leucocitúria (41%) e proteinúria (11%), reforçando a relevância destes parâmetros como elementos discriminadores de doença (PLATA *et al.*, 1998).

A análise de correlação apresentada na Figura 6, mostra que a concentração da esterase leucocitária foi o parâmetro que apresentou a correlação positiva e significativa ( $r > 0,10$  e  $p < 0,001$ ) com maior número de outros parâmetros (leucócitos, hemoglobina e cilindros). Esta observação era esperada e confirma a importância do parâmetro como elemento diagnóstico, como descrito por (PENDERS, 2002).

### 6.3 REDES NEURONAIS

Em estudos iniciais para esta pesquisa uma seleção e triagem de redes neuronais foram realizadas. Foram estudadas quatro redes neuronais: Multi Layer Perceptron (MLP), Free Associative Neurons (FAN), Radial basis functions (RBF) e Support Vector Machine (SVM). Considerando características como facilidade de uso e bom desempenho em estudos preliminares, foram selecionadas as redes FAN e MLP para este estudo.

Rede FAN

A aplicação da rede neuronal FAN esta bem disseminada no laboratório de Bioinformática da UFPR. Esta rede está incorporada no software easyFAN que permite a entrada de dados e a visualização dos resultados (RAITTZ, 2002). Para a rede FAN, foram utilizados protocolos já validados e de domínio público disponíveis no Laboratório de Bioinformática da UFPR (RAITTZ, 1997).

### Rede MLP

A rede neuronal MLP é bem difundida na literatura (WEI *et al.*, ; WEST, 2000). No entanto, a aplicação desta rede na identificação de padrões de parâmetros laboratoriais não está disponível na literatura, assim, ensaios de validação foram realizados.

Em primeira abordagem, o treinamento, teste e validação da rede MLP foi realizado com três bases de dados iguais de 34.292 registros, sendo estas amostras escolhidas aleatoriamente e mantendo as proporções similares dos grupos “liberado” e “retido” definidos pelos profissionais (Figura 8). Com 45.000 épocas a rede apresentou estabilidade, ausência de overfitting e acerto satisfatório acima de 90%. No entanto, foi observado um tempo de treinamento elevado, superior a 6 horas (Figura 8).

Em outro experimento, o treinamento, teste e validação da rede MLP foi realizado com arquivos contendo 17.200 registros, portanto 50% menor que o procedimento inicialmente utilizado. Para este estudo, foram gerados arquivos contendo os grupos “liberado” e “retido” distribuídos em proporções iguais, também designados “balanceados”. Com 45.000 épocas esta formatação de arquivos permitiu atingir cerca de 94% de acerto (Figura 9). O tempo de treinamento neste caso foi pouco superior a 3 horas, portanto 50% menor que a abordagem sem amostra “balanceada” (Figura 10).

Quando indicadores de desempenho da rede MLP, validada pelos dois procedimentos descritos anteriormente, foi realizado, comparou-se o procedimento de validação com amostra “balanceada”, e a resultante foi superior (Tabela 1).

Entre os indicadores, os resultados de falso-negativo foram significativamente maiores ( 1,9 vezes) com o treinamento “balanceado”, que também apresentou 60% menos resultados verdadeiro-positivos. Esta característica desfavorável, no entanto, não é crítica ao sistema, pois estes resultados serão retidos e analisados em outra etapa, evitando a liberação inconsistente.

É possível concluir que a composição do grupo de treinamento afeta o

desempenho da rede MLP (MAZUROWSKI *et al.*, 2008). Estudando redes neuronais para apoio a decisão em câncer, mostraram que a utilização de treinamento da rede com dados não balanceados (“imbalance”) tem efeito progressivamente deletério na qualidade da classificação fornecida pela rede, resultado compatível com os obtidos em nosso trabalho.

É relevante ressaltar que as duas formas de validação estudadas para a rede MLP mostraram acurácia similar, superior a 93%, como predito pela porcentagem de acerto nos ensaios de validação (Figuras 8 e 9).

Neste contexto, propomos que a rede MLP seja treinada, testada e validada com arquivos “balanceados” das características desejadas, o que permite menor tempo de processamento e mantém o mesmo grau de acerto para a aplicação em tela. Esta forma de validação foi empregada para os experimentos em sequência.

#### 6.4 DESEMPENHO DOS SISTEMAS DE LIBERAÇÃO DE RESULTADOS

Uma amostra de 484 registros de urinálise, representando a rotina de um dia do LMC, foi analisada para avaliar o grau de divergência entre os processos em estudo (Tabela 7). Um profissional experiente analisou os resultados no início do seu dia de trabalho (Avaliador A) classificando as amostras em “liberada” e “retida”. Esta classificação, para fins de estudo, foi considerada como referência para as demais comparações. As mesmas amostras foram na sequência analisadas novamente pelo avaliador A no final da jornada de trabalho (Avaliador A1), outro profissional do laboratório (Avaliador B) e pelas redes FAN e MLP, esta última validada com amostra “balanceada”.

O avaliador A apresentou uma divergência entre as duas análises consigo mesmo da ordem de 5% e o avaliador B divergiu em 8%. As divergências entre analistas humanos são esperadas, uma vez que muitas variáveis são analisadas em conjunto em cada registro, os critérios de liberação são flexíveis e o tempo de tomada de decisão para cada amostra é usualmente pequeno (10-20 segundos por amostra).

As redes neuronais estudadas apresentaram divergência superior aos analistas humanos, sendo o melhor desempenho obtido com a rede MLP (Tabela 7). A divergência da rede FAN com a referência (Avaliador A) foi cerca de 3 vezes maior (17,1%), enquanto a rede MLP foi de cerca de 2 vezes superior (9,1%). Portanto, a rede MLP proporcionou resultados melhores quando comparada a rede FAN.

A análise de desempenho entre humanos e redes neuronais em estudo reforça a observação que a rede neuronal MLP é superior à rede FAN, considerando os parâmetros estudados, com destaque para sensibilidade (100%) e valor preditivo negativo (100%), como observado na Tabela 8.

Na análise das amostras, o avaliador B apresentou sensibilidade e valor preditivo positivo pouco superior a 50%, sugerindo que este profissional quando comparado ao avaliador A (tomado como referência) tem critérios mais conservadores, ou seja, tende a caracterizar a amostra como “retida” com mais frequência, característica esta mais próxima ao desempenho das redes neuronais (Tabela 6).

Com as características, análises de divergência e desempenho das redes neuronais FAN e MLP conhecidas e quantificadas, estas foram utilizadas para avaliar uma base de dados grande, superior a 100.000 registros (Tabela 9). Nesta análise, o desempenho da rede MLP quanto às amostras retidas (21,6%) se aproxima ao dos profissionais. A rede FAN reteve apenas 18,2% das amostras.

Uma vez que as redes neuronais foram validadas com os critérios de liberação dos profissionais, o melhor desempenho está relacionado a uma frequência de retenção similar a este critério. Neste contexto é possível concluir que a rede MLP apresentou desempenho superior à rede FAN. A menor frequência de amostras retidas pela rede FAN pode ser atribuída a sua menor sensibilidade (39%, Tabela 8) quando comparada a rede MLP (100%).

A Tabela 10 apresenta uma síntese para a comparação entre as principais características das redes neuronais em estudo. Embora algumas características apresentadas tenham elementos subjetivos de análise, estes foram considerados relevantes para comparações com futuros trabalhos com estes sistemas.

De modo seguro e conclusivo a rede neuronal MLP apresentou melhor desempenho que a rede FAN sendo recomendada para os objetivos desta pesquisa.

A iniciativa de utilizar redes neuronais nos procedimentos do laboratório clínico é antiga (CICCHETTI, 1992; EKLUND; FORSSSTRÖM, 1995; TANGRI *et al.*, 2008)), No entanto, a aplicação de redes neuronais associada a processos pertinentes ao laboratório clínico está nos seus primórdios, considerando a escassez de publicações na área. A rápida evolução desta metodologia na análise de padrões de imagens de uso diagnóstico sugere que a transposição deste processo como auxiliar na decisão de liberação de resultados no laboratório clínico é esperada e previsível(CUNNINGHAM *et al.*, 2000).

Segundo (PLEBANI; CARRARO, 1997) a fase pós-analítica, que contempla a liberação de resultados, é responsável por 18-47% dos erros associados ao laboratório clínico. Neste contexto é previsível que a utilização de redes neuronais como uma tecnologia de auxílio à liberação do resultado, com as características mostradas neste trabalho, tenha impacto positivo e relevante na qualidade do laboratório clínico e satisfação do cliente. O trabalho desenvolvido nesta dissertação é pioneiro e reflete os primeiros passos para implementação e difusão da aplicação de redes neuronais no laboratório de análises clínicas.

## 7 CONCLUSÕES

- As características demográficas e laboratoriais, sexo, idade, concentração da esterase leucocitária, concentração de hemoglobina, contagem de leucócitos e contagem de eritrócitos na urina são os principais fatores discriminantes para a liberação de resultados em urinálise.
- Os critérios para a configuração dos arquivos de treinamento, teste e validação para a rede MLP afetam o desempenho desta, sendo arquivos “balanceados” definidos pela presença de distribuição dos grupos de forma equitativa recomendados.
- A rede neuronal *Multi Layer Perceptron* (MLP) validada com uma base de dados “balanceada” permite identificar padrões liberação e retenção de resultados de forma a mimetizar os critérios de profissionais para o ensaio de urinálise.
- A rede neuronal MLP apresentou desempenho superior à rede *Free Associative Neurons* (FAN).
- O programa computacional em linguagem Python, utilizando a rede MLP, desenvolvido para acessar o banco de dados e classificar a amostra apresentou bom desempenho e segurança, sendo recomendado para uso no laboratório clínico.

## REFERÊNCIAS BIBLIOGRÁFICAS

ADDIS, T. The number of formed elements in the urinary sediment of normal individuals. **J. Clin. Investig.**, v. 2, n. 5, p. 409–15, jun. 1926. ISSN 0021-9738. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=434597&tool=pmcentrez&rendertype=abstract>.

AGATONOVIC, S.; BERESFORD, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. **J. Pharm. Biomed. Anal.**, Elsevier, v. 22, n. 5, p. 717–727, jun. 2000. ISSN 0731-7085. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0731708599002721>.

AMORIM, A.; PACHECO, J.; FERNANDES, T. Exame de urina tipo I: frequência percentual de amostras que sugerem infecção urinária. *Anuário da Produção Iniciação ificaDiscente.*, v. 11, n. 12, p. 57, 2009. ISSN 1517-6444. Disponível em: <http://www.sare.unianhanguera.edu.br/index.php/anuic/article/viewArticle/492>.

BATES, D.; GAWANDE, A. Improving safety with information technology. **New. Engl. J. Med.**, Mass Med Soc, v. 348, n. 25, p. 2526, 2003. Disponível em: <http://content.nejm.org/cgi/content/abstract/348/25/2526>.

BECK, J. Laboratory decision science applied to chemometrics: strategic testing of thyroid function. **Clin. Chem.**, Am Assoc Clin Chem, v. 32, n. 9, p. 1707, 1986. Disponível em: <http://www.clinchem.org/cgi/content/abstract/32/9/1707>.

BEN-EZRA, J.; BORK, L. Evaluation of the Sysmex UF-100 automated urinalysis analyzer. **Clin. chemistry.**, v. 44, n. 1, p. 92–5, jan. 1998. ISSN 0009-9147. Disponível em: <http://www.clinchem.org/cgi/content/abstract/44/1/92>.

BIOMEDICA, B. **Braile**. Disponível em: <http://www.braile.com.br/cientifica/pesqciem.htm>.

BOTTINI, P.; GARLIPP, C. Urinálise: comparação entre microscopia óptica e citometria de fluxo. **J. Bras. Patol. Med. Lab.**, v. 42, n. 3, p. 157–162, jun. 2006. ISSN 1676-2444. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1676-24442006000300003&lng=pt&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1676-24442006000300003&lng=pt&nrm=iso&tlng=pt).

BRILHA, S.; PROENCA, H.; CRISTINO, J. M.; HÄNSCHEID, T. Use of flow cytometry (Sysmex) UF-100) to screen for positive urine cultures: in search for the ideal cut-off. **Clin. Chem. Lab. Med.**, v. 48, n. 2, p. 289–92, fev. 2010. ISSN 1434-6621. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/19961394>.

CALVINO, I. **Mr. Palomar**. [S.l.]: Mariner Books, 1986. 144 p. ISBN 0156627809.

CAREL, R.; SILVERBERG, D.; KAMINSKY, R.; AVIRAM, A. Routine urinalysis (dipstick) findings in mass screening of healthy adults. **Clin. Chem.**, Am Assoc Clin



Chem, v. 33, n. 11, p. 2106, 1987. Disponível em: <http://www.clinchem.org/cgi/content/abstract/33/11/2106>.

CHILDS, S.; EGAN, R. Bacteriuria and urinary infections in the elderly. *Urol. Clin. N. Am.*, v. 23, n. 1, p. 43, 1996. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/8677536>.

CHRISTENSON, R.; TUCKER, J.; ALLEN, E. Results of dipstick tests, visual inspection, microscopic examination of urine sediment, and microbiological cultures of urine compared for simplifying urinalysis. *Clin. Chem.*, Am Assoc Clin Chem, v. 31, n. 3, p. 448, 1985. Disponível em: <http://www.clinchem.org/cgi/content/abstract/31/3/448>.

CICCHETTI, D. Neural networks and diagnosis in the clinical laboratory: state of the art. *Clin. Chem.*, Am Assoc Clin Chem, v. 38, n. 1, p. 9, 1992. Disponível em: <http://www.clinchem.org/cgi/reprint/38/1/9.pdf>.

COSTAVAL, J. da; PADUA, M. de; CERQUEIRA, C.; COSTAVAL, A. da; AULER, A.; MARTINS, G. Qual o valor da sedimentoscopia em urinas com características físico-químicas normais? *J. Bras. Patol. Med. Lab.*, SciELO Brasil, v. 37, n. 4, 2001. Disponível em: [http://www.scielo.br/scielo.php?pid=S1676-24442001000400007&script=sci\\_arttext&tlng=in](http://www.scielo.br/scielo.php?pid=S1676-24442001000400007&script=sci_arttext&tlng=in).

CUNNINGHAM, A.; CARNEY, J.; JACOB, S. Stability problems with artificial neural networks and the ensemble solution. *Artif. Intell. Med.*, v. 20, p. 217–225, 2000.

DAYHOFF, J.; DELEO, J. Artificial neural networks. *Cancer.*, Wiley Online Library, v. 91, n. S8, p. 1615–1635, 2001. ISSN 1097-0142. Disponível em: [http://onlinelibrary.wiley.com/doi/10.1002/1097-0142\(20010415\)91:8+1615::AID-CNCR1175>3.0.CO;2-L/full](http://onlinelibrary.wiley.com/doi/10.1002/1097-0142(20010415)91:8+1615::AID-CNCR1175>3.0.CO;2-L/full).

DELANGHE, J. New screening diagnostic techniques in urinalysis. *Acta. Clin. Belg.*, v. 62, n. 3, p. 155, 2007. ISSN 0001-5512. Disponível em: <http://www.actaclinicabelgica.be/V62,Nr3/155-161Newscreening.pdf>.

DRAKE, F. L. An Introduction to Python Guido van Rossum. *Tech.*

EASYFAN. **EasyFan**. Disponível em: <http://easyfan.sourceforge.net/>.

EKLUND, P.; FORSSTRÖM, J. Computational intelligence for laboratory information systems. *Scand. J. Clin. Lab. Investig.*, Informa UK Ltd UK, v. 55, n. S222, p. 21–30, 1995. ISSN 0036-5513. Disponível em: <http://informahealthcare.com/doi/abs/10.3109/00365519509088447>.

EVANS, R.; DAVIDSON, M. M.; SIM, L. R. W.; HAY, a. J. Testing by Sysmex UF-100 flow cytometer and with bacterial culture in a diagnostic laboratory: a comparison. *J. Clin. Pathol.*, v. 59, n. 6, p. 661–2, jun. 2006. ISSN 0021-9746. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1860387&tool=pmcentrez&rendertype=abstract>.

FERNEDA, E. Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ci. Inf.*, SciELO Brasil, v. 35, n. 1, p. 25–30, 2006. Disponível em: <http://www.scielo.br/pdf/ci/v35n1/v35n1a03.pdf>.

FERREIRA, C. E. D. S.; ANDRIOLO, A. Intervalos de referência no laboratório clínico. *J. Bras. Patol. Med. Lab.*, v. 44, n. 1, p. 11–16, fev. 2008. ISSN 1676-2444. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1676-24442008000100004&lng=pt&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1676-24442008000100004&lng=pt&nrm=iso&tlng=pt).

FRIZE, M.; ENNETT, C. M.; STEVENSON, M.; TRIGG, H. C. E. Clinical decision support systems for intensive care units : using artificial neural networks. *Med. Eng. Phys.*, v. 23, p. 217–225, 2001.

GEIGER, B. D. L. U.; ASCP, D. L. M. LAB MANAGEMENT A labs strategy to reduce errors depends on automation. *Med. Lab. Obs.*, 1995.

GOLDSCHMIDT, H. M. J. A review of autovalidation software in laboratory medicine. *Accredit. Qual. Assur.*, v. 7, n. 11, p. 431–440, nov. 2002. ISSN 0949-1775. Disponível em: <http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1007/s00769-002-0547-y>.

GOULART, D. A.; TACLA, M. A.; MARBACK, P. M. F.; SOLE, D.; Paranhos Junior, A.; PEREZ, H. B.; FREITAS, D. D.; SATO, E. H. Redes neurais artificiais aplicadas no estudo de questionário de varredura para conjuntivite alérgica em escolares. *Arq. Bras. Oftalmol.*, v. 69, n. 5, out. 2006. ISSN 0004-2749. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0004-27492006000500017&lng=pt&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0004-27492006000500017&lng=pt&nrm=iso&tlng=pt).

GRAUPE, D. **Principles of artificial neural networks**. [S.l.]: World Scientific Pub Co Inc, 2007. 258 p. ISBN 9812706240.

HAUSS, O. Bringing Urinalysis into the 21st Century: From Uroscopy to Automated Flow Cytometry. *Sysmex. J. Int.*, v. 18, n. 2, p. 38–43, 2008. Disponível em: [http://www.sysmex.ru/files/f8/Image/pic/\\_1849/Oliver/\\_Hausse/\\_Sysmex/\\_Journal.pdf](http://www.sysmex.ru/files/f8/Image/pic/_1849/Oliver/_Hausse/_Sysmex/_Journal.pdf).

HAWKER, C. D. Laboratory automation: total and subtotal. *Clin. Lab. Med.*, v. 27, n. 4, p. 749–70, vi, dez. 2007. ISSN 0272-2712. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/17950896>.

HAYKIN, S. Neural networks: a comprehensive foundation. *Neural. Networks.*, 1994. Disponível em: <http://portal.acm.org/citation.cfm?id=541500>.

HEBB, D. The organization of behaviour: a neuropsychological theory. *J. Appl. Behav. Anal.*, Wiley, 1949. Disponível em: <http://www.citeulike.org/group/2050/article/1282862>.

HECHT N, R. Theory of the backpropagation neural network. In: **International Joint Conference on Neural Networks**. [s.n.], 1989. p. 593–605. Disponível em: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=118638](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=118638).

HOPFIELD, J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. Usa.*, National Acad Sciences, v. 79, n. 8, p. 2554, 1982. Disponível em: <http://www.pnas.org/content/79/8/2554.short>.

HÖRNER, R.; , Aline Mastella , Adenilde Salla , Roselene Alves Righi, B. M. . L. O. d. O. **Prevalência de microrganismos em infecções do trato urinário de**

**pacientes atendidos no Hospital Universitário de Santa Maria Prevalence of the microorganisms in urinary tract infections of patients attended in the Hospital.** 147–150 p. Tese (Doutorado), 2006.

JOHNSTONE, L. A practical approach to the implementation of autoverify. **Accredit. Qual. Assur.**, 2004. Disponível em: <http://www.springerlink.com/index/CDA4TJUE001PT88K.pdf>.

KELEMEN, J. From Artificial Neural Networks to Emotion Machines with Marvin Minsky. **Acta. Polytech. Hungarica.**, v. 4, n. 4, p. 5–16, 2007. Disponível em: [http://old.bmf.hu/journal/Kelemen\\\_12.pdf](http://old.bmf.hu/journal/Kelemen\_12.pdf).

KROENKE, K.; HANLEY, J.; COPLEY, J.; MATTHEWS, J.; DAVIS, C.; FOULKS, C.; CARPENTER, J. The admission urinalysis. **J. Gen. Intern. Med.**, Springer, v. 1, n. 4, p. 238–242, 1986. ISSN 0884-8734. Disponível em: <http://www.springerlink.com/index/G0645U67V4511P62.pdf>.

KÜHNEL, W. **Color atlas of cytology, histology, and microscopic anatomy.** [S.l.]: George Thieme Verlag, 2003. ISBN 1588901750.

LANGLOIS, M. R.; DELANGHE, J. R.; STEYAERT, S. R.; EVERAERT, K. C.; BUYZERE, M. L. de. Automated flow cytometry compared with an automated dipstick reader for urinalysis. **Clin. Chem.**, v. 45, n. 1, p. 118–22, jan. 1999. ISSN 0009-9147. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/9895347>.

LEONARD, J.; KRAMER, M. Improvement of the backpropagation algorithm for training neural networks. **Comput. Chem. Eng.**, Elsevier, v. 14, n. 3, p. 337–341, 1990. ISSN 0098-1354. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0098135490870706>.

LISBOA, P. A review of evidence of health benefit from artificial neural networks in medical intervention. **Neural. Networks.**, v. 15, n. 1, p. 11–39, jan. 2002. ISSN 08936080. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608001001113>.

LISBOA, P.; TAKTAK, A. The use of artificial neural networks in decision support in cancer: a systematic review. **Neural. Networks.**, Elsevier, v. 19, n. 4, p. 408–415, 2006. ISSN 0893-6080. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608005002844>.

LOHR, J. A.; PORTILLA, M. G.; GEUDER, T. G.; DUNN, M. L.; DUDLEY, S. M. Making a presumptive diagnosis of urinary tract infection by using a urinalysis performed in an on-site laboratory. **J. Pediatr.**, v. 122, n. 1, p. 22–5, jan. 1993. ISSN 0022-3476. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/8419610>.

LORITZ, D. Cerebral and cerebellar models of language learning. **Appl. Linguist.**, Am Assoc Appl Ling, v. 12, n. 3, p. 299, 1991. ISSN 0142-6001. Disponível em: <http://applij.oxfordjournals.org/content/12/3/299.abstract>.

MACHADO, M.; GONÇALVES, E.; LARGURA, M. Automação do exame de urina: comparação do Urisys 2400 com a rotina manual (Microscopia do sedimento urinário). **RBAC.**, v. 35, n. 4, p. 165–167, 2003. Disponível em:

[http://www.proead.unit.br/professor/jose\\\_melquiades/arquivos/atividades/Automacaodoexamedeurina.pdf](http://www.proead.unit.br/professor/jose\_melquiades/arquivos/atividades/Automacaodoexamedeurina.pdf).

MANONI, F.; FORNASIERO, L.; ERCOLIN, M.; TINELLO, A.; FERRIAN, M.; HOFFER, P.; VALVERDE, S.; GESSONI, G. Cutoff values for bacteria and leukocytes for urine flow cytometer Sysmex UF-1000i in urinary tract infections. *Diagn. Microbiol. Infect. Dis.*, Elsevier Inc., v. 65, n. 2, p. 103–7, out. 2009. ISSN 1879-0070. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/19748419>.

MARQUES, A.; PAULO, M. D. Diagnóstico auxiliado por computador na radiologia. *Radiol. Bras.*, v. 34, n. 5, p. 285–293, out. 2001. ISSN 0100-3984. Disponível em: [http://www.scielo.br/scielo.php?script=sci\\\_arttext&pid=S0100-39842001000500008&lng=pt&nrm=iso&tlng=pt](http://www.scielo.br/scielo.php?script=sci\_arttext&pid=S0100-39842001000500008&lng=pt&nrm=iso&tlng=pt).

MATHWORKS. **Matlab r2009a**. Inc, 2009. Disponível em: <http://www.mathworks.com/products/matlab/>.

MAZUROWSKI, M.; HABAS, P.; ZURADA, J.; LO, J.; BAKER, J.; TOURASSI, G. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural. Networks.*, Elsevier, v. 21, n. 2-3, p. 427–436, 2008. ISSN 0893-6080. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0893608007002407>.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biol.*, Springer, v. 5, n. 4, p. 115–133, 1943. ISSN 0092-8240. Disponível em: <http://www.springerlink.com/index/61446605110620KG.pdf>.

MELO, M.; MELO, K. de; FARIA, C. Utilidade de regras booleanas aplicadas à liberação de resultados de exames hormonais e marcadores tumorais. *J. Bras. Patol. Med. Lab.*, SciELO Brasil, v. 42, n. 4, p. 257–264, 2006. Disponível em: <http://www.scielo.br/pdf/jbpml/v42n4/a06v42n4.pdf>.

MUSEN, M.; SHAHAR, Y.; SHORTLIFFE, E. Clinical decision-support systems. *Biomed. Inf.*, Springer, p. 698–736, 2006. Disponível em: <http://www.springerlink.com/index/r28046876k610342.pdf>.

NETWORKX, P. networkx 1.4. . Disponível em: <http://pypi.python.org/pypi/networkx>.

NUMPY, P. **Python numpy**. Disponível em: <http://numpy.scipy.org/>.

OKADA, H.; HORIE, S.; INOUE, J. The basic performance of bacteria counting for diagnosis of urinary tract infection using the fully automated urine particle analyser UF 1000i. *Sysmex. J.*, v. 17, n. 2, p. 95–101, 2007. Disponível em: [http://www.centralizeddiagnostics.cl/Literatura/docs/literatura/Performance\\\_bacteria\\\_counting\\\_UF\\\_1000\\\_vol17\\\_2\\\_03.pdf](http://www.centralizeddiagnostics.cl/Literatura/docs/literatura/Performance\_bacteria\_counting\_UF\_1000\_vol17\_2\_03.pdf).

OTTENBACHER, K.; LINN, R.; SMITH, P.; ILLIG, S.; MANCUSO, M.; GRANGER, C. Comparison of logistic regression and neural network analysis applied to predicting living setting after hip fracture. *Ann. Epidemiol.*, Elsevier, v. 14, n. 8, p. 551–559, 2004. ISSN 1047-2797. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S1047279703003521>.

PENDERS, J. Quantitative evaluation of urinalysis test strips. *Clin. Chem.*, 2002. Disponível em: <http://www.clinchem.org/cgi/content/abstract/48/12/2236>.

PLATA, R.; SILVA, C.; YAHUITA, J. The first clinical and epidemiological programme on renal disease in Bolivia: a model for prevention and early diagnosis of renal diseases in the developing countries. *Nephrol. Dial.*, 1998. Disponível em: <http://ndt.oxfordjournals.org/content/13/12/3034.abstract>.

PLEBANI, M.; CARRARO, P. Mistakes in a stat laboratory: types and frequency. *Clin. Chem.*, v. 43, n. 8 Pt 1, p. 1348–51, ago. 1997. ISSN 0009-9147. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/9267312>.

PYTHON. **Python for Windows extensions**. Disponível em: <http://sourceforge.net/projects/pywin32/files/>.

PYTHON. **Python 2.6 Released: 01-Oct-2008**. 2008. Disponível em: <http://www.python.org/download/releases/2.6/>.

QUINLAN, J. Simplifying decision trees. *Int. J. Man. Mach. Stud.*, Elsevier, v. 27, n. 3, p. 221–234, 1987. ISSN 0020-7373. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0020737387800536>.

RAITZ, R. Free Associative Neurons Fan Uma Abordagem Para Reconhecimento de Padrões. , mar. 1997. Disponível em: <http://en.scientificcommons.org/7867623>.

RAITZ, R. **Universidade Federal de Santa Catarina Programa de Pós-Graduação em Engenharia de Produção FAN 2002 : UM MODELO NEURO-FUZZY PARA RECONHECIMENTO DE PADRÕES**. Tese (Doutorado), 2002.

Renal Cancer, B. **Renal Cancer Blog: urine test strips**. 2010. Disponível em: <http://kidney-renal-cancer.blogspot.com/2010/10/urine-test-strips-detect-blood-in-urine.html>.

ROGGEMAN, S.; ZAMAN, Z. by Combining Urine Flow Cytometer and Strip Results. *Am. J. Clin. Pathol.*, p. 872–878, 2001.

ROJAS, R. **Neural networks: a systematic introduction**. [S.l.]: Springer, 1996. ISBN 3540605053.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 1958. Disponível em: [http://www.ling.upenn.edu/courses/Fall\\\_2007/cogs501/Rosenblatt1958.pdf](http://www.ling.upenn.edu/courses/Fall\_2007/cogs501/Rosenblatt1958.pdf).

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Prentice hall, 2009. ISBN 0136042597.

SAMUELSON, W.; ZECKHAUSER, R. Status quo bias in decision making. *J. Risk. Uncertain.*, v. 1, n. 1, p. 7–59, mar. 1988. ISSN 0895-5646. Disponível em: <http://www.springerlink.com/index/10.1007/BF00055564>.

SANGRADOR, C. O.; REDONDO, F. C.; INVESTIGADOR, G. Utilidad de los distintos parámetros del perfil urinario en el diagnóstico de infección urinaria IN THE DIAGNOSIS OF URINARY TRACT INFECTIONS. *Ann. Pediatr.*, v. 67, n. 5, p. 450–460, 2007.

SCIPY. **python 2.6 scipy**. Disponível em: <http://www.scipy.org/Download>.

SIMERVILLE, J.; MAXTED, W.; PAHIRA, J. Urinalysis: a comprehensive review. **Am. Fam. Phys.**, v. 71, n. 6, p. 1153–62, 2005. Disponível em: <http://cchs.ua.edu/system/files/uploads/documents/fmr/curriculum/urinalysis.pdf>.

SIVASANKAR, E.; RAJESH, R.; VENKATESWARAN, S. Diagnosing Appendicitis Using Backpropagation Neural Network and Bayesian Based Classifier. **Int. J. Comput. Theor. Eng.**, v. 1, n. 4, p. 358–363, 2009. Disponível em: <http://www.ijcte.org/papers/056.pdf>.

STATSOFT. **Statistica V8.0**. Disponível em: <http://www.statsoft.com/>.

STRASINGER, S.; DI LORENZO, M.; WANG, B.; BONOMELLI, S. **Urinalysis and body fluids**. [S.l.]: FA Davis, 2008. ISBN 0803681038.

TANGRI, N.; ANSELL, D.; NAIMARK, D. Predicting technique survival in peritoneal dialysis patients: comparing artificial neural networks and logistic regression. **Nephrol. Dial. Transpl.**, ERA-EDTA, v. 23, n. 9, p. 2972, set. 2008. ISSN 0931-0509. Disponível em: <http://ndt.oxfordjournals.org/content/23/9/2972.abstract>.

TU, J. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomesâ 1. **J. Clin. Epidemiol.**, Elsevier, v. 49, n. 11, p. 1225–1231, 1996. ISSN 0895-4356. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0895435696000029>.

UETTWILLER-GEIGER, D. A labs strategy to reduce errors depends on automation. **Med. Lab. Obs.**, v. 37, n. 12, p. 26, 2005. ISSN 0580-7247. Disponível em: [http://www.mlo-online.com/articles/1205/1205lab\\\_mgmt.pdf](http://www.mlo-online.com/articles/1205/1205lab\_mgmt.pdf).

WEI, J. T.; ZHANG, Z.; BARNHILL, S. D.; MADYASTHA, K. R.; ZHANG, H.; OESTERLING, J. E. THE PRACTICING UROLOGIST. **Urol.**, v. 4295, n. 98.

WEST, D. Neural network credit scoring models. **Comput. Oper. Res.**, Elsevier, v. 27, n. 11-12, p. 1131–1152, 2000. ISSN 0305-0548. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0305054899001495>.

WOJCIECHOWSKI, M. ffnet: Feed-forward neural network for python. , 2007. Disponível em: <http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:ffnet\#5>.

ZHANG, Z.; FRIEDRICH, K. Artificial neural networks applied to polymer composites: a review. **Compos. Sci. Technol.**, Elsevier, v. 63, n. 14, p. 2029–2044, 2003. ISSN 0266-3538. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0266353803001064>.

## APÊNDICE A – INTERFACE GRÁFICA

A figura 11, mostra a interface gráfica construída para a validação do processo.

The screenshot displays a software interface for urine analysis. At the top, there is a 'Controle' field with a value of '1' and a 'tabela' link. Below this, a header section shows 'Modificado 0' and a timestamp '13:20'. The main data entry area includes fields for 'Codigo' (002233000104), 'Rank' (021403), and 'Idade' (61a). A 'Nome' field is present but empty. The test results are organized into a grid:

Densidade	1.014	pH	5	Est	0	Nit	NEGATIVO	Alb	NEGATIVO
Glicose	50	Acetona	NEGATIVO	Uro	NORMAL	Billi	NEGATIVO	Hem	10
Eritrocitos	2.80	Leu	7.70	Cel	5.70	Cll	1.35	Bac	2671.40

Below the grid, there is a 'Classe' field and a 'Rev' field. A list of findings is displayed in a scrollable area, including items like 'Bilirrubina confirmada', 'Excesso de celulas epiteliais', and 'Filamentos de muco'. At the bottom, there is a 'Gravar' button and navigation links for '<< Anterior' and 'Próximo >>'.

**FIGURA 11:** INTERFACE GRÁFICA INTUITIVA QUE APRESENTA OS DADOS DOS ENSAIOS DE URINÁLISE DESENVOLVIDO PARA PROPORCIONAR AOS PROFISSIONAIS A CLASSIFICAÇÃO DAS AMOSTRAS.

## APÊNDICE B – PASSO A PASSO PARA O SISTEMA DE APOIO À DECISÃO

Dentro do prompt do python e no mesmo diretório em que estiver o arquivo do sistema de apoio à decisão (sad.py), importa a classe sad para o ambiente e a instância:

```
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> from sad import sad
>>> sad = sad()
```

Apas, carrega-se os arquivos de treino, teste e validação, passando como argumento o nome do arquivo e o separador dos campos, neste caso a a tabulação ('\t'):

```
>>> sad.carregarTreino('reduzido/treino.txt', '\t')
-> Base de treino carregada de arquivo
>>> sad.carregarTeste('reduzido/teste.txt', '\t')
-> Base de teste carregada de arquivo
>>> sad.carregarvalidação('reduzido/validação.txt', '\t')
-> Base de validação carregada de arquivo
```



Exemplo de algumas linhas de um dos arquivos, onde cada um dos campos refere-se a uma característica: idade, sexo, esterase leucocitária, leucócitos, hemoglobina, eritrócitos e classe.

78	0	10	5.90	100	674.10	1
69	0	0	3.10	500	202.6	0
46	1	25	57.50	0	6.90	0
24	1	25	13.30	100	1024.20	1
56	1	0	3.10	0	37.70	1
76	1	10	16.00	100	21.60	0
48	1	0	5.10	25	10.5	0
24	1	10	9.40	500	257.1	0
2	1	0	5.10	0	10.30	1
77	1	50	8.80	25	27.30	1

Define-se as camadas ocultas da rede passando como parâmetro um array com a quantidade de perceptrons de cada camada. A primeira camada possui perceptrons conforme o número de características e a definido automaticamente quando se carrega o arquivo de treino e a última camada a sempre um, pois a o perceptrons que apresenta a classificação.

```
>>> sad.defCamadas([15,15])
>>> sad.camadas
[6, 15, 15, 1]
```

Depois dessas configurações básicas vem o treinamento da rede, que pode ser interrompido a qualquer momento. Durante o treinamento é apresentado a taxa de acertos da rede em relação a classificação dos arquivos:

```
>>> sad.treinar()
Início: 2010-06-23 10:15:55.867856
TREINANDO REDE.(pressione ENTER para parar o treinamento)
apocas: 1000 | acertos – Treino: 96.08% Teste: 95.74%
Validação: 90.70% /tempo: 0:13:20.107251
apocas: 2000 | acertos – Treino: 96.09% Teste: 95.73%
Validação: 90.29% /tempo: 0:26:42.369600
apocas: 3000 | acertos – Treino: 96.11% Teste: 95.75%
Validação: 90.50% /tempo: 0:40:05.227460
```

apocas: 4000 | acertos – Treino: 96.09%    Teste: 95.73%  
 Validação: 90.50% /tempo: 0:53:30.732171  
 parando treinamento ...

As matrizes de confusão, onde o sistema mostra os erros e acertos nas classificações, são mostradas através da função desempenho():

```
>>> sad.desempenho()
```

BASE DE TREINO com 17200 amostras

acertos: 16665 96.89% (erros: 535 3.11%)

Matriz de confusão:

	0	1
0	8185(47.59%)	415( 2.41%)
1	120( 0.70%)	8480(49.30%)

BASE DE TESTE com 17200 amostras

acertos: 16606 96.55% (erros: 594 3.45%)

Matriz de confusão:

	0	1
0	8175(47.53%)	425( 2.47%)
1	169( 0.98%)	8431(49.02%)

BASE DE VALIDAÇÃO com 17210 amostras

acertos: 16550 96.17% (erros: 660 3.83%)

Matriz de confusão:

	0	1
0	8131(47.25%)	475( 2.76%)
1	185( 1.07%)	8419(48.92%)

Para carregar dados para um banco de dados (no caso MySQL), primeiro define-se as informações para conexão com o servidor:

```
>>> sad.defConSQL()
```

Entre com o servidor SQL: localhost

Entre com o usuário: bioinfo

Entre com a senha: bioinfo

Entre com a base de dados: bioinfo

→ Conexão com banco de dados ok !

Com o teste de conexão funcionando, define-se então a tabela e os campos a serem buscados do banco de dados:

```
>>> sad.defCamposSQL()
```

Nome da Tabela: TB\_Resul

Já existe uma rede carregada com 6 características

Coluna da característica 1/6: DATE\_FORMAT(Date, '%Y')–

DATE\_FORMAT(DN, '%Y') –(DATE\_FORMAT(Date, '00-%m-%d') <

DATE\_FORMAT(DN, '00-%m-%d'))

Coluna da característica 2/6: SX

Coluna da característica 3/6: campo10

Coluna da característica 4/6: campo11

Coluna da característica 5/6: campo3

Coluna da característica 6/6: campo12

Entre com a coluna da classe: Campo22

Entre com a coluna do identificador: Sample\_NR

Caso deseje, entre com algum filtro SQL: where Campo22 = NULL

or Campo22 = ''

→ OK, query valida

Note que não há um campo com a idade do paciente na tabela. Então define-se a característica 1 com uma query que traz o cálculo deste valor. A coluna na classe onde está a classificação do exame ou o campo onde será armazenada a classificação realizada pela rede posteriormente. A coluna do identificador é o campo de identificação de cada exame para que seja corretamente atualizado. No caso do filtro SQL, foi solicitado apenas os exames que ainda não foram classificados. Também pode passar como parâmetro se deseja que os dados carregados sejam definidos como base de classificação (parâmetro definido por omissão), treino, teste ou validação.

O comando carregarDeSQL importa os dados do banco de dados conforme os parâmetro de conexão e de campos definidos:

```
>>> sad.carregarDeSQL()
```

→ Dados carregados do banco de dados na base de classificação

Após carregar a base de classificação, executa-se a classificação da mesma:

```
>>> sad.classificar()
-> Base de classificação classificada !
```

E, finalmente para salvar essa classificação no banco de dados, que servirá como sugestão de diagnóstico:

```
»> sad.salvarClassificadoSQL() ->Classificações salvas com sucesso !
```

Então, para que o sistema salve as configurações já executadas:

```
>>> sad.salvarEsquema('esquema')
->Rede salva em esquema.net
->Esquema salvo em esquema.esq
```

Toda vez que o esquema salvo for carregado, ele importará todas as configurações salvas e executará a busca de dados, classificação e salvará no banco de dados:

```
>>> sad.carregarEsquema('esquema')
->Base de treino carregada de arquivo
->Base de teste carregada de arquivo
->Base de validação carregada de arquivo
->Rede carregada
->Dados carregados do banco de dados na base de classificada
->Base de classificação classificada !
->Classificações salvas com sucesso !
```

Para automatizar ainda mais, a função de criar executável, monta um arquivo que basta apenas executá-lo para realizar toda a operação:

```
>>> sad.criarExecutavel('exe')
-> OK – Executavel exe.py criado !
```

Exemplo do executável criado:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from sad import sad
sad = sad()
sad.carregarEsquema('esquema')
```

## ANEXO A – SAD

Aqui está o código fonte do programa SAD (sistema de apoio à decisão) desenvolvido em python.

```

1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  from ffnet import ffnet , mlgraph , readdata , loadnet , savenet
5  from numpy import array
6  from datetime import datetime
7  import select , sys , MySQLdb , os
8
9  class sad() :
10
11     def __init__( self ) :
12         self.rede = ""
13         self.camadaInicial = ""
14         self.baseTreino = ""
15         self.camadas = ""
16         self.baseTeste = ""
17         self.baseValidacao = ""
18         self.conSQL = {}
19         self.baseSQL = ""
20         self.camposSQL = ""
21         self.arquivos = {}
22         self.baseClassificacao = ""
23         self.classificado = False
24         self.esquema = ""
25
26     def reiniciar( self ) :
```

```

27     self.__init__()
28
29     # salvar as configuracoes atuais em um arquivo para futura
30     recuperacao
31     def salvarEsquema(self , nome):
32         s = ''
33         if self.rede:
34             self.salvarRede(nome + '.net')
35             s += 'rede=' + nome + '.net\n'
36
37         if self.arquivos:
38             for i in self.arquivos.keys():
39                 s += i + '=' + self.arquivos[i][0] + ':' + self
40                 .arquivos[i][1] + '\n'
41
42         if self.baseSQL:
43             s += 'baseSQL=' + self.baseSQL + '\n'
44
45         if self.conSQL:
46             s += 'conSQL='
47             for i in self.conSQL.keys():
48                 s += i + ':' + self.conSQL[i] + ', '
49             s = s[:-1] + '\n'
50         if self.camposSQL:
51             s += 'camposSQL='
52             for i in self.camposSQL:
53                 s += i + ', '
54             s = s[:-1] + '\n'
55
56         res = open(nome+'.esq' , 'w')
57         res.write(s)
58         res.close
59         self.esquema = nome
60         print 'esquema salvo em %s.esq' % nome
61
62     # carregar as configuracoes de um esquema salvo em arquivo
63     def carregarEsquema(self ,nome):

```

```
61     self.__init__ ()
62     texto = open(nome+'.esq')
63     p = {}
64     for linha in texto:
65         linha = linha.replace('\n', '')
66         pos = linha.find('=')
67         p[linha[:pos]] = linha[(pos+1):]
68
69     if p.has_key('conSQL'):
70         c = p['conSQL'].rsplit(',')
71         for i in c:
72             i = i.rsplit(':')
73             self.conSQL[i[0]] = i[1]
74
75     for t in ['treino', 'teste', 'validacao', 'classificacao'
76             ]:
77         if p.has_key(t):
78             a, s = p[t].rsplit(':')
79             if t == 'treino':
80                 self.carregarTreino(a,s)
81             elif t == 'teste':
82                 self.carregarTeste(a,s)
83             elif t == 'validacao':
84                 self.carregarValidacao(a,s)
85             elif t == 'classificacao':
86                 self.carregarClassificacao(a,s)
87
88     if p.has_key('baseSQL'):
89         self.baseSQL = p['baseSQL']
90
91     if p.has_key('rede'):
92         self.carregarRede(p['rede'])
93
94     if p.has_key('camposSQL'):
95         self.camposSQL = p['camposSQL'].rsplit(',')
```

```

96
97     if self.baseSQL == 'classificada':
98         self.carregarDeSQL()
99         if self.baseClassificacao:
100             self.classificar()
101             self.salvarClassificadoSQL()
102
103     self.esquema = nome
104     #rede, arquivos, baseSQL, conSQL
105     #if dic.has_key():
106
107
108     # definir base de treino, mostrando o arquivo e o tipo de
109     separador das características
110     def carregarTreino(self, arquivo, separador=" "):
111         self.baseTreino = carregarDeArquivo(arquivo, separador)
112         #caso exista uma rede ja carregada, compara-se o numero
113         de características do arquivo com a da rede
114         if self.rede and len(self.baseTreino[0][0]) != self.
115         camadaInicial:
116             print 'N de características do arquivo diferente do
117             da rede carregada'
118             self.baseTreino = ''
119         else:
120             self.camadaInicial = len(self.baseTreino[0][0])
121             self.arquivos['treino'] = [arquivo, separador]
122             print 'Base de treino carregada de arquivo'
123             #se as camadas intermediarias ja estiverem definidas, e
124             inserida a primeira camada
125             if self.camadas:
126                 self.camadas.insert(0, self.camadaInicial)
127
128     def carregarTeste(self, arquivo, separador=" "):
129         self.baseTeste = carregarDeArquivo(arquivo, separador)
130         if len(self.baseTeste[0][0]) != len(self.baseTreino
131         [0][0]):

```



```

126         print 'Invalido – qtde de carac. diferente da base
           de treino '
127         self.baseTeste = ""
128     else:
129         self.arquivos['teste'] = [arquivo, separador]
130         print 'Base de teste carregada de arquivo '
131
132     def carregarValidacao(self, arquivo, separador=" "):
133         self.baseValidacao = carregarDeArquivo(arquivo,
           separador)
134         if len(self.baseValidacao[0][0]) != len(self.baseTreino
           [0][0]):
135             print 'Invalido – qtde de caracteristicas diferente
           da base de treino '
136             self.baseValidacao = ""
137         else:
138             self.arquivos['validacao'] = [arquivo, separador]
139             print 'Base de validacao carregada de arquivo '
140
141     def carregarClassificacao(self, arquivo, separador):
142         if self.baseSQL or self.baseClassificacao:
143             r = raw_input('Ja existe uma base carregada,
           sobrescrever ? (S)im ou (N)ao: ')
144         else:
145             r = 's'
146         if r.lower() == 's':
147             self.baseClassificacao = carregarDeArquivo(arquivo,
           separador, tipo='classificado')
148             self.arquivos['classificacao'] = [arquivo,
           separador]
149             print 'Base de classificacao carregada de arquivo '
150             self.classificado = False
151             for i in range(len(self.baseClassificacao[1])):
152                 self.baseClassificacao[1][i] = 0.
153         if self.baseSQL:
154             self.baseSQL = 'classificacao '

```

```

155
156 def salvarClassificado(self, arquivo, separador):
157     if self.classificado == True:
158         s = ''
159         for i in range(len(self.baseClassificacao[0])):
160             for j in range(len(self.baseClassificacao[0][i
161                 ])):
162                 s += self.baseClassificacao[0][i][j] +
163                     separador + ','
164                 if j == (len(self.baseClassificacao[0][i][j
165                     ]) - 1):
166                     s += self.baseClassificacao[1][j] + '\n
167                     ,
168
169         arq = open(arquivo, 'w')
170         arq.write(s)
171         arq.close()
172
173
174
175
176
177
178
179
180
181
182
183
184
185 def desempenho(self):
186     print 60 * '-'
187     print 'BASE DE TREINO',
188     if self.baseTreino:
189         testarRede(self.baseTreino, self.rede, 'desempenho')
190     else:
191         print ' - Nao ha base de treino carregada'
192     print 'BASE DE TESTE',
193     if self.baseTeste:
194         testarRede(self.baseTeste, self.rede, 'desempenho')
195     else:
196         print ' - Nao ha base de teste carregada'
197     print 'BASE DE VALIDACAO',
198     if self.baseValidacao:
199         testarRede(self.baseValidacao, self.rede, '
200             desempenho')
201     else:
202         print ' - Nao ha base de teste carregada'

```

```

186
187 def treinar(self, ciclos = 1000, tipo = 'tnc'):
188     #ciclos = 1000 #intervalo de epocas de treinamento para
189         avaliacao
189     if not self.baseTreino:
190         print 'Nao ha base de treino carregada'
191     elif not self.rede:
192         print 'Nao ha nenhuma rede treinada ou criada'
193     else:
194         treinar = True
195         #p = select.epoll()
196         #p.register(sys.stdin, select.EPOLLIN)
197         inicio = datetime.today()
198         print 'inicio: ', inicio
199         print 'TREINANDO REDE.(pressione ENTER para parar o
200             treinamento)'
201         it = 0
201     while treinar:
202         if tipo == 'tnc':
203             self.rede.train_tnc(self.baseTreino[0], self
204                 .baseTreino[1], maxfun=ciclos)
205         #elif tipo == 'rprop':
206         # self.rede.train_rprop(self.baseTreino[0],
207             self.baseTreino[1], maxfun=ciclos)
208         #r = p.epoll(1000)
209         it += 1
210         print 'ciclos:', it * ciclos, '| acertos -
211             Treino:',
212         testarRede(self.baseTreino, self.rede, 'treino')
213         if self.baseTeste:
214             print ' Teste:',
215             testarRede(self.baseTeste, self.rede, '
216                 treino')
217         if self.baseValidacao:
218             print ' Validacao:',

```

```

215         testarRede(self.baseValidacao, self.rede, '
                treino')
216         print '/tempo:', datetime.today() - inicio
217         #if len(r):
218         #    print 'parando treinamento...'
219         #    break
220
221     def salvarRede(self, arquivo):
222         if self.rede:
223             savenet(self.rede, arquivo)
224             print 'rede salva em ', arquivo
225         else:
226             print 'nao existe rede para salvar'
227
228     def carregarRede(self, arquivo):
229         self.rede = loadnet(arquivo)
230         self.camadaInicial = self.rede.inno[0]#define a camada
                inicial conforme a rede
231         print 'rede carregada'
232
233
234     # definicao das camadas ocultas da rede passado por um
                array
235     def defCamadas(self, camadas):
236         self.camadas = camadas
237         if self.camadaInicial:
238             self.camadas.insert(0, self.camadaInicial)
239         self.camadas.append(1)
240         self.rede = ffnet(mlgraph(tuple(self.camadas)))
241
242     def defConSQL(self, servidor = False, usuario = False,
                senha = False, base = False):
243
244         if not servidor:
245             servidor = raw_input('Entre com o servidor SQL: ')
246         if not usuario:

```

```

247         usuario = raw_input('Entre com o usuario: ')
248     if not senha:
249         senha = raw_input('Entre com a senha: ')
250     if not base:
251         base = raw_input('Entre com a base de dados: ')
252     try:
253         conexaoBD(servidor , usuario , senha , base)
254     except:
255         print 'falha na conexao, verifique os dados'
256     else:
257         self.conSQL = {'servidor':servidor , 'usuario':
258             usuario , 'senha':senha, 'base':base}
259         print 'conexao com banco de dados ok !'
260
261     # campos = [tabela , campo1, ... , campoN, campo
262     identificador ou campo classe, filtro sql]
263     def defCamposSQL(self , campos = False , base = '
264     classificacao'): #base pode ser classificacao , treino ,
265     teste ou validacao
266     if not self.conSQL:
267         print 'e necessario informar os dados de conexao
268         com .defConSQL()'
269     else:
270     if not campos:
271         campos = []
272         campos.append(raw_input('Nome da Tabela: '))
273         if not self.camadaInicial:
274             print 'Ainda nao ha uma rede ou base de
275             treino carregados'
276             self.camadaInicial = int(raw_input('Quantas
277             caracteristicas tem essa base ?:'))
278         else:
279             print 'Ja existe uma rede carregada com %i
280             caracteristicas'% self.camadaInicial
281         for i in range(self.camadaInicial):

```

```

274         pergunta = 'Coluna da caracteristica %i/%i:
                ' % (i+1, self.camadaInicial)
275         campos.append(raw_input(pergunta))
276         #if base == 'classificacao':
277
278         campos.append(raw_input('Entre com a coluna da
                classe: '))
279         campos.append(raw_input('Entre com a coluda do
                identificador: '))
280         campos.append(raw_input('Caso deseje, entre com
                algum filtro SQL: '))
281     c = conexaoBD(self.conSQL['servidor'], self.conSQL['
        usuario'], self.conSQL['senha'], self.conSQL['base
        '])
282     cursor = c.cursor()
283     query = montarQuerySQL(campos)
284     try:
285         cursor.execute(query)
286     except:
287         print 'Query invalida – verificar:'
288         print query
289     else:
290         print 'OK, query valida'
291         self.baseSQL = base
292         self.camposSQL = campos
293
294
295     def carregarDeSQL(self):
296         if not self.conSQL:
297             print 'e necessario informar os dados de conexao
                com .defConSQL()'
298         else:
299             try:
300                 conexaoBD(self.conSQL['servidor'], self.conSQL['
                usuario'], self.conSQL['senha'], self.conSQL['
                base'])

```



```

325         self.baseTeste = base
326     elif self.baseSQL == 'validacao':
327         self.baseValidacao = base
328     if self.baseSQL in self.arquivos.keys():
329         self.arquivos.pop(self.baseSQL)#
330         limpar o caminho do nome do
331         arquivo de classificacao
332     self.classificado = False
333     print 'Dados carregados do banco de
334     dados na base de', self.baseSQL
335
336 def salvarClassificadoSQL(self):
337     if self.baseSQL == 'classificada':
338         try:
339             conexaoBD(self.conSQL['servidor'], self.conSQL['
340             usuario'], self.conSQL['senha'], self.conSQL['
341             base'])
342         except:
343             print 'falha na conexao, verifique os dados e
344             conexoes'
345         else:
346             c = conexaoBD(self.conSQL['servidor'], self.
347             conSQL['usuario'], self.conSQL['senha'], self.
348             conSQL['base'])
349             cursor = c.cursor()
350             for i in range(len(self.baseClassificacao[0])):
351                 sql = 'update %s set %s = %i where %s = "%s
352                 ";' % ( self.camposSQL[0], self.
353                 camposSQL[-3], self.baseClassificacao
354                 [1][i] , self.camposSQL[-2], self.
355                 baseClassificacao[2][i] )
356                 cursor.execute(sql)
357             print 'classificacoes salvas com sucesso !'
358     else:

```



```

347         print 'A base sql carregada ainda nao foi
           classificada. Ver .classificar()'
348
349
350
351     def classificar(self):
352         if not self.rede:
353             print 'Nao ha uma rede treinada ou carregada para
           classificar dados.'
354         elif not self.baseClassificacao:
355             print 'Nao ha base de classificacao carregada. Ver
           .carregarDeSQL() ou .carregarClassificacao()'
356         else:
357             #self.baseClassificacao[1] = classificarBase(self.
           baseClassificacao, self.rede)
358             a,b = self.rede.test(self.baseClassificacao[0], self
           .baseClassificacao[1], iprint=0)
359             res = []
360             for i in range(len(a)):
361                 self.baseClassificacao[1][i] = int(round(a[i
           ][0]))
362                 #res.append(int(round(i[0])))
363                 #res.append(round(i[0],2))
364             if self.baseSQL == 'classificacao':
365                 self.baseSQL = 'classificada'
366             self.classificado = True
367             print 'Base de classificacao classificada !'
368
369
370     def montarQuerySQL(self):
371         query = 'select '
372         for i in range(1, (len(self.camposSQL) - 1) ):
373             query += self.camposSQL[i] + ', '
374         # inserir o nome da tabela e o filtro sql na query
375         query = query[0:-2] + ' from ' + self.camposSQL[0] + '
           ' + self.camposSQL[-1] + ' ;'

```

```

376         return query
377
378     def criarExecutavel(self, esquema = ''):
379         if not self.esquema or esquema:
380             self.esquema = esquema
381         t = "#!/usr/bin/python\n# -*- coding: utf-8 -*-\nfrom
           sad import sad\nsad = sad()\nsad.carregarEsquema('%s
           ')" % self.esquema
382         arq = open(self.esquema+'.py', 'w')
383         arq.write(t)
384         arq.close
385         cmd = 'chmod 770 '+ self.esquema + '.py'
386         os.system(cmd)
387         print 'OK - Executavel', self.esquema + '.py criado !'
388
389     def formatarParaFAN(self):
390         s = ''
391         if self.baseClassificacao:
392             for l in self.baseClassificacao[0]:
393                 for i in range(len(l)):
394                     s += '%s,' % l[i]
395                 s = s[:-1]
396                 s += '\n'
397             s = s[:-1]
398             arq = open('fan.txt', 'w')
399             arq.write(s)
400             arq.close()
401         else:
402             print 'nao ha base de classificacao para formatar
           para FAN'
403
404
405
406     def classificarFAN(self, redeFAN):
407
408         self.formatarParaFAN()

```

```

409     cmd = 'java -jar SLQuery.jar -i fan.txt -f %s -o
        classFAN.txt > /dev/null' % redeFAN
410     os.system(cmd)
411     arq = open('classFAN.txt')
412     res = []
413     for linha in arq:
414         if linha != '<classes>\n' and linha != '</classes>\n':
415             linha = linha.rsplitt('</nome>')
416             linha = linha[0].rsplitt('<nome>')
417             #print linha, linha[1]
418             res.append(float(linha[1]))
419
420     for i in range(len(self.baseClassificacao[1])):
421         #print i, len(res), len(self.baseClassificacao[1]),
            self.baseClassificacao[1][i], res[i]
422         self.baseClassificacao[1][i] = res[i]
423     self.classificado = True
424     print 'Ok - Base de Classificacao classificada na rede
        FAN'
425
426
427
428     #####FUNCOES AUXILIARES#####
429     def montarQuerySQL(campos):
430         query = 'select '
431         for i in range(1, (len(campos) - 1) ):
432             query += campos[i] + ', '
433         # inserir o nome da tabela e o filtro sql na query
434         query = query[0:-2] + ' from ' + campos[0] + ' ' + campos
            [-1] + ' ;'
435         return query
436
437     def carregarDeArquivo(arq, sep=" ", tipo = 'classificado'):
438         texto = open(arq)
439         res = []

```

```

440     for linha in texto:
441         linha = linha.replace("\n","")
442         linha = linha.rsplit(sep)
443         if linha[0] == "":
444             linha.pop(0)
445         if not tipo == 'classificado':
446             linha.append(0)
447         for i in range(0,len(linha)):
448             try:
449                 linha[i] = float(linha[i])
450             except ValueError:
451                 print 'Atencao, o conjunto carregado possui
452                     strings e \
453                     rede aceita apenas inteiros ou decimais'
454                 continue
455             res.append(linha)
456     texto.close()
457     #introduzir um tratamento de conversao de string para
458     inteiro ou float, pois a rede nao suporta strings
459     dados = array(res)
460     c1 = dados[:,:(len(dados[0])-1)]
461     c2 = dados[:, -1]
462     #print 'camada inicial deve ter %i neuronios' % (len(dados
463     [0])-1)
464     return c1,c2
465
466 def testarRede(dados, rede, tipo):
467     a,b = rede.test(dados[0],dados[1],iprint=0)
468     #print b
469     #return a
470     for i in range(len(a)):
471         a[i] = round(a[i][0])
472     ## a - dados do arquivo classificado
473
474     # classesTeste - possui todas as classes do arquivo a ser
475     testado

```

```

472     c = set(a)
473     classesTeste = []
474     for i in c:
475         classesTeste.append(int(i))
476     classesTeste.sort()
477
478     # classesTreino – possui todas as classes treinadas
479     c = set(dados[1])
480     classesTreino = []
481     for i in c:
482         classesTreino.append(int(i))
483     classesTreino.sort()
484
485     #caso as classes do arquivo testado extrapolem as do treino
486     (valores maiores ou menores)
487     #as classes das extremidades sao arredondadas para o valor
488     mais proximo
489     for i in range(len(a)):
490         if a[i] < classesTreino[0]:
491             a[i] = classesTreino[0]
492         if a[i] > classesTreino[len(classesTreino) - 1]:
493             a[i] = classesTreino[len(classesTreino) - 1]
494
495     matrizConfusao = []
496     for i in range(len(classesTreino)):
497         matrizConfusao.append([])
498         for j in range(len(classesTreino)):
499             matrizConfusao[i].append(0)
500
501     erros = 0
502     acertos = 0
503     linhasErradas = []
504
505     for i in range(len(a)):
506         for z in range(len(classesTreino)):
507             if classesTreino[z] == int(dados[1][i]):

```

```

506         x = z
507         if classesTreino[z] == int(a[i]):
508             y = z
509         matrizConfusao[x][y] += 1
510         if dados[1][i] == a[i]:
511             acertos += 1
512
513         else:
514             erros += 1
515             linhasErradas.append(i+1)
516
517     if tipo == 'treino':
518         print '%.2f%%' % (float(acertos)/len(a) * 100),
519
520     if tipo == 'desempenho':
521         #print 'classes: treinadas =', classesTreino,
522         #print 'resultantes =', classesTeste
523         print 'com %i amostras' % len(a)
524         print 'acertos: %i' % acertos, '%.2f%%' % (float(
525             acertos)/len(a) * 100),
526         print '(erros: %i' % erros, '%.2f%%)' % (float(erros)/
527             len(a) * 100)
528         #print 'linhas Erradas: ', len(linhasErradas), ' - ',
529             linhasErradas
530         print 'Matriz de confusao:'
531         print ' ',
532         for s in classesTreino:
533             print '%15i' % s,
534         print ''
535         l = 0
536         for i in matrizConfusao:
537             print '%3i' % classesTreino[l],
538             l += 1
539             for j in i:
540                 print '%6i(%6.2f%%)' % (j, float(j)/len(a) *
541                     100),

```

```

538         print ' '
539         print 60 * '–'
540
541 def conexaoBD(servidor , usuario , senha , base) :
542     # inserir aqui os dados de conexao com BD – servidor ,
543     # usuario , senha , base de dados
544     return MySQLdb.connect(servidor , usuario , senha , base)
545
546 def carregarBD(query , cursor , base = 'classificacao') :
547     #c = conexaoBD()
548     #cursor = c.cursor()
549     #sql = "select DATE_FORMAT(Date, '%Y')–DATE_FORMAT(DN, '%Y')
550     #       –(DATE_FORMAT(Date, '00–%m–%d ') < DATE_FORMAT(DN, '00–%m–%d
551     #       ')) as id , SX, campo10, campo11, campo3, campo12,
552     #       Sample_NR from TB_Resul"
553     cursor.execute(query)
554     r = cursor.fetchall()
555     #c.close()
556     dados = []
557     classe = []
558     chave = []
559     for l in range(len(r)) :
560         campos = []
561         for i in range(len(r[l])–2) :
562             campos.append(r[l][i])
563         dados.append(campos)
564         if base == 'classificacao' or base == 'classificada' :
565             classe.append(0)
566         else :
567             classe.append(int(r[l][–2]))
568             chave.append(r[l][len(r[l])–1])
569     #substituir os valores que sao strings por valores inteiros
570     #como sexo (M=0/F=1) e NEGATIVO=0
571     for i in range(len(dados)) :
572         for j in range(len(dados[i])) :

```

```

569         try :
570             dados[i][j] = float(dados[i][j])
571         except ValueError:
572             #continue
573             if dados[i][j] == 'NEGATIVO':
574                 dados[i][j] = 0.
575             elif dados[i][j] == 'M':
576                 dados[i][j] = 0.
577             elif dados[i][j] == 'F':
578                 dados[i][j] = 1.
579         return array(dados), array(classe), array(chave)
580
581 #funcao para salvar os dados ja classificados em um banco de
582 dados
583 def salvarBD(dados): #m e o multiplicador , para diferenciar
584 exames
585     c = conexaoBD()
586     cursor = c.cursor()
587     for i in range(len(dados[0])):
588         sql = 'update TB_Resul set Campo22 = %i where Sample_Nr
589             = "%s";' % ( dados[1][i] ,dados[2][i] )
590         cursor.execute(sql)
591
592 def classificarBase(dados, rede):
593     a,b = rede.test(dados[0],dados[1],iprint=0)
594     res = []
595     for i in a:
596         res.append(int(round(i[0])))
597         #res.append(round(i[0],2))
598     return array(res)
599
600 classificacÃ£o
601 '''
602 from tcc import tcc
603 a = tcc()
604 a.carregarTreino('reduzido/treino.txt', '\t')
605 a.carregarTeste('reduzido/teste.txt', '\t')

```



```
602 a.carregarValidacao('reduzido/validacao.txt', '\t')
603 a.defCamadas([15,15])
604 a.treinar()
605 '''
```