

Mario Antonio Bortoli Filho

**Indicadores Baseados em Diferenças de
Primeira Ordem para Orientar Robôs no
Mercado *FOREX***

Salvador, Bahia

2015

Mario Antonio Bortoli Filho

Indicadores Baseados em Diferenças de Primeira Ordem para Orientar Robôs no Mercado *FOREX*

Monografia apresentada ao Colegiado de
Sistemas de Informação da Universidade
do Estado da Bahia como parte integrante
dos requisitos necessários para obtenção
do diploma de Bacharel em Sistemas da
Informação

Universidade do Estado da Bahia – UNEB

Departamento de Ciências Exatas e da Terra I – DCET I

Programa de Graduação em Sistemas de Informação

Orientador Leandro Santos Coelho de Souza

Orientador Diego Gervasio Frías Suárez

Salvador, Bahia

2015

Mario Antonio Bortoli Filho

Indicadores Baseados em Diferenças de Primeira Ordem para Orientar Robôs no Mercado *FOREX*

Monografia apresentada ao Colegiado de Sistemas de Informação da Universidade do Estado da Bahia como parte integrante dos requisitos necessários para obtenção do diploma de Bacharel em Sistemas da Informação

Trabalho aprovado. Salvador, Bahia, 19 de novembro de 2015:

Leandro Santos Coelho de Souza
Orientador

Diego Gervasio Frías Suárez
Orientador

Eduardo Manuel de Freitas Jorge
Convidado

Julian Hermógenes Quezada
Convidado

Salvador, Bahia
2015

Esse trabalho é dedicado À minha mãe, Margarete. Ao meu pai, Mario. E à minha
irmã, Michele.

AGRADECIMENTOS

À minha família pelo amor e apoio incondicional em todos os momentos da minha vida, mas principalmente ao longo dessa jornada, mesmo que distantes uns dos outros.

À minha namorada e companheira Lorena Pereira pela compreensão, por me motivar nos momentos mais difíceis e me acalmar naqueles de incerteza.

Aos meus colegas e amigos Alan Deivite, Rafael Factum, Elton Fraga, Marília Moraes, Leone Jesus, Adailton Cerqueira, Fagner Pimentel, José Grimaldo, Rafael Guimarães, Camila Laranjeira, Ayran Costa, George Dias, Flávio Sapucaia, Marlina Martins, Emmanuel Argollo, Augusto Lima, Lucas Freitas, pelas pesquisas, conversas, discussões e momentos de descontração.

Aos meus orientadores Diego Frías e Leandro Coelho, que me fizeram aprender enquanto superava desafios e onde as orientações foram valiosas não apenas para esse trabalho, mas para a vida.

E a todos os professores e funcionários desse colegiado, que foram tão importantes para o meu desenvolvimento acadêmico e profissional.

*Antes que a tarde amanheça
e a noite vire dia
põe poesia no café
e café na poesia
Paulo Leminski*

RESUMO

Técnicas da inteligência artificial têm se estabelecido para previsão de séries temporais em detrimento das técnicas tradicionais econométricas, principalmente nas séries estocásticas e não lineares, como séries financeiras. Dentre as técnicas, destaca-se o uso de redes neurais artificiais (RNAs) para extrapolar e encontrar tendências e padrões em séries temporais. Nesse sentido, acredita-se que extrair dados e informações das séries provenham resultados e previsões melhores ao serem utilizados com RNAs do que a utilização dos dados da própria série. Uma das formas de se extrair informações é através de análises técnicas. Esse trabalho propôs e desenvolveu uma análise técnica a ser aplicada em séries do mercado financeiro *FOREX* para extração de indicadores que possam ser utilizados para suporte a decisão de operação no mercado através de redes neurais artificiais. Pôde-se observar que o modelo preditivo – composto pelos indicadores, rede neural e robô negociador – obteve taxas de sucesso interessantes em certas configurações da análise e do robô, comprovando a viabilidade da análise criada e dos indicadores extraídos. Os resultados preliminares ainda não permitem chance de ganho real nesse modelo, não obstante, os resultados são encorajadores para a sequência da pesquisa nesse tema, havendo demonstrado a possibilidade de maiores ganhos.

Palavras-chaves: Análise Técnica. FOREX. Redes Neurais Artificiais. Predição. Séries Temporais.

ABSTRACT

The techniques of artificial intelligence has been established for time series forecasting at the expense of traditional econometric models, especially when the series has stochastic and nolinear behaviour such as financial series. Among the techniques, artificial neural networks (ANN) stands out as a way to extrapolate and to find tendencies and patterns in time series. Accordingly, it is believed that data and information extracted from the series provides better results and predictions when used with ANN than using just the series. Technical analysis is a way in which it is possible to extract data and information from time series. This research proposed and developed a technical analysis to be applied in time series from financial market FOREX, in order to extract indicators that can be used in decision support in the market through the use of neural networks. Given this perspective, a prediction model composed of the technical analysis, neural networks and a negotiator robot was created. The results obtained were compared to a random model and an informed random model. It was observed that the created model provides interesting sucess rates in certain configurations for the the analysis and the robot, proving the feasibility of the analysis and created of the extracted indicators. Preliminary results do not yet permit chance of real monetary gain in this model, however they encourage the continuity of the research in this subject, having demonstrated the possibility of higher gain.

Key-words: Technical Analysis. FOREX. Artificial Neural Networks. Prediction. Time Series.

LISTA DE ILUSTRAÇÕES

Figura 1 – Gráficos da série <i>ask</i>	30
Figura 2 – Interação de um agente com o ambiente (Adaptado de: Norvig e Russell (2014, p. 32))	31
Figura 3 – Modelo de neurônio artificial (Adaptado de: Haykin (2001, p. 36)).	34
Figura 4 – Função limiar	35
Figura 5 – Função Logística	35
Figura 6 – Exemplo de uma rede neural artificial com alimentação adiante e diversas camadas. Cada x representa uma entrada e cada y uma saída. (Extraído de: Barbosa et al. (2005, p. 249))	36
Figura 7 – Representação da arquitetura de uma rede FTDNN (Adaptado de: Haykin (2001, p. 695))	38
Figura 8 – Fluxo Metodológico	41
Figura 9 – O modelo preditivo	46
Figura 10 – Exemplo de funcionamento das SEQs, contabilizando três SEQs positivas e três SEQs negativas.	47
Figura 11 – Resultados das redes neurais e dos modelos aleatórios por parâmetro do robô.	59
Figura 12 – Resultados das redes neurais e dos modelos aleatórios por parâmetro do robô.	61
Figura 13 – Agente proposto (Adaptado de: Norvig e Russell (2014, p. 32)) . .	65
Figura 14 – Gráficos dos indicadores extraídos na série do dia 05/05/2014, no período das 12:00 às 13:00. Dados extraídos de uma corretora real.	75

LISTA DE TABELAS

Tabela 1	– Os 4 pares de moedas mais negociados no mercado <i>FOREX</i>	26
Tabela 2	– Exemplo demonstrando a abertura de operação de compra e depois fechando a operação como venda (Fonte: Adaptado de Comissão de Valores Mobiliários (2014)).	27
Tabela 3	– Descrição do PEAS para um robô aspirador de pó	32
Tabela 4	– Propriedades do ambiente de tarefa para um agente inteligente de aspirador de pó	33
Tabela 5	– Valores a serem utilizados para o cálculo dos indicadores	48
Tabela 6	– Conjuntos de parâmetros utilizados no robô	50
Tabela 7	– Os dois conjuntos de limiares	52
Tabela 8	– Topologias testadas para cada arquitetura	54
Tabela 9	– Registros feitos pelo robô na Semana 1 com janela em modo sequencial, utilizados no treinamento das redes e do modelo Aleatório Informado.	57
Tabela 10	– Topologias escolhidas para cada tipo de rede para o indicadores sequenciais.	57
Tabela 11	– Registros feitos pelo robô na Semana 2 com janela em modo sequencial, utilizados nos testes das redes e dos modelos aleatórios. 58	
Tabela 12	– Comparativo dos modelos com janela em modo sequencial. FTDNN-A utiliza o conjunto de limiares A e assim por diante.	58
Tabela 13	– Registros feitos pelo robô na Semana 1 com janela no modo em saltos, utilizados no treinamento das redes e do modelo Aleatório Informado.	60
Tabela 14	– Topologias escolhidas para cada tipo de rede para janela no modo em saltos.	60
Tabela 15	– Registros feitos pelo robô na Semana 2 com janela no modo em saltos, utilizados no teste das redes e dos modelos aleatórios.	60
Tabela 16	– Comparativo dos modelos com indicadores em saltos.	61
Tabela 17	– Descrição do PEAS para o agente proposto.	65
Tabela 18	– Propriedades do ambiente de tarefa para o agente inteligente proposto	66

LISTA DE ABREVIATURAS E SIGLAS

DSS	Sistemas de apoio à decisão
IA	Inteligência Artificial
IDSS	Sistemas inteligentes de apoio à decisão
<i>FOREX</i>	<i>FOReign EXchange</i>
FTDNN	Rede Neural Alimentada Adiante Focada Atrasada no Tempo
RNA	Redes Neurais Artificial
RNAs	Redes Neurais Artificiais
SL	<i>Stop-Loss</i>
TP	<i>Take-Profit</i>

SUMÁRIO

1	INTRODUÇÃO	21
2	O <i>FOREX</i>, SUAS SÉRIES E PREVISÃO	25
2.1	O Mercado <i>FOREX</i>	25
2.1.1	Pares de moedas e operações	26
2.2	Análises de Tendências do Mercado Financeiro	28
2.3	Séries Temporais	28
2.4	Agentes Inteligentes	31
2.5	Redes Neurais Artificiais	33
2.5.1	A arquitetura FTDNN	37
2.5.2	Treinamento da rede	38
3	INDICADORES BASEADOS EM DIFERENÇAS DE PRIMEIRA ORDEM	41
3.1	Trabalhos Correlatos	42
3.2	Contextualização do Problema	44
3.3	Caracterização do Problema	45
3.4	O Modelo Preditivo Proposto	46
3.5	A Análise Técnica	46
3.6	O Robô	49
3.7	Redes Neurais Utilizadas	50
3.7.1	Pré-processamento dos dados	50
3.7.2	Transformação das saídas das redes	51
3.7.3	Estratégia de treinamento	52
3.8	Estratégia de Teste	53
3.9	Avaliação de Desempenho e Apresentação dos Resultados	54
4	ANÁLISE DOS RESULTADOS	57
4.1	Janela no Modo Sequencial	57
4.1.1	Treinamento e escolha das redes	57
4.1.2	Resultados com a Semana 2	58
4.2	Janela no Modo em Saltos	59
4.2.1	Treinamento e escolha das redes	59
4.2.2	Resultados com a Semana 2	60

4.3	Comparativo	62
5	CONCLUSÕES	63
5.1	Trabalhos Futuros	64
5.1.1	Agente inteligente proposto	64
	REFERÊNCIAS	67
	APÊNDICES	71
	APÊNDICE A – PROCESSO DE CRIAÇÃO DOS INDICADORES UTILIZADOS	73

1 INTRODUÇÃO

Prever o futuro sempre intrigou a humanidade. Saber o que ocorrerá em momentos a frente do seu tempo pode auxiliar a tomada de decisão e impactar no resultado da situação. Nesse contexto, a predição de mercados financeiros é um problema clássico, com grande apelo científico (dada a complexidade desses mercados) e monetário, permitindo maiores ganhos.

Desta maneira, a análise de séries temporais constitui uma forma importante de analisar o passado para tentar prever o futuro, principalmente em mercados financeiros, onde é utilizada para prover um maior suporte nas decisões dos investidores e assim possibilitando maiores ganhos. Todavia, a estocasticidade e a não linearidade, características intrínsecas às séries financeiras, tornam a tarefa de prevê-las um grande desafio, sobretudo em previsões de longo prazo. Essas características ainda podem ser agravadas caso o mercado financeiro tenha grande volume de transações, apresentando alta volatilidade, como é o caso do mercado *FOREX*¹.

Supõe-se ainda que o *know-how* dos métodos de previsão de ativos de mercado financeiro está restrito à poucas pessoas, além de representar um problema que exige uma decisão rápida e precisa sobre as operações a serem realizadas, principalmente no mercado de câmbio *FOREX*, que é um dos maiores mercados do mundo, com grande liquidez, volatilidade e volume de transações. Assim sendo, prever o futuro de um ativo nesse tipo de mercado representa um desafio, sobretudo aliado às restrições de tempo. Acredita-se na possibilidade em prever comportamentos de ativos nesse mercado e assim realizar operações automatizadas mais lucrativas e minimizando as perdas, a partir do uso de técnicas da inteligência artificial, estatística e da economia.

Não obstante, é possível utilizar modelos estatísticos para tentar extrapolar séries temporais, contudo, os modelos tradicionais não conseguem “replicar a dinâmica inerente às séries econômicas e financeiras” (COELHO et al., 2008, p. 1). Pode-se, então, recorrer aos modelos computacionais, mais precisamente redes neurais artificiais, para realizar tal tarefa.

Redes neurais são consideradas modelos não lineares, auto-adaptáveis, capazes de generalizar ou aproximar funções através de um treinamento com uma amostra de dados, sendo especialmente interessantes utilizá-las em tarefas de predição de séries temporais (KHASHEI; BIJARI, 2010; LI; MA, 2010).

¹ O mercado *FOREX* está detalhado na seção 2.1

A literatura existente mostra que redes neurais muitas vezes superam a precisão e o ganho dos modelos econométricos tradicionais quando se tenta extrapolar uma série econômica-financeira para operar no mercado. Os trabalhos de Coelho et al. (2008), Oliveira (2007), Bahrammirzaee (2010), Alon et al. (2001) e Wood e Dasgupta (1996) avaliam o uso de técnicas da computação e da estatística para prever séries temporais financeiras, verificando que os resultados das redes neurais são melhores que os modelos tradicionais econométricos. Nelson et al. (1999) verificam que tanto métodos estatísticos quanto redes neurais obtiveram melhores resultados quando as séries não possuíam elementos sazonais, sendo que as redes proferiram melhores resultados. Já Bahrammirzaee (2010) faz uma revisão de diversos trabalhos que abordam sistemas especialistas, redes neurais ou modelos híbridos no mercado financeiro, concluindo que as redes neurais obtêm bons resultados principalmente em previsões.

Pode-se, ainda, utilizar redes neurais com indicadores do mercado para buscar padrões e tendências, como visto em Kayal (2010), onde foram utilizados diversos indicadores em diversas frequências como entrada para rede neural, obtendo resultados interessantes. Yao e Tan (2000) avaliam o uso de indicadores e dados da própria séries para previsão de diversos pares de moeda, mostrando ser possível utilizar indicadores simples como entradas para redes neurais e obter boas previsões do mercado. Em uma abordagem mais complexa, Korczak et al. (2013), fazem uso de um sistema multiagente para reduzir o risco de investimento, onde os agentes devem decidir sobre a operação no mercado, sendo que alguns deles fazem uso de indicadores.

Contudo, as abordagens vistas geralmente fazem uso de dados discretos, utilizando dados do mercado agrupados de minuto a minuto ou até mesmo a cada semana. Isso representa um problema para estratégias de muito curto prazo, como a estratégia *Scalper*, em que deve-se realizar diversas ordens em curtos espaços de tempo, fazendo proveito de uma oscilação do mercado, lucrando um pouco em diversas ordens. Mesmo um modelo que utilize dados discretos por minuto pode representar perda para um robô *Scalper*, visto que possíveis momentos de operação intra-minuto não são considerados e podem ser perdidas.

Neste contexto, e por ainda não existir um modelo adequado à esses requisitos, se faz necessário a construção de um modelo para previsão e decisão capaz de analisar séries temporais não lineares estocásticas com eficácia e eficiência. Esse modelo deve viabilizar o reconhecimento de padrões e comportamentos, assim possibilitando a previsão de tendências, para que seja tomada uma decisão autônoma em tempo real sobre as operações a serem feitas nesse mercado. Não há a intenção de extrapolar

a série com esse modelo, ou seja, não são previstos valores do mercado. Através desse modelo, visa-se uma maior possibilidade de ganho sobre as operações feitas no mercado. No futuro, pode-se aplicá-lo em outras áreas onde seja interessante decidir com base no comportamento de uma série.

O objetivo desse trabalho é realizar um estudo sobre as séries temporais do mercado *FOREX* afim de criar indicadores que possam ser utilizados para prever se há possibilidade de ganho em dado momento com uma estratégia a curto prazo, através do uso de redes neurais artificiais. Nessa perspectiva, criou-se um modelo preditivo composto pela análise técnica, redes neurais e um robô negociador. Os resultados obtidos foram comparados com um modelo aleatório e outro modelo aleatório informado, através de uma métrica criada nesse trabalho. Também pretende-se propor um modelo de agente inteligente que possa ser utilizado futuramente em outros trabalhos.

Logo, espera-se que o *know-how* adquirido através do trabalho de pesquisa apresentado seja aplicado no mercado para possibilitar melhores resultados financeiros ou em problemas que sejam necessários decidir a partir da previsão de comportamentos. Para isso, o trabalho foi dividido em 5 capítulos. O Capítulo 2 refere-se à conceitos do mercado *FOREX*, de séries temporais e suas análises, da inteligência artificial e de redes neurais artificiais.

O Capítulo 3 aborda o trabalho realizado e seu desenvolvimento, apresentando o problema, a metodologia, os *datasets* utilizados, os indicadores criados, os mecanismos e técnicas utilizadas para criação do modelo preditivo, bem como as estratégias de teste e avaliação de desempenho.

Os resultados são vistos e comparados no Capítulo 4, avaliando o desempenho das redes neurais em conjunto com os indicadores criados.

E, por fim, no Capítulo 5 são apresentadas as conclusões finais, assim como trabalhos futuros e a proposta de um agente inteligente criado através do *know-how* obtido com esse trabalho.

2 O *FOREX*, SUAS SÉRIES E PREVISÃO

2.1 O Mercado *FOREX*

Sendo utilizado por diversos bancos, governos e empresas para compra e venda de diversos pares de moedas, o *FOREX* é um dos maiores mercados financeiros do mundo, podendo atingir mais de 4 trilhões de dólares em transações diárias (INVESTOPEDIA, 2014). É considerado um mercado derivativo, com diversos instrumentos financeiros, dentre eles, o mercado de opções é o mais comum no varejo e também o que possui maior liquidez. Para facilitar o entendimento, pode-se pensar que um mercado derivativo é um mercado de aposta, neste caso, a aposta pode ser entendida por sendo: se a relação de uma moeda por outra valorizará ou desvalorizará. No início, o mercado servia apenas à bancos, governos e empresas que interagem através de corretoras especializadas. Com a expansão da Internet, o mercado evoluiu para permitir que qualquer pessoa invista no mercado através de corretoras à varejo.

Atualmente, o *FOREX* não possui regulamentação própria e cada país é responsável pela regulamentação das corretoras e bancos que atuam no mercado. No Brasil, por exemplo, uma corretora seria regulamentada pela Comissão de Valores Mobiliários, entretanto ainda não é permitido que corretoras brasileiras atuem no mercado *FOREX*.

As transações feitas no mercado também são descentralizadas, ou seja, não existe uma entidade ou organização que seja o centro de todas as transações, como acontece com bolsas de valores, por exemplo. Isso permite que ele funcione 24 horas por dia, durante 5 dias da semana, abrindo ao domingo 21:00 GMT e fechando sexta feira 21:00 GMT, onde suas operações migram, dependendo do horário, entre os 4 maiores centros financeiros do mundo: Sidney, Tóquio, Londres e Nova Iorque.

Um investidor pode comprar ou vender um par de moeda, ao vender ele espera que o par desvalorize e, ao comprar, que o par valorize. Essas operações serão melhor detalhadas na subseção 2.1.1. Geralmente, cotações monetárias não variam muito em um período curto, o que geraria pouquíssimo lucro/prejuízo em cada operação (COMISSÃO DE VALORES MOBILIÁRIOS, 2014). Para contornar esse obstáculo, as corretoras utilizam o sistema de alavancagem (*leverage*, em inglês) onde as corretoras “completam” o volume investido pelo cliente. Por exemplo, uma alavancagem 1:1 proporciona que, a cada dólar aplicado pelo investidor em uma

operação, serão negociados \$100.000 dólares pela corretora, isso é possível graças ao *margin cut*¹, que é uma proteção contra perdas que não permite ao negociador que o balanço em sua conta fique negativo através de um percentual máximo de exposição.

Caso o investidor aplique \$100 dólares, a corretora estará negociando \$10.000 dólares, permitindo maior lucro ou prejuízo a ser compartilhado entre a corretora e o investidor.

Esse mercado também está sujeito as leis de oferta e procura, podendo sofrer influências externas de fatores políticos, econômicos e outros eventos, como por exemplo guerras, mudança de presidentes, leis, etc. Sendo assim, é um mercado estocástico, com alto nível de aleatoriedade.

2.1.1 Pares de moedas e operações

A compra e venda de moedas no mercado *FOREX* funciona através de pares de moeda. A quantidade disponível de pares de moedas para negociação depende de cada corretora. Os principais pares negociados e presentes nas maiorias das corretoras são vistos na Tabela 1.

Tabela 1 – Os 4 pares de moedas mais negociados no mercado *FOREX*.

Par	Moeda	Contraparte
EUR/USD	EURO	DÓLAR
USD/JPY	DÓLAR	YEN
GBP/USD	LIBRA	DÓLAR
USD/CHF	DÓLAR	FRANCO SUIÇO

Para negociação, a compra é referenciada no valor de *ask* e a venda no valor de *bid*. O *ask* é o valor que o mercado está disposto a vender uma moeda e sempre será um valor maior ou igual ao *bid*. Já o *bid* é o valor que o mercado está disposto a comprar uma moeda. A diferença entre os dois valores é chamada de *spread*. Esses valores estão sempre em pares e a qualquer momento podem ser atualizados pelo mercado². Essa atualização é chamada de *tick*. Cada *tick* possui um valor de *ask*, *bid* e *spread*, além do horário em que essas cotações ocorreram.

Para facilitar o entendimento do mercado de opções, basta pensar que ao criar uma ordem, deve-se selecionar se será compra ou venda, entretanto o fechamento

¹ Para mais informações sobre o funcionamento de alavancagem e *margin cut* acesse (em inglês): <http://www.investopedia.com/ask/answers/06/forexmargin.asp>

² Uma corretora recebe cotações do mercado e normalmente um servidor é responsável por enviar esses dados

da ordem é do tipo oposto ao de abertura. Para realizar essa tarefa de abertura e fechamento de ordens, várias corretoras permitem a criação de *Expert Advisors*, que são programas para operar de forma automatizada no mercado, seguindo as estratégias implementadas nele, como a de fechar uma ordem caso atinja um nível de prejuízo (*Stop-Loss*) ou de ganho (*Take-Profit*). Os valores dispostos aos investidores correspondem ao valor que se pode comprar ou vender uma moeda pelo preço de outra. No par EUR/USD (Euro/Dólar), por exemplo, o valor apresentado é o custo em dólares para comprar € 1,00. Contudo, essa compra não existe fisicamente, já que neste caso o investidor especula na valorização ou desvalorização do par, comprando o direito de compra ou venda de uma moeda pelo preço determinado naquele momento, podendo até mesmo "vender" euros que não possui. Nesse caso, o objeto de especulação é o euro e servirá apenas como um pivô, na esperança de uma liquidação com lucro, devido à diferença de preços em uma data futura.

A Tabela 2 demonstra como uma operação de compra funciona, utilizando um sistema de alavancagem 1:100. Ao comprar, o investidor espera que o valor de venda do par de moeda aumente.

Tabela 2 – Exemplo demonstrando a abertura de operação de compra e depois fechando a operação como venda (Fonte: Adaptado de Comissão de Valores Mobiliários (2014)).

Exemplo Simplificado	ASK	BID	Equivalente em EUR	Equivalente em USD
Compra de 1 lote EUR/USD	1,30000	1,25000	+100.000	-130.000
Venda de 1 lote EUR/USD	1,35500	1,35000	-100.000	+135.000
Resultado Bruto			0	+5000

Nota: Neste exemplo foi utilizado a alavancagem de 1 para 100, ou seja, o investidor precisou investir \$1300 para conseguir operar com \$130.000. Para simplificação, não foram aplicadas taxas de corretagem.

Para mensurar os valores é comum a terminologia *pips*³. No exemplo da Tabela 2, no momento da compra o valor do *spread* era de 5000 *pips*.

Para tentar prever o mercado pode-se fazer uso de análises, que são estudos sobre as influências em um ativo. Um ativo é tudo aquilo que pode ser comprado, vendido ou especulado no mercado financeiro. No mercado de ações, a ação é um ativo, já no mercado cambial, as moedas são ativos.

³ Um *pip* é o valor unitário da última casa decimal do par de moeda negociado, no exemplo EUR/USD, 1 *pip* equivaleria à 0,00001.

2.2 Análises de Tendências do Mercado Financeiro

Existem alguns tipos de análises, dentre elas as mais vistas atualmente são a análise fundamentalista e a análise técnica. Essas análises tentam criar algum tipo de indicador que possa quantificar a direção e força da tendência de um ativo. Podemos então dizer que o produto de uma análise são os indicadores, e que permitem uma noção sobre o estado atual e futuro de um ativo.

A análise fundamentalista é uma análise qualitativa dos ativos que pretende-se investir a médio e longo prazo. É voltada para o contexto político, econômico, social e empresarial, que podem influenciar na tendência do ativo. Contudo, como as informações necessárias para realizar esse tipo de análise são fornecidas por empresas, governos e outras fontes de informação (e.g., jornais e revistas), tais informações podem ser imprecisas (RUSCHEL, 2012), afetando assim a credibilidade.

Por sua vez, a análise técnica, ou análise quantitativa, geralmente é voltada para aplicações de curto prazo, tentando encontrar padrões matemáticos e estatísticos nas tendências dos preços e volumes dos ativos, através do estudo dos dados passados (RUSCHEL, 2012). Nesse tipo de análise tenta-se simular o comportamento de um ativo, mas ambas análises convergem na tentativa de criar estratégias que realizem previsões das tendências. Em suma, a análise técnica utiliza dados de séries temporais para criar indicadores.

2.3 Séries Temporais

Box et al. (2013) esclarecem que uma série temporal é uma sequência de observações feitas num determinado período de tempo e sua análise permite prever, com certo grau de acurácia, a própria série; estimar a função transferência entre duas séries dinamicamente relacionadas; detecção e análise dos efeitos de eventos anormais que possam influenciar a série; controle estatístico de processos através de ajustes nas variáveis que influenciam o sistema (e conseqüentemente a série). Os autores ainda constataam que séries temporais estão presentes em diversas áreas do conhecimento, tais como economia, negócios e ciências naturais, e que predize-las “pode proporcionar uma base para (1) planejamentos econômicos e de negócios, (2) planejamento de produção, (3) controle de produção e inventário e (4) controle e otimização de processos industriais”⁴ (BOX et al., 2013, p. 1, tradução nossa).

⁴ Originalmente: “*can provide a basis for (1) economic and business planning, (2) production planning, (3) inventory and production control, and (4) control and optimization of industrial processes.*”

Winters (1960) esclarece que as previsões através do estudo das séries temporais podem ser feitas para uma variedade de propósitos e cita como exemplo questões orçamentárias, metas de vendas, gastos e controle e planejamento de inventário, sendo que os métodos para previsão variam conforme seus usos. As séries ainda podem apresentar componentes sazonais (OLIVEIRA, 2007) e comportamentos cíclicos, podendo ser interessante investigar esses comportamentos nos dados e o mecanismo gerador de uma série temporal (BEZERRA, 2006).

Séries temporais possuem características que as classificam e influenciam na forma em que a análise pode ser feita. Uma série temporal pode ser considerada:

- **Determinística x Estocástica:** Quando um modelo matemático descreve com exatidão o valor de uma série em qualquer momento, então ela é determinística, porém é estocástica se seus valores futuros podem ser descritos apenas através de probabilidade (OLIVEIRA, 2007; BOX et al., 2013)
- **Linear x Não Linear:** São lineares quando a trajetória da série temporal pode ser descrita por uma função linear. E não lineares quando a trajetória da série temporal pode ser descrita, mesmo que aproximadamente, através de uma ou mais funções não lineares (OLIVEIRA, 2007; BOX et al., 2013).
- **Estacionária:** É considerado uma classe dentro das séries estocásticas, sendo estacionária caso os valores estatísticos, como média e variância, da série não se alterem com o tempo, caso isso não ocorra, a série é chamada de não estacionária.

Quando se aborda estacionariedade, existem dois tipos: a estrita e a fraca (COSTA, 2012; COSTA; BAIDYA, 2001; MONTGOMERY et al., 2015). Diz-se um processo estritamente estacionário quando, em uma série temporal X_t , as distribuições finito-dimensionais de $X_i, X_{i+1}, \dots, X_{i+n}$ é idêntica à de $X_j, X_{j+1}, \dots, X_{j+n}$, ou seja, não dependem do tempo. Já um processo estacionário fraco possui uma média constante (Equação 2.1), variância finita e autocovariância dependente apenas de um Δt (Equação 2.1).

$$E[X_t] = \mu \quad (2.1)$$

$$Cov(X_t, X_{t+k}) = Cov(|t - t + k|) \quad (2.2)$$

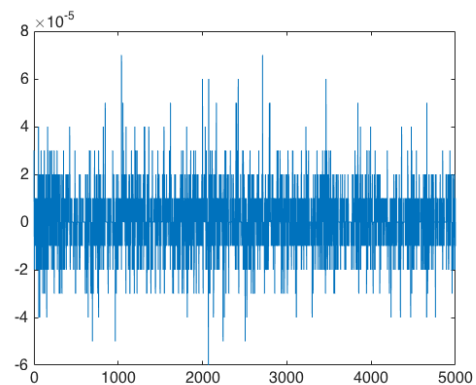
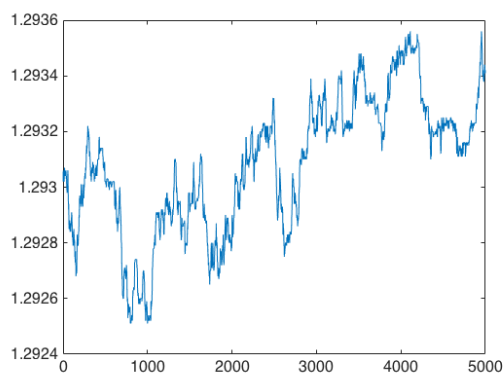
Qualquer referência à um processo estacionário neste trabalho é uma referência à estacionariedade fraca.

Uma série financeira é considerada não linear, estocástica, dotada de sazonalidade e tendências, com comportamentos pseudo-cíclicos e valores atípicos que destoam da série, convergindo assim para a não estacionariedade e tornando difícil a sua previsão (BOX et al., 2013; COSTA, 2012). É possível transformar a série para minimizar essas complexidades e assim obter melhores resultados em sua predição (NELSON et al., 1999). Mesmo com transformações, existe uma preocupação de que os modelos lineares e os econométricos tradicionais não consigam ser satisfatórios na previsão de modelos tão dinâmicos como as séries financeiras e por isso há uma busca nos métodos da inteligência artificial para que melhores modelos sejam criados (COELHO et al., 2008).

O método das diferenças de primeira ordem permite transformar uma série não estacionária em estacionária, obtendo resultados melhores que uma série não transformada quando usado em conjunto com redes neurais (COSTA, 2012; QI; ZHANG, 2008). A diferenciação é feita pela subtração entre uma observação e seu valor antecedente. Como visto em Hyndman e Athanasopoulos (2013), a equação das diferenças de primeira ordem é dada por:

$$X' = X_t - X_{t-1} \quad (2.3)$$

Onde X é uma série temporal com N observações, e X' é a sua diferenciação com $N - 1$ valores. O gráfico na Figura 1a mostra os primeiros cinco mil pontos da série *ask* no dia 05/09/2014, e na Figura 1b está o gráfico da série transformada por diferenças de primeira ordem.



- (a) Primeiros 5000 pontos da série *ask* do dia 05/09/2014, obtidos em uma base de dados de testes de uma corretora real. (b) A mesma série transformada por diferenças de primeira ordem

Figura 1 – Gráficos da série *ask*.

Observando os gráficos verifica-se a capacidade de, uma vez transformados os

dados, serem analisados através de técnicas da inteligência artificial, como agentes inteligentes.

2.4 Agentes Inteligentes

Agentes são entidades computacionais autônomas, capazes de perceber o ambiente em que estão inseridos, seja este físico (e.g. uma sala de estar) ou virtual (e.g. mercado de ações), e agir sobre eles através de atuadores (ou efetadores, pois efetivam a decisão do agente) (WEISS, 1999).

Os Agentes Inteligentes herdam as características de Agentes e devem ser ser racionais, no sentido que não são perfeitos e podem falhar (WEISS, 1999). Eles agem racionalmente conforme a sequência de percepções recebida até o momento; seu conhecimento interno (e.g., heurística, árvores, estados possíveis); a capacidade de seus efetadores; e sua medida de desempenho, tentando sempre maximizá-la (WEISS, 1999; NORVIG; RUSSELL, 2014). A Figura 2 exemplifica o funcionamento de um agente inserido em um ambiente

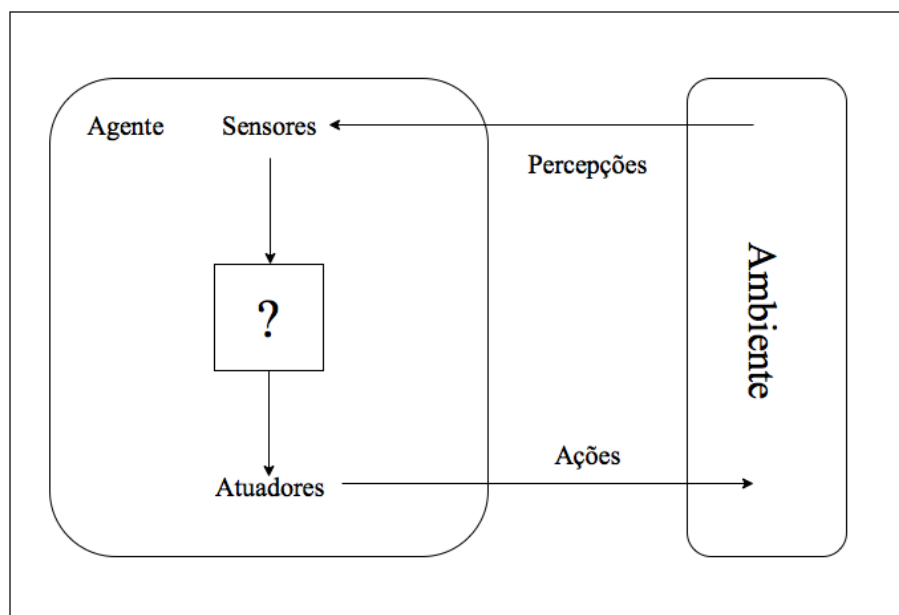


Figura 2 – Interação de um agente com o ambiente (Adaptado de: Norvig e Russell (2014, p. 32))

Para que um Agente Inteligente seja construído, deve-se primeiro entender e descrever o ambiente da tarefa, que engloba a medida de desempenho, o ambiente, os efetadores e os sensores do agente. Essa descrição é por vezes chamada de PEAS (em inglês, *Performance, Environment, Actuators, Sensors*) (NORVIG; RUSSELL,

2014). A Tabela 3 exemplifica o PEAS para um agente aspirador de pó. Neste caso, o aspirador de pó pode se mover por todos os cômodos da casa, mas conhece apenas o estado do espaço em que está posicionado. Após aspirar, ele limpa totalmente aquele local e move-se para o próximo.

Tabela 3 – Descrição do PEAS para um robô aspirador de pó

Tipo de Agente	Aspirador de pó
Medida de Desempenho	Deixar o chão limpo
Ambiente	Pessoas, animais, móveis, cômodos
Atuadores	Direção, aspiração
Sensores	Sensores de sujeira, de obstáculos, de distância e de bateria.

Para o ambiente de tarefa, Norvig e Russell (2014) definem as seguintes propriedades:

- **Completamente observável ou parcialmente observável:** Um ambiente é considerado observável se os sensores do agente possibilitam o acesso ao estado completo de um ambiente em cada momento.
- **Agente único ou multiagente:** Se o ambiente possuirá outros agentes, competindo ou cooperando entre si.
- **Determinístico ou estocástico:** Um ambiente é considerado determinístico caso seja determinado por seu estado atual e pela atuação do agente. Caso contrário, ele será estocástico. Não se deve levar em conta as ações de outros agentes em um ambiente multiagente.
- **Episódico ou sequencial:** Se o agente recebe uma percepção e executa uma ação, então o ambiente é episódico. É importante notar que as ações em um episódio não influenciam episódios futuros.
- **Estático ou dinâmico:** É considerado estático caso não ocorra alterações no ambiente enquanto o agente está tomando uma decisão.
- **Discreto ou contínuo:** Se existe uma quantidade finita de ações e percepções, então o ambiente é discreto (WEISS, 1999).

Para o problema do aspirador de pó, o ambiente de tarefa é caracterizado e explicado na Tabela 4

Tabela 4 – Propriedades do ambiente de tarefa para um agente inteligente de aspirador de pó

Propriedade	Explicação
Parcialmente Observável	Neste caso, o robô não tem noção dos estados dos outros locais ou cômodos da casa
Agente único	Não há outros agentes competindo ou cooperando com
Determinístico	O agente conhece o estado atual e sabe que se aspirar, ficará limpo
Episódico	O agente recebe a percepção se está sujo ou não. Se não estiver ele passa para o próximo local. Se estiver, ele limpa o local. Limpar ou não o espaço atual não interfere na ação futura, em outro local
Estático	Difícilmente o ambiente será alterado enquanto o robô está funcionando
Discreto	Há um número finito de estados e ações para o robô. Mover ou não mover, limpar ou não limpar.

Como pode-se observar, a construção de um agente inteligente demanda um módulo computacional que consiga prover essa inteligência em determinado grau. Assim sendo, as redes neurais artificiais se configuram na literatura como opção válida para agregar esse módulo de inteligência ao agente.

2.5 Redes Neurais Artificiais

As redes neurais artificiais, ou simplesmente redes neurais, são objeto de estudo desde antes de 1943, quando surgiu o primeiro modelo artificial para o neurônio, criado por McCulloch e Pitts (1943) e representado pelo modelo na Figura 3. Haykin (2001) define um neurônio artificial como uma unidade, ou nó, de processamento necessária para o funcionamento de uma rede neural e identifica três elementos básicos:

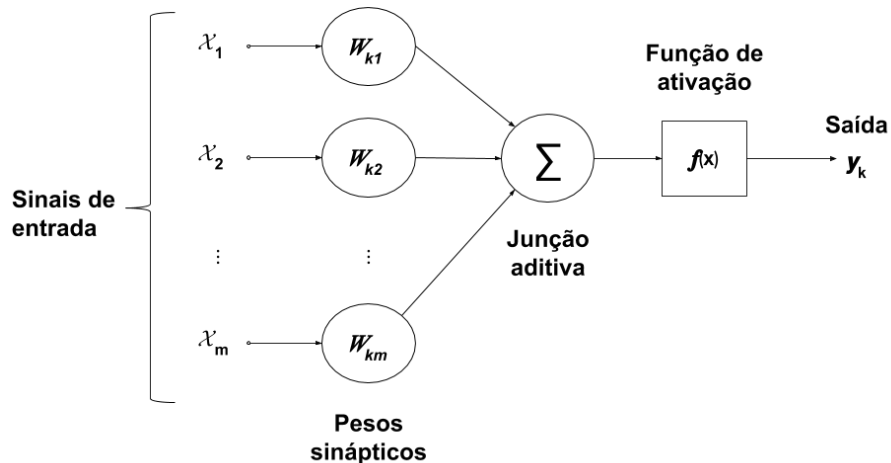


Figura 3 – Modelo de neurônio artificial (Adaptado de: Haykin (2001, p. 36)).

- Sinapses: Cada sinal x_j deve passar por uma sinapse, que possui um peso w_{kj} , onde k refere-se ao neurônio que está conectada e j ao sinal de entrada. O sinal de entrada é multiplicado por esse peso e então repassado para o próximo elemento do neurônio.
- Junção aditiva: Realiza uma soma dos sinais após passarem pelas suas respectivas sinapses.
- Função de ativação: Processa o resultado da soma para restringir o valor em um intervalo, que pode ser entre $[0,1]$ ou $[-1,1]$. O resultado gerado é conhecido como saída do neurônio, que pode servir como entrada para outros neurônios ou como saída da rede.

Para a função de ativação, o ideal seria utilizar uma função que pudesse dividir os dados em dois subconjuntos disjuntos, gerando uma “previsão completamente confiante de 1 ou 0” (NORVIG; RUSSELL, 2014, p. 633). A função limiar (Equação 2.4) atende a esses requisitos e seu gráfico pode ser observado na Figura 4.

$$Degrau(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (2.4)$$

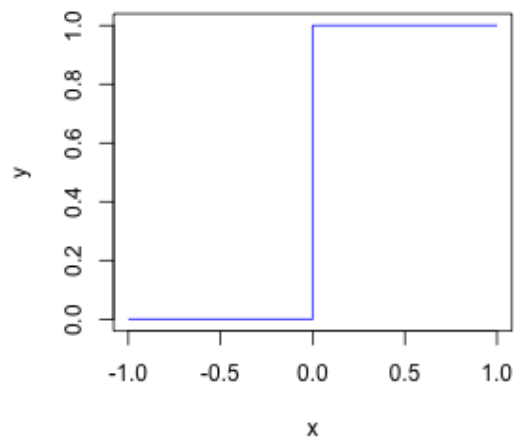


Figura 4 – Função limiar

Contudo, a função limiar não é satisfatória quando não é possível separar linearmente um conjunto de dados (NORVIG; RUSSELL, 2014). Assim se faz necessário o uso de uma função que possa lidar com esse comportamento não linear, como a função sigmóide (HAYKIN, 2001). A função logística (Equação 2.5) é um exemplo de função sigmóide muito utilizada em redes voltadas para problemas de classificação.

$$\text{Logística}(z) = \frac{1}{1 + e^{(-z)}} \quad (2.5)$$

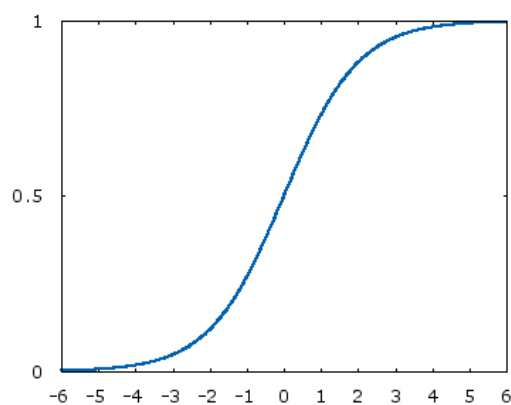


Figura 5 – Função Logística

As funções de ativação a serem utilizadas nesse trabalho são a função tangente hiperbólica (Equação 2.6), função *softmax* (Equação 2.7), e uma função linear

(Equação 2.8).

$$F_1(z) = \frac{2}{1 + e^{(-2*z)}} - 1 \quad (2.6)$$

$$F_2(z_i) = \frac{e^{(z_i)}}{\sum_{j=1}^K e^{(z_j)}} \quad (2.7)$$

onde z_i é a entrada de um neurônio i , sendo K o número de neurônios na camada.

$$F_3(z) = z \quad (2.8)$$

É possível, então, criar uma rede neural organizando os neurônios em camadas. Um dos tipos de rede mais comum é a rede de múltiplas camadas com alimentação para frente (*feedforward*), também chamada de *perceptrons* de múltiplas camadas (*multilayer perceptron - MLP*), exemplificada na Figura 6, possuindo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída da rede. O sinal de entrada é então propagado para as camadas à frente e seus pesos são ajustados conforme os erros obtidos através do algoritmo de treinamento (HAYKIN, 2001).

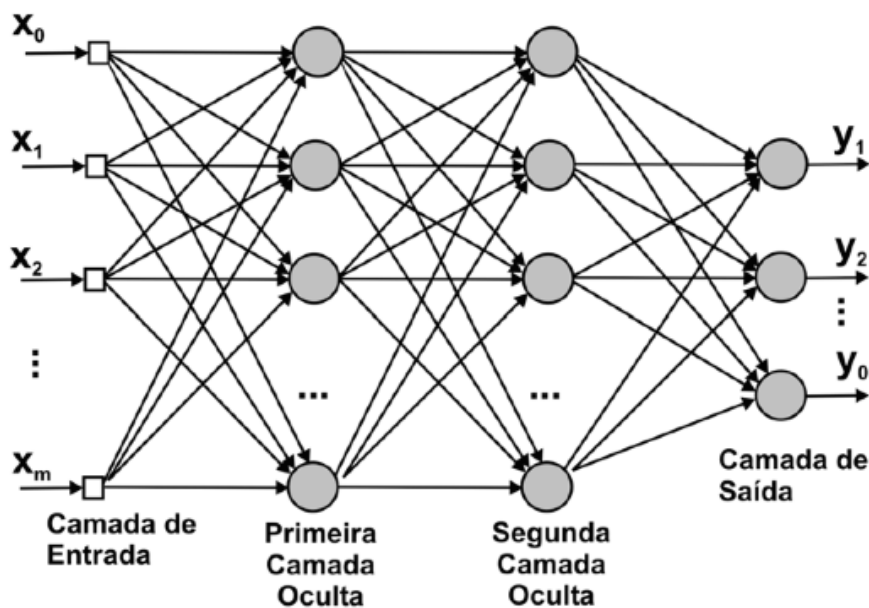


Figura 6 – Exemplo de uma rede neural artificial com alimentação adiante e diversas camadas. Cada x representa uma entrada e cada y uma saída. (Extraído de: Barbosa et al. (2005, p. 249))

Em termos gerais, uma RNA adquire conhecimento através de um processo de aprendizagem, onde um algoritmo de treinamento atualiza seus pesos sinápticos conforme os erros da rede até que uma condição de parada seja atingida. Pode-se

dizer então que o conhecimento da rede está armazenado em seus pesos (HAYKIN, 2001).

Dentre as propriedades das RNA enumeradas por Haykin (2001), pode-se citar:

- Não linearidade: Essa característica é importante pois permite que ela seja utilizada no estudo de mecanismos mais complexos e que geram dados não lineares.
- Mapeamento de entradas e saídas: Permite que o treinamento supervisionado ocorra.
- Adaptabilidade: Uma rede pode ser facilmente re-treinada para lidar com pequenas mudanças no ambiente em que foi projetada.
- Tolerância a erros: Permite que erros ocorram sem invalidar todos os resultados da rede, entretanto, ao longo do tempo, o erros podem expandir pela rede e comprometer seu funcionamento.
- Capacidade de generalização: Se o conjunto de dados para treinamento for considerado bom, a rede neural poderá generalizar e inferir sobre dados que não constavam no conjunto de treinamento.

Norvig e Russell (2014) argumentam que, mesmo após a introdução de modelos mais detalhados e realistas, as propriedades das redes neurais ainda permanecem interessantes para diversos problemas, sendo a previsão de valores em séries temporais um deles (OLIVEIRA, 2007).

Nesse trabalho as redes serão utilizadas para (1) classificar e (2) prever valores através de processamento temporal. Em específico, para processamento temporal, será utilizado uma rede alimentada adiante focada atrasada no tempo (*focused time-delay neural network* – FTDNN).

2.5.1 A arquitetura FTDNN

Enquanto uma rede MLP pode reconhecer padrões e assim classificar, uma rede para processamento temporal necessita da introdução do aspecto de tempo na rede. No caso da FTDNN, o objetivo é inserir o aspecto de tempo em uma rede MLP através de suas entradas, proporcionando dinamismo e uma memória de curto prazo (HAYKIN, 2001; COSTA, 2012).

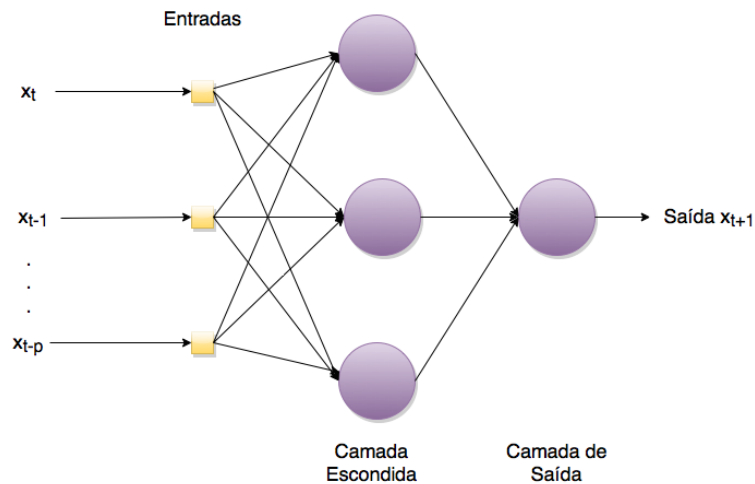


Figura 7 – Representação da arquitetura de uma rede FTDNN (Adaptado de: Haykin (2001, p. 695))

Para realizar a previsão de um momento x_{t+1} , deve-se primeiro escolher a quantidade de atrasos p (valores passados de x) que servirão de entrada juntamente com o momento x_t , sendo assim, uma rede FTDNN possui como entrada um vetor de $p + 1$ valores, expressado pela equação Equação 2.9.

$$x_{t+1} = FTDNN(x_t, x_{t-1}, x_{t-2}, \dots, x_{t-p}) \quad (2.9)$$

Essa característica da FTDNN faz que ela possa ser utilizada não apenas para prever o futuro de uma série, mas também prever o futuro de uma série y que seja relacionada com a série x .

2.5.2 Treinamento da rede

O treinamento da rede é uma das partes mais críticas a serem realizadas e deve-se prosseguir com cautela. Normalmente, para o treinamento supervisionado, é escolhido um conjunto de dados que possa ser separado em três conjuntos disjuntos: um para treinar a rede; outro para validação; e outro para teste. Em certas situações, pode ser interessante dividir os conjuntos de forma contínua, onde cada conjunto recebe um bloco de dados, tal que o conjunto treinamento recebe o primeiro, o conjunto validação recebe o segundo e o conjunto teste recebe o último; intercalada, onde os dados são “fatiados” em partes menores e essas partes menores são divididas entre os conjuntos; e aleatória, onde cada conjunto recebe dados selecionados aleatoriamente. Contudo, cada caso deve ser analisado separadamente, visto que redes

neurais devem ser treinadas para um problema específico. No caso de uma FTDNN, por exemplo, é interessante que os dados sejam contínuos.

É necessário ter cuidado com o conjunto utilizado para treinamento, para que a rede não “decore” os dados, mas passe a “entender” as relações matemáticas entre eles. Caso a rede seja muito grande, ela pode estar sujeita a superadaptação, ou seja, ela apenas decorou os dados do treinamento (NORVIG; RUSSELL, 2014). Ainda pode ser interessante normalizar os valores de treinamento em um domínio específico para melhorar o desempenho da rede (HAYKIN, 2001). O trabalho atual realiza a normalização através da Equação 2.10.

$$\bar{x}_i = \frac{(D_{max} - D_{min}) * (x_i - x_{min})}{(x_{max} - x_{min})} + D_{min} \quad (2.10)$$

Onde D é o domínio em que o valor x_i deve pertencer após a normalização, x_{max} e x_{min} são, respectivamente, o maior e menor valor apresentado por x . Essa equação é implementada pela função `mapminmax()` no software MATLAB[®] versão R2015b (8.6.0.267246).

No treinamento supervisionado, cada entrada possui uma saída desejável e a saída da rede, onde os erros da rede são calculados (Equação 2.12) e, utilizando um algoritmo de treinamento, como o *backpropagation*, os pesos da rede são reajustados através dos erros obtidos. Uma forma de se calcular os erros é através do erro quadrático médio (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N (er_i)^2 \quad (2.11)$$

$$er_i = alvo_i - saída_i \quad (2.12)$$

Onde N é o número de exemplos apresentados à rede, *alvo* é o resultado esperado da rede e *saída* é a resultado real da rede.

As entradas são novamente computadas, o erro é recalculado e o processo de reajuste é refeito. Esse processo de treinamento continua até que uma das seguintes condições sejam atingidas (HAYKIN, 2001):

- Erro mínimo: Quando a taxa de erro atinge um valor pequeno
- Número máximo de épocas: Quantidade de vezes que pode-se reajustar os pesos da rede.

- Variação da Taxa de Aprendizagem: Quando a variação for muito pequena ou muito grande.
- Parada antecipada: A cada K épocas, testa-se a rede no conjunto de validação, se o erro crescer por N vezes seguidas, então o treinamento é parado.

Esse trabalho faz uso do algoritmo de treinamento Levenberg-Marquardt, que utiliza a aproximação da matriz Hessiana através da matriz Jacobiana constituída pelas derivadas parciais dos erros (Equação 2.12) em relação aos pesos, tal que

$$H \approx J^T J \quad (2.13)$$

Hagan e Menhaj (1994) demonstram que, seja Equação 2.14 o método de Newton para minimizar uma função $V(x)$ com relação ao vetor x , sendo $V(x)$ como a função erro segundo a Equação 2.11 e $\Delta V(x)$ como o gradiente

$$\Delta x = -[H(x)]^{-1} \Delta V(x) \quad (2.14)$$

E que

$$\Delta V(x) = J^t(x)e(x) \quad (2.15)$$

Sendo $e(x)$ o vetor de erros conforme a Equação 2.12. A modificação de Levenberg-Marquardt na Equação 2.14 é:

$$\Delta x = [J^T(x)J(x) + \mu I]^{-1} J^T(x)e(x) \quad (2.16)$$

Onde μ é multiplicado por um fator β arbitrário toda vez que $V(x)$ aumentar e é dividido por β toda vez que $V(x)$ diminuir (HAGAN; MENHAJ, 1994).

3 INDICADORES BASEADOS EM DIFERENÇAS DE PRIMEIRA ORDEM

Os indicadores foram criados para servirem de entrada para uma rede neural, que decide sobre a operação de um robô no mercado. Para desenvolvê-los se fez necessário um fluxo metodológico (Figura 8) que será detalhado a seguir.

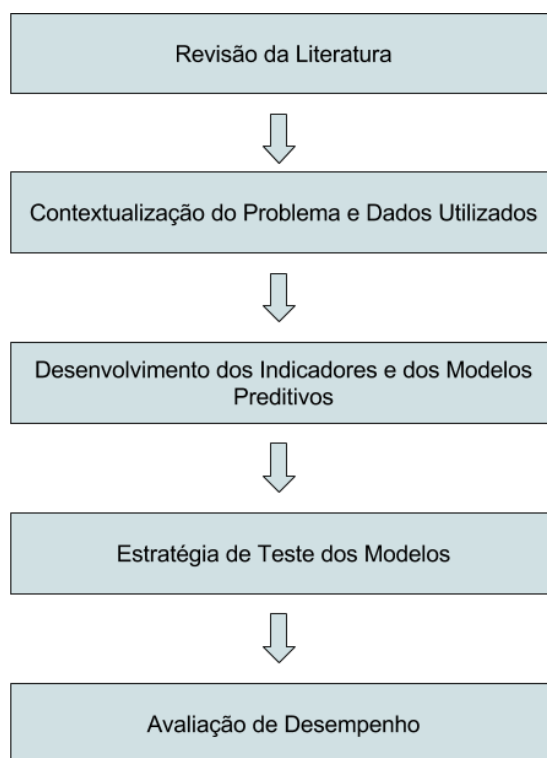


Figura 8 – Fluxo Metodológico

A primeira etapa é composta pela seção 3.1 onde é apresentado um apanhado geral sobre trabalhos na literatura que estão relacionados com o trabalho desenvolvido.

A segunda é composta pelas seções 3.2 e 3.3, que contextualiza o problema abordado e os dados utilizados, justificando o uso e a criação dos indicadores como objetivo principal desse trabalho.

A terceira etapa aborda o modelo preditivo proposto (seção 3.4), desenvolvimento dos indicadores (seção 3.5), do robô (seção 3.6) e das redes neurais utilizadas (seção 3.7), além dos seus mecanismos e especificidades, de forma que seja possível entender como essas partes se relacionam e funcionam.

Na quarta etapa, seção 3.8, apresenta-se como os modelos foram executados e testados, ou seja, como se deu a experimentação dos modelos criados.

A quinta etapa, contida na seção 3.9 contém a forma como foi feita a avaliação de desempenho, através das métricas definidas, e como se dá a apresentação dos resultados obtidos, para facilitar a organização do trabalho.

3.1 Trabalhos Correlatos

Para que o trabalho pudesse ser iniciado, foi realizada uma pesquisa extensa na literatura por trabalhos que pudessem servir de referência, principalmente por aqueles que utilizassem redes neurais na busca por tendências em séries temporais de mercados financeiros.

Nesse sentido, o trabalho de Kohzadi et al. (1996) compara a utilização de uma rede neural *feedforward* com a técnica ARIMA para previsão de preços de gado e trigo. A rede neural mostrou desempenho melhor em todos os casos, sendo considerada muito superior ao modelo ARIMA devido a capacidade de captar o comportamento não linear das séries.

O estudo feito por Alon et al. (2001) analisa os resultados obtidos por rede neural na previsão de vendas no varejo com três modelos estatísticos de referência (ARIMA, regressão múltipla e suavização exponencial). Foram realizadas previsões de um passo e múltiplos passos a frente em duas séries, uma menos volátil que a outra. No geral as redes neurais obtiveram menor erro que os outros modelos, principalmente na série mais volátil, onde a não linearidade era maior, apresentando taxa de erro de até 1,32% menor na previsão um passo à frente e até 0,99% na previsão de múltiplos passos à frente. Na série mais estável os modelos estatísticos demonstraram viabilidade, apresentando menor taxa de erro que a rede neural.

Wood e Dasgupta (1996) fazem uso de três técnicas (redes neurais, regressão e ARIMA) para prever o *Morgan Stanley Capital Index* (MSCI) um passo à frente. Os autores preferiram utilizar as mudanças percentuais das médias móveis do índice ao invés de utilizar a série original, dessa forma fizeram uso de outras moedas para tentar prever as mudanças percentuais da média móvel do dólar no MSCI no mercado americano. Verificou-se que as redes neurais obtiveram resultados melhores que decisões aleatórias, obtendo uma taxa de sucesso de 65% na previsão de tendência do MSCI, contra 56% à 61% nos outros modelos avaliados.

A tese de doutorado de Oliveira (2007) faz uso de rede neural *feedforward*, RNA treinada com filtro de Kalman estendido, RNA treinada com filtro de Kalman

unscented e o modelo estatístico ARIMA-GARCH na previsão de um passo à frente de séries da bolsa de valores brasileira. A rede neural treinada com filtro de Kalman *unscented* obteve melhores resultados.

A investigação feita por Nelson et al. (1999) verifica se redes neurais obtêm melhores resultados em séries onde os comportamentos sazonais foram removidos (transformadas). Observa-se que tanto o método estatístico utilizado como referência quanto a rede neural obtiveram melhores resultados com as séries sem elementos sazonais, o que não acontece quando a rede neural utiliza séries com elementos sazonais. Os autores atribuem à esse resultado a necessidade das redes aprenderem apenas as tendências e componentes cíclicos quando as séries são transformadas, enquanto nas séries originais também se faz necessário aprender a sazonalidade.

Bahrammirzaee (2010) faz uma revisão de diversos trabalhos que abordam sistemas especialistas, redes neurais ou modelos híbridos no mercado financeiro. No geral, verifica-se que as redes neurais obtêm bons resultados principalmente em predições. Todavia o autor deixa claro que nem sempre as redes obtêm os melhores resultados e não se pode concluir que são modelos muito precisos, visto que a maioria dos trabalhos investigados utilizavam *backpropagation* como treinamento, que é mais propenso a encontrar um mínimo local, além disso, suas topologias e arquiteturas são selecionadas via tentativa e erro, estando sujeitas a inúmeros erros. É concluído que modelos híbridos possuem melhores resultados, pois visam combinar técnicas diferentes para superar as deficiências de cada uma.

Coelho et al. (2008) tentam prever as taxa de câmbio brasileira através de redes neurais *perceptron* multicamadas (MLP), redes neurais função de base radial (RBF), sistema nebuloso Takagi-Sugeno (TS), modelo ARMA e modelo ARMA-GARCH. Foram utilizadas séries de retorno da taxa real/dólar com frequências de 15 minutos, 60 minutos, 120 minutos, diária e semanal. As séries foram transformadas por diferenças de logaritmos. As redes neurais obtiveram resultados melhores que os outros modelos, especialmente a arquitetura MLP, sendo que os modelos ARMA e ARMA-GARCH foram piores. Verificou-se que quanto menor a frequência, melhor são as previsões das séries pelos modelos.

Costa (2012) realiza uma comparação entre duas arquiteturas de redes dinâmicas no mercado de ações: a FTDNN e a rede não linear auto-regressiva com entradas exógenas (NARX). Em cada uma das arquiteturas foram utilizadas quatro tipos de pré-processamento (transformação da série). Verificou-se que as redes tipo NARX obtiveram melhores resultados, mas com um custo computacional muito maior. Das quatro formas de pré-processamento, a melhor realizou transformação

por retornos de logarítmicos, seguida por diferenças de primeira ordem.

Verifica-se que são poucos os trabalhos que propõem a utilização de indicadores para reconhecimento de padrões no mercado ou ainda aqueles que utilizem alguma estratégia de negociação automatizada; a maioria opta apenas pela extrapolação das séries do mercado. Na pesquisa realizada não foi encontrado nenhum trabalho recente que propusesse e desenvolvesse novos indicadores a serem utilizados com redes neurais. Por fim, a maioria dos trabalhos que realizaram comparações entre redes neurais e métodos tradicionais econométricos concluíram que as redes proveem resultados melhores.

Já que nenhum não encontrou-se nenhum trabalho que fosse similar ao desenvolvido nessa pesquisa, se faz necessário então contextualizar e caracterizar o problema abordado.

3.2 Contextualização do Problema

É crucial para um robô negociador saber quando deve **Não Operar** ou abrir uma ordem (**Compra** ou **Venda**) e assim aproveitar uma tendência do mercado. Para que se possa decidir o que o robô deve fazer, adotou-se, dentre as diversas técnicas da inteligência artificial, o uso de redes neurais para identificação e previsão de tendências do mercado, baseando-se nos fatos de que são relativamente fáceis de entendê-las e aplicá-las, com uso de poder computacional significativo apenas na fase de treinamento, sendo executada rapidamente, além de terem sua eficácia comprovada no mercado financeiro.

Optou-se, como entrada da rede, o uso de indicadores para que a rede possa encontrar padrões e tendências, já que o uso da série *tick-a-tick* não provê bons resultados na previsão de múltiplos passos à frente. Preferiu-se pela criação de indicadores ao invés de utilizar indicadores existentes disponíveis, pois pressupõe-se em duas hipóteses:

1. Mercado Eficiente: Como diversos investidores possuem acesso aos indicadores existentes, utilizando-os para criar estratégias, acredita-se que o mercado é, de certa forma, imune à essas estratégias.
2. Série Discreta é Ineficiente para Previsões de Curto Prazo: Observou-se que a maioria dos indicadores utiliza séries discretas, seja em minutos, horas ou dias, para o seu cálculo. Acredita-se que esses indicadores não possam inferir sobre períodos de curto prazo, sendo ineficientes para a estratégia adotada.

Dessa forma, a previsão da rede neural não é diretamente sobre o mercado, mas sobre as ações de um robô negociador que atuaria no mercado.

Logo, esse trabalho tem como objetivo principal criar e avaliar indicadores que utilizassem dados *tick-a-tick* do mercado financeiro *FOREX* para uma estratégia de operação a curto prazo. Os objetivos específicos são listados abaixo:

1. Criação de uma análise técnica própria
2. Criação de um modelo preditivo para avaliar os indicadores, necessitando-se:
 - 2.1. Criar um robô negociador que atenda a estratégia de curto prazo utilizada
 - 2.2. Explorar o uso de Redes Neurais Artificiais para previsão do mercado através dos indicadores criados, provendo uma decisão automatizada para o robô
3. Avaliar se o modelo preditivo desenvolvido é satisfatório

Como objetivos secundários deseja-se verificar e comprovar a utilização de redes neurais no mercado financeiro e a aquisição do *know-how* para criação de agentes inteligentes que operem em mercados financeiros.

3.3 Caracterização do Problema

Para o estudo realizado foram utilizadas duas séries distintas, contendo o *bid* e o *ask*. A primeira série contém os *ticks* de 07/09/2014 até 12/09/2014 e a segunda de 14/09/2014 até 19/09/2014. Para facilitar o entendimento, a primeira série será chamada de Semana 1 e a segunda de Semana 2. O par de moeda utilizado foi o EUR/USD. Os dados foram obtidos em uma base de dados de testes de uma corretora real.

Para criar e avaliar os modelos, algumas restrições foram consideradas:

- O mercado é simulado através de um programa desenvolvido contendo o robô negociador e o cálculo dos indicadores.
- Já que a estratégia adotada para operação é do tipo *Scalper*, as ordens não devem ficar expostas por muito tempo, então um limite de tempo máximo foi imposto de exposição da ordem, após isso a ordem é fechada.
- As ordens devem ser fechadas antes do final de semana, já que o mercado não está aberto nesse período e é difícil inferir sobre seu retorno.

- Não se deseja extrapolar a série. A RNA deve dizer ao robô qual operação deve iniciar (Venda, Compra, ou Não Operar), mas é o robô que fechará a ordem criada, conforme seus parâmetros.

As RNAs foram desenvolvidas e treinadas utilizando o software MATLAB[®] versão R2015b (8.6.0.267246). A simulação do mercado, o robô e os indicadores serão criados através de um programa desenvolvido em linguagem C e compilado utilizando Apple LLVM versão 6.0 (clang-600.0.57). O trabalho foi realizado em um MacBook Pro Mid 2012, no sistema operacional OS X 10.9 (Mavericks), com 8gb de memória RAM e processador i7 3520M 2,9Ghz.

3.4 O Modelo Preditivo Proposto

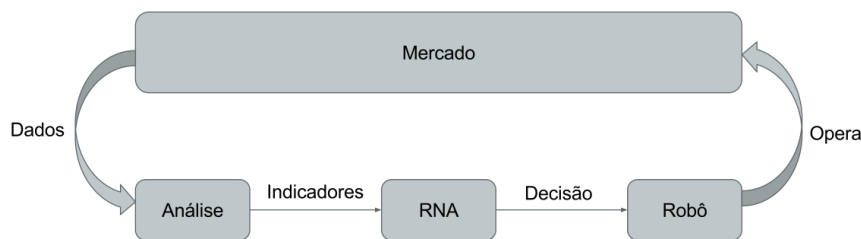


Figura 9 – O modelo preditivo

A Figura 9 apresenta o modelo preditivo proposto, onde os dados são gerados no mercado, processados pela análise técnica, que resulta nos indicadores como entrada da rede neural, que por sua vez deve decidir qual ação o robô deve tomar, influenciando no mercado.

3.5 A Análise Técnica

Para criar a análise técnica, foram observadas, empiricamente, algumas características do mercado financeiro e assim tentar quantificar essas características. Verificou-se que:

1. A variação *tick-a-tick* do mercado é pequena ou nula, e às vezes oscila em torno de um valor, demorando para apresentar uma tendência.
2. Algumas poucas variações são realmente expressivas e apresentam oportunidade de ganho imediata.

Assim sendo, iniciou-se o desenvolvimento de uma análise técnica em que fosse possível extrair indicadores que fossem úteis para previsão do mercado em curto prazo através da série de diferenças *tick-a-tick*¹.

Nessa seção são apresentados a configuração atual da análise e dos indicadores extraídos. A análise atual permite quantificar não apenas variações expressivas ou inexpressivas, mas sequências de variações, em *pips*, através das diferenças de primeira ordem entre os *ticks*.

Para medir essas sequências de variações, foram criadas as SEQs. Uma SEQ é definida pela observação de variações consecutivas em um mesmo sentido (apenas positivas ou negativas). É iniciada quando uma variação diferente de 0 ocorre e é finalizada quando uma variação em outro sentido ocorre, assim reiniciando o processo para uma nova SEQ.

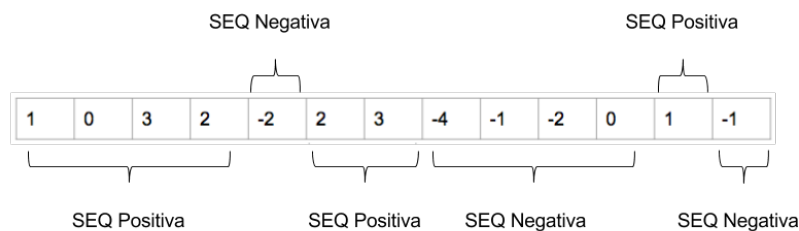


Figura 10 – Exemplo de funcionamento das SEQs, contabilizando três SEQs positivas e três SEQs negativas.

Para que se possa mensurar as SEQs expressivas, com possibilidades maiores de ganho, e as SEQs inexpressivas, foram criados dois tipos de SEQs:

- Se $|SEQ| > V$ então SEQ_{maior} , considerada expressiva
- Se $|SEQ| < V$ então SEQ_{menor} , considerada inexpressiva

Sendo $|SEQ|$ o valor absoluto da soma das variações da SEQ, e V um valor arbitrário a ser definido.

Se cada tipo de SEQ possui dois tipos de variação (positiva ou negativa), então foram criados quatro indicadores:

- Número de $SEQ_{s_{menor-}}$, com variações negativas

¹ O desenvolvimento da análise técnica está detalhado no Apêndice A

- Número de $SEQ_{s_{menor+}}$, com variações positivas
- Número de $SEQ_{s_{maior-}}$, com variações negativas
- Número de $SEQ_{s_{maior+}}$, com variações positivas

O sinal ao lado de *maior* ou *menor* indica se as variações são negativas ou positivas, para facilitar a notação no trabalho. Definiu-se que as SEQs negativas sempre são registradas com valores negativos e as SEQs positivas sempre valores positivos.

Os indicadores devem ser calculados dentro de uma janela de tamanho \mathbf{L} e são registrados, então a janela deve ser atualizada para refazer o processo. Identificou-se a possibilidade de atualizar a janela de duas maneiras:

1. Janela Sequencial: após calcular os indicadores de uma janela, ela é movida para incluir o próximo *tick* e remove-se os *ticks* mais antigos para atingir a restrição do tamanho da janela.
2. Janela em Saltos: a janela move-se um período de tempo à frente, recebendo *ticks* até que sua condição de tempo seja atingida, mantendo apenas o *tick* que era considerado o mais recente na janela anterior.

Ressalta-se que no modo sequencial a janela deve possuir um tamanho máximo \mathbf{L} , com exceção da primeira janela, que pode extrapolar esse tamanho, visto que não há como saber se o *tick* seguinte estará dentro do tempo definido, portanto decidiu-se que a primeira janela do modo sequencial teria seu tempo extrapolado, ou seja, assim que um *tick* recebido extrapolasse o tempo, ela seria "fechada" e os cálculos realizados. Na janela em saltos, ela possui um tempo mínimo, devido ao mesmo problema encontrado com a primeira janela do modo sequencial, então quando um *tick* recebido extrapola o tempo mínimo, a janela é fechada e os cálculos realizados.

A tabela apresenta os valores utilizados para o cálculo dos indicadores:

Tabela 5 – Valores a serem utilizados para o cálculo dos indicadores

V	L
7 + Spread Médio	300 segundos

O valor 7 foi escolhido pois verificou-se que a comissão a ser paga por ordem é estimada nesse valor. O *spread* médio citado é referente aos *ticks* da janela. Ele

é considerado no valor, pois o *spread* também influencia na perda de uma ordem e dificulta para fechá-la com ganho, principalmente se o valor for alto. O tamanho da janela foi escolhido de forma arbitrária, acreditando que 300 segundos seriam suficientes para uma previsão de curto prazo.

A série a ser utilizada para computar os indicadores é a média do *ask* com o *bid*, garantindo que a informação das duas séries esteja presente.

3.6 O Robô

O robô criado é caracterizado como um negociador automatizado, realizando operações no mercado em nome de um investidor. Ele é parametrizado e possui apenas critérios de fechamento de ordem, já que a abertura da ordem deve ser dada pela rede neural. O primeiro parâmetro é o *Take-Profit* (TP), que é o ganho mínimo em *pips* a ser atingido para fechar a ordem. O segundo é o *Stop-Loss* (SL), que é a perda máxima aceitável em *pips* em uma ordem. Enquanto o *Take-Profit* é sempre posicionado em relação ao valor utilizado para abrir a ordem, o *Stop-Loss* é posicionado em relação ao valor utilizado para fechar a ordem, considerando assim o *spread*. Por exemplo, ao abrir uma ordem de Compra considerando o *ask* igual à 1,30000, e o *bid* igual à 1,29997, sendo o *Take-Profit* igual à 14 e o *Stop-Loss* igual à 7, a ordem seria fechada por *Take-Profit* caso o *bid* alcançasse o valor de 1,30014 ou fechada por *Stop-Loss* caso alcançasse o valor de 1,29990.

Ele também possui uma trava de tempo para exposição de uma ordem. Se o tempo de exposição de uma ordem ultrapassar 5 minutos, a ordem é fechada, independente do ganho ou da perda. Ressalta-se que uma ordem só é considerada bem sucedida se for fechada através do *Take-Profit*. Não são registradas ordens que não fecham por *Take-Profit*, *Stop-Loss*, ou pela trava de exposição, pois não há como inferir sobre os resultados delas.

O negociador também pode ser executado em modo *dummie*, ou seja, em um determinado *tick* são abertas duas ordens, uma de **Compra** e outra de **Venda**, então é registrado qual ordem houve sucesso, ou se não houve sucesso em nenhuma das ordens (**Não Operar**). O modo *dummie* é utilizado após o cálculo dos indicadores, no último *tick* da janela, registrando o resultado do robô com os valores dos indicadores daquela janela.

Foi decidido variar os parâmetros do robô, para que se possa identificar valores que sejam melhores para o modelo, neste caso decidiu-se testar com três conjuntos de parâmetros diferentes:

Tabela 6 – Conjuntos de parâmetros utilizados no robô

TP	SL
14	7
10	7
10	20

Os valores para **TP** permitem um ganho pequeno, garantindo o lucro nessas ordens. Nos dois primeiros conjuntos, os valores para **SL** foram escolhidos para que a perda não fosse tão maior quanto a comissão adicionada ao *spread*. Já no último conjunto, fora escolhido para que as ordens pudessem ter uma “elasticidade” maior, permitindo mais tempo para serem fechadas.

3.7 Redes Neurais Utilizadas

Utilizam-se duas arquiteturas de rede nesse trabalho: A *feedforward* classificadora simples e a rede focada atrasada no tempo. Tentou-se manter as duas arquiteturas parecidas para fins de comparações. O algoritmo de treinamento utilizado foi Levenberg-Marquardt; a função de ativação na camada escondida é a Tangente Hiperbólica (Equação 2.6); e a função custo é a MSE (Equação 2.11). Para a função de ativação na camada de saída, a rede FTDNN utiliza a função linear (Equação 2.8) e a rede classificadora a função *softmax* (Equação 2.7).

O número de neurônios (e de atrasos, no caso da FTDNN) foram variados para que se pudesse utilizar as melhores topologias.

3.7.1 Pré-processamento dos dados

Foi realizado um pré-processamento dos dados utilizados para que fossem condizentes com as arquiteturas utilizadas.

Para as entradas, nas duas redes, decidiu-se por normalizar a série de cada indicador entre 0 e 1, ou -1 e 0, dependendo do sinal da série. Os dados de entrada foram normalizados dentro da imagem da função de ativação para obter melhores resultados e acelerar o processo de treinamento das redes (SOLA; SEVILLA, 1997).

- Séries negativas \rightarrow -1 e 0
- Séries positivas \rightarrow 0 e 1

Já os dados registrados pelo robô negociador precisam de um pré-processamento diferenciado para cada arquitetura.

Uma saída da rede classificadora, por causa da função *softmax* (Equação 2.7), será sempre um vetor com o número de posições igual o número de classes, onde cada posição contém uma probabilidade das entradas processadas pertencerem à classe representada pela aquela posição no vetor. A soma dos valores do vetor será sempre igual a 1. Logo, os registros do robô, para a rede classificadora, serão transformados em:

- Venda $\rightarrow (0; 0; 1)$
- Não Operar $\rightarrow (0; 1; 0)$
- Compra $\rightarrow (1; 0; 0)$

Por exemplo, sendo $(0,3; 0,5; 0,2)$ uma saída da rede, 30% é a probabilidade das entradas pertencerem à classe **Compra**, 50% pertencerem à classe **Não Operar**, e 20% pertencerem à classe **Venda**.

Já uma saída da rede FTDNN é a previsão de um valor, que gira em torno dos valores apresentados no treinamento da rede. Portanto decidiu-se por transformar os resultados do robô da seguinte forma para essa arquitetura:

- Venda $\rightarrow -1$
- Não Operar $\rightarrow 0$
- Compra $\rightarrow 1$

3.7.2 Transformação das saídas das redes

Como visto na subseção 3.7.1, as saídas das redes não são valores que definem diretamente se são **Compra**, **Venda** ou **Não Operar**, portanto devem ser transformadas para que possam ser utilizadas.

Para a rede classificadora isso é mais fácil, basta considerar a posição do vetor de saída que possui maior probabilidade. No exemplo $(0,3; 0,5; 0,2)$, seria transformado em $(0; 1; 0)$, que é **Não Operar**.

Na rede FTDNN o processo é mais complexo, visto que os valores previstos não são exatos e podem ser maiores que 1 ou menores que -1. Contudo, espera-se que as **Vendas** sejam previstas com valores mais próximos à -1, as **Compras** com valores

mais próximos à 1, e Não Operar próximo à 0. Nesse caso optou-se utilizar limiares para transformar as saídas. Serão testados três conjuntos de limiares, apresentados na tabela abaixo.

Tabela 7 – Os dois conjuntos de limiares

Conjuntos	Limiar Inferior	Limiar Superior
A	-0,15	0,15
B	-0,3	0,3
C	-0,5	0,5

- Saída \leq Limiar Inferior \rightarrow Venda
- Saída \geq Limiar Superior \rightarrow Compra
- Limiar Inferior $<$ Saída $<$ Limiar Superior \rightarrow Não Operar

3.7.3 Estratégia de treinamento

Os *datasets* utilizados para treinamento serão pré processados e então divididos em três partes: 70% para treinamento, 15% para validação e 15% para teste. Esses valores foram escolhidos pois o conjunto de treinamento não pode ser nem muito grande para que não ocorra superadaptação e nem muito pequeno para que o desempenho não seja ruim. O conjunto de validação também é considerado como dados vistos pela rede e serve para verificar a parada do treinamento por parada antecipada (*early-stopping*). Para a rede classificadora os conjuntos foram montados aleatoriamente, já na rede FTDNN os dados são contínuos devido ao atraso inserido na rede.

Após a divisão dos conjuntos, a rede é inicializada com pesos aleatórios e inicia-se o processo de treinamento. A cada época de treinamento, o conjunto de treinamento é executado pela rede, onde os pesos são atualizados conforme os erros. Assim que os pesos são atualizados, o conjunto de validação é executado na rede e registra-se o erro (MSE) da rede nesse conjunto. Esse processo se repete até que uma das condições sejam atingidas:

- Limite máximo de 50 épocas.
- Parada antecipada quando o erro subir seis vezes seguidas no conjunto de validação. Essa verificação é feita após cada época.
- MSE inferior à 10^{-5} .

- Taxa de aprendizagem menor que 10^{-3} ou maior que 10^{10} .

Após atingir alguma dessas condições o treinamento é parado e executa-se 100% do *dataset* na rede para avaliar seu desempenho geral em cima das saídas originais da rede, sem transformação. Como os pesos iniciais das redes são valores aleatórios, é interessante treinar a mesma rede diversas vezes e escolher a que apresentar melhor desempenho. O conjunto de teste é importante nessa etapa pois introduz dados não vistos e permite a escolha de uma rede que não tenha sofrido *overfitting*. Nesse trabalho cada rede foi treinada 10 vezes e escolhida a melhor.

3.8 Estratégia de Teste

A estratégia adotada foi pensada com o objetivo de verificar se os indicadores criados são viáveis, testando diferentes ajustes e parâmetros.

Primeiramente o robô e a análise técnica foram executados em cada série do mercado separadamente para gerar registros em arquivos, que contém os indicadores relacionados com os resultados do robô. Como foram testados três parâmetros para o robô para cada um dos dois tipos de janelas, então foram gerados ao total doze *datasets*. Desses, os referentes à Semana 1 serão utilizados para o treino das redes e os da Semana 2 serão utilizados para testar as redes após o treino. Cada *dataset* de treino será utilizado para treinar duas redes: uma Classificadora e outra FTDNN, e para cada um deles existe um *dataset* equivalente para teste.

Com dois tipos de janelas, três parâmetros do robô e duas arquiteturas de redes, isso resulta em doze redes a serem utilizadas.

Antes de iniciar o treinamento, os dados da Semana 1 são pré-processados, conforme subseção 3.7.1, e as configurações P são armazenados para normalizar a Semana 2, que servirá de teste.

Para a escolha da topologia de cada rede, no treinamento foram variados os números de neurônios na camada de saída e os atrasos, conforme a Tabela 8. Então cada arquitetura é treinada com diferentes topologias conforme a subseção 3.7.3 e é selecionada a topologia que apresentar o menor MSE ao ser executada com os dados de treinamento, sem aplicar a transformação nas saídas da rede. Foram testadas quantidades menores de neurônios na rede FTDNN, visto que ela exige maior poder de processamento.

Tabela 8 – Topologias testadas para cada arquitetura

Tipo de Rede	Neurônios	Atrasos
FTDNN	6 até 10	3 até 5
Classificadora	11 até 15	N/A

Após o treinamento, os *datasets* referentes à Semana 2 também foram pré-processados, mas a normalização se deu através da utilização das configurações P obtidas com as normalizações realizadas com os *datasets* referentes à Semana 1, já que deve-se considerar a Semana 2 como dados não vistos e que são recebidos aos poucos do mercado.

Então cada rede é executada com seu respectivo *dataset* da Semana 2. As transformações conforme a subseção 3.7.2 são realizadas e os resultados apresentados para comparação.

3.9 Avaliação de Desempenho e Apresentação dos Resultados

Os resultados obtidos com as redes foram comparados com dois modelos de referência, um Aleatório e outro Aleatório Informado. O modelo Aleatório apenas escolhe randomicamente as decisões de **Compra**, **Venda** e **Não Operar**. O Aleatório Informado escolhe as decisões com base nas distribuições apresentadas nos *datasets* de treinamento. Ao todo, serão doze resultados, um para cada modelo aleatório, com cada tipo de janela, em cada parâmetro do robô. Cada um dos modelos são executados dez vezes e escolhe-se o que apresentar o maior *SCORE*.

O *SCORE* também será utilizado para comparar todos os resultados obtidos. Para obter o *SCORE*, calcula-se primeiro os *SCORES* parciais de **Compra** (Equação 3.1), **Venda** (Equação 3.2) e **Não Operar** (Equação 3.3), e então calcula-se a média das parciais (Equação 3.4).

$$SCORE_{Compra} = \frac{N^{\circ} \text{ Compras Previstas Corretas}}{N^{\circ} \text{ Compras no Dataset}} \quad (3.1)$$

$$SCORE_{Venda} = \frac{N^{\circ} \text{ Vendas Previstas Corretas}}{N^{\circ} \text{ Vendas no Dataset}} \quad (3.2)$$

$$SCORE_{N\grave{a}oOperar} = \frac{N^{\circ} \text{ N\grave{a}o Operar Previstas Corretas}}{N^{\circ} \text{ N\grave{a}o Operar no Dataset}} \quad (3.3)$$

$$SCORE = \frac{SCORE_{Compra} + SCORE_{Venda} + SCORE_{NãoOperar}}{3} \quad (3.4)$$

Quanto mais alto o valor do *SCORE*, melhor. As redes neurais, que utilizam os indicadores, devem possuir um *SCORE* com pelo menos quatro pontos percentuais a mais que o melhor modelo aleatório.

Devido à quantidade de resultados apresentados, eles serão separados em seções: Os resultados com a janela sequencial são apresentados e discutidos seção 4.1; A seção 4.2 contém os resultados com a janela em saltos; e A seção 4.3 apresenta uma comparação entre os resultados obtidos com os dois tipos de janelas.

4 ANÁLISE DOS RESULTADOS

4.1 Janela no Modo Sequencial

4.1.1 Treinamento e escolha das redes

As seis redes referentes aos indicadores com janela em modo sequencial foram treinadas utilizando as amostras da Semana 1. A Tabela 9 apresenta a distribuição de *Compra*, *Venda* e *Não Operar* na Semana 1. O robô com parâmetros em $(TP, SL) = (10, 7)$ apresenta uma distribuição mais igualitária entre as decisões. Em cada conjunto de parâmetros a distribuição é parecida entre *Compra* e *Venda*, apresentando diferenças de no máximo 0,46%.

Parâm. robô (TP, SL)	Amostra			Total
	Compra (%)	Venda (%)	Não Operar (%)	
(14, 7)	127029 (28,90%)	127759 (29,06%)	184793 (42,04%)	439581
(10, 7)	153706 (34,97%)	155817 (35,45%)	130067 (29,59%)	439590
(10, 20)	194796 (44,33%)	196801 (44,78%)	47869 (10,89%)	439466

Tabela 9 – Registros feitos pelo robô na Semana 1 com janela em modo sequencial, utilizados no treinamento das redes e do modelo Aleatório Informado.

As topologias escolhidas para cada rede são apresentadas na Tabela 10. Como as saídas originais das duas arquiteturas de rede são apresentadas de formas diferentes, não há uma compatibilidade com o erro quadrático médio entre elas, portanto não se pode fazer comparações diretas através do MSE. Verifica-se que os menores erros quadráticos são das redes que fizeram uso do robô com parâmetros $(TP, SL) = (14, 7)$

Parâm. robô (TP, SL)	Classificadora		FTDNN		
	MSE	Neurônios	MSE	Neurônios	Atrasos
(14, 7)	0,1876	14	0,5744	9	3
(10, 7)	0,2173	15	0,7	7	5
(10, 20)	0,2137	13	0,8858	7	5

Tabela 10 – Topologias escolhidas para cada tipo de rede para o indicadores sequenciais.

4.1.2 Resultados com a Semana 2

A distribuição de Compra, Venda e Não Operar com o robô executado em cada conjunto de parâmetros a Semana 2 é mostrada na Tabela 11.

Parâm. robô (TP, SL)	Amostra				Total
	Compra (%)	Venda (%)	Não Operar (%)		
(14, 7)	125569 (28,62%)	126769 (28,89%)	186465 (42,49%)		438803
(10, 7)	150742 (34,35%)	151241 (34,46%)	136842 (31,18%)		438825
(10, 20)	191609 (43,67%)	194857 (44,41%)	52332 (11,93%)		438798

Tabela 11 – Registros feitos pelo robô na Semana 2 com janela em modo sequencial, utilizados nos testes das redes e dos modelos aleatórios.

A Tabela 12 apresenta os resultados dos modelos por parâmetro do robô. A Figura 11 apresenta o gráfico com os valores apresentados na tabela.

Parâm. robô (TP, SL)	Modelos	Compras			Vendas			Não Operar			SCORE Total
		Prev.	Certas	SCORE Parc.	Prev.	Certas	SCORE Parc.	Prev.	Certas	SCORE Parc.	
(14, 7)	Class.	33731	10436	8,31%	36821	11365	8,97%	368251	160274	85,95%	34,41%
	FTDNN-A	16256	4822	3,84%	22298	6777	5,35%	400246	171750	92,11%	33,76%
	FTDNN-B	5597	1809	1,44%	10686	3159	2,49%	422517	179974	96,52%	33,48%
	FTDNN-C	1595	632	0,50%	4409	1282	1,01%	432796	183896	98,62%	33,38%
	Aleatório	146457	42130	33,55%	145763	42167	33,26%	146583	62609	33,58%	33,46%
	Aleatório Informado	126968	36353	28,95%	127548	37038	29,22%	184287	78778	42,25%	33,47%
(10, 7)	Class.	207761	75236	49,91%	155136	57913	38,29%	75928	33883	24,76%	37,65%
	FTDNN-A	8868	3318	2,20%	26858	9548	6,31%	403094	126898	92,73%	33,75%
	FTDNN-B	2329	912	0,61%	11091	3896	2,58%	425400	132717	96,99%	33,39%
	FTDNN-C	611	207	0,14%	5099	1764	1,17%	433110	135221	98,82%	33,37%
	Aleatório	146636	50253	33,34%	146339	50639	33,48%	145850	45799	33,47%	33,43%
	Aleatório Informado	153163	52718	34,97%	155793	53845	35,60%	129869	40535	29,62%	33,40%
(10, 20)	Class.	210752	94009	49,06%	201173	90972	46,69%	26873	10056	19,22%	38,32%
	FTDNN-A	16563	8246	4,30%	34143	15821	8,12%	388087	49727	95,02%	35,82%
	FTDNN-B	6186	3186	1,66%	19246	8776	4,50%	413361	51276	97,98%	34,72%
	FTDNN-C	1431	993	0,52%	10159	4715	2,42%	427203	52036	99,43%	34,12%
	Aleatório	146263	64265	33,54%	146274	65072	33,39%	146261	17540	33,52%	33,48%
	Aleatório Informado	194617	85213	44,47%	196527	87742	45,03%	47654	5699	10,89%	33,46%

Tabela 12 – Comparativo dos modelos com janela em modo sequencial. FTDNN-A utiliza o conjunto de limiares **A** e assim por diante.

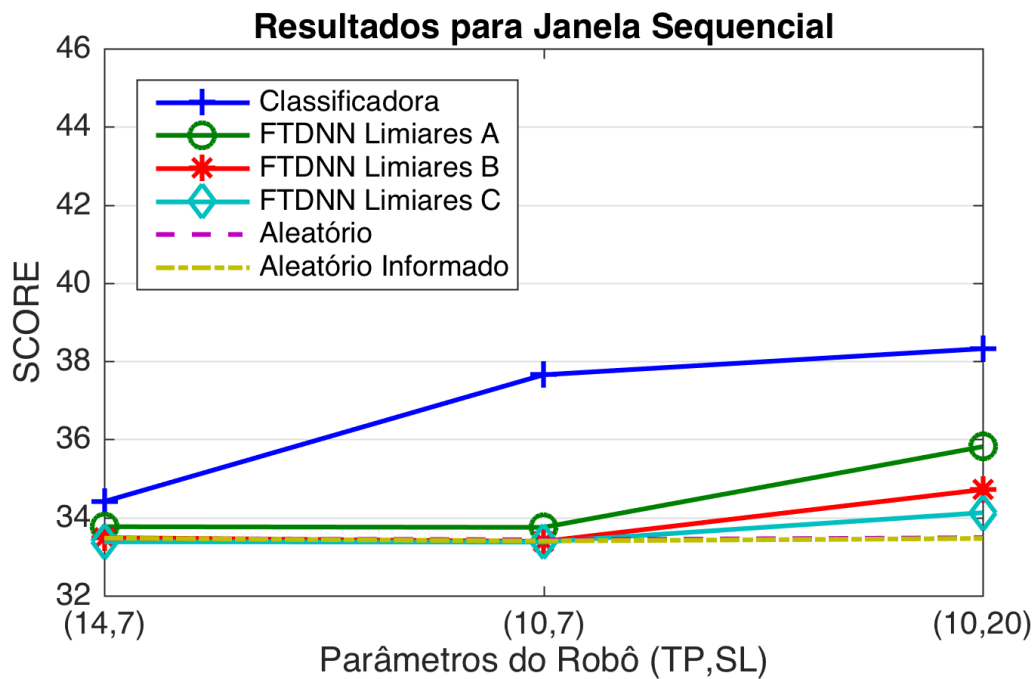


Figura 11 – Resultados das redes neurais e dos modelos aleatórios por parâmetro do robô.

Observa-se que as redes Classificadoras obtiveram melhores resultados que os outros modelos, e as redes com os parâmetros do robô eram (10, 7) e (10, 20) ficaram acima dos 4 pontos percentuais, sendo a rede que faz uso do último parâmetro a melhor nesse caso.

A melhor rede FTDNN fez uso do conjunto **A** de limiares com os parâmetros do robô em (10, 20), porém não ficou acima da meta desejada. O conjunto de limiares **A** apresentou melhores resultados do que os outros dois conjuntos, sendo que os três conjuntos apresentaram resultados melhores quando os parâmetros do robô eram (10, 20). Contudo, as redes com atraso não apresentaram um desempenho expressivo se comparadas com o modelos aleatórios, portanto não se equiparam aos resultados obtidos pelas redes Classificadoras. Os dois modelos aleatórios se equiparam e se mantiveram relativamente constantes nos diferentes parâmetros.

4.2 Janela no Modo em Saltos

4.2.1 Treinamento e escolha das redes

As seis redes referentes aos indicadores em saltos foram treinadas utilizando as amostras da Semana 1 (Tabela 13), com a janela no modo em saltos. Apesar de da distribuição mais igualitária também ser com o robô em parâmetros $(TP, SL) =$

(10, 7), a distribuição entre **Compra** e **Venda** apresenta um desvio padrão maior nos três conjuntos de parâmetros, ficando entre 1,68% e 2,77% contra 0,12% à 0,34% no modelo sequencial, sendo menos uniforme nesse aspecto.

Parâm. robô	Amostra				
	(TP, SL)	Compra (%)	Venda (%)	Não Operar (%)	Total
(14, 7)		371 (25,96%)	405 (28,34%)	653 (45,70%)	1429
(10, 7)		450 (31,49%)	506 (35,41%)	473 (33,10%)	1429
(10, 20)		570 (39,89%)	620 (43,39%)	239 (16,72%)	1429

Tabela 13 – Registros feitos pelo robô na Semana 1 com janela no modo em saltos, utilizados no treinamento das redes e do modelo Aleatório Informado.

As topologias escolhidas para cada rede são apresentadas na Tabela 14. Observa-se os menores erros quadráticos nas redes classificadoras são com os parâmetros do robô em (10, 20) e das redes FTDNN em (14, 7). Da mesma forma que na janela sequencial, não se pode comparar os MSE entre as duas arquiteturas de rede.

Parâm. robô	Classificadora		FTDNN			
	(TP, SL)	MSE	Neurônios	MSE	Neurônios	Atrasos
(14, 7)		0,2074	14	0,5268	10	5
(10, 7)		0,2148	12	0,632	9	4
(10, 20)		0,1962	12	0,7874	9	5

Tabela 14 – Topologias escolhidas para cada tipo de rede para janela no modo em saltos.

4.2.2 Resultados com a Semana 2

A Tabela 15 mostra as distribuições de **Compra**, **Venda** e **Não Operar** na Semana 2 com janela no modo em saltos.

Parâm. robô	Amostra				
	(TP, SL)	Compra (%)	Venda (%)	Não Operar (%)	Total
(14, 7)		365 (25,56%)	404 (28,29%)	659 (46,15%)	1428
(10, 7)		447 (31,30%)	476 (33,33%)	505 (35,36%)	1428
(10, 20)		578 (40,48%)	593 (41,53%)	257 (18,00%)	1428

Tabela 15 – Registros feitos pelo robô na Semana 2 com janela no modo em saltos, utilizados no teste das redes e dos modelos aleatórios.

O resultados das redes e dos modelos aleatórios são apresentados na Tabela 16 e graficamente na Figura 12

Parâm. robô (TP,SL)	Modelos	Compras			Vendas			Não Operar			SCORE
		Prev.	Certas	SCORE Parc.	Prev.	Certas	SCORE Parc.	Prev.	Certas	SCORE Parc.	Total
(14,7)	Class.	125	35	9,59%	108	40	9,90%	1195	575	87,25%	35,58%
	FTDNN-A	231	64	17,53%	423	124	30,69%	769	363	55,08%	34,43%
	FTDNN-B	73	19	5,21%	186	48	11,88%	1164	545	82,70%	33,26%
	FTDNN-C	19	6	1,64%	69	24	5,94%	1335	626	94,99%	34,19%
	Aleatório	520	141	38,63%	464	139	34,41%	444	221	33,54%	35,52%
	Aleatório Informado	372	105	28,77%	399	123	30,45%	657	306	46,43%	35,21%
(10,7)	Class.	321	105	23,49%	518	182	38,24%	589	252	49,90%	37,21%
	FTDNN-A	260	87	19,46%	451	162	34,03%	713	259	51,29%	34,93%
	FTDNN-B	125	44	9,84%	184	73	15,34%	1115	400	79,21%	34,80%
	FTDNN-C	46	20	4,47%	76	29	6,09%	1302	467	92,48%	34,35%
	Aleatório	490	169	37,81%	467	160	33,61%	471	173	34,26%	35,23%
	Aleatório Informado	411	152	34,00%	506	160	33,61%	511	195	38,61%	35,41%
(10,20)	Class.	543	229	39,62%	612	281	47,39%	273	107	41,63%	42,88%
	FTDNN-A	389	160	27,68%	384	166	27,99%	650	119	46,30%	33,99%
	FTDNN-B	210	91	15,74%	156	69	11,64%	1057	193	75,10%	34,16%
	FTDNN-C	94	47	8,13%	69	30	5,06%	1260	232	90,27%	34,49%
	Aleatório	456	190	32,87%	507	223	37,61%	465	90	35,02%	35,17%
	Aleatório Informado	562	238	41,18%	624	282	47,55%	242	54	21,01%	36,58%

Tabela 16 – Comparativo dos modelos com indicadores em saltos.

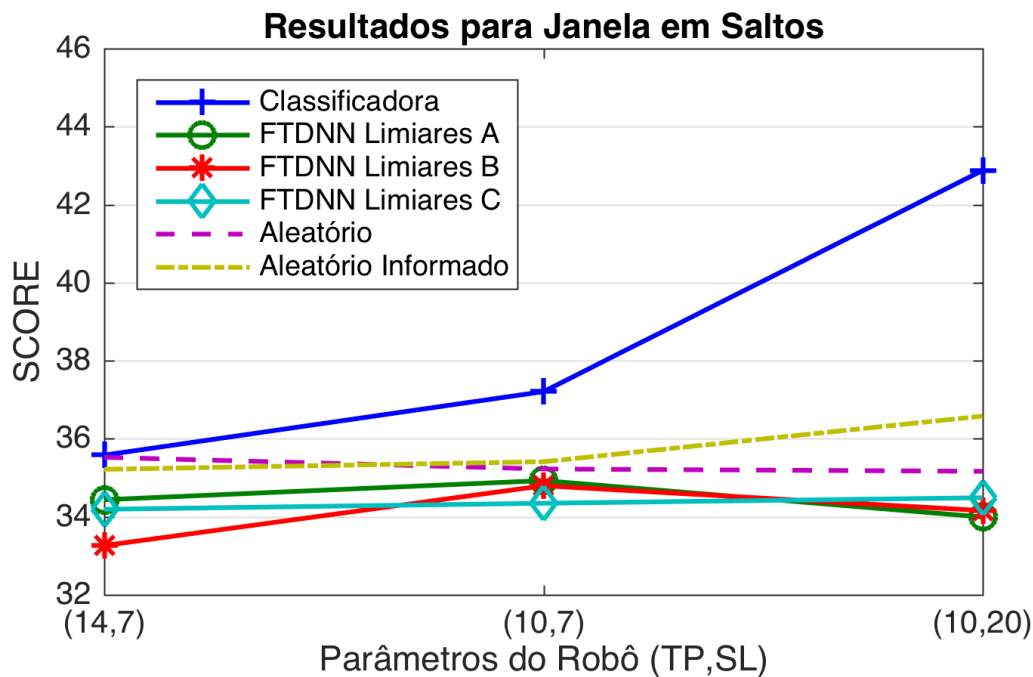


Figura 12 – Resultados das redes neurais e dos modelos aleatórios por parâmetro do robô.

As redes FTDNN tiveram resultados piores que os modelos aleatórios de referência, para quaisquer parâmetros do robô. A rede Classificadora com parâmetros em (14,7) foi a que obteve pior resultado nessa arquitetura, aproximando-se dos

modelos de referência. O melhor resultado foi com os parâmetros em (10, 20), com uma diferença de 6,3 pontos percentuais do melhor modelo aleatório, ultrapassando a meta desejada. A rede Classificadora com robô em parâmetros (10, 7) apresentou um *SCORE* superior aos modelos referência, mas não atingiu a meta desejada. Os modelos referência possuem resultados próximos e constantes nos parâmetros (14, 7) e (10, 7), já no parâmetro (10, 20) o modelo Aleatório Informado possui um *SCORE* mais alto que o modelo Aleatório puro, que continuou constante nesse parâmetro.

4.3 Comparativo

Há uma consistência nos resultados no que se diz respeito ao desempenho das redes Classificadoras nos dois tipos de janela. A rede do robô com parâmetros em (10, 20) foi melhor que com parâmetros (10, 7), que por sua vez foi melhor do que com parâmetros em (14, 7), sendo os resultados desse último sempre próximos aos modelos referência. Observa-se ainda os resultados inexpressivos das redes com atraso, que esperava-se um melhor desempenho.

Verifica-se que os modelos preditivos com janela em saltos possuem um desempenho superior àqueles com janela sequencial, isso vale até mesmo para os modelos aleatórios, que, salvo as proporções e grandezas vistas em cada *dataset*, obtiveram *SCORES* mais altos quando utilizaram *datasets* em que a janela estava nesse modo.

5 CONCLUSÕES

Esse trabalho teve como objetivo principal criar e avaliar indicadores que utilizassem dados do mercado financeiro *FOREX* para uma estratégia de operação a curto prazo. A avaliação se deu na forma de criação e comparação de resultados de modelos preditivos que utilizassem os indicadores criados em conjunto com redes neurais classificadoras ou redes neurais focadas atrasadas no tempo, tentando-se também, como objetivo secundário, comprovar a eficiência de redes neurais na predição e detecção de padrões no mercado financeiro. Para verificar se os modelos realmente reconheciam padrões, eles foram comparados com dois modelos aleatórios de referência, criados para simular a operação de duas pessoas no mercado: uma informada sobre as distribuições de momentos ótimos para **Venda**, **Compra** e **Não Operar** passados e outra não informada.

Constatou-se que as redes Classificadoras possuem desempenho melhor que os modelos aleatórios e as redes FTDNN, particularmente com o robô em parâmetros (10, 20). Supõe-se que a estratégia adotada no robô pode ser refinada para melhorar no desempenho geral do modelo. No caso das redes FTDNN, elas apresentaram resultados inexpressivos, equiparando-se ou sendo piores aos modelos aleatórios.

Os resultados evidenciam que os indicadores criados podem ser utilizados para encontrar padrões no mercado, se utilizados em conjunto com um modelo preditivo adequado. Isso é particularmente importante, pois o contexto do mercado é dinâmico, ou seja, se os indicadores possuíam um significado no passado, pode ser que esse significado (ou contexto) seja diferente atualmente. Constatou-se ainda que os indicadores proveem melhores resultados se utilizada a janela em saltos, condizente com o trabalho de Coelho et al. (2008), o que gera indícios de que agrupar ou discretizar dados para obter informações pode ser útil em estratégias de curto prazo.

Também acredita-se que os modelos aleatórios, principalmente o informado, podem ser aperfeiçoados com a utilização de aprendizagem de máquina.

Por fim, os indicadores, o robô e os modelos preditivos com melhores resultados devem ser estudados mais a fundo, para que se possa realizar ajustes e melhorar o desempenho geral, visto que atualmente não representam uma chance de ganho real, mesmo se desconsiderar as comissões impostas pelas corretoras. Alguns experimentos realizados apontam que utilizar apenas a série de *ask* ou de *bid* pode ser melhor que a

interpolação das duas séries, como fora feito nesse trabalho. Presume-se ainda, através dos melhores resultados obtidos com *Stop-Loss* mais alto, que seria mais interessante o robô apenas verificar e registrar se uma ordem é fechada por *Take-Profit* em até cinco minutos, evitando fechá-la prematuramente por *Stop-Loss*.

5.1 Trabalhos Futuros

Sugere-se como trabalhos futuros realizar melhorias nos indicadores propostos, ajustando o tamanho da janela e tentar comprovar se o uso de uma única série (*ask* ou *bid*) é melhor que a interpolação feita. Também deseja-se estudar outras estratégias de negociação, que sejam mais refinadas quanto as realizadas pelo robô utilizado.

Seria interessante, também, investigar sobre outros indicadores que possam ser utilizados em conjunto os criados nesse trabalho, isso ajudaria a mitigar o problema com a mudança de contexto do mercado e provavelmente proveria melhores resultados. Para tanto é desejável um sistema multiagente, em que cada agente faça uso de diferentes indicadores e proponha uma decisão, com a decisão final através de um consenso, como proposto por Korczak et al. (2013). Os agentes também devem ser capazes de retrainar seus modelos, avaliar as mudanças necessárias e acompanhar o dinamismo do mercado.

É possível, ainda, investigar outras áreas em que seja possível aplicar os indicadores criados. Por exemplo, uma provável aplicação seria na manutenção de máquinas na indústria, onde pode-se utilizar o som emitido por elas como séries temporais para verificar problemas. Contudo, os problemas não estão restritos apenas na indústria, por isso se faz necessário um estudo para encontrar aplicações nas diversas áreas do conhecimento.

5.1.1 Agente inteligente proposto

A pesquisa desenvolvida até o momento demonstra resultados interessantes, contudo ainda há a necessidade de melhoria dos modelos desenvolvidos e investigação de novos métodos e possibilidades.

Pensando no desenvolvimento de um sistema multiagente, propõe-se então um agente inteligente com redes neurais que pudesse ser utilizado em trabalhos futuros.

O agente será composto de três módulos: percepção, que receberá os dados da análise técnica toda vez que o agente necessitar; decisão, que será composto pela

rede neural e decidirá sobre a atuação no mercado naquele momento; e execução (atuadores), que ativará o negociador automatizado proprietário. A Figura 13 apresenta a organização do agente e também seu fluxo de execução.

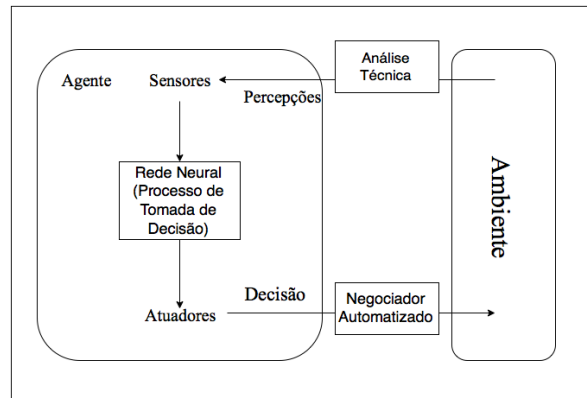


Figura 13 – Agente proposto (Adaptado de: Norvig e Russell (2014, p. 32))

O PEAS do agente é apresentado na Tabela 17 e o detalhamento do ambiente de tarefas é apresentado na Tabela 18.

Tabela 17 – Descrição do PEAS para o agente proposto.

Tipo de Agente	Negociador inteligente
Medida de Desempenho	No caso dos indicadores criados, o <i>SCORE</i> mínimo ainda deve ser estimado.
Ambiente	Séries temporais estocásticas, valores de ASK e BID , indicadores
Atuadores	Operar ou não operar
Sensores	<i>Input</i> dos valores dos indicadores

Tabela 18 – Propriedades do ambiente de tarefa para o agente inteligente proposto

Propriedade	Explicação
Observável	O agente tem acesso a todas as informações necessárias para tomar a decisão em um instante
Multiagente	No mercado há outros agentes competindo entre si
Estocástico	O ambiente não será determinado pela ação executada pelo agente
Episódico	A decisão de operar em um momento não é afetada por decisões passadas e não afetará as futuras.
Estático ou dinâmico	Dinâmico.
Discreto	Pois o agente decide apenas se realizará uma operação de compra, venda ou não fará nada.

Portanto conclui-se esse trabalho e ficam definidos os próximos passos a serem realizados.

REFERÊNCIAS

- ALON, I.; QI, M.; SADOWSKI, R. J. Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, Elsevier, v. 8, n. 3, p. 147–156, 2001. Citado 2 vezes nas páginas 22 e 42.
- BAHRAMMIRZAEI, A. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, Springer, v. 19, n. 8, p. 1165–1195, 2010. Citado 2 vezes nas páginas 22 e 43.
- BARBOSA, A. H.; FREITAS, M. S. da R.; NEVES, F. de Assis das. Confiabilidade estrutural utilizando o método de monte carlo e redes neurais. *Rem: Revista Escola de Minas*, scielo, v. 58, p. 247 – 255, 09 2005. ISSN 0370-4467. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0370-44672005000300011&nrm=iso>. Citado 2 vezes nas páginas 13 e 36.
- BEZERRA, M. I. S. Apostila de análise de séries temporais. *Curso de Estatística, available in <http://people.ufpr.br/lucambio/CE017/1S2010/5515941-Apostila-Series-Temporais.pdf>*, 2006. Citado na página 29.
- BOX, G. E.; JENKINS, G. M.; REINSEL, G. C. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2013. Citado 3 vezes nas páginas 28, 29 e 30.
- COELHO, L. dos S.; SANTOS, A. A. P.; JR, N. C. A. da C. Podemos prever a taxa de cambio brasileira? evidência empírica utilizando inteligência computacional e modelos econométricos. *Gestão & Produção*, SciELO Brasil, v. 15, p. 635–647, 2008. Citado 5 vezes nas páginas 21, 22, 30, 43 e 63.
- COMISSÃO DE VALORES MOBILIÁRIOS. *Mercado FOREX: Série Alertas*. [S.l.], 2014. Disponível em: <<http://www.cvm.gov.br/port/taxas/mercadoForex.pdf>>. Acesso em: 30.11.2014. Citado 3 vezes nas páginas 15, 25 e 27.
- COSTA, A. F. da. Previsão de séries temporais financeiras por meio de redes neurais dinâmicas e processos de transformação de dados: uma abordagem empírico-comparativa. Universidade Estadual Paulista (UNESP), 2012. Citado 4 vezes nas páginas 29, 30, 37 e 43.
- COSTA, P. H. S.; BAIDYA, T. K. N. Propriedades estatísticas das séries de retornos das principais ações brasileiras. *Pesquisa Operacional*, SciELO Brasil, v. 21, n. 1, p. 61–87, 2001. Citado na página 29.
- HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on*, IEEE, v. 5, n. 6, p. 989–993, 1994. Citado na página 40.

- HAYKIN, S. S. Redes neurais. Bookman, 2001. Citado 8 vezes nas páginas 13, 33, 34, 35, 36, 37, 38 e 39.
- HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. 2013. Disponível em: <<https://www.otexts.org/>>. Acesso em: 06.09.2015. Citado na página 30.
- INVESTOPEDIA. *Forex Walkthrough*. 2014. Disponível em: <<http://www.investopedia.com/walkthrough/forex/>>. Acesso em: 30.11.2014. Citado na página 25.
- KAYAL, A. A neural networks filtering mechanism for foreign exchange trading signals. In: IEEE. *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*. [S.l.], 2010. v. 3, p. 159–167. Citado na página 22.
- KHASHEI, M.; BIJARI, M. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, Elsevier, v. 37, n. 1, p. 479–489, 2010. Citado na página 21.
- KOZHADI, N.; BOYD, M. S.; KERMANSHAHI, B.; KAASTRA, I. A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing*, Elsevier, v. 10, n. 2, p. 169–181, 1996. Citado na página 42.
- KORCZAK, J.; HERNES, M.; BAC, M. Risk avoiding strategy in multi-agent trading system. In: IEEE. *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. [S.l.], 2013. p. 1131–1138. Citado 2 vezes nas páginas 22 e 64.
- LI, Y.; MA, W. Applications of artificial neural networks in financial economics: a survey. In: IEEE. *Computational Intelligence and Design (ISCID), 2010 International Symposium on*. [S.l.], 2010. v. 1, p. 211–214. Citado na página 21.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 33.
- MONTGOMERY, D. C.; JENNINGS, C. L.; KULAHCI, M. *Introduction to time series analysis and forecasting*. [S.l.]: John Wiley & Sons, 2015. Citado na página 29.
- NELSON, M.; HILL, T.; REMUS, W.; O'CONNOR, M. Time series forecasting using neural networks: Should the data be deseasonalized first? *Journal of forecasting*, Wiley Online Library, v. 18, n. 5, p. 359–367, 1999. Citado 3 vezes nas páginas 22, 30 e 43.
- NORVIG, P.; RUSSELL, S. *Inteligência Artificial*. Tradução: Regina Célia Simille de Macedo. Consultoria Editorial e Revisão técnica: Dr. Flávio Soares Corrêa da Silva, Dra. Leliane Nunes de Barros and Dra. Renata Wassermann. 3. ed. Rio de Janeiro: Elsevier Brasil, 2014. v. 1. Citado 8 vezes nas páginas 13, 31, 32, 34, 35, 37, 39 e 65.

- OLIVEIRA, M. A. de. *Aplicação de redes neurais artificiais na análise de séries temporais econômico-financeiras*. Tese (Doutorado) — Universidade de São Paulo, 2007. Citado 4 vezes nas páginas 22, 29, 37 e 42.
- QI, M.; ZHANG, G. P. Trend time-series modeling and forecasting with neural networks. *Neural Networks, IEEE Transactions on*, IEEE, v. 19, n. 5, p. 808–816, 2008. Citado na página 30.
- RUSCHEL, L. *Os caminhos da Análise Técnica*. 2012. Disponível em: <<http://exame.abril.com.br/rede-de-blogs/blog-da-analise-tecnica/2012/12/13/os-caminhos-da-analise-tecnica/>>. Acesso em: 11.12.2014. Citado na página 28.
- SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *Nuclear Science, IEEE Transactions on*, IEEE, v. 44, n. 3, p. 1464–1468, 1997. Citado na página 50.
- WEISS, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: MIT press, 1999. Citado 2 vezes nas páginas 31 e 32.
- WINTERS, P. R. Forecasting sales by exponentially weighted moving averages. *Management Science*, INFORMS, v. 6, n. 3, p. 324–342, 1960. Citado na página 29.
- WOOD, D.; DASGUPTA, B. Classifying trend movements in the msci usa capital market index—a comparison of regression, arima and neural network methods. *Computers & Operations Research*, Elsevier, v. 23, n. 6, p. 611–622, 1996. Citado 2 vezes nas páginas 22 e 42.
- YAO, J.; TAN, C. L. A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, Elsevier, v. 34, n. 1, p. 79–98, 2000. Citado na página 22.

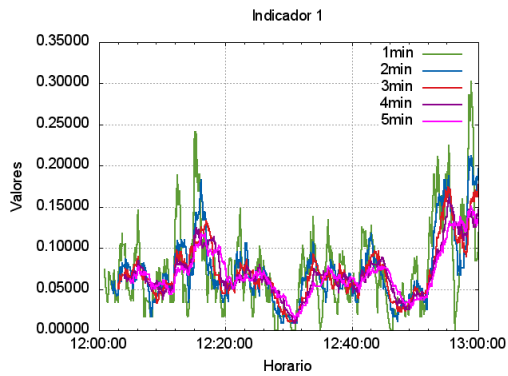
Apêndices

APÊNDICE A – PROCESSO DE CRIAÇÃO DOS INDICADORES UTILIZADOS

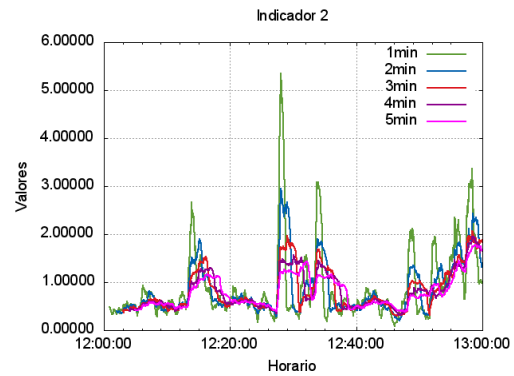
No início do desenvolvimento da análise tentou-se, a princípio, extrair indicadores que pudessem quantificar grandes variações nas séries. Decidiu-se analisar dois tipos de ocorrências: variações maiores que um valor V (VARs) e sequências de variações (SEQs). Nesse caso, uma VAR só seria aberta se houvesse uma variação entre dois *ticks* e caso não houvesse uma sequência aberta ainda. Ela só seria fechada e registrada caso a próxima variação fosse inversa a ocorrida, se fosse no mesmo sentido, então ela se tornaria uma SEQ. Nesse etapa do desenvolvimento criaram-se sete indicadores:

- Indicador 1: A soma do número absoluto de VARs com o número de absoluto SEQs dividido pelo tamanho da janela em segundos.
- Indicador 2: O número de *ticks* por segundo na janela.
- Indicador 3: O valor absoluto da soma das variações, em *pips*, das SEQs naquela janela, considerando o sinal das variações. A quantidade absoluta de *pips* de todas as variações pertencentes às SEQs.
- Indicador 4: A média de *pips* das SEQs.
- Indicador 5: O valor absoluto da soma das variações, em *pips*, das VARs naquela janela, considerando o sinal das variações.
- Indicador 6: A média de *pips* das VARs.
- Indicador 7: Número de VARs dividido pelo número de SEQs.

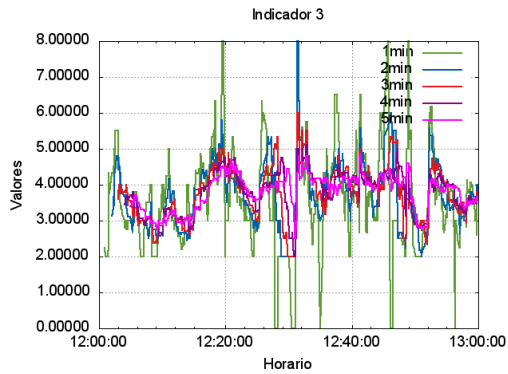
Não eram separadas as VARs e SEQs negativas, todos valores eram absolutos. As janelas utilizadas eram todas em modos sequenciais e foram testadas com cinco tamanhos diferentes: 60, 120, 180, 240 e 300 segundos.



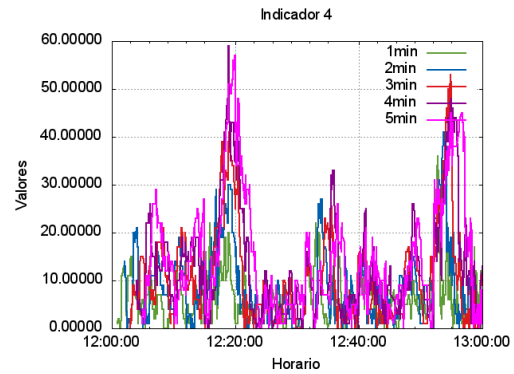
(a) soma do número absoluto de VARs com o número de absoluto SEQs dividido pelo tamanho da janela em segundos.



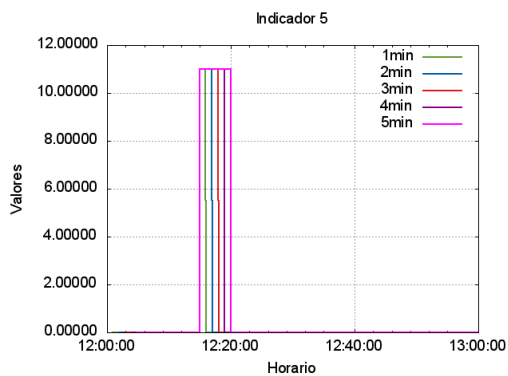
(b) O número de *ticks* por segundo na janela.



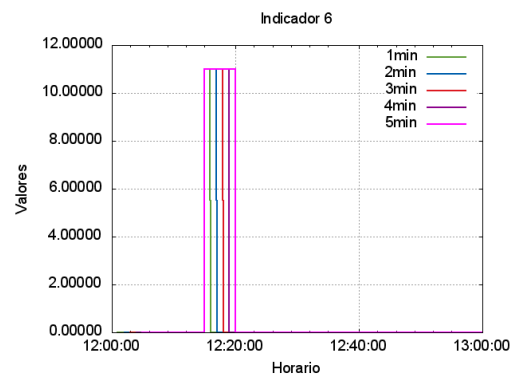
(c) O valor absoluto da soma das variações, em *pips*, das SEQs naquela janela, considerando o sinal das variações.



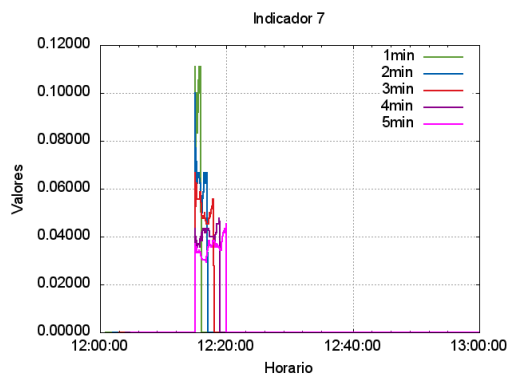
(d) A média de *pips* das SEQs.



(e) O valor absoluto da soma das variações, em *pips*, das VARs naquela janela, considerando o sinal das variações.



(f) A média de *pips* das VARs.



(g) Número de VARs dividido pelo número de SEQs.

Figura 14 – Gráficos dos indicadores extraídos na série do dia 05/05/2014, no período das 12:00 às 13:00. Dados extraídos de uma corretora real.

Através dos gráficos (Figura 14) verificou-se que as janelas de 300 segundos eram consideradas as melhores, visto que provêm curvas mais suaves. Contudo, também verificou-se que as VARs não contribuíam com muitas informações, já que aconteciam poucas vezes. Abrangeu-se, então, as condições em que as VARs ocorrem. As VARs poderiam ser registradas mesmo se aparecessem no meio de uma SEQ, fechando-se a SEQ e abrindo uma VAR. Também não seria necessário que a próxima variação fosse negativa, uma VAR ocorreria sempre que a variação fosse maior que o valor determinado.

As SEQs e VARs passaram a ser separadas por variações positivas e negativas. Os indicadores de média de *pips* para VARs e SEQs foram replicados para medir as VARs e SEQs positivas e negativas, mas também mantiveram-se as médias para todas as VARs e todas as SEQs. Também foram adicionados outros indicadores: Número de VARs e SEQs positivas e negativas. Com isso, totalizaram-se 15 indicadores criados.

Ainda assim as VARs, com registros esporádicos, não proveram informações suficientes. Optou-se então por remover as VARs e observou-se que seria mais interessante registrar apenas sequências de diferenças em que as variações totais fossem maior ou menor que V , assim existindo dois tipos de SEQs. É dessa forma que a análise técnica está configurada atualmente.

Algumas alterações finais foram realizadas nos indicadores extraídos da análise técnica, que agora são: A soma da quantidade de todas as SEQs dividido pelo tamanho da janela; *ticks* por segundo; número total de $SEQs_{menor}$; número de $SEQs_{menor-}$ e $SEQs_{menor+}$; *pips* totais das $SEQs_{menor-}$ e $SEQs_{menor+}$; número de $SEQs_{maior}$; número de $SEQs_{maior-}$ e $SEQs_{maior+}$; *pips* totais das $SEQs_{maior-}$ e $SEQs_{maior+}$. Ao total foram extraídos doze indicadores.

Foram realizados alguns experimentos utilizando *clusters* para selecionar os indicadores que fossem mais apropriados e que provessem mais informações úteis. Verificou-se que os indicadores que expressassem a quantidade de $SEQ_{s_{menor-}}$, $SEQ_{s_{menor+}}$, $SEQ_{s_{maior-}}$ e $SEQ_{s_{maior+}}$ na janela serviriam para a finalidade desejada.