

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO PARA PREVISÃO DO MERCADO DE AÇÕES
UTILIZANDO BANDAS DE BOLLINGER

ADRIANO CASSANIGA PETRY

BLUMENAU
2010

2010/2-01

ADRIANO CASSANIGA PETRY

PROTÓTIPO PARA PREVISÃO DO MERCADO DE AÇÕES

UTILIZANDO BANDAS DE BOLLINGER

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Prof. Roosevelt dos Santos Junior - Orientador

**BLUMENAU
2010**

2010/2-01

PROTÓTIPO PARA PREVISÃO DO MERCADO DE AÇÕES UTILIZANDO BANDAS DE BOLLINGER

Por

ADRIANO CASSANIGA PETRY

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Roosevelt dos Santos Junior – Orientador, FURB

Membro: _____
Prof. Fernando dos Santos – FURB

Membro: _____
Prof. Roberto Heinzle, Titulação – FURB

Blumenau, 13 de dezembro de 2010

Dedico este trabalho a minha família, amigos e especialmente a todos que de alguma forma me contribuíram durante a sua realização.

AGRADECIMENTOS

À minha família, que esteve sempre presente em minha vida.

Aos meus amigos, por sempre me apoiarem nos momentos difíceis.

À minha namorada, Naiara, pela motivação e compreensão nas horas difíceis.

Ao meu orientador, Roosevelt dos Santos Junior, por aceitar o desafio e acreditar na conclusão deste trabalho.

RESUMO

Este trabalho descreve o desenvolvimento de um protótipo para previsão do mercado de ações, utilizando o indicador Bandas de Bollinger como peça fundamental na mineração de dados. Para realizar as previsões, são utilizadas redes neurais artificiais com os algoritmos *Backpropagation*, *Linear Regression* e *Learning Vector Quantisation*. Os resultados são apresentados através de um relatório, onde são englobados os desempenhos dos algoritmos. Além disso, os dados reais e as previsões podem ser visualizados e analisados graficamente, de maneira individual ou em conjunto.

Palavras-chave: Inteligência artificial. Redes neurais artificiais. Bandas de Bollinger. Bolsa de valores. Mercado de ações.

ABSTRACT

This work describes the development of a prototype to predict a stock market, using the indicator Bollinger Bands as a key in data mining. To make predictions, artificial neural networks are used with the algorithms Backpropagation, Linear Regression and Learning Vector Quantisation. The results are presented through a report, which are addressed by the performances of the algorithms. Moreover, the actual data and forecasts can be graphically displayed and analyzed, either individually or together.

Key-words: Artificial intelligence. Artificial neural networks. Bollinger Bands. Stock exchanges. Stock market.

LISTA DE ILUSTRAÇÕES

Figura 1 - Alicerces da análise fundamentalista.....	15
Figura 2 – Tendências.....	18
Figura 3 – Suportes e resistências.....	19
Figura 4 – Gráfico de linha.....	20
Figura 5 – Gráfico de barras e volume	20
Figura 6 – Gráfico de <i>candlesticks</i> obtido através da ferramenta Operador	20
Figura 7 – Gráfico de cotações da Petrobras com bandas de Bollinger obtido através da ferramenta Operador.....	22
Figura 8 – Gráfico de cotações da Petrobras com bandas de Bollinger identificando pontos de super valorização e desvalorização obtido através da ferramenta Operador.....	23
Quadro 1 – Fórmulas utilizadas no cálculo das BBs.....	24
Figura 9 – Gráfico com análise das BBs combinado com o fluxo de volume obtido através da ferramenta Profitchart.....	25
Figura 10 – Análise das BBs combinada com o IFR.....	26
Quadro 2 – Código resumido demonstrando o cálculo do IFR.....	27
Quadro 3 – Código demonstrando os cálculos para definir as BBs	28
Figura 11 – Exemplos de falhas nas BBs obtidas através da ferramenta Operador	29
Figura 12 – Tendências de uma ação em relação as BBs obtidas através da ferramenta Operador	30
Quadro 4 – Tendências de uma ação em relação as BBs	30
Figura 13 – Partes do neurônio.....	32
Figura 14 – Diagrama esquemático de um neurônio artificial	33
Figura 15 – Rede alimentada adiante ou acíclica com uma única camada de neurônios.....	34
Figura 16 – Rede alimentada adiante ou acíclica totalmente conectada com uma camada oculta e uma camada de saída.....	35
Figura 17 – Rede recorrente sem laços de auto-realimentação e sem neurônios ocultos.....	36
Figura 18 – Diagrama de casos de uso	40
Quadro 5 – Detalhamento do caso de uso UC01 - Importar arquivo texto com histórico de cotações.....	40
Quadro 6 – Detalhamento do caso de uso UC02 - Definir parâmetros para o cálculo dos indicadores.....	41

Quadro 7 – Detalhamento do caso de uso UC03 - Treinar a rede neural.....	41
Quadro 8 – Detalhamento do caso de uso UC04 - Treinar a rede neural.....	41
Figura 19 – Diagrama de classes	42
Figura 20 – Bibliotecas adicionadas no projeto.....	44
Figura 21 – Depuração do algoritmo LVQ.....	45
Quadro 9 – LR: testes realizados com o parâmetro <code>attributeSelectionMethod</code>	46
Quadro 10 – LR: testes realizados com o parâmetro <code>eliminateColinearAttributes</code>	46
Quadro 11 – Código demonstrando a parametrização, treinamento e classificação utilizando LR	46
Quadro 12 – LVQ: testes realizados com o parâmetro <code>initialisationMode</code>	47
Quadro 13 – LVQ: testes realizados com o parâmetro <code>learningFunction</code>	48
Quadro 14 – LVQ: testes realizados com o parâmetro <code>learningRate</code>	48
Quadro 15 – LR: testes realizados com o parâmetro <code>useVoting</code>	48
Quadro 16 – Código demonstrando a parametrização, treinamento e classificação utilizando LVQ	49
Quadro 17 – MLP: testes realizados com o parâmetro <code>hiddenLayers</code>	50
Quadro 18 – MLP: testes realizados com o parâmetro <code>learningRate</code>	50
Quadro 19 – MLP: testes realizados com o parâmetro <code>momentum</code>	50
Quadro 20 – MLP: testes realizados com o parâmetro <code>trainingTime</code>	50
Quadro 21 – Código demonstrando a parametrização, treinamento e classificação utilizando MLP	51
Figura 22 – Histórico de cotações em arquivo texto	52
Figura 23 – Tela do protótipo para inserir os papéis	53
Figura 24 – Tela do protótipo com o gráfico da Petrobras, BB's e média móvel	54
Figura 25 – Tela do protótipo com análise do valor de fechamento da Vale do Rio Doce.....	55
Figura 26 – Tela do protótipo com análise de tendência da Vale do Rio Doce	56
Quadro 22 – Desempenho da ação da Petrobras testada no período de 1998 a 2010	57
Quadro 23 – Desempenho da ação da Vale do Rio Doce testada no período de 1998 a 2010.	58

LISTA DE SIGLAS

BB – Bandas de Bollinger

CVM – Comissão de Valores Mobiliários

IFR – Índice de Força Relativa

LR – *Linear Regression*

LVQ – *Learning Vector Quantisation*

MLP – *Multi Layer Perceptron*

RF – Requisito Funcional

RNA – Rede Neural Artificial

RNF – Requisito Não Funcional

SA – Sociedade Anônima

UML – *Unified Modeling Language*

WEKA - *Waikato Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	13
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 MERCADO DE AÇÕES.....	14
2.1.1 Análise fundamentalista.....	15
2.1.1.1 Análise da empresa	16
2.1.1.2 Análise da indústria	16
2.1.1.3 Análise geral da economia.....	17
2.1.2 Análise técnica	17
2.1.2.1 Tendências	18
2.1.2.2 Suportes e resistências	18
2.1.2.3 Volume	19
2.1.2.4 Gráficos.....	19
2.1.2.5 Indicadores.....	21
2.2 BANDAS DE BOLLINGER.....	22
2.2.1 Cálculo das BBs	24
2.2.2 Análise combinada com volume	24
2.2.3 Análise combinada com IFR.....	25
2.2.4 Cálculo das cotações	26
2.3 REDES NEURAIS ARTIFICIAIS.....	31
2.3.1 Neurônio biológico.....	31
2.3.2 Neurônio artificial	32
2.3.3 Arquiteturas de rede	33
2.3.3.1 Redes alimentadas adiante com camada única	33
2.3.3.2 Redes alimentadas diretamente com múltiplas camadas.....	34
2.3.3.3 Redes recorrentes.....	35
2.3.4 Treinamento	36
2.3.5 Técnicas utilizando mineração de dados e algoritmos de redes neurais	37
2.4 TRABALHOS CORRELATOS	38
2.4.1 Desenvolvimento de um protótipo para predição de índice de ações	38

2.4.2	Previsão do mercado de ações brasileiro utilizando redes neurais artificiais	38
3	DESENVOLVIMENTO DO PROTÓTIPO	39
3.1	REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	39
3.2	ESPECIFICAÇÃO	39
3.2.1	Diagrama de casos de uso	40
3.2.2	Diagrama de classes	42
3.3	IMPLEMENTAÇÃO	43
3.3.1	Técnicas e ferramentas utilizadas.....	43
3.3.1.1	IKVM.NET	44
3.3.1.2	Linear Regression	45
3.3.1.3	Learning Vector Quantisation.....	47
3.3.1.4	Multi Layer Perceptron.....	49
3.3.2	Operacionalidade da implementação	51
3.3.2.1	Preparação dos dados.....	52
3.3.2.2	Análise das ações	53
3.4	RESULTADOS E DISCUSSÃO	56
4	CONCLUSÕES.....	59
4.1	EXTENSÕES	59
	REFERÊNCIAS BIBLIOGRÁFICAS	61

1 INTRODUÇÃO

Nos dias atuais, os investimentos relacionados às bolsas de valores vêm se tornando cada vez mais populares e acessíveis. Através de sistemas denominados *home brokers*, é possível investir e acompanhar as ações de qualquer lugar do mundo através da internet. Porém, como ações são investimentos de renda variável, torna-se necessário que o investidor saiba como o mercado funciona, e como ele deve analisar a ação ao qual está prestes a investir (CALDERARO, 2009).

Uma característica comum a todas as bolsas de valores é a incerteza quanto ao comportamento do valor das suas ações. A avaliação do investimento em ações é explicada pelas correntes fundamentalista e técnica. A análise fundamentalista baseia-se na análise econômica e financeira de uma companhia, enquanto que a análise técnica assume que os preços das ações apresentam tendência de movimento condicionada a uma dependência significativa dos preços observados no passado (MUELLER, 1996, p. 7).

Segundo Baião (2008), entre os indicadores utilizados na análise técnica, as Bandas de Bollinger (BB) estão entre as mais utilizadas pelos analistas técnicos. Criado por John Bollinger em 1980, elas mantêm uma relação com a variação dos preços das ações, podendo ajudar a antecipar movimentos e identificar pontos de compra e venda. Porém a maior dificuldade em analisar as BBs é reconhecer suas tendências para prever os próximos movimentos do mercado.

As tendências e padrões de uma ação são analisados pelo histórico das cotações, e podem ser explorados através de uma solução computacional. A área de reconhecimento de padrões, objetivo maior de uma Rede Neural Artificial (RNA), resolve uma vasta classe de problemas reais como: processamento de voz, processamento de imagem, processamento de conhecimento inexato, processamento de linguagem natural, previsão e otimização, entre outros, sendo interessante ressaltar que muitos destes problemas ainda não possuem soluções satisfatórias dentro dos métodos convencionais existentes (ABELEM, 2004, p. 2).

Desta forma, com base no contexto acima, tem-se como meta desenvolver um protótipo que utilizará as BBs como base para realizar a previsão das tendências das ações, e RNAs para aprender o comportamento das BBs a partir do histórico das cotações. Para isso, será utilizado o histórico das empresas Petrobras e Vale do Rio Doce para constituir as BBs, que por sua vez servirão como entrada para criação de RNAs. Por fim, o resultado obtido com as RNAs será comparado com as tendências reais, a fim de obter um acerto favorável para

aplicação em operações reais.

1.1 OBJETIVOS DO TRABALHO

Este trabalho tem por objetivo desenvolver um protótipo que viabilize a utilização de RNAs para prever movimentos futuros de ações na bolsa de valores, utilizando como entradas as BBs.

Os objetivos específicos do trabalho são:

- a) desenvolver uma base de dados com as cotações, Índice de Força Relativa (IFR), média móvel e BBs das empresas Petrobras e Vale do Rio Doce;
- b) criar RNAs com diferentes configurações, utilizando as BBs como entrada;
- c) utilizar a combinação de parâmetros como o volume negociado no dia, IFR, média móvel e valor de fechamento para auxiliar no treinamento e análise das RNAs;
- d) desenvolver o protótipo utilizando as RNAs, para detectar se a tendência indicada pela BB será verdadeira ou falsa;
- e) analisar a eficiência dos resultados obtidos com as RNAs, em comparação com o sucesso das BBs nas tendências reais.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos: introdução, fundamentação teórica, desenvolvimento e conclusão.

O capítulo 2 apresenta as tecnologias envolvidas e os aspectos teóricos estudados para o desenvolvimento deste trabalho. São relatados temas como o mercado de ações, conceitos e informações a respeito da análise das BBs nas cotações, RNAs, tecnologias envolvidas e trabalhos correlatos a este.

No capítulo 3 é abordado o desenvolvimento do presente trabalho, detalhando a especificação e implementação.

O capítulo 4 apresenta-se as conclusões deste trabalho, bem como as sugestões para possíveis extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os assuntos e técnicas utilizadas para o desenvolvimento e entendimento deste trabalho. Na seção 2.1 são expostas informações sobre como funciona o mercado de ações. Na seção 2.2 são explanados conceitos a respeito de como funcionam as BBs, e como elas são utilizadas em análises de ações. Na seção 2.3 são abordadas as RNAs, suas arquiteturas de rede e técnicas de treinamento. Por fim, na seção 2.4, são descritos trabalhos correlatos.

2.1 MERCADO DE AÇÕES

Segundo Gonçalves (2008, p. 23), a regra número um para entender o mercado de ações é saber o que é uma bolsa de valores e sua finalidade. Pode-se dizer que a bolsa é um ambiente onde ações de diferentes empresas são compradas e/ou vendidas. Quem concretiza as operações do mercado acionário são as corretoras de valores, que recebem ordens diretamente de seus clientes. Além de funcionar como um elo entre pessoas interessadas em investir em ações e empresas que precisam de recursos para crescer, a Bolsa é um centro seguro, rápido e transparente de intermediação de negociações.

Ações são títulos emitidos por uma empresa com o objetivo de captar recursos financeiros. Uma ação representa a menor parcela em que se divide o capital de uma empresa constituída como Sociedade Anônima (SA). O acionista passa a ser “dono” de uma parte daquela empresa, tornando-se sócio do negócio e assumindo os riscos. Os acionistas somente terão ganhos se a empresa tiver lucros (e distribuir dividendos) e/ou se a ação tiver uma valorização no mercado.

Os preços são formados através da dinâmica das forças de oferta e procura de cada ação. A maior ou menor oferta e procura por uma ação está diretamente relacionada ao comportamento histórico dos preços e, sobretudo, às perspectivas futuras da empresa emissora, incluindo-se aí sua política de dividendos, prognósticos de expansão de seu mercado e dos seus lucros, influência da política econômica sobre as atividades da empresa, etc.

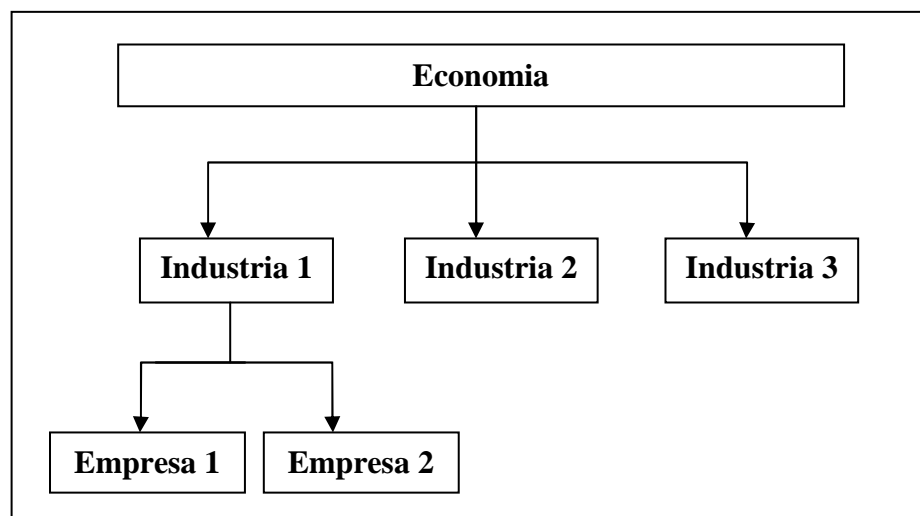
Segundo Montanari (2009), existem duas análises que estudam o comportamento do

mercado e auxiliam o investidor na escolha das ações e no melhor momento para comprar ou vender. São a análise fundamentalista e a análise gráfica ou técnica.

2.1.1 Análise fundamentalista

Análise fundamentalista é o estudo dos fatores que afetam as situações de oferta e demanda de um mercado, com o objetivo de determinar o valor intrínseco de um ativo. Através dessa análise, o analista está apto a comparar o preço encontrado com o preço do mercado e classificá-lo como sobre-avaliado com sinalização de venda, sub-avaliado com sinalização de compra ou que seu preço é condizente (justo) com o praticado pelo mercado. Em vista disso, a análise fundamentalista define qual o ativo deve ser comprado ou vendido. Esta análise geralmente está atrelada a grandes movimentos no longo prazo e não prediz o tempo certo para entrar ou sair do mercado (CHAVES, 2004, p. 16).

Segundo Winger e Frasca (1995, pg. 216), a análise fundamentalista tem em seu alicerce a análise de três fatores: análise da empresa, análise da indústria em que a empresa está inserida e a análise geral da economia, conforme exemplificado na Figura 1. É essencial que estes três fatores sejam analisados, não importando a ordem de suas utilizações. Apenas como ressalva, esses fatores estão relacionados ao ativo escolhido para o estudo, portanto, somente no caso de avaliação de ações esses três fatores serão considerados. Para os demais ativos, a análise econômica balizará o estudo.



Fonte: Winger e Frasca (1995, p. 237).

Figura 1 - Alicerces da análise fundamentalista

2.1.1.1 Análise da empresa

A análise da empresa é aplicada somente no mercado de ações para precificação das ações de uma empresa. Essa análise exige uma grande quantidade de informações sobre a empresa analisada. As principais fontes de dados para essa análise são: os balanços contábeis onde estão detalhados ativos e passivos da empresa e seu valor contábil no momento da divulgação; e as demonstrações de resultados que oferecem as despesas e receitas da empresa além dos resultados obtidos no período. No Brasil, para as empresas de capital aberto, a divulgação desses dados é obrigatória assim como, tem sua periodicidade de publicação definida e regulamentada pela Comissão de Valores Mobiliários (CVM), órgão fiscalizador do mercado (CHAVES, 2004, p. 17).

Através dessas duas fontes de dados, os analistas podem derivar outras informações e balizar seu estudo, todavia, devem levar em consideração que esses demonstrativos são elaborados sob a ótica contábil, regulamentada por leis e convenções inerentes ao país onde está situada a empresa analisada, o que pode ocasionar uma avaliação equivocada dos valores reais da empresa.

2.1.1.2 Análise da indústria

A indústria representa o conjunto de empresas de um mesmo setor onde está inserida a empresa analisada. Essa indústria é a conexão entre a própria empresa com toda a economia. O fundamentalista frequentemente desenvolve previsões para as indústrias que irão sustentar suas projeções para o ativo objeto. A análise da indústria ou mercado consiste em três atividades: correlacionar o desempenho da indústria com o desempenho da economia, isto é, identificar os ciclos econômicos e sua relação com a indústria; encontrar novos desenvolvimentos dentro da indústria que possam alavancar vendas ou revolucionar o mercado atual; e por fim, correlacionar o desempenho da indústria com o desempenho da empresa (CHAVES, 2004, p. 18).

A obtenção desses dados para a análise é complicada e apresenta um custo elevado. Atualmente, estes serviços são prestados pelas grandes corretoras e agências especializadas, que realizam estudos nas diferentes indústrias de forma detalhada e aprofundada os quais permitem aos fundamentalistas delinear suas análises para a empresa em questão.

2.1.1.3 Análise geral da economia

Na análise fundamentalista, os impactos da economia para com a empresa ou ativo objeto são essenciais. Qualquer previsão estará baseada nas projeções econômicas tanto do país onde está inserido como do mercado analisado. Até mesmo a análise econômica mundial e de seus principais participantes devem ser consideradas. Para isso, o fundamentalista deve estar atento às publicações sobre a economia a fim de obter conhecimento geral de como a economia está e eventualmente captar as ideias dos principais especialistas do mercado. Dois métodos podem ajudar nessa tarefa: acompanhar as publicações governamentais, dentre os quais estão os relatórios de indicadores econômicos e analisar diferentes opiniões de especialistas a fim de alcançar um consenso sobre a perspectiva da economia (CHAVES, 2004, p. 19).

2.1.2 Análise técnica

Segundo Montanari (2009), a análise técnica visa estudar gráficos, formados a partir de cálculos estatísticos, para identificar um determinado padrão histórico no comportamento de uma ação. O que todos procuram nos gráficos é saber o que vai acontecer no futuro. Quando uma ação está em alta há algumas semanas o desafio é identificar quais seus movimentos dali em diante.

Estudando historicamente o comportamento de uma ação, percebe-se que alguns fatos se repetem ao longo do tempo e sempre que determinado fato ocorre, em seguida ocorre outro. Para exemplificar o que é um padrão, quando o clima numa grande cidade como São Paulo fica seco durante muitos dias, já se sabe que em seguida a poluição fica concentrada e a qualidade do ar piora.

O mesmo ocorre com determinadas formações nos gráficos. O desafio é entender algumas regras que ajudam a reconhecer um padrão quando ele aparece. Por isso foram criados diversos indicadores formados por cálculos estatísticos para reconhecerem esses padrões.

Para entender melhor como a análise técnica funciona, torna-se necessário conhecer seus conceitos básicos, como tendências, suportes e resistências, volume, gráficos e indicadores.

2.1.2.1 Tendências

Segundo Achelis (2001, p. 25), tendência representa uma mudança consistente nos preços, isto é, mudanças nas expectativas dos investidores. Murphy (1986, p. 53) define tendência de alta como uma reta traçada por mínimos crescentes no tempo, isto é, o seguinte superior ao anterior. Já tendência de baixa é definida por uma reta traçada por máximos ou topos decrescentes no tempo, ou o seguinte inferior ao anterior. Já tendência lateral, ou em linha, quando ocorre uma sucessão de topos e fundos lateralmente irregulares. Essas tendências podem ser visualizadas na Figura 2, com seus topos (T) e fundos (F).



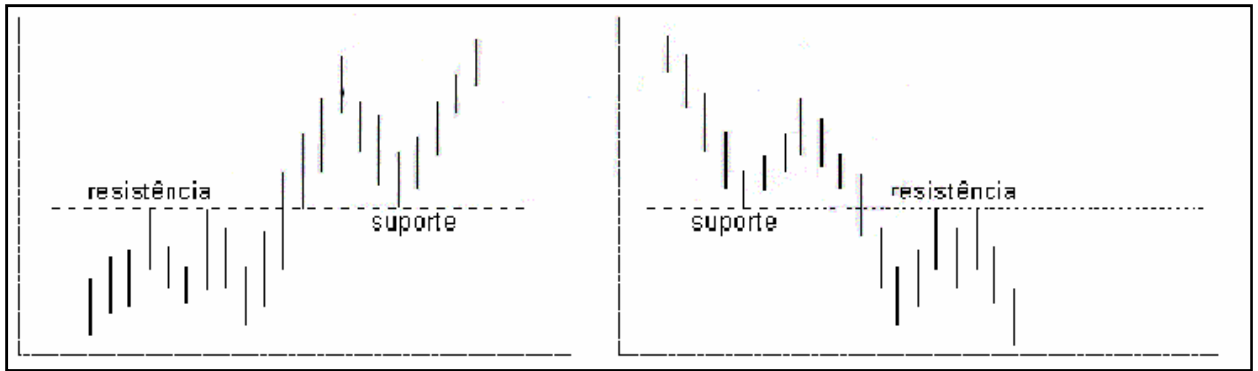
Fonte: Chaves (2004, p.43).

Figura 2 – Tendências

Um ponto é considerado máximo quando os pontos que lhe sucedem são inferiores, e considerado mínimo quando os pontos que lhe sucedem são superiores, sendo usual aguardar um intervalo de três dias para a confirmação (CHAVES, 2004, p. 43).

2.1.2.2 Suportes e resistências

São níveis de preço onde as compras e as vendas, respectivamente, são fortes o suficiente para interromper um processo de queda ou de alta durante algum tempo e possivelmente revertê-lo. Assim, topos são zonas de resistência e fundos, zonas de suporte. Em outras palavras, um suporte é o nível de pontos abaixo do qual uma cotação não cai em tendências de queda ou em deslocamentos laterais. Uma resistência é um nível de pontos que não é ultrapassado em tendências de alta ou em deslocamentos laterais conforme pode ser visualizada na Figura 3 (CHAVES, 2004, p. 43).



Fonte: Chaves (2004, p. 43).

Figura 3 – Suportes e resistências

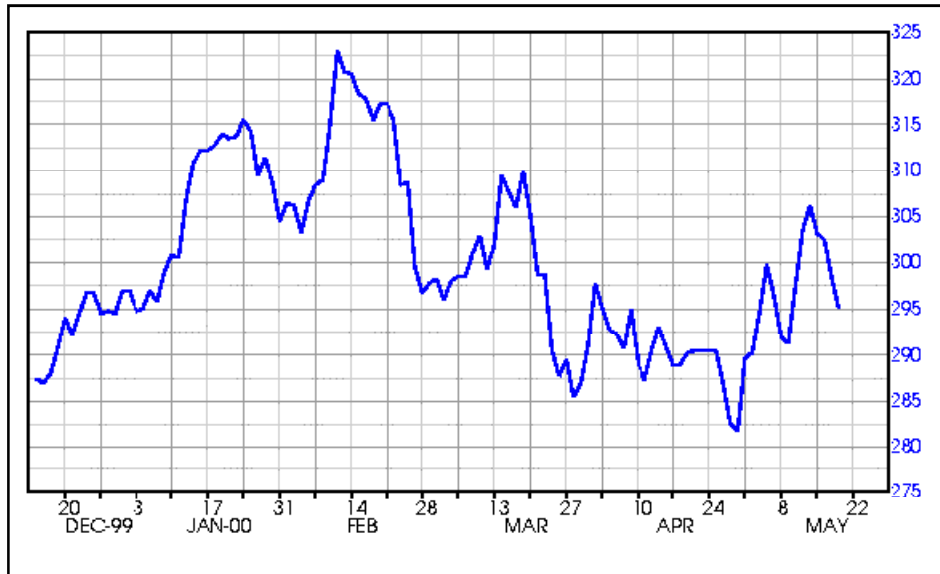
2.1.2.3 Volume

Noronha (1995, pg. 159) define a quantidade de negócios dos ativos como um fator de muita importância na negociação, um parâmetro que serve para corroborar o que ocorre no gráfico de barras. O volume deve ser crescente quando a evolução dos preços se dá na mesma direção da tendência principal, diminuindo nas correções. Durante a formação de figuras tende a diminuir, aumentando quando de seu rompimento.

2.1.2.4 Gráficos

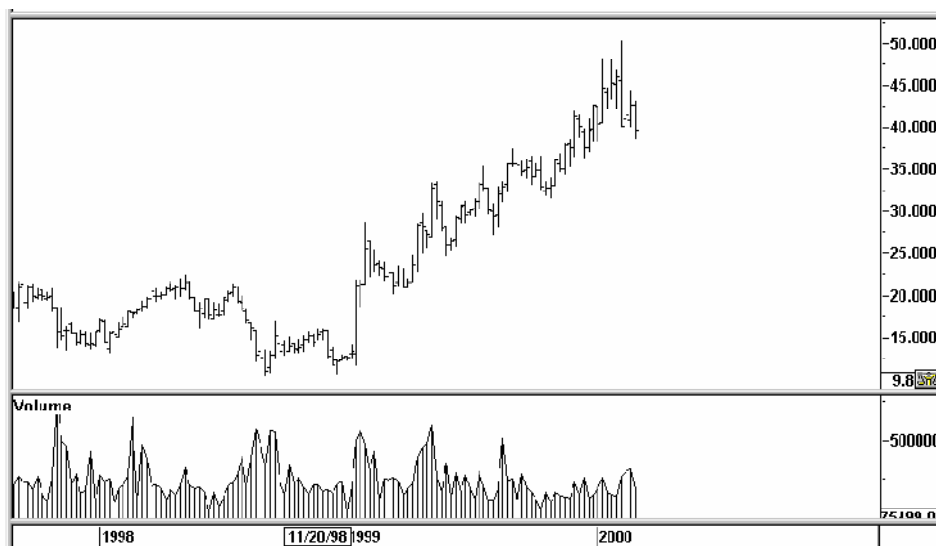
Os gráficos são a principal ferramenta da análise técnica. Eles têm sido desenvolvidos em diferentes formas e estilos para representar graficamente, além dos preços, todos os influenciadores do mercado. Os gráficos podem ser encontrados na escala mensal, diária, semanal e até mesmo em períodos de uma hora. Quanto à escala, podem ser encontrados sob a forma aritmética, logarítmica ou quadrática (CHAVES, 2004, p. 44).

Os tipos mais utilizados são os gráficos de linha (Figura 4), barras (Figura 5), ou *candlesticks* (Figura 6).



Fonte: Chaves (2004, p. 45).

Figura 4 – Gráfico de linha



Fonte: Chaves (2004, p. 46).

Figura 5 – Gráfico de barras e volume



Figura 6 – Gráfico de *candlesticks* obtido através da ferramenta Operador

2.1.2.5 Indicadores

Noronha (1995, p. 179) afirma que os indicadores contribuem na identificação de tendências e dos pontos de reversão. Além de fornecer uma visão aprofundada dentro do equilíbrio de forças entre compradores e vendedores. Alguns analistas afirmam que os indicadores são mais objetivos quando comparados aos padrões gráficos. O principal revés da utilização desses indicadores é que, muitas vezes, eles se contradizem uns aos outros.

Alguns funcionam melhor em mercados sem tendência e outros em mercados com tendência. Deste modo surge a definição de dois grupos de indicadores, os rastreadores e osciladores de tendência.

Os rastreadores funcionam melhor quando o mercado está se movendo numa direção, porém fornecem sinais ruins quando o mercado apresenta-se estável, sem tendência. Esses indicadores contribuem para a confirmação de uma tendência quando esta se encontra em seu início.

Os osciladores identificam com certa precisão os pontos de retorno em mercados sem tendência (2/3 do tempo), porém fornecem sinais prematuros e perigosos quando uma tendência se inicia no mercado.

Os analistas técnicos reúnem vários indicadores e tentam obter uma média dos seus sinais. A utilização de apenas um desses indicadores pode levar a erros que comprometerão o sucesso das operações (CHAVES, 2004, p. 56).

Segundo Noronha (1995, p. 224) o IFR é o oscilador mais utilizado pelo mercado. Sua função é medir a força de qualquer ativo monitorando as mudanças nos seus preços de fechamento variando entre 0 e 100.

Quando o IFR cai abaixo dos 30, diz-se que se inicia uma zona de compra, e o mercado está desvalorizado, sugerindo a proximidade de uma reversão da tendência de baixa para alta, e um próximo ponto de compra.

Quando o IFR sobe acima dos 70, diz que se inicia uma zona de venda, e o mercado está supervalorizado, sugerindo a proximidade de uma reversão de tendência de alta para baixa, e um próximo ponto de venda.

2.2 BANDAS DE BOLLINGER

Segundo Baião (2008), entre os indicadores utilizados na análise técnica, as BBs estão entre as mais utilizadas pelos analistas técnicos. Criado por John Bollinger em 1980, elas mantêm uma relação com a variação dos preços das ações, podendo ajudar a antecipar movimentos e identificar pontos de compra e venda.

Observando uma ação ao longo do tempo, é possível perceber períodos de alta variação de preços e outros bem tranquilos. Na análise, é verificado que a ação dificilmente foge de uma determinada região, sendo atraída para um canal. Este canal é identificado com a utilização das médias dos preços em determinado período como pode ser visto na Figura 7 (BAIÃO, 2008).



Figura 7 – Gráfico de cotações da Petrobras com bandas de Bollinger obtido através da ferramenta Operador

Segundo Matsura (2006, p. 77), as BBs, são formadas por três linhas. A linha central é uma média móvel simples; a linha ou banda superior é a linha central mais duas vezes o seu desvio padrão; a linha ou banda inferior é igual à linha central menos duas vezes o seu desvio padrão. O desvio padrão representa o nível de variação dos preços da ação. A ideia é que a variação dos preços sempre reverta para o seu ponto médio.

Uma das maneiras de se analisar as BBs é observando o seu estreitamento. Muitas vezes ocorre uma diminuição na variação de preços de uma ação em razão do equilíbrio entre

demanda e oferta. Esta diminuição tem reflexo direto nas bandas, uma vez que elas aproximam-se deixando um canal muito mais estreito. A melhor analogia para o que esta formação representa é a calmaria que antecede uma tempestade, uma vez que se trata de um considerável sinal de que um forte movimento está vindo.

Assim, à medida que o gráfico toca ou cruza a banda superior, o mercado pode ser considerado super valorizado, em sentido contrário, quando o gráfico toca ou cruza a banda inferior, o mercado pode ser considerado desvalorizado conforme visualizado na Figura 8.

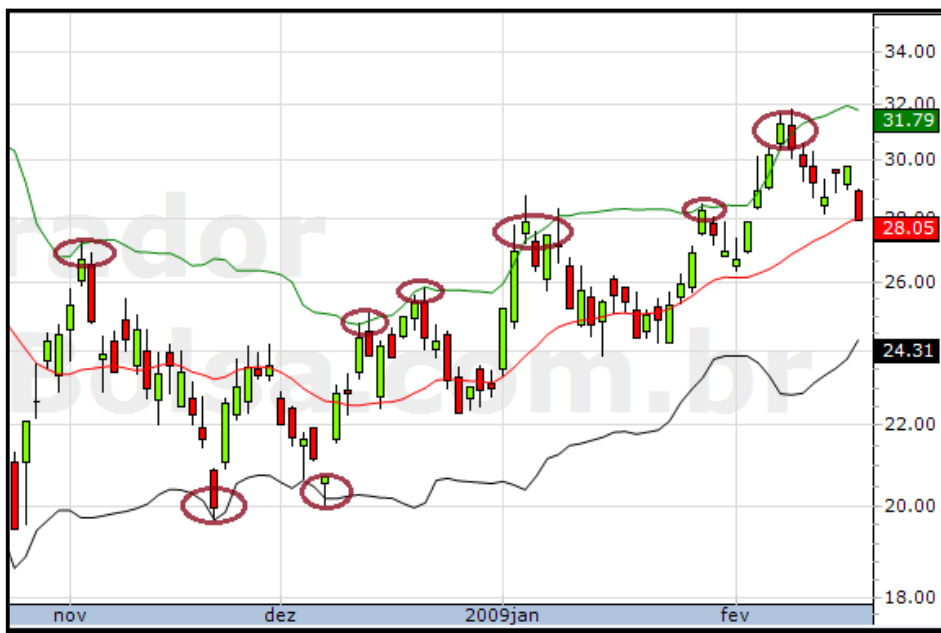


Figura 8 – Gráfico de cotações da Petrobras com bandas de Bollinger identificando pontos de super valorização e desvalorização obtido através da ferramenta Operador

Entre outros usos, ainda pode-se destacar o fato de que as bandas são bons alvos de preços. Assim, um movimento que se inicia sobre uma banda tende a percorrer todo o caminho até a outra.

Segundo Dobson (2000, p. 9), os pontos-chave para interpretar e definir as BBs são os seguintes:

- a) BBs são calculadas a partir de uma média móvel simples;
- b) para uma representação efetiva, é recomendado utilizar 20 períodos para o cálculo da média móvel;
- c) a distância das bandas em relação à média móvel varia de acordo com o número de desvios-padrão utilizados para o cálculo;
- d) dois desvios-padrão são recomendados para formar a largura das bandas em relação à média móvel;
- e) as bandas variam de acordo com a volatilidade do mercado;
- f) as BBs podem ser utilizadas em qualquer tipo de mercado;

- g) qualquer período de tempo pode ser analisado, sejam períodos de diários, semanais, mensais ou semestrais;
- h) um dos principais usos das BBs são identificar se os preços estão altos ou baixos de acordo com uma base relativa;
- i) as bandas estreitam-se conforme o mercado está calmo, e expandem-se conforme o mercado torna-se volátil;
- j) movimentos rápidos de preços tendem a ocorrer após as bandas estreitarem-se;
- k) as BBs devem ser utilizadas em conjunto com outros indicadores técnicos a ponto de detectar inversões de tendências;
- l) quando um movimento teve origem em uma banda, é provável que ele busque a banda na outra extremidade.

2.2.1 Cálculo das BBs

As BBs são calculadas a partir das médias móveis conforme a fórmula \bar{X} descrita no Quadro 1. A banda central é formada unicamente pela média móvel, a banda superior é formada pela banda central mais dois desvios-padrão, e finalmente a banda inferior é formada pela banda central menos dois desvios-padrão. A fórmula do desvio-padrão pode ser visualizada também no Quadro 1, através do símbolo σ (INVESTPEDIA, 2010).

Fórmulas Bandas de Bollinger	
$\sigma = \frac{\sqrt{\sum_{j=1}^N (X_j - \bar{X})^2}}{N}$	
$\bar{X} = \frac{\sum_{j=1}^N X_j}{N}$	Banda superior: $\bar{X} + 2\sigma$ Banda central: \bar{X} Banda inferior: $\bar{X} - 2\sigma$

Quadro 1 – Fórmulas utilizadas no cálculo das BBs

2.2.2 Análise combinada com volume

A combinação entre o volume e as BBs, também é poderosa para a análise de preços.

Quando o *candle* toca a banda superior, somado a uma tendência de alta, a confirmação de um indicador de volume é uma boa sinalização.

Ocorrências de toques na banda superior sem confirmação por parte dos indicadores são sinais de enfraquecimento da tendência de alta em curso, assim como a ocorrência de toques sem confirmação da banda inferior é um sinal de enfraquecimento da tendência de baixa em curso. A Figura 9 exemplifica este fato, mostrando o indicador acumulação/distribuição cuja função é identificar o fluxo de volume, combinado com as BBs (RIBEIRO, 2010).



Figura 9 – Gráfico com análise das BBs combinado com o fluxo de volume obtido através da ferramenta Profitchart

2.2.3 Análise combinada com IFR

Segundo Investpedia (2010), uma das combinações mais poderosas em análise técnica, e que poucas pessoas conhecem ou sabem utilizar com eficácia, é a união das BBs com o uso do IFR (Índice de Força Relativa).

Em seus estudos, John Bollinger notou que quando os preços fechavam acima da banda superior e o IFR estava abaixo de 70, então a tendência era forte e novas máximas

estariam por vir.

De maneira análoga, quando os preços fechavam abaixo da banda inferior e o IFR estava acima de 30, então a tendência era forte, só que dessa vez para baixo, e novas mínimas estariam por vir. O momento de saída da operação também era determinado através do IFR.



Fonte: Investpedia (2010).

Figura 10 – Análise das BBs combinada com o IFR

Na Figura 10 a seta clara superior indica um fechamento fora da banda superior após um período de estreitamento das bandas com IFR abaixo de 70 (seta clara inferior). Logo após essa formação os preços iniciam uma forte tendência de alta, que gera um alargamento na banda superior.

A saída da operação é dada pelo IFR, quando este rompe a linha de 70 para baixo, indicando fraqueza da tendência de alta (seta escura inferior) e os preços começam a se distanciar da banda superior (seta escura superior). Repare que nesse caso, no momento da indicação de saída, os preços estavam distantes da média móvel de 20 períodos, o que reforçava ainda mais a possível queda em direção à média (INVESTPEDIA, 2010).

2.2.4 Cálculo das cotações

As cotações das ações são importadas através de arquivo texto, com os valores de abertura, fechamento, máxima, mínima e volume. Porém é necessário calcular as BBs, além de outros indicadores que irão auxiliar as redes neurais a encontrar padrões mais facilmente.

A classe responsável por calcular estes indicadores é a `CalculoCotacao`. Após as

cotações serem importadas, o primeiro indicador a ser calculado é o IFR. Após ser definido o número de dias que o IFR deve composto ($iDiasIFR$), o cálculo é feito conforme Quadro 2.

```
...
dMediaGanhos = ((dMediaGanhos * (iDiasIFR-1)) + dGanho ) / iDiasIFR;
dMediaPerdas = ((dMediaPerdas * (iDiasIFR-1)) + dPerda ) / iDiasIFR;

if (dMediaPerdas == 0)
    dIFR = 100;
else
    dIFR = 100 - (100/(1 + (dMediaGanhos / dMediaPerdas)));

cotacaoAtual.VlIFR = dIFR;
...
```

Quadro 2 – Código resumido demonstrando o cálculo do IFR

As BBs, indicador mais relevante do projeto, é calculado juntamente com o desvio padrão, média móvel, e percentual da BB. O número de desvios, e a quantidade de dias da média móvel são definidos pelo usuário no momento da importação das cotações. Todos os cálculos podem ser vistos no código-fonte do Quadro 3.

```

...
double dSomaMM = 0;
double dSomaAoQuadrado = 0;
Cotacao cotacaoAtual = null;
foreach (KeyValuePair<DateTime, Cotacao> kvp in cotacoes)
{
    cotacaoAtual = (Cotacao)kvp.Value;
    dSomaMM += cotacaoAtual.VlFechamento;
    dSomaAoQuadrado += Math.Pow(cotacaoAtual.VlFechamento, 2);
    cotacoes[cotacaoAtual.Data].VFalha = 0;

    if ((cotacoes.IndexOfValue(cotacaoAtual)+1) >= iDiasMM)
    {
        //Media Móvel
        cotacoes[cotacaoAtual.Data].MediaMovel = dSomaMM / iDiasMM;

        //Desvio Padrão
        cotacoes[cotacaoAtual.Data].DesvioPadrao =
            Math.Sqrt((1.0 / (iDiasMM - 1)) *
                (dSomaAoQuadrado - (Math.Pow(dSomaMM, 2) / iDiasMM)));

        //Banda Superior
        cotacoes[cotacaoAtual.Data].BandaSuperior =
            cotacoes[cotacaoAtual.Data].MediaMovel + iNumDesvios *
            cotacoes[cotacaoAtual.Data].DesvioPadrao;

        //Banda Inferior
        cotacoes[cotacaoAtual.Data].BandaInferior =
            cotacoes[cotacaoAtual.Data].MediaMovel - iNumDesvios *
            cotacoes[cotacaoAtual.Data].DesvioPadrao;

        //Percentual BB
        double vCotacao = 0;
        if( cotacoes[cotacaoAtual.Data].VlFechamento >
            cotacoes[cotacaoAtual.Data].MediaMovel )
            vCotacao = cotacoes[cotacaoAtual.Data].VlMaxima;
        else
            vCotacao = cotacoes[cotacaoAtual.Data].VlMinima;

        cotacoes[cotacaoAtual.Data].PercentualBB =
            (vCotacao - cotacoes[cotacaoAtual.Data].BandaInferior) /
            (cotacoes[cotacaoAtual.Data].BandaSuperior -
            cotacoes[cotacaoAtual.Data].BandaInferior);

        KeyValuePair<DateTime, Cotacao> primeiraCotacao =
            cotacoes.ElementAt((cotacoes.IndexOfValue(cotacaoAtual)+1)-iDiasMM);
        dSomaMM -= primeiraCotacao.Value.VlFechamento;
        dSomaAoQuadrado -= Math.Pow(primeiraCotacao.Value.VlFechamento, 2);
    }
}
...

```

Quadro 3 – Código demonstrando os cálculos para definir as BBs

Para calcular a média móvel, a variável `iDiasMM` é definida pelo usuário do protótipo, assim como a variável `iNumDesvios`, que é utilizada para calcular as BBs.

Como qualquer indicador, as BBs podem indicar falhas nos gráficos, ou seja, as bandas podem indicar um movimento até determinado ponto, mas a cotação pode não seguir esta indicação. Desta maneira, foi desenvolvido um algoritmo que identifica estas falhas, e armazena a intensidade da falha em um novo atributo. A Figura 11 demonstra alguns exemplos de falhas.

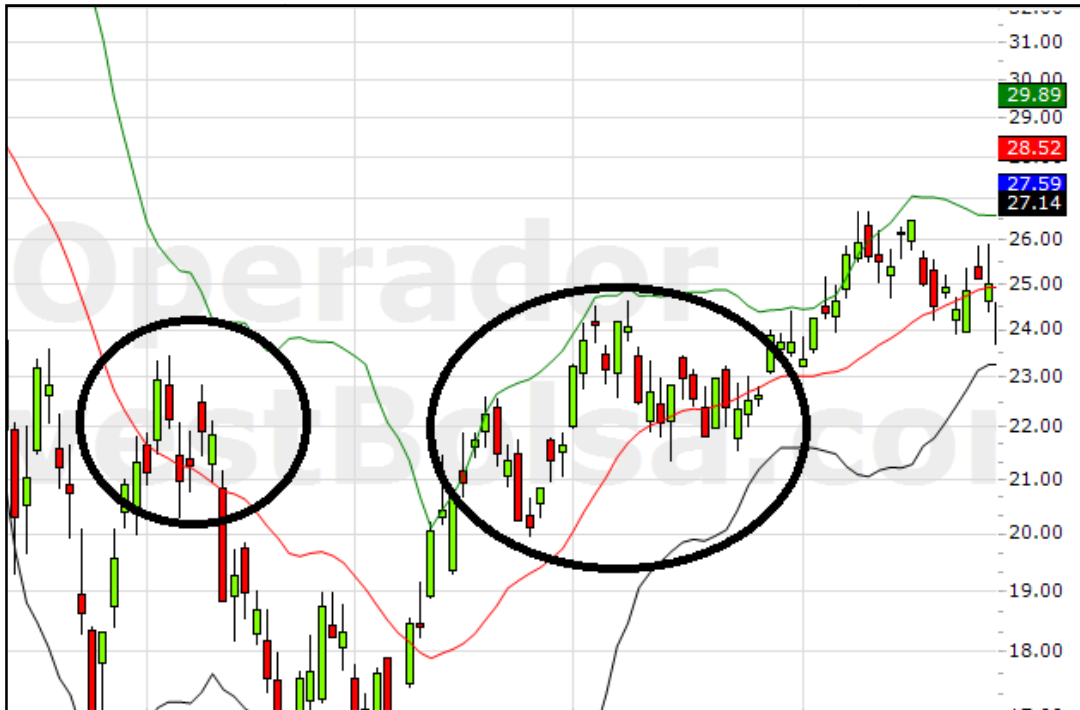


Figura 11 – Exemplos de falhas nas BBs obtidas através da ferramenta Operador

Conforme as regras das BBs (Seção 2.2), é considerada falha quando a ação inicia um movimento de alta, ultrapassa a banda central, porém não consegue obter força suficiente para tocar na banda superior e volta a se movimentar abaixo da banda central. Da mesma maneira, a falha ocorre quando a ação inicia um movimento de baixa, ultrapassa a banda central, porém não consegue obter força suficiente para tocar na banda inferior e volta a se movimentar acima da banda central.

Para medir a intensidade desta falha, foi calculado o percentual da BB. Este número indica em qual percentual a ação encontra-se, em relação à distância das bandas superior e inferior. Desta maneira, quando um movimento é falho, identifica-se o ponto da falha e atribui-se a distância da ação até a banda que deveria ter sido atingida.

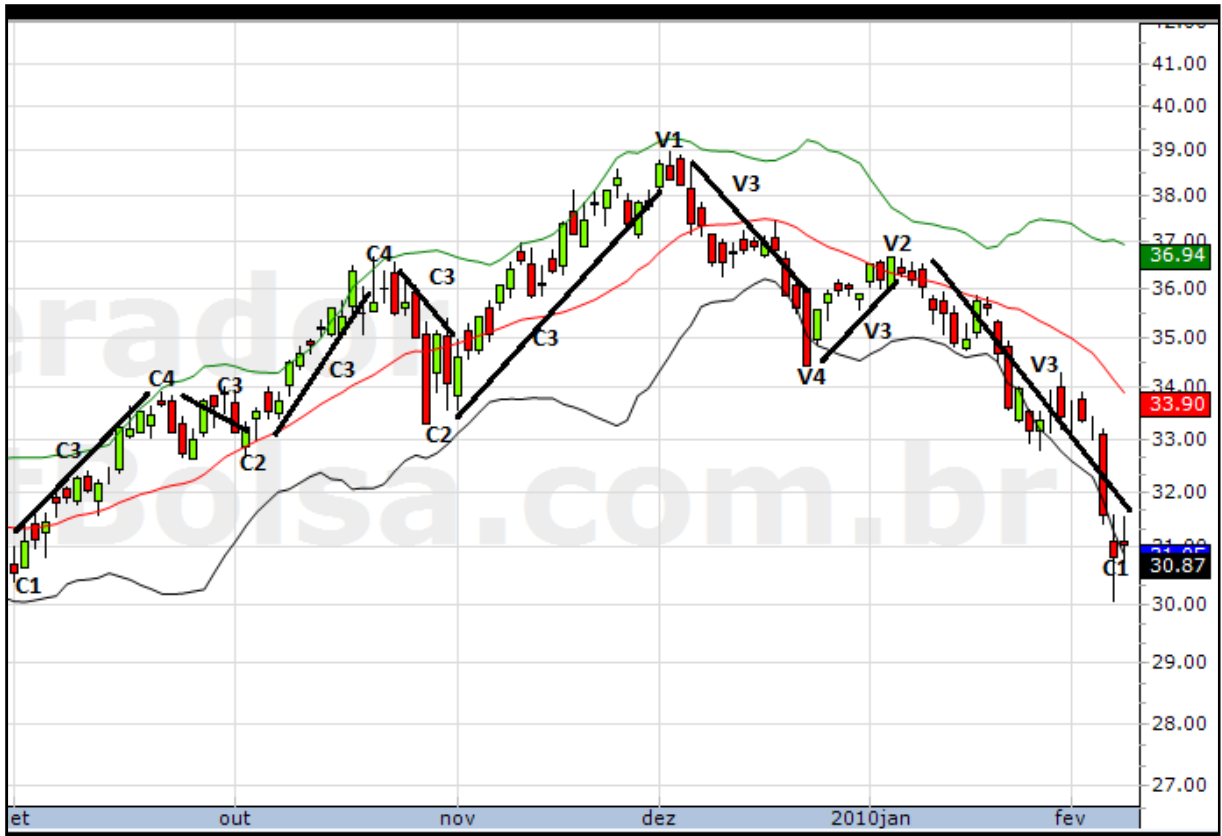


Figura 12 – Tendências de uma ação em relação as BBs obtidas através da ferramenta Operador

Na figura 12, encontra-se exemplificado como identificar as tendências. Os pontos identificados pela letra c indicam um movimento de compra, enquanto os pontos indicados pela letra v indicam um movimento de venda.

Os pesos destas letras, são explicados no Quadro 4. Estes pesos são armazenados em um atributo denominado tendência, e será utilizado como principal parâmetro para o treinamento das redes. Após as redes treinadas, elas poderão identificar em qual tendência a ação encontra-se a fim de indicar uma compra ou venda, objetivo este do protótipo proposto.

Tendências de uma ação em relação as BBs	
Valor	Significado
V1 -> 0,00	Corresponde a venda imediata
V2 -> 0,25	Ponto de venda dentro de um movimento de baixa
V3 -> 0,50	Manter a posição de venda
V4 -> 0,75	Ponto de compra dentro de um movimento de baixa
C1 -> 1,00	Corresponde a compra imediata
C2 -> 1,25	Ponto de compra dentro de um movimento de alta
C3 -> 1,50	Manter a posição de compra
C4 -> 1,75	Ponto de venda dentro de um movimento de alta

Quadro 4 – Tendências de uma ação em relação as BBs

Ao final da execução dos cálculos da classe *CalculoCotacao*, a classe *Cotacao* está completa com os seguintes atributos populadas: *BandaInferior*, *BandaSuperior*, *Data*, *DesvioPadrao*, *MediaMovel*, *PercentualBB*, *Tendencia*, *VFalha*, *VlAbertura*, *VlFechamento*, *VlIFR*, *VlMaxima*, *VlMinima* e *Volume*.

Após isso, as redes serão treinadas com estes atributos e posteriormente executadas a fim de prever em qual tendência encontra-se a ação, podendo indicar uma possível compra ou venda.

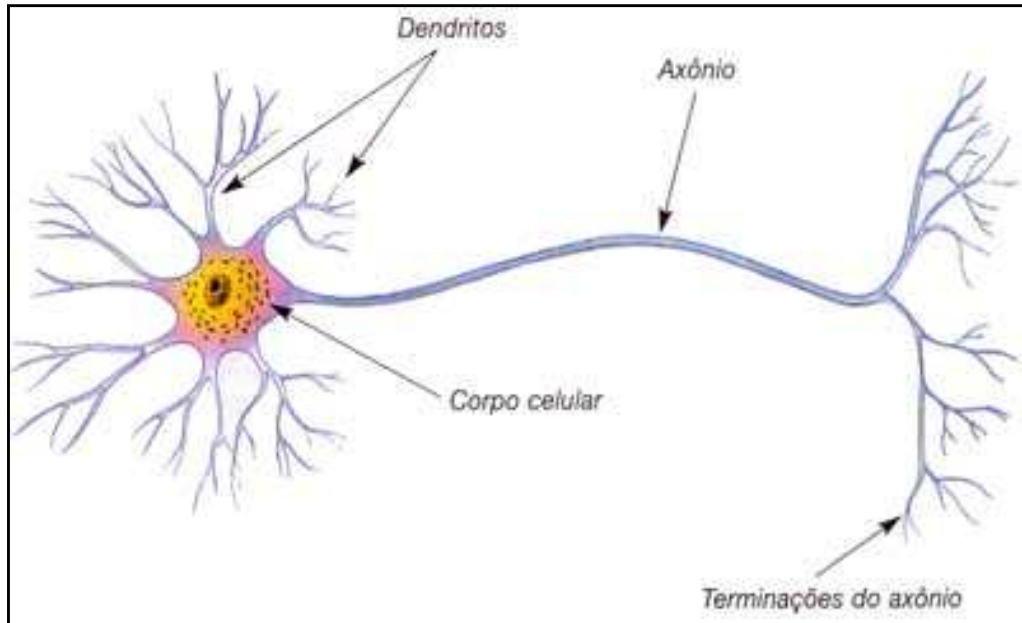
2.3 REDES NEURAIS ARTIFICIAIS

Para entender o funcionamento de uma RNA é necessário conhecer basicamente como é o funcionamento interno do cérebro. O cérebro humano é composto por aproximadamente 10 bilhões de neurônios e é responsável por grande quantidade de controle sobre as funções do corpo. Os neurônios estão conectados uns aos outros através de sinapses, e juntos formam uma grande rede. As sinapses transmitem estímulos através de diferentes concentrações de Na^+ (Sódio) e K^+ (Potássio), e o resultado disto pode ser estendido por todo o corpo humano. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informações (CARNEIRO, 2009).

2.3.1 Neurônio biológico

Segundo Kovács (1996, p. 13), o neurônio é delimitado por uma fina membrana celular que além da sua função biológica normal, possui determinadas propriedades que são essenciais para o funcionamento elétrico da célula nervosa. A partir do corpo celular, ou soma, projetam-se extensões filamentosas, os dendritos, e o axônio, como pode ser visualizado na Figura 13.

Um neurônio faz conexões com 10 a 100.000 outros neurônios, em junções chamadas sinapses. Os sinais se propagam de um neurônio para outro por meio de uma complicada reação eletroquímica. Os sinais controlam a atividade cerebral em curto prazo, e também permitem mudanças a longo prazo na posição e na conectividade dos neurônios.



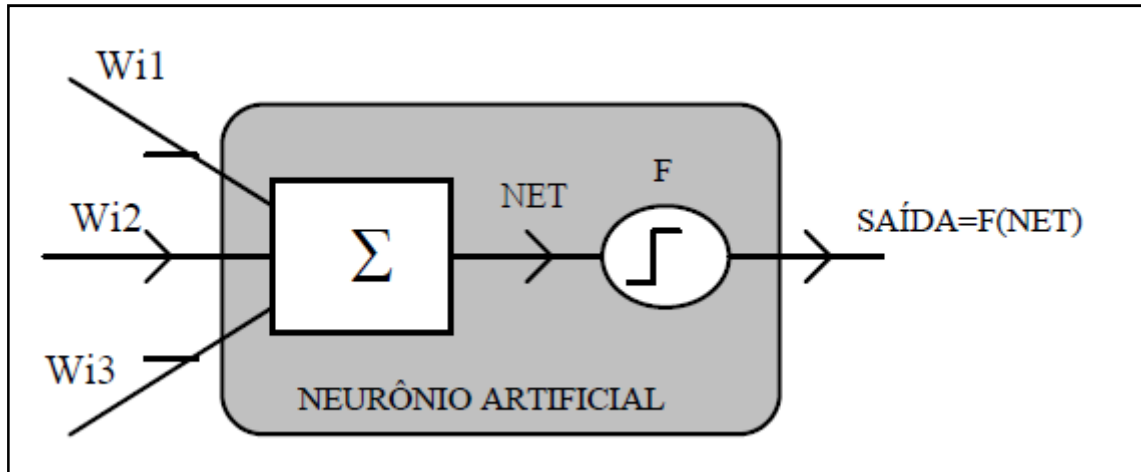
Fonte: Sogab (2010).

Figura 13 – Partes do neurônio

As manifestações elétricas de neurônios biológicos foram observadas pela primeira vez no século 19. Apenas na década de 60, como resultado do trabalho de vários pesquisadores, passou-se a entender o neurônio biológico como sendo basicamente o dispositivo computacional elementar do sistema nervoso, que possuía diversas entradas e uma única saída. As entradas ocorrem através das conexões sinápticas, que conectam a árvore dendrital aos axônios de outras células nervosas. Os sinais que chegam por estes axônios são pulsos elétricos conhecidos como impulsos nervosos ou potenciais de ação, e constituem a informação que o neurônio processará, de alguma forma, para produzir como saída um impulso nervoso no seu axônio (KOVÁCS, 1996, p. 15).

2.3.2 Neurônio artificial

Segundo Abelem (2004, p. 18), o neurônio artificial foi projetado para imitar as características de primeira ordem de um neurônio biológico. Essencialmente, um conjunto de entradas são aplicadas ao neurônio artificial, cada uma representando a saída de outros neurônios. Cada entrada é multiplicada por um peso correspondente (W_{ij}), gerando entradas ponderadas, de forma análoga à força das sinapses. Em seguida todas estas entradas ponderadas são somadas, obtendo-se um valor NET que será utilizado na função de ativação do neurônio (F). Caso este valor alcance o valor limite de ativação do neurônio, ele se ativará, caso contrário ele ficará inativo. A Figura 14 mostra o modelo que implementa esta ideia.



Fonte: Abelem (2004, p.19).

Figura 14 – Diagrama esquemático de um neurônio artificial

Dentre as principais funções de ativação podemos destacar as funções: linear, degrau e tangente hiperbólica. Outros tipos de funções podem ser usados dependendo do problema específico (FARIA, 2008, p. 6).

2.3.3 Arquiteturas de rede

As camadas são os elementos básicos que formam a RNA. Uma camada é composta de neurônios, sendo que todos eles possuem as mesmas características. Ele transforma a sua entrada em uma saída executando uma função de ativação. Cada neurônio processa a sua entrada aplicando uma função e enviando o resultado para a sinapse que a conecta à próxima camada. As sinapses, por sua vez, são as conexões entre duas camadas, permitindo que os valores sejam transmitidos de uma camada para outra. A sinapse é a memória da rede neural. Durante o treinamento o peso de cada conexão é alterado de acordo com o algoritmo de aprendizagem utilizado. Uma sinapse é tanto a sinapse de saída de uma camada quanto a sinapse de entrada da próxima camada (SESTREM, 2005, p. 22).

2.3.3.1 Redes alimentadas adiante com camada única

Em uma rede neural em camadas, os neurônios estão organizados na forma de camadas. Na forma mais simples de uma rede em camadas, temos uma camada de entrada que

se projeta sobre uma camada de saída de neurônios, mas não vice-versa. Em outras palavras, esta rede é estritamente do tipo alimentada adiante ou acíclica. Ela é ilustrada na Figura 15 para o caso de três nós tanto na camada de entrada como na de saída. Esta rede é chamada de rede de camada única, sendo que a designação “camada única” se refere à camada de saída de nós computacionais (neurônios). A camada de entrada não é contada, porque lá não é realizada qualquer computação (HAYKIN, 1999, p. 46).

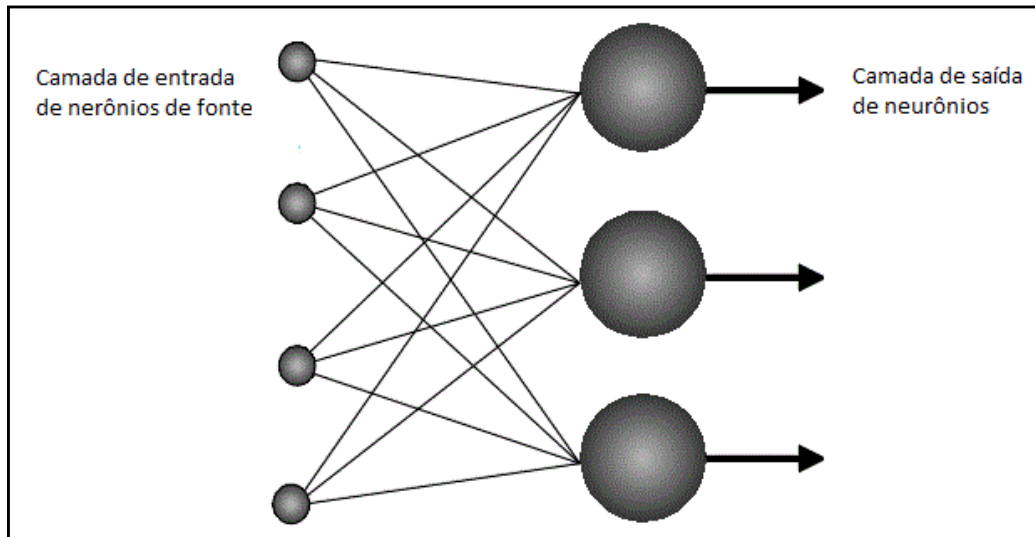


Figura 15 – Rede alimentada adiante ou acíclica com uma única camada de neurônios

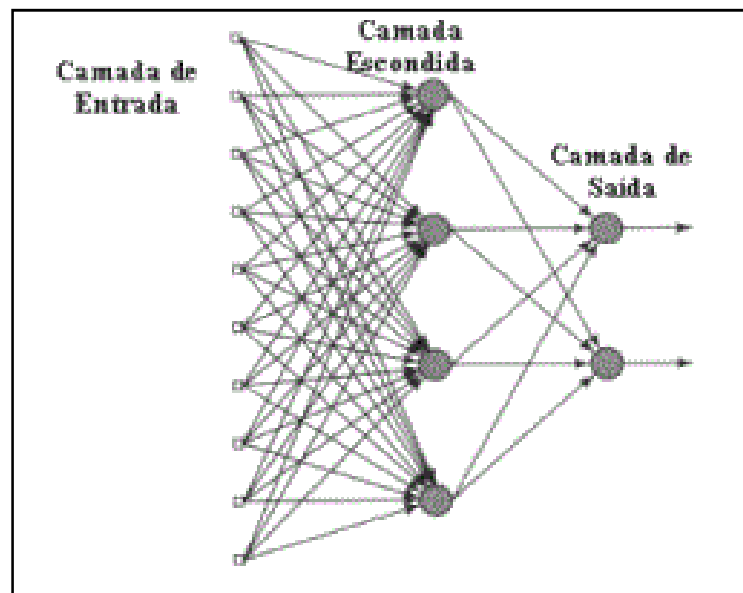
2.3.3.2 Redes alimentadas diretamente com múltiplas camadas

A segunda classe de uma rede neural alimentada adiante se distingue pela presença de uma ou mais camadas ocultas, cujos nós computacionais são chamados correspondentemente de neurônios ocultos ou unidades ocultas. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de uma maneira útil. Adicionando-se uma ou mais camadas ocultas, tornamos a rede capaz de extrair estatísticas de ordem elevada. Em um sentido bastante livre, a rede adquire uma perspectiva global apesar de sua conectividade local, devido ao conjunto extra de conexões sinápticas e da dimensão extra de interações neurais. A habilidade de os neurônios ocultos extraírem estatísticas de ordem elevada é particularmente valiosa quando o tamanho da camada de entrada é grande (HAYKIN, 1999, p.46).

Os nós da camada de entrada da rede fornecem os respectivos elementos do padrão de ativação (vetor de entrada), que constituem os sinais de entrada aplicados aos neurônios (nós computacionais) na segunda camada. Os sinais de saída da segunda camada são utilizados

como entradas para a terceira camada, e assim por diante para o resto da rede. Tipicamente, os neurônios em cada camada da rede têm como suas entradas apenas os sinais de saída da camada precedente. O conjunto de sinais de saída dos neurônios da camada de saída (final) da rede constitui a resposta global da rede para o padrão de ativação fornecido pelos nós de fonte da camada de entrada (primeira). O grafo arquitetural na Figura 14 ilustra a planta de uma rede neural de múltiplas camadas alimentada adiante para o caso de uma única camada oculta (HAYKIN, 1999, p.46).

A rede neural da Figura 16 é dita totalmente conectada, no sentido de que cada um dos nós de uma camada da rede está conectado a todos os nós da camada adjacente seguinte. Entretanto, se alguns dos elos de comunicação (conexões sinápticas) estiverem faltando na rede, dizemos que a rede é parcialmente conectada.



Fonte: Scielo (2005).

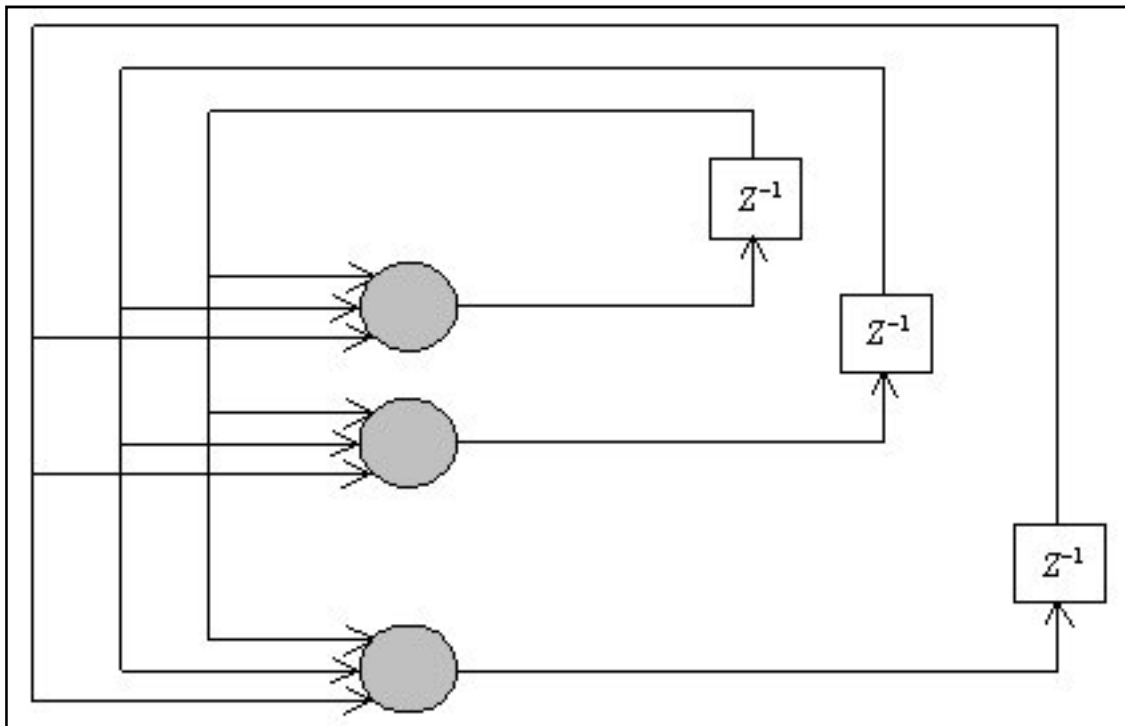
Figura 16 – Rede alimentada adiante ou acíclica totalmente conectada com uma camada oculta e uma camada de saída

2.3.3.3 Redes recorrentes

Uma rede neural recorrente distingue-se de uma rede neural alimentada diretamente pelo fato de possuir ao menos um laço de realimentação. Uma rede recorrente pode se constituir, por exemplo, de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para as entradas de todos os outros neurônios, como ilustrado no grafo arquitetural da Figura 17. Na estrutura apresentada nesta figura não existem neurônios

ocultos.

A presença de laços de realimentação tem um impacto profundo na capacidade de aprendizagem da rede e no seu desempenho. Além disso, os laços de realimentação envolvem o uso de ramos particulares compostos de elementos de atraso unitário (representados por z^{-1}), o que resulta em um comportamento dinâmico não-linear, admitindo-se que a rede neural contenha unidades não-lineares (HAYKIN, 1999, p. 48).



Fonte: Natcomp (2010).

Figura 17 – Rede recorrente sem laços de auto-realimentação e sem neurônios ocultos

2.3.4 Treinamento

Segundo Bittencourt (2006, p. 315), treinar uma rede neural significa ajustar sua matriz de pesos de forma que o vetor de saída coincida com certo valor desejado para cada vetor de entrada. O processo de treinamento pode ser visto como um problema de otimização, quando deseja-se minimizar a taxa de erro ao treinar a RNA. Pode-se resolver este problema de diversas formas, desde otimizações heurísticas como *simulated annealing*, algoritmos genéticos, até *backpropagation*.

Segundo Witten e Frank (2000, p. 265), o algoritmo *backpropagation*, além de outros algoritmos utilizados no treinamento de RNAs pode ser implementado utilizando a *Waikato Environment for Knowledge Analysis* (WEKA), uma biblioteca de uso livre desenvolvida pela

universidade de Waikato na Nova Zelândia.

2.3.5 Técnicas utilizando mineração de dados e algoritmos de redes neurais

A mineração de dados é o assunto da indústria de tecnologia, na medida em que as empresas estão gerando milhões de pontos de dados sobre seus usuários e buscando um modo de transformar estas informações em mais receita. A mineração de dados é um termo coletivo para dúzias de técnicas que retiram informações dos dados e as transformam em algo significativo (ABERNETHY, 2010).

Uma das técnicas de mineração de dados bastante utilizada é a regressão linear. Esta técnica possui uma ou mais entradas, uma única saída, e é usada para prever o resultado de uma variável dependente desconhecida, dados os valores das variáveis independentes (ABERNETHY, 2010).

Além da regressão linear como técnica para predição de valores, o *backpropagation* perceptron e o *learning vector quantisation* são exemplos de redes neurais para esta finalidade.

O algoritmo mais conhecido para treinamento de redes neurais é o *backpropagation*. Ele funciona apresentando-se um exemplo à rede e obtendo-se a saída correspondente. Em seguida, é calculado o vetor de erro que consiste na diferença entre a saída obtida e a esperada, e calculado também o gradiente do vetor de erro, atualizando os pesos da camada de saída. Para finalizar, propaga-se para trás os valores desejados de modo a atualizar os pesos para as demais camadas (BITTENCOURT, 2006, p. 317).

Segundo Loesch e Sari (1996, p.49), o *learning vector quantisation* é uma rede que possui como aplicações básicas a revocação autoassociativa (complementação do padrão a um padrão parcial apresentado), compressão de dados e classificação de padrões. Como vantagens, é capaz de auto-organizar representações vetoriais de distribuições aleatórias em dados apresentados, além de rápida execução após treinamento completado. Como desvantagens, possui características não resolvidas na seleção do número de vetores usados e tempo de treinamento apropriado, além do tempo de treinamento ser lento.

2.4 TRABALHOS CORRELATOS

Atualmente existe pouco conteúdo sobre a previsão de ações utilizando inteligência artificial. Dentre os trabalhos encontrados neste contexto, pode-se citar: o desenvolvido por Sestrem (2005) e o descrito pelo Centro Brasileiro de Pesquisas Físicas (2008).

2.4.1 Desenvolvimento de um protótipo para previsão de índice de ações

Em Sestrem (2005) é avaliado o uso de RNAs para prever o valor da cotação da ação da Petrobras. Foram coletados neste trabalho, o histórico das ações da Petrobras em um determinado período e feito a previsão baseado neste histórico. Os melhores resultados foram alcançados utilizando como parâmetros o valor de fechamento e o volume negociado da ação. Os resultados encontrados neste trabalho não permitem a utilização das RNAs como única ferramenta na análise de investimentos, pois uma taxa de acerto de 31% não garante sucesso na maior parte dos casos, mas apresentaram potencial para serem utilizadas como uma ferramenta auxiliar.

2.4.2 Previsão do mercado de ações brasileiro utilizando redes neurais artificiais

Em Faria et al. (2008), é apresentado o estudo das RNAs na previsão da série temporal do indicador mais importante do mercado de ações brasileiro, mais precisamente a série Ibovespa, que é o índice da bolsa de valores do estado de São Paulo. Para este estudo foi implementada uma rede neural artificial *multilayer perceptron* e treinada com o algoritmo *backpropagation* para fazer previsões do índice Ibovespa. Por fim, é destacado que estudos envolvendo redes neurais artificiais envolvem muito tempo na busca de parâmetros para a rede que melhor representa os resultados almejados.

3 DESENVOLVIMENTO DO PROTÓTIPO

Este capítulo detalha as etapas do desenvolvimento do protótipo para previsão do mercado de ações utilizando BBs. São apresentados os requisitos, a especificação e a implementação do mesmo. Por fim, são indicados os resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos apresentados encontram-se classificados em Requisito Funcional (RF) e Requisito Não-Funcional (RNF), os quais são:

- a) importar o arquivo TXT com o histórico de cotações nele definido (RF);
- b) efetuar os cálculos dos indicadores (BBs e IFR) com as cotações definidas no arquivo (RF);
- c) gerar a previsão das cotações baseadas em um período definido pelo usuário (RF);
- d) disponibilizar um relatório de acompanhamento das previsões, indicando os pontos de acertos, pontos de erros e seus percentuais (RF);
- e) ser implementado em C#, utilizando o ambiente de desenvolvimento Microsoft Visual Studio (RNF);
- f) utilizar a biblioteca WEKA na implementação das RNAs (RNF);
- g) utilizar banco de dados MySQL para armazenar os dados referentes as cotações (RNF);
- h) funcionar no sistema operacional Windows XP ou Superior (RNF).

3.2 ESPECIFICAÇÃO

A especificação do protótipo é apresentada seguindo a análise orientada a objetos, e utilizando a notação *Unified Modeling Language* (UML) para a criação dos diagramas de casos de uso, classe e de sequência. A modelagem dos diagramas foi desenvolvida utilizando a ferramenta *Enterprise Architect*.

3.2.1 Diagrama de casos de uso

A Figura 18 apresenta o diagrama de casos de uso com as principais interações do usuário com o protótipo.

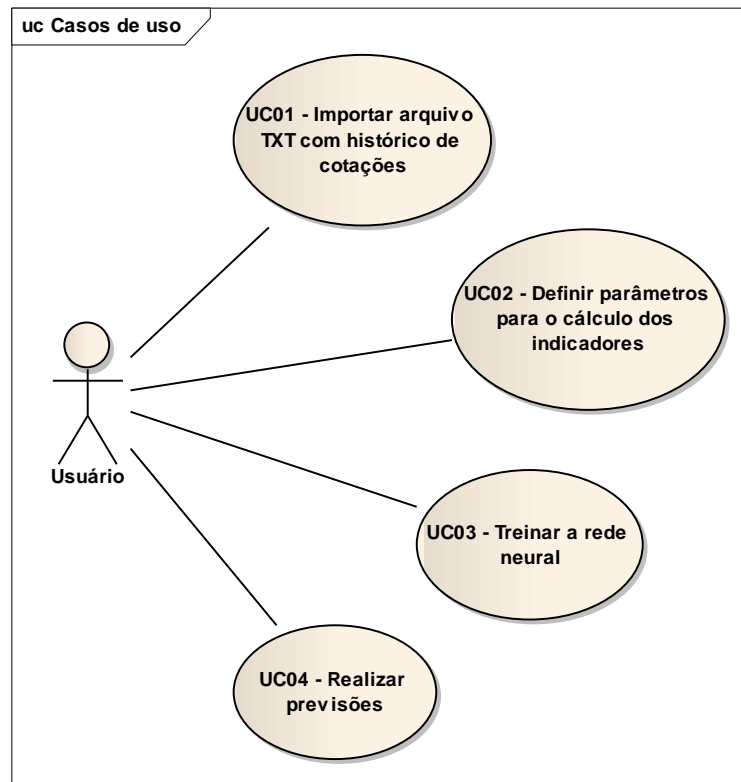


Figura 18 – Diagrama de casos de uso

O caso de uso UC01 - Importar arquivo texto com histórico de cotações (Quadro 5), permite que o usuário importe as cotações de determinado papel para formar sua base de dados.

UC01 – Importar arquivo texto com histórico de cotações	
Pré-condição	O arquivo texto deve estar com as cotações conforme leiaute exigido pelo sistema.
Cenário principal	<ol style="list-style-type: none"> 1) o usuário seleciona a aba inserir papéis; 2) o usuário informa o código e nome do papel, além de selecionar o arquivo; 3) o usuário clica no botão atualizar; 4) o protótipo interpreta o arquivo, calcula os indicadores, e salva o papel com suas cotações no banco de dados; 5) o protótipo informa ao usuário a quantidade de cotações adicionadas.
Exceção	No passo 3, caso o código, nome do papel, ou arquivo não estejam informados, o protótipo consiste com uma mensagem de erro e aborta o processo.
Pós-condição	O protótipo importou as cotações com sucesso.

Quadro 5 – Detalhamento do caso de uso UC01 - Importar arquivo texto com histórico de cotações

O caso de uso UC02 - Definir parâmetros para o cálculo dos indicadores (Quadro 6), permite que o usuário defina a forma como os indicadores serão calculados.

UC02 – Definir parâmetros para o cálculo dos indicadores	
Pré-condição	A base de dados deve possuir ao menos um papel cadastrado com suas cotações.
Cenário principal	<ol style="list-style-type: none"> 1) o usuário seleciona a aba inserir papéis; 2) o usuário informa o código do papel; 3) o usuário define o número de dias da média móvel, do IFR, e a quantidade de desvios-padrão; 4) o usuário clica no botão atualizar; 5) o protótipo faz o cálculo dos indicadores com todas as cotações do papel, e salva no banco de dados.
Exceção	No passo 4, caso o código do papel não seja informado, o protótipo consiste com uma mensagem de erro e aborta o processo.
Pós-condição	O protótipo calculou os indicadores com sucesso.

Quadro 6 – Detalhamento do caso de uso UC02 - Definir parâmetros para o cálculo dos indicadores

O caso de uso UC03 - Treinar a rede neural (Quadro 7), permite que o usuário treine a rede utilizando os papéis cadastrados, para posteriormente realizar as previsões.

UC03 – Treinar a rede neural	
Pré-condição	A base de dados deve possuir ao menos um papel cadastrado com suas cotações.
Cenário principal	<ol style="list-style-type: none"> 1) o usuário seleciona o papel; 2) o usuário informa o período de cotações que será utilizado para treinar a rede neural; 3) o usuário define o atributo de saída que se espera da rede neural; 4) o usuário clica no botão treinar; 5) o protótipo treina as redes e exibe o resultado graficamente e textualmente.
Exceção	No passo 2, caso o período esteja inválido, o protótipo consiste com uma mensagem de erro e aborta o processo.
Pós-condição	As redes neurais estão treinadas para realizarem as previsões.

Quadro 7 – Detalhamento do caso de uso UC03 - Treinar a rede neural

O caso de uso UC04 - Realizar previsões (Quadro 8), permite que o usuário realize previsões a partir de uma rede neural treinada.

UC04 – Realizar previsões	
Pré-condição	O papel deve possuir uma rede neural treinada.
Cenário principal	<ol style="list-style-type: none"> 1) O usuário seleciona o período; 2) O usuário clica no botão executar; 3) O protótipo faz a previsão, e informa os resultados graficamente e textualmente.
Exceção	No passo 2, caso o período esteja inválido ou a rede neural não esteja treinada, o protótipo consiste com uma mensagem de erro e aborta o processo.
Pós-condição	O protótipo exibe os resultados das previsões.

Quadro 8 – Detalhamento do caso de uso UC04 - Realizar a rede neural

3.2.2 Diagrama de classes

O diagrama de classes apresentado na Figura 19 exibe as principais classes utilizadas no protótipo.

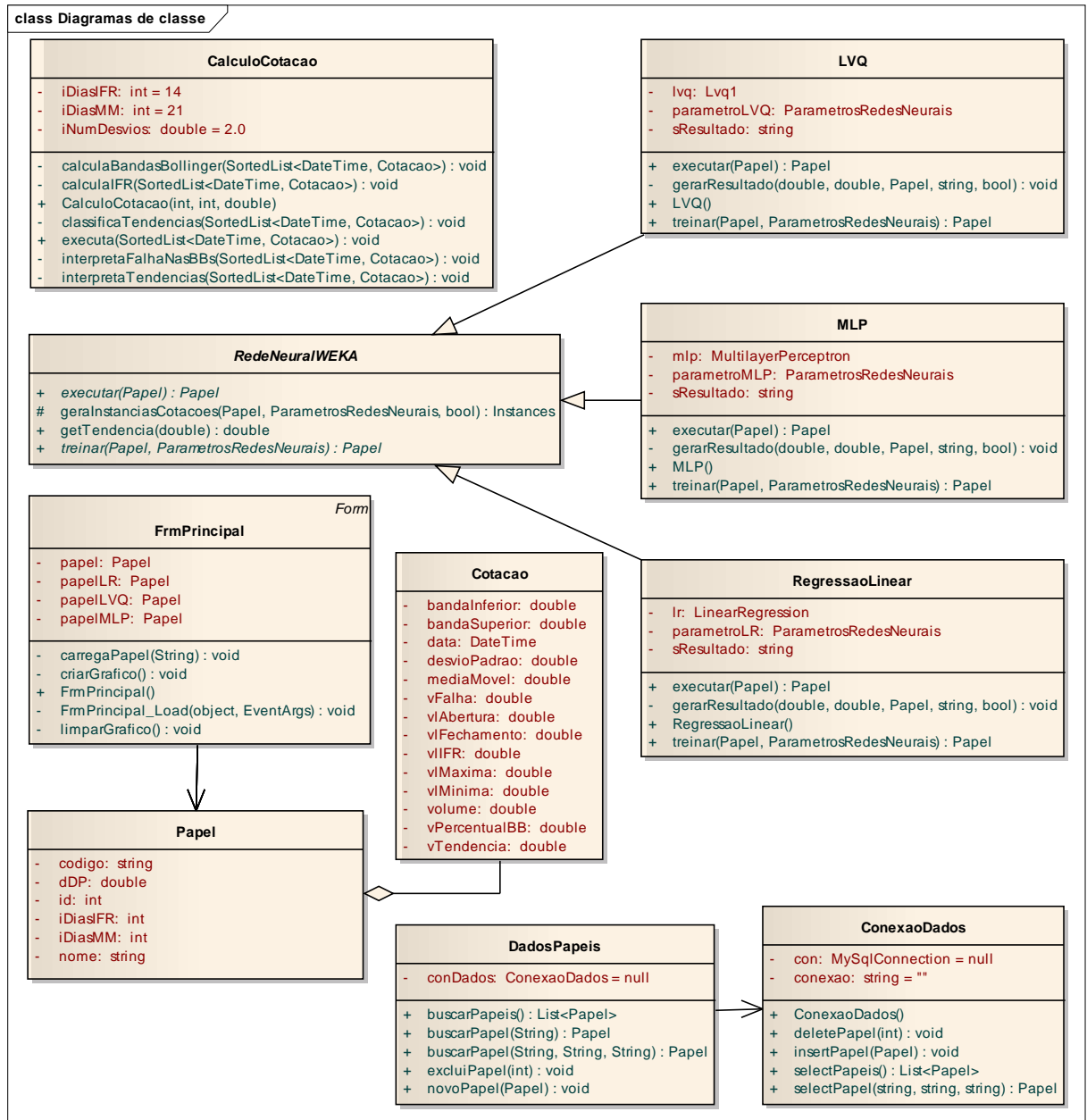


Figura 19 – Diagrama de classes

Segue o detalhamento das classes contidas no protótipo:

- ConexaoDados: classe que implementa a conexão e acesso as informações com o banco de dados;
- DadosPapeis: classe intermediária responsável pela ligação entre a interface do protótipo e a classe ConexaoDados;

- c) `Papel`: esta classe é responsável por armazenar as informações e parâmetros referentes à ação que será analisada;
- d) `Cotacao`: esta classe é responsável por armazenar as informações referentes a cotação da ação, bem como todos os atributos derivados de uma cotação, como as BBs, o IFR, a média móvel, o desvio padrão, o volume, entre outros;
- e) `FrmPrincipal`: esta classe é responsável pela interação do usuário com o protótipo;
- f) `CalculoCotacao`: classe que implementa o cálculo de todos os atributos derivados da cotação para serem utilizados como entrada nas redes neurais;
- g) `RedeNeuralWEKA`: classe abstrata que provê métodos que serão utilizados e implementados pelas redes neurais do protótipo;
- h) `LVQ`: classe que implementa a rede neural *Learning Vector Quantisation* (LVQ);
- i) `MPL`: classe que implementa a rede neural *Multi Layer Perceptron* (MLP) utilizando o algoritmo *Backpropagation*;
- j) `RegressaoLinear`: classe que implementa a rede neural *Linear Regression* (LR).

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas informações sobre as técnicas e ferramentas utilizadas para a implementação da aplicação, bem como os processos de implementação, detalhando com trechos do código fonte quando necessário.

3.3.1 Técnicas e ferramentas utilizadas

O protótipo foi desenvolvido utilizando a linguagem de programação C# através do ambiente Microsoft Visual Studio 2008. Para realizar a conexão com o banco de dados foi utilizada a biblioteca *MySQL Connector .NET*. A predição do mercado acionário foi realizada através de redes neurais, utilizando algoritmos da biblioteca WEKA, e o resultado da execução das redes é visualizado através da criação de gráficos, utilizando a biblioteca ZedGraph.

A biblioteca WEKA foi desenvolvida em linguagem Java pela Universidade de Wakato na Nova Zelândia, que tem em sua composição inúmeros algoritmos provenientes da área de inteligência artificial.

3.3.1.1 IKVM.NET

O IKVM.NET é uma aplicação que permite utilizar bibliotecas Java em aplicações .NET. Como a biblioteca WEKA foi desenvolvida em Java, foi necessário utilizar o IKVM para que os algoritmos da WEKA possam ser utilizados no protótipo.

Antes de utilizar a biblioteca WEKA no projeto, foi necessário converter o arquivo weka.jar para weka.dll. Para isso, o IKVM possui um utilitário chamado ikvmc. Este utilitário é executado através da seguinte linha de comando: `ikvmc -target:library weka.jar`.

A biblioteca convertida é adicionada junto com as dll's `IKVM.Runtime.dll`, `IKVM.OpenJDK.SwingAWT.dll`, `IKVM.OpenJDK.Text.dll` e `IKVM.OpenJDK.Core.dll` conforme Figura 20.

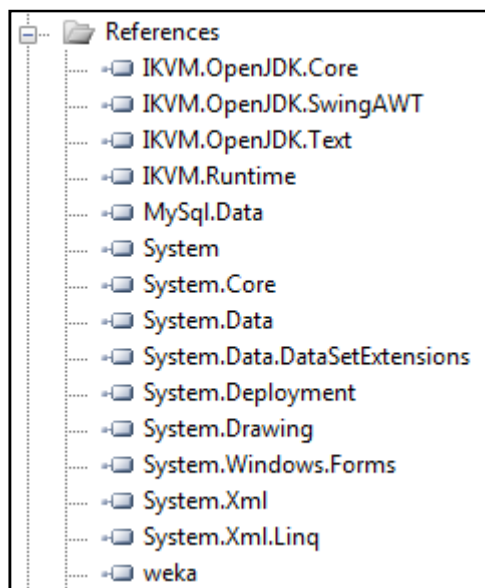


Figura 20 – Bibliotecas adicionadas no projeto

Após a WEKA estar incluída no projeto, o IKVM executa uma máquina virtual Java na aplicação e a biblioteca é utilizada em linguagem C#. A Figura 21 demonstra o algoritmo LVQ em modo de depuração, onde é possível visualizar a derivação do Java no interior de uma instância em linguagem C#.

```

Instances cotacoesTreinamento = geraInstanciasCotacoes(papelTreinamento, parametro, true);

// Define os parametros para classifica o
lvq = new Lvq1();
lvq.setInitialisationMode(new SelectedTag(3, lvq.getInitialisationMode().getTags()));
lvq.setSelectedTag(1, lvq.getLearningFunction().getTags());
lvq.setAlgorithmAncestor(new LvqAlgorithmAncestor());
lvq.setClassifier(new Classifier());
lvq.setUserDefinedOptions(new HashMap());
lvq.setUseDebug(false);

// Treina a rede
lvq.buildClassifier();

// Serializa a rede treinada
BinaryFormatter sFormatter = new BinaryFormatter();

```

Figura 21 – Depura o do algoritmo LVQ

3.3.1.2 Linear Regression

O algoritmo LR implementado por `weka.classifiers.functions.LinearRegression`,   utilizado para predi o de valores. Utiliza o crit rio *Akaike* para modelos de sele o e   capaz de lidar com inst ncias ponderadas. Seus atributos s o:

- `attributeSelectionMethod`: os m todos dispon veis s o nenhuma sele o, sele o de atributos usando o m todo M5 (percorre os atributos e remove aquele com o menor coeficiente padronizado at  que nenhuma melhoria   observada na estimativa do erro dada pelo crit rio de informa o *Akaike*), e sele o Greedy que usa as informa oes m tricas de *Akaike*;
- `eliminateColinearAttributes`: eliminar atributos colineares;
- `ridge`: valor de alcance (Padr o: 1.0e-8).

Para definir quais valores utilizar nos par metros, foram realizados diversos testes com as a oes da Petrobras e Vale do Rio Doce, no per odo de janeiro de 1998 at  setembro de 2010, totalizando 3120 cota oes para cada a o. Os Quadros 9 e 10 demonstram quais valores se adequaram melhor para os par metros `attributeSelectionMethod` e `eliminateColinearAttributes`. O par metro `ridge` n o demonstrou nenhuma relev ncia nos testes.

Testes realizados com o parâmetro <code>attributeSelectionMethod</code>		
Ação	Valor	Cotações corretas
Petrobras	No attribute Selection	317 (10,16%)
Petrobras	M5 Method	316 (10,13%)
Petrobras	Greedy Method	319 (10,22%)
Vale do Rio Doce	No attribute Selection	383 (12,28%)
Vale do Rio Doce	M5 Method	387 (12,40%)
Vale do Rio Doce	Greedy Method	389 (12,47%)

Quadro 9 – LR: testes realizados com o parâmetro `attributeSelectionMethod`

Testes realizados com o parâmetro <code>eliminateColinearAttributes</code>		
Ação	Valor	Cotações corretas
Petrobras	True	216 (6,92%)
Petrobras	False	319 (10,22%)
Vale do Rio Doce	True	342 (10,96%)
Vale do Rio Doce	False	389 (12,47%)

Quadro 10 – LR: testes realizados com o parâmetro `eliminateColinearAttributes`

Após realizados os testes, ficou comprovado que a melhor configuração do algoritmo LR para predição da tendência de determinada ação, é utilizar o método *Greedy*, não permitir a eliminação de atributos colineares e utilizar o valor default para o *ridge*. O Quadro 11 demonstra o código-fonte da parametrização, treinamento e classificação da rede.

```

...
Instances cotacoesTreinamento =
    geraInstanciasCotacoes(papelTreinamento, parametro, false);

// Define os parametros para classificação
lr = new LinearRegression();
lr.setAttributeSelectionMethod(new SelectedTag(2,
lr.getAttributeSelectionMethod().getTags()));
lr.setEliminateColinearAttributes(false);

// Treina a rede
lr.buildClassifier(cotacoesTreinamento);

//Classifica a rede com os dados de treinamento
Papel papelLR = papelTreinamento;
int corretas = 0;
for (int i = 0; i < cotacoesTreinamento.numInstances(); i++)
{
    Instance instancia = cotacoesTreinamento.instance(i);
    double valor = (double)(lr.classifyInstance(instancia));
    valor = getTendencia(valor);

    if (valor == instancia.classValue())
        corretas++;
}
...

```

Quadro 11 – Código demonstrando a parametrização, treinamento e classificação utilizando LR

3.3.1.3 Learning Vector Quantisation

O algoritmo LVQ implementado por `weka.classifiers.neural.lvq.Lvq1`, é também utilizado para predição de valores e possui os seguintes atributos em sua essência:

- a) `initialisationMode`: os modos de inicialização são *Random Training Data Proportional*, *Random Training Data Even*, *Random Values In Range*, *Simple KMeans*, *Farthest First*, *K-Nearest Neighbour Even*;
- b) `learningFunction`: possui as funções *Linear Decay*, *Inverse*, *Static*;
- c) `learningRate`: valor da taxa inicial de aprendizagem;
- d) `totalCodebookVectors`: número total de vetores *Codebook* utilizados no treinamento;
- e) `totalTrainingIterations`: número total de iterações no treinamento;
- f) `useVoting`: utilizado dinamicamente para selecionar a classe atribuída de cada vetor *Codebook*.

Para definir quais valores utilizar nos parâmetros, foram realizados diversos testes com as ações da Petrobras e Vale do Rio Doce, no período de janeiro de 1998 até setembro de 2010. Os Quadros 12, 13, 14 e 15 demonstram quais valores se adequaram melhor para os parâmetros `initialisationMode`, `learningFunction`, `learningRate` e `useVoting`. Os parâmetros `totalCodebookVectors` e `totalTrainingIterations` não demonstraram nenhuma relevância nos testes.

Testes realizados com o parâmetro <code>initialisationMode</code>		
Ação	Valor	Cotações corretas
Petrobras	Random Training Data Proportional	1570 (50,32%)
Petrobras	Random Training Data Even	1387 (44,46%)
Petrobras	Random Values In Range	1646 (52,76%)
Petrobras	K-Nearest Neighbour Eve	1647 (52,79%)
Vale do Rio Doce	Random Training Data Proportional	1431 (45,87%)
Vale do Rio Doce	Random Training Data Even	1551 (49,71%)
Vale do Rio Doce	Random Values In Range	1645 (52,72%)
Vale do Rio Doce	K-Nearest Neighbour Eve	1475 (47,28%)

Quadro 12 – LVQ: testes realizados com o parâmetro `initialisationMode`

Testes realizados com o parâmetro <code>learningFunction</code>		
Ação	Valor	Cotações corretas
Petrobras	Linear Decay	1646 (52,76%)
Petrobras	Inverse	1626 (52,12%)
Petrobras	Static	1648 (52,82%)
Vale do Rio Doce	Linear Decay	1645 (52,72%)
Vale do Rio Doce	Inverse	1606 (51,47%)
Vale do Rio Doce	Static	1623 (52,02%)

Quadro 13 – LVQ: testes realizados com o parâmetro `learningFunction`

Testes realizados com o parâmetro <code>learningRate</code>		
Ação	Valor	Cotações corretas
Petrobras	0,1	1645 (52,72%)
Petrobras	0,5	1646 (52,76%)
Petrobras	1,0	1645 (52,72%)
Vale do Rio Doce	0,1	1644 (52,69%)
Vale do Rio Doce	0,5	1660 (53,21%)
Vale do Rio Doce	1,0	1653 (52,98%)

Quadro 14 – LVQ: testes realizados com o parâmetro `learningRate`

Testes realizados com o parâmetro <code>useVoting</code>		
Ação	Valor	Cotações corretas
Petrobras	True	1670 (53,53%)
Petrobras	False	1646 (52,76%)
Vale do Rio Doce	True	1672 (53,59%)
Vale do Rio Doce	False	1660 (53,21%)

Quadro 15 – LR: testes realizados com o parâmetro `useVoting`

Ao final dos testes, a configuração para o algoritmo LVQ ficou definida da seguinte forma: *Random Values In Range* para o parâmetro `initialisationMode`, *Linear Decay* para o parâmetro `learningFunction`, 0,5 para o parâmetro `learningRate` e *True* para o parâmetro `useVoting`. O Quadro 16 demonstra o código-fonte da parametrização, treinamento e classificação da rede.

```

...
Instances cotacoesTreinamento =
    geraInstanciasCotacoes(papelTreinamento, parametro, true);

// Define os parametros para classificação
lvq = new Lvql();
lvq.setInitialisationMode(new SelectedTag(3,
    lvq.getInitialisationMode().getTags()));
lvq.setLearningFunction(new SelectedTag(1,
    lvq.getLearningFunction().getTags()));
lvq.setLearningRate(0.5);
lvq.setTotalCodebookVectors(20);
lvq.setTotalTrainingIterations(1000);
lvq.setUseVoting(true);

// Treina a rede
lvq.buildClassifier(cotacoesTreinamento);

//Classifica a rede com os dados de treinamento
Papel papelLVQ = papelTreinamento;
int corretas = 0;
for (int i = 0; i < cotacoesTreinamento.numInstances(); i++)
{
    Instance instancia = cotacoesTreinamento.instance(i);
    int valor = (int)(lvq.classifyInstance(instancia));

    if (valor == instancia.classValue())
        corretas++;
}
...

```

Quadro 16 – Código demonstrando a parametrização, treinamento e classificação utilizando LVQ

3.3.1.4 Multi Layer Perceptron

O MLP é um classificador que utiliza o algoritmo *backpropagation*, implementado por `weka.classifiers.functions.MultilayerPerceptron`, possui diversos parâmetros de configuração sendo os mais relevantes os seguintes:

- a) `autoBuild`: adiciona e conecta-se as camadas ocultas da rede;
- b) `decay`: decrementa a taxa de aprendizagem;
- c) `hiddenLayers`: número de camadas ocultas da rede;
- d) `learningRate`: taxa de aprendizado;
- e) `momentum`: dinâmica aplicada aos pesos durante a atualização;
- f) `normalizeAttributes`: normaliza os atributos para melhorar o desempenho da rede;
- g) `trainingTime`: número de épocas para o treinamento;

h) `validationThreshold`: utilizado para finalizar o teste de validação.

Para definir quais valores utilizar nos parâmetros, foram realizados diversos testes com as mesmas ações e períodos demonstrados nos algoritmos LR e LVQ. Os Quadros 17, 18, 19 e 20 demonstram quais valores se adequaram melhor para os parâmetros `hiddenLayers`, `learningRate`, `momentum` e `trainingTime`.

Testes realizados com o parâmetro <code>hiddenLayers</code>		
Ação	Valor	Cotações corretas
Petrobras	"a" = (atributos+classes)/2	565 (18,11%)
Petrobras	"i" = atributos	586 (18,78%)
Petrobras	"t" = atributos + classes	637 (20,42%)
Vale do Rio Doce	"a" = (atributos+classes)/2	1157 (37,08%)
Vale do Rio Doce	"i" = atributos	1180 (37,82%)
Vale do Rio Doce	"t" = atributos + classes	1131 (36,25%)

Quadro 17 – MLP: testes realizados com o parâmetro `hiddenLayers`

Testes realizados com o parâmetro <code>learningRate</code>		
Ação	Valor	Cotações corretas
Petrobras	0,3	586 (18,78%)
Petrobras	0,6	632 (20,26%)
Petrobras	0,9	153 (4,90%)
Vale do Rio Doce	0,3	1180 (37,82%)
Vale do Rio Doce	0,6	1580 (50,64%)
Vale do Rio Doce	0,9	1523 (48,81%)

Quadro 18 – MLP: testes realizados com o parâmetro `learningRate`

Testes realizados com o parâmetro <code>momentum</code>		
Ação	Valor	Cotações corretas
Petrobras	0,20	632 (20,26%)
Petrobras	0,23	723 (23,17%)
Petrobras	0,27	593 (19,01%)
Vale do Rio Doce	0,20	1580 (50,64%)
Vale do Rio Doce	0,23	1571 (50,35%)
Vale do Rio Doce	0,27	1547 (49,58%)

Quadro 19 – MLP: testes realizados com o parâmetro `momentum`

Testes realizados com o parâmetro <code>trainingTime</code>		
Ação	Valor	Cotações corretas
Petrobras	500	723 (23,17%)
Petrobras	1000	786 (25,19%)
Petrobras	2000	856 (27,44%)
Vale do Rio Doce	500	1580 (50,64%)
Vale do Rio Doce	1000	1561 (50,03%)
Vale do Rio Doce	2000	1553 (49,78%)

Quadro 20 – MLP: testes realizados com o parâmetro `trainingTime`

Ao final dos testes, a configuração para o algoritmo MLP ficou definida da seguinte

forma: “i” para o parâmetro `hiddenLayers`, 0,6 para o parâmetro `learningRate`, 0,23 para o parâmetro `momentum` e 2000 para o parâmetro `trainingTime`. Os parâmetros `autoBuild`, `normalizeAttributes`, `decay` e `validationThreshold` foram definidos com o valor padrão, pois a modificação dos mesmos não desempenharam uma mudança positiva no treinamento. O Quadro 21 demonstra o código-fonte da parametrização, treinamento e classificação da rede.

```

...
Instances cotacoesTreinamento =
    geraInstanciasCotacoes(papelTreinamento, parametro, false);

// Define os parametros para classificação
mlp = new MultilayerPerceptron();
mlp.setAutoBuild(true);
mlp.setDecay(false);
mlp.setNormalizeAttributes(true);
mlp.setLearningRate(0.6);
mlp.setMomentum(0.23);
mlp.setTrainingTime(2000);
mlp.setHiddenLayers("i");

// Treina a rede
mlp.buildClassifier(cotacoesTreinamento);

//Classifica a rede com os dados de treinamento
string sClasse = "";
Papel papelMLP = papelTreinamento;
int corretas = 0;
for (int i = 0; i < cotacoesTreinamento.numInstances(); i++)
{
    Instance instancia = cotacoesTreinamento.instance(i);
    double valor = (double)(mlp.classifyInstance(instancia));
    valor = getTendencia(valor);

    if (valor == instancia.classValue())
        corretas++;
}
...

```

Quadro 21 – Código demonstrando a parametrização, treinamento e classificação utilizando MLP

3.3.2 Operacionalidade da implementação

Esta seção tem por finalidade mostrar a operacionalidade da implementação em nível de usuário. Nas próximas seções serão abordadas as principais funcionalidades do protótipo.

3.3.2.1 Preparação dos dados

Antes de iniciar qualquer análise, é necessário primeiramente adicionar o histórico de cotações no banco de dados, além de calcular as BBs, IFR, falhas e tendências. Para isto, deve-se estar de posse do arquivo texto contendo o histórico de cotações conforme mostra a Figura 22. Os valores para formar o arquivo podem ser extraídos através da página da Bovespa.

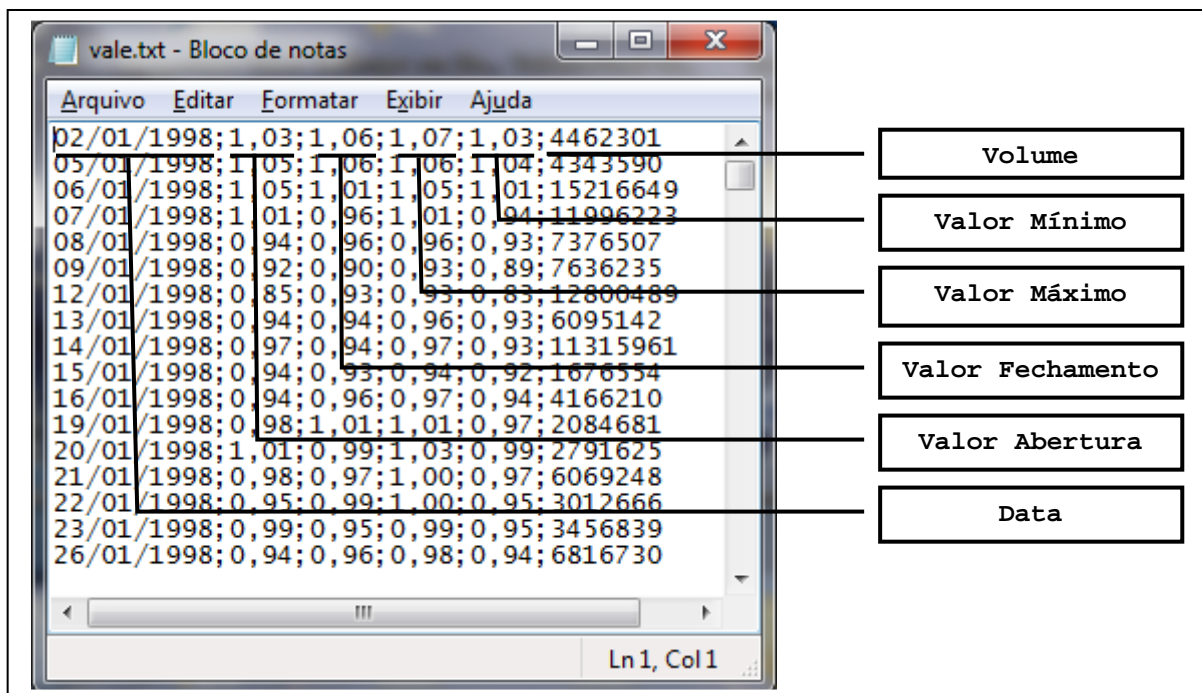


Figura 22 – Histórico de cotações em arquivo texto

Ao iniciar a aplicação, deve-se selecionar a aba Inserir papéis, preencher as informações de acordo com a Figura 23 e clicar no botão Atualizar. O protótipo interpretará o arquivo texto, calculará os indicadores, falhas e tendências, e apresentará uma mensagem informando a quantidade de cotações que foram adicionadas/calculadas.

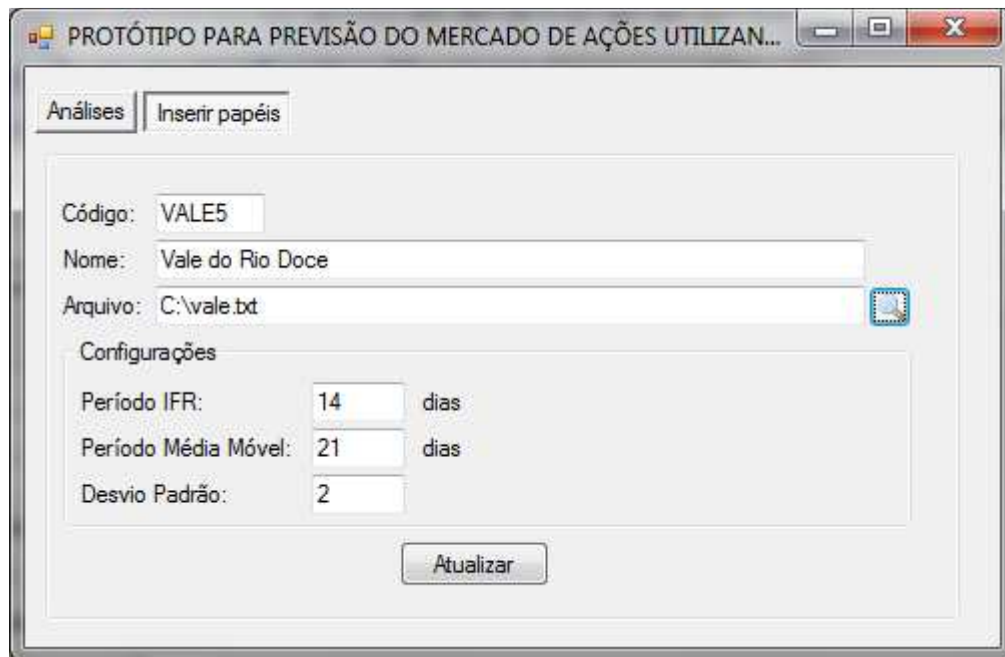


Figura 23 – Tela do protótipo para inserir os papéis

3.3.2.2 Análise das ações

Selecionando a aba *Análises*, é apresentando ao usuário a possibilidade de escolher uma ação/papel, selecionar um período e um atributo para previsão. Após o preenchimento destes campos, é possível observar o gráfico do papel baseado no atributo de previsão selecionado. A Figura 24 demonstra o gráfico do valor de fechamento da Petrobras com as BB's e a média móvel.

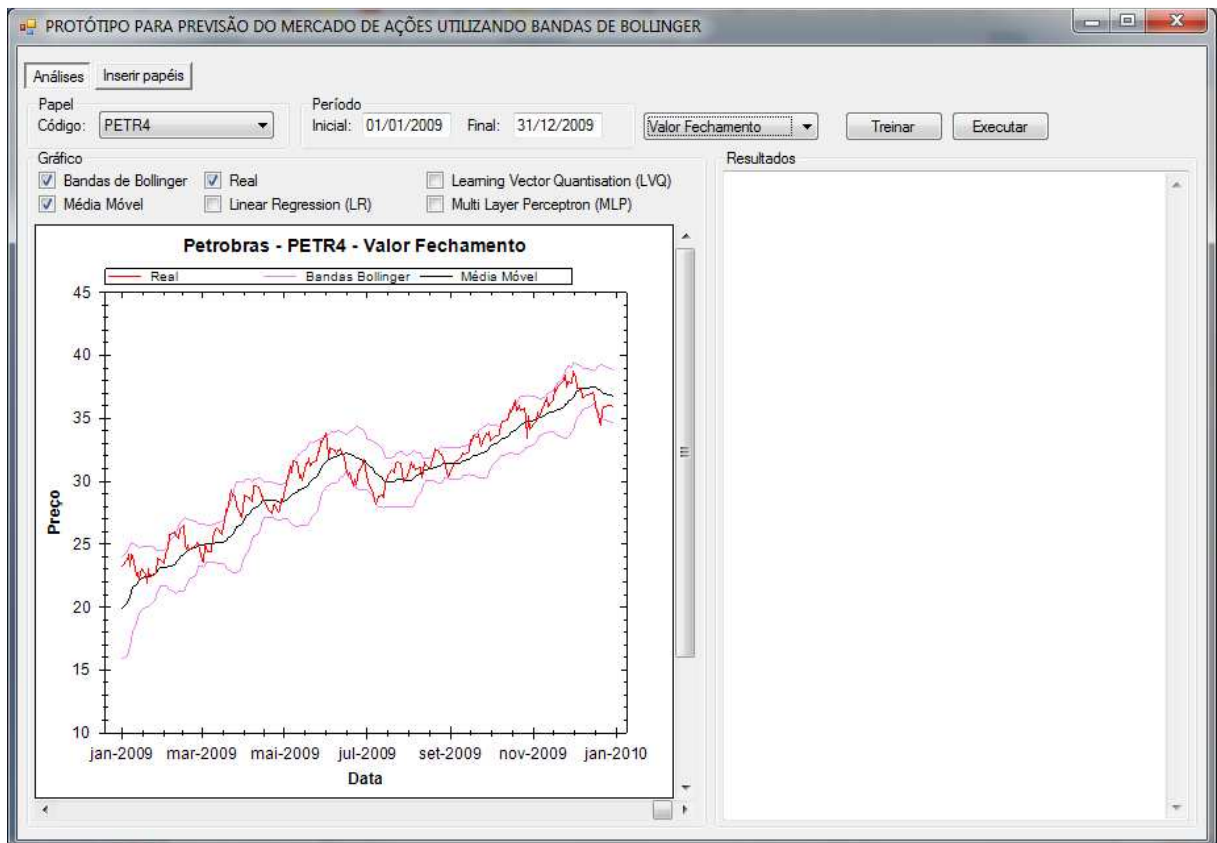


Figura 24 – Tela do protótipo com o gráfico da Petrobras, BB's e média móvel

Para realizar o treinamento da rede, basta clicar no botão *Treinar*. O protótipo realiza o treinamento de todas as redes suportadas pelo mesmo e apresenta um gráfico referente às informações reais *versus* informações previstas, de acordo com o algoritmo de treinamento. A Figura 25 demonstra o treinamento da ação Vale do Rio Doce para o período de 01/01/2009 até 31/12/2009, utilizando o valor de fechamento da ação para classificação (atributo de saída das redes).

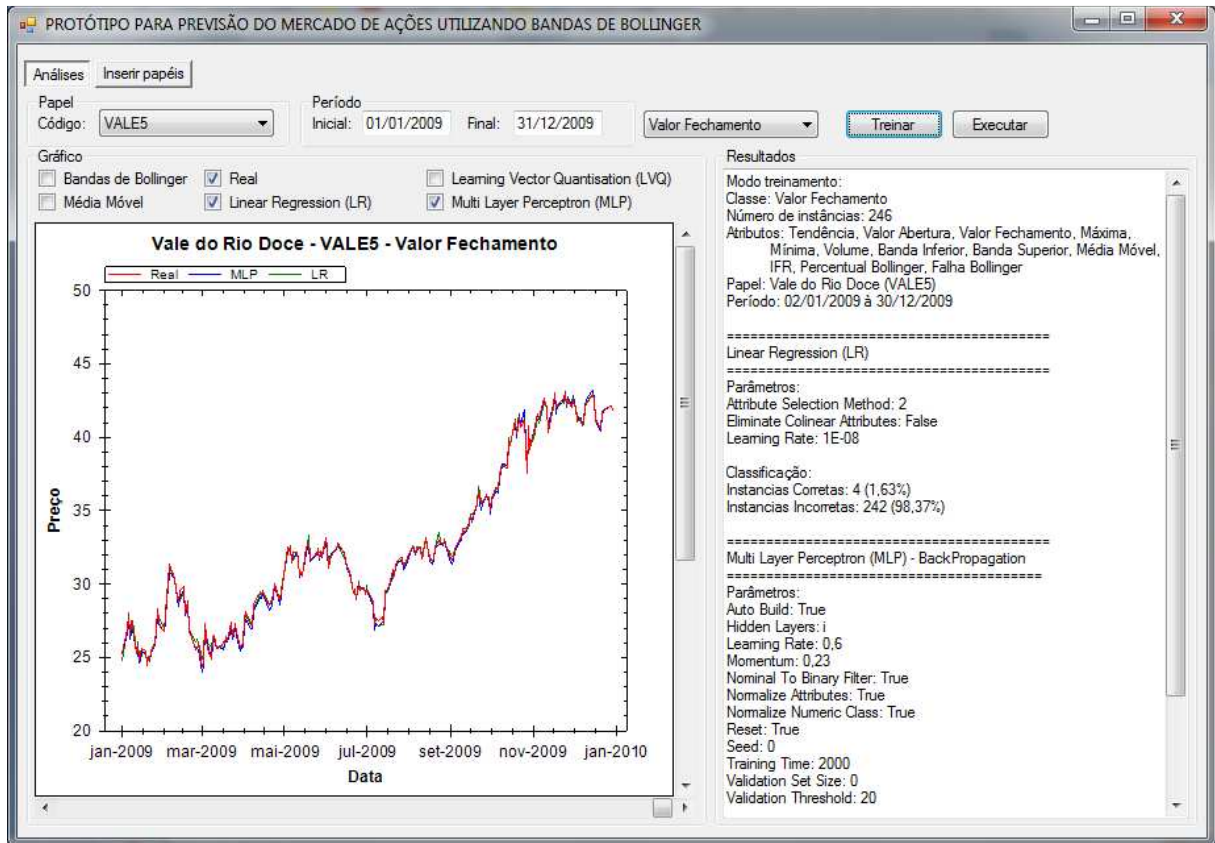


Figura 25 – Tela do protótipo com análise do valor de fechamento da Vale do Rio Doce

Visualizando a Figura 25, é possível notar que o percentual de classificações corretas pela LR foi muito baixo (1,63%), porém o gráfico da LR ficou idêntico ao real. Apesar desta disparidade, pode-se considerar que as redes estão bem treinadas, pois seus gráficos estão muito próximos da cotação real, o que indica que este percentual de erro gira em torno de poucos centavos.

No entanto, o valor de fechamento, assim como o valor de abertura, máximo e mínimo, tem utilidade somente para auxiliar o treinamento das redes MLP e LR. O objetivo final do protótipo é indicar ao usuário uma atitude a ser tomada, seja de compra, venda ou manutenção de posição. Este objetivo é alcançado utilizando a tendência para treinar e executar a rede, conforme pode ser exemplificado na Figura 26.

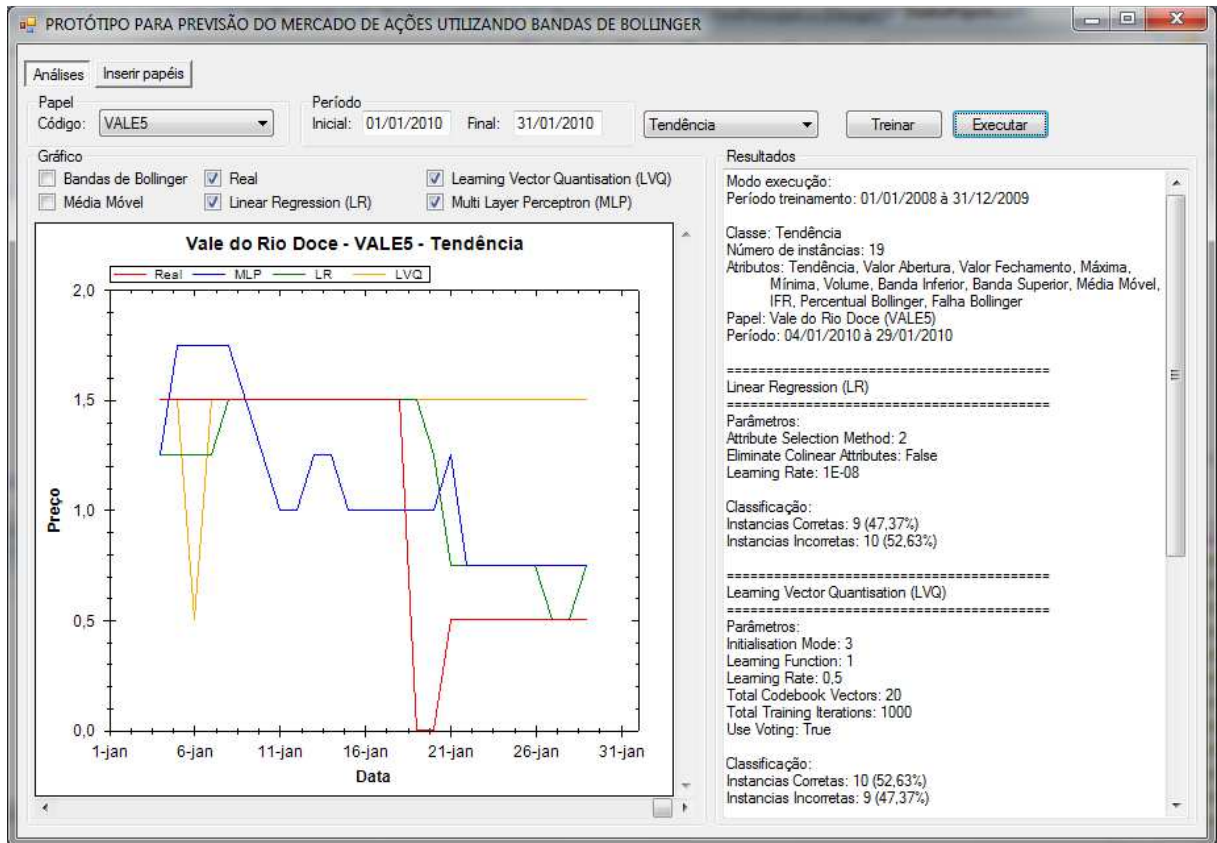


Figura 26 – Tela do protótipo com análise de tendência da Vale do Rio Doce

Após treinar as redes no período de 01/01/2008 à 31/12/2009 e executar as redes em janeiro de 2010, os gráficos da MLP e LR indicam possibilidade de compra dentro de um movimento de baixa, o gráfico da LVQ indica a manutenção da posição, enquanto o gráfico de tendência real utilizando as BB's também indica a manutenção da posição. O desempenho geral de cada rede em comparação com os estudos reais pode ser visualizado na lateral direita do protótipo, e o método para interpretar o gráfico é demonstrado no Quadro 8.

3.4 RESULTADOS E DISCUSSÃO

O presente trabalho apresentou o projeto e desenvolvimento de um protótipo para previsão do mercado de ações utilizando BB's, onde foram utilizadas diferentes redes neurais artificiais para identificar os padrões das BB's e demais indicadores auxiliares. Assim, foi possível indicar previsões sobre quais atitudes devem ser tomadas a respeito de determinada ação e em determinado momento do mercado.

O Quadro 22 apresenta o desempenho dos algoritmos em relação à ação da Petrobras

sobre diversos períodos de treinamentos e testes. O desempenho médio do algoritmo LR foi 16,75% de cotações classificadas corretamente, enquanto o LVQ classificou 48,25% das cotações corretamente e o MLP classificou 11,37%.

Desempenho da ação da Petrobras testada no período de 1998 a 2010					
Treinamento	Teste	Cotações	LR	LVQ	MLP
01/1998 à 12/2009	01/1998 à 12/2009	2949	10,99%	54,36%	9,46%
	01/2009 à 12/2009	246	19,92%	61,79%	4,88%
	01/2009 à 08/2010	411	16,30%	49,64%	9,73%
	01/2010 à 08/2010	165	10,91%	31,52%	16,97%
	06/2010 à 08/2010	64	9,38%	39,06%	10,94%
01/2004 à 12/2009	01/2004 à 12/2009	1484	17,72%	57,21%	8,49%
	01/2009 à 12/2009	246	19,11%	61,79%	9,35%
	01/2009 à 08/2010	411	16,55%	49,64%	11,92%
	01/2010 à 08/2010	165	12,73%	31,52%	15,76%
	06/2010 à 08/2010	64	12,50%	39,06%	21,88%
01/2004 à 12/2007	01/2004 à 12/2007	989	22,45%	59,15%	7,79%
	01/2007 à 12/2008	494	22,06%	52,63%	5,26%
	01/2008 à 12/2008	249	17,27%	44,18%	4,42%
	01/2009 à 12/2009	246	30,49%	61,79%	6,10%
	01/2010 à 08/2010	165	21,82%	32,12%	11,52%
01/2007 à 12/2009	01/2007 à 12/2009	740	22,30%	55,95%	20,41%
	01/2009 à 12/2009	246	17,89%	62,20%	13,82%
	01/2009 à 08/2010	411	15,33%	50,12%	13,38%
	01/2010 à 08/2010	165	11,52%	32,12%	12,73%
	06/2010 à 08/2010	64	7,81%	39,06%	12,50%

Quadro 22 – Desempenho da ação da Petrobras testada no período de 1998 a 2010

O Quadro 23 apresenta o desempenho dos algoritmos em relação à ação da Vale do Rio Doce sobre diversos períodos de treinamentos e testes. O desempenho médio do algoritmo LR foi 11,85% de cotações classificadas corretamente, enquanto o LVQ classificou 55,32% e o MLP classificou 15,84%.

Desempenho da ação da Vale do Rio Doce testada no período de 1998 a 2010					
Treinamento	Teste	Cotações	LR	LVQ	MLP
01/1998 à 12/2009	01/1998 à 12/2009	2949	12,58%	54,15%	7,49%
	01/2009 à 12/2009	246	12,20%	58,94%	6,91%
	01/2009 à 08/2010	411	12,41%	57,91%	7,79%
	01/2010 à 08/2010	165	12,73%	58,18%	9,09%
	06/2010 à 08/2010	64	7,81%	53,13%	9,38%
01/2004 à 12/2009	01/2004 à 12/2009	1484	16,24%	54,72%	8,56%
	01/2009 à 12/2009	246	12,60%	56,91%	4,88%
	01/2009 à 08/2010	411	12,41%	59,37%	7,54%
	01/2010 à 08/2010	165	12,12%	62,42%	11,52%
	06/2010 à 08/2010	64	4,69%	59,38%	7,81%
01/2004 à 12/2007	01/2004 à 12/2007	989	16,99%	59,35%	56,83%
	01/2007 à 12/2008	494	12,35%	53,44%	33,60%
	01/2008 à 12/2008	249	11,65%	47,39%	17,27%
	01/2009 à 12/2009	246	12,60%	51,22%	49,19%
	01/2010 à 08/2010	165	9,70%	58,79%	26,06%
01/2007 à 12/2009	01/2007 à 12/2009	740	15,95%	50,14%	11,76%
	01/2009 à 12/2009	246	11,79%	53,25%	13,01%
	01/2009 à 08/2010	411	11,44%	56,69%	10,46%
	01/2010 à 08/2010	165	10,91%	55,76%	6,67%
	06/2010 à 08/2010	64	7,81%	45,31%	10,94%

Quadro 23 – Desempenho da ação da Vale do Rio Doce testada no período de 1998 a 2010

Referente aos trabalhos correlatos apresentados, o trabalho de Sestrem (2005), utilizou-se os algoritmos *RProp* e *QuickProp* (variações do *Backpropagation*) para treinar as redes neurais. Como entradas foram feitas variações entre o valor de abertura, fechamento, máxima, mínima e volume, retornando como saída (valor de previsão) o valor de fechamento. Este trabalho obteve uma taxa de acerto de 31%, o que não garante sucesso na maioria dos casos.

O trabalho de Faria et al. (2008), implementa uma rede neural MLP, utilizando o algoritmo *backpropagation* para fazer a previsão do Índice Bovespa. Como entrada foi utilizado o valor de fechamento do Índice Bovespa em diferentes dias, retornando como saída o próprio valor de fechamento. Desta forma, o Índice Bovespa pode ser previsto com erros médios em torno de 2%.

Ambos os trabalhos apresentaram como previsão, o valor de fechamento. Este tipo de previsão pode ser simulada no protótipo proposto conforme mostra a Figura 24, apresentando visualmente resultados semelhantes. Porém o que realmente difere este protótipo dos demais trabalhos é a previsão de uma tendência, indicando ao investidor a possibilidade de compra, venda ou de se manter uma posição. Desta forma, o investidor consegue aplicar a previsão do protótipo em suas operações reais, baseadas nestas indicações.

4 CONCLUSÕES

Este trabalho propôs o desenvolvimento de um protótipo para a previsão da tendência de determinada ação, utilizando para isso as BBs como principal parâmetro para designar uma tendência e RNAs para realizarem os treinamentos e obterem as previsões. Os resultados obtidos com o desenvolvimento deste protótipo foram satisfatórios, os requisitos propostos foram cumpridos, e assim o objetivo principal do trabalho foi alcançado.

A biblioteca WEKA, utilizada na implementação dos algoritmos MLP, LR e LVQ, foi de grande importância no trabalho, pois assim puderam ser feitas comparações de desempenho entre as RNAs, garantindo assim maior clareza e confiabilidade na conclusão dos resultados. Além disso, a implementação de gráficos utilizando a biblioteca *ZedGraph*, proporciona ao usuário uma visualização e interpretação melhor dos resultados.

Desta forma, o algoritmo LVQ obteve os melhores resultados com acertos nas previsões superiores a 50%, indicando a possibilidade de utilização inclusive como ferramenta auxiliar na tomada de decisões por parte dos investidores. Entretanto, o MLP obteve uma média de acertos inferior a 15%, além de ocupar um tempo elevado de processamento em comparação com o LVQ e LR. Isto demonstra que algoritmos com técnicas mais básicas como é o caso do MLP com *backpropagation*, não se mostraram eficientes, ao contrário de algoritmos com técnicas mais especializadas como o LVQ e LR.

Como limitação, o protótipo lista em seus resultados as previsões das tendências apenas em forma numérica. Assim, o usuário necessita interpretar estes números para definir se o protótipo está indicando compra, venda ou manutenção de tendência.

Por fim, apesar dos resultados divergentes entre as redes neurais, é possível utilizá-las como ferramentas para filtrar e eliminar padrões que afetam as decisões nas operações na bolsa de valores. Inclusive os resultados obtidos com o algoritmo LVQ, sugerem seu uso em operações reais, uma vez que as chances de sucesso ultrapassam os 50%.

4.1 EXTENSÕES

A seguir são apresentados alguns pontos que podem ser agregados ou melhorados no protótipo. Segue as seguintes sugestões:

- a) acrescentar outras técnicas para predição de ações no protótipo e no estudo, utilizando sistemas especialistas (como árvores de decisão, por exemplo) e baseados em conhecimento (lógica de predicados);
- b) melhorar o relatório para que seja apresentado ao usuário de forma clara qual operação deverá ser feita baseada nas previsões, de modo com que não seja necessária a interpretação gráfica nas tomadas de decisões;
- c) utilizar outros indicadores em conjunto com as BBs como ondas de Elliott e Fibonacci, afim de formular uma melhor definição de tendência;
- d) implementar métodos para salvar os estudos e obter um acompanhamento das operações que estão sendo realizadas baseadas nas previsões do protótipo.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABELEM, Antonio. J. G. **Redes neurais artificiais na previsão de séries temporais**. 2004. 100 f. Dissertação (Mestrado em Engenharia Elétrica) – Curso de Mestrado em Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- ABERNETHY, Michael. **Mineração de dados com WEKA, Parte 1: Introdução e regressão**. [S.l.], 2010. Disponível em: <<http://www.ibm.com/developerworks/br/opensource/library/os-weka1/>>. Acesso em: 11 out. 2010.
- ACHELIS, S. **Technical analysis from A to Z**. New York: McGraw-Hill, 2001.
- BAIÃO, Leonardo. **Bandas de Bollinger**. Rio de Janeiro, 2008. Disponível em: <<http://www.conexaodineiro.com/analise-tecnica/bandas-Bollinger>>. Acesso em: 03 abr. 2010.
- BITTENCOURT, Guilherme. **Inteligência artificial: ferramentas e teorias**. 3. ed. Florianópolis: Editora da UFSC, 2006.
- CALDERARO, Fábio. **Por que utilizar Home Broker?** [S.l.], 2009. Disponível em: <<http://www.meumilhao.com.br/por-que-utilizar-home-broker/>>. Acesso em: 11 out. 2010.
- CARNEIRO, Leandra. **Redes neurais, inteligência artificial e aplicações na medicina**. [S.l.], 2009. Disponível em: <<http://www.webtuga.com/redes-neurais-inteligencia-artificial-e-aplicacoes-na-medicina/>>. Acesso em: 04 out. 2010.
- CHAVES, Daniel A. T. **Análise técnica e fundamentalista: divergências, similaridades e complementaridades**. 2004. 119 f. Monografia (Bacharel em Administração) – Curso de Administração, Universidade de São Paulo, São Paulo.
- DOBSON, Edward D. **Understanding Bollinger bands**. Cedar Falls: Trader Press, 1994.
- FARIA, Elisângela L. et al. **Previsão do mercado de ações brasileiro utilizando redes neurais de índices de ações**. Rio de Janeiro, 2008. Disponível em: <http://cbpfindex.cbpf.br/publication_pdfs/nt00208.2009_01_16_12_38_46.pdf>. Acesso em: 03 abr. 2010.
- GONÇALVES, Fábio. **Aprenda a analisar e investir em ações na bolsa de valores usando análise gráfica e fundamentalista**. Rio de Janeiro: Ciência Moderna, 2008.
- KOVÁCS, Zsolt L. **Redes neurais artificiais: fundamentos e aplicações**. 2. ed. São Paulo: Collegium Cognito, 1996.

INVESTPEDIA. **Indicadores:** aprenda a utilizar as bandas de Bollinger. [S.l.], 2010. Disponível em: <<http://www.investpedia.com.br/artigo/Indicadores+Aprenda+a+utilizar+as+Bandas+de+Bollinger.aspx>>. Acesso em: 29 set. 2010.

LOESCH, Claudio; SARI, Solange T. **Redes neurais artificiais fundamentos e modelos.** Blumenau : Editora da FURB, 1996.

MONTANARI, Fernando. **Como combinar análise técnica e fundamentalista.** São Paulo, [2009]. Disponível em: <<http://www.meumilhao.com.br/como-combinar-analise-tecnica-e-fundamentalista/>>. Acesso em: 25 abr. 2010.

MUELLER, Alessandro. **Uma aplicação de redes neurais artificiais na previsão do mercado acionário.** 1996. 98 f. Dissertação (Mestrado em Engenharia de Produção) – Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

MURPHY, John. **Technical analysis of the futures markets.** New York: New York Institute of Finance, 1986.

NATCOMP. **Redes neurais artificiais.** [S.l.], 2010. Disponível em: <<http://www.natcomp.com.br/lvcon/tema?tema=2>>. Acesso em: 29 set. 2010.

NORONHA, Márcio. **Análise técnica:** teorias, ferramentas, estratégias. Rio de Janeiro: Editec, 1995.

RIBEIRO, Rafael S. **Análise técnica:** como utilizar com maior eficiência as bandas de Bollinger. São Paulo, 2010. Disponível em: <<http://web.infomoney.com.br/templates/news/view.asp?codigo=1765458&path=/investimentos/>>. Acesso em: 29 set. 2010.

SCIELO. **Usando redes neurais artificiais e regressão logística na predição da hepatite A.** [S.l.], 2005. Disponível em: <http://www.scielo.br/scielo.php?pid=S1415-790X2005000200004&script=sci_arttext>. Acesso em: 29 set. 2010.

SESTREM, Evandro. **Desenvolvimento de um protótipo para a predição de índices de ações.** 2005. 42 f. Monografia (Especialização em Tecnologias para Desenvolvimento de Aplicações Web) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SOGAB. **Origem e divisão do sistema nervoso.** [S.l.], 2010. Disponível em: <<http://www.sogab.com.br/anatomia/sistemanervosojonas.htm>>. Acesso em: 29 set. 2010.

WINGER, Bernard.; FRASCA, Ralph. **Investments:** introduction to analysis and planning, 3.ed. Englewood Cliffs: Prentice Hall, 1995.

WITTEN, Ian H.; FRANK, Eibe. **Data mining:** practical machine learning tools and techniques with Java implementations. San Francisco : Morgan Kaufmann, 2000.