UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Mateus Drigo da Silva

UM MODELO PARA ANÁLISE E TRATAMENTO DE DADOS DE DEMANDA DE ENERGIA ELÉTRICA

Alfenas, 27 de junho de 2011.

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

UM MODELO PARA ANÁLISE E TRATAMENTO DE DADOS DE DEMANDA DE ENERGIA ELÉTRICA

Mateus Drigo da Silva

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Alfenas como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Prof. Nome do Orientador(a)

Alfenas, 27 de junho de 2011.

Mateus Drigo da Silva

UM MODELO PARA ANÁLISE E TRATAMENTO DE DADOS DE DEMANDA DE ENERGIA ELÉTRICA

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Dr. Denismar Alves Nogueira

Universidade Federal de Alfenas

Prof. Dr. Luiz Alberto Beijo

Universidade Federal de Alfenas

Prof. Dr. Ricardo Menezes Salgado (Orientador)

Universidade Federal de Alfenas

Alfenas, 27 de junho de 2011.

AGRADECIMENTO

Aos meus pais, **Onivaldo Aparecido da Silva** e **Rosana Aparecida Drigo da Silva** por me apoiarem nos estudos.

Agradeço ao **Prof. Dr. Ricardo Menezes Salgado** pela orientação no desenvolvimento do projeto e na elaboração desta monografia.

Agradeço ao **Laboratório de Inteligência Computacional** (LInC) e toda a equipe de trabalho do laboratório, em especial a Mariana, Mateus e Renata, por me ajudarem neste trabalho.

Aos demais professores do **Curso de Ciência da Computação da Unifal-MG**, que possibilitaram a minha formação.

A todos os meus amigos, por estarem sempre ao meu lado em momentos difíceis e também pelos momentos de distrações, alegrias, festas.

Enfim, agradeço a todos que de alguma forma contribuíram para o resultado deste trabalho.

Muito obrigado a todos!

RESUMO

A energia elétrica ocupa um lugar de destaque no país, gerando grandes quantidades de dados que ficam sujeitos a falhas dos equipamentos, erros de medição, dados incompletos, erros humanos, entre outros. Assim, análise dos dados se torna uma tarefa crítica, principalmente em ambientes que exijam segurança e confiabilidade dos dados, uma vez que a presença de dados inconsistente prejudica na tomada de decisão, além de afetar o desempenho do sistema, a segurança e a confiabilidade das informações. Em muitos problemas do setor elétrico, como na previsão de carga ou na previsão por barramento, a qualidade dos dados reflete diretamente no resultado obtido.

O processo KDD, por sua vez, tem um papel importante por estar apto a lidar com grandes volumes de dados, que, sem a ajuda da inteligência computacional, seriam inviáveis se fossem investigados por pessoas, por maior que fosse a equipe de trabalho. Assim, neste trabalho é proposto um modelo para identificação e tratamento de *outliers*, utilizando o processo KDD, dando ênfase aos métodos estatísticos para identificação de *outliers*, e aos operadores média e redes neurais artificiais (RNA) para o devido tratamento.

Neste modelo, o tratamento dos dados de demanda de carga, de certo dia, baseia-se nas cargas elétricas dos dias próximos, já que as cargas elétricas apresentam comportamento similar em dias semelhantes como dias úteis, sábados, domingos e feriados. Porém, certos eventos, como feriados, devem ser analisados com atenção especial para não serem tratados como *outliers,* já que apresentam grande variação de comportamento em relação a um dia útil, por exemplo. Para a execução dos experimentos deste trabalho, serão utilizados dados de medição obtidos do sistema elétricos brasileiros compreendidos no período de 01/01/2010 a 31/12/2010, com a medição efetuada com discretização por minutos.

Palavras-Chave: *outliers*, energia elétrica, identificação de *outliers,* tratamento de *outliers,* redes neurais artificiais.

ABSTRACT

Electricity occupies a prominent place in the country, generating large amounts of data that are subject to equipment failure, measurement errors, incomplete data, human error, among others. Thus, data analysis becomes a critical task, especially in environments that require security and reliability of the data, since the presence of inconsistent data affect decision making and affect system performance, safety and reliability of the information . In many problems in the electricity sector, such as global load forecasting or bus load forecasting, the data quality is directly reflected in the result.

The KDD process, in turn, has an important role by being able to handle large volumes of data, without the help of computational intelligence, would be unviable if they were investigated by people, were bigger than the team. Thus, this research proposes a model for identifying and treating outliers using the KDD process, with emphasis on statistical methods to identify outliers, and average operators and artificial neural networks (ANN) for treatment.

In this model, the treatment of load demand, in a day, based on electrical charges in the coming days, since the electrical charges present similar behavior in similar days as working days, Saturdays, Sundays and holidays. However, certain events such as holidays, should be analyzed with special attention to not being treated as outliers, since wide variation in behavior toward a working day. For the execution of experiments in this paper, we used the historical data obtained from the Brazilian electrical system within a period of 01/01/2010 to 21/31/2010, with the measurement made with minutes time interval.

Keywords: outliers, electricity, identification of outliers, outlier treatment, artificial neural networks.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

CV	Coeficiente de Variação
EMR	Erro Relativo Médio
KDD	*Knowledge Discovery in Database*
LMA	*Levenberg-Marquardt*
NL	*Nest-Loop*
RNA	Rede Neural Artificial
RNAs	Redes Neurais Artificiais

SUMÁRIO

1
Introdução

Este capítulo apresenta a justificativa, a motivação e os objetivos para a realização deste trabalho de conclusão de curso juntamente com a organização dos capítulos subsequentes.

A necessidade de armazenar e tratar dados de diversas fontes é um desafio cada vez maior no setor elétrico (Wehenkel, 1998). Tendo em vista a grande quantidade de dados produzidos, erros de medições, dados incompletos, amostras distorcidas, falhas humanas ou dos equipamentos utilizados, dentre muitos outros fatores, contribuem para falta de confiabilidade destas informações.

A análise e validação dos dados medidos de energia elétrica permitem o acompanhamento da carga, detectando possíveis problemas. Contudo, devido ao grande número de pontos de medição, e consequentemente, de uma significativa quantidade de informações, a análise e validação tornam-se praticamente inviáveis de serem realizadas manualmente (Filho, 2008).

A análise da consistência dos dados é considerada uma das etapas mais importantes do processo denominado Descoberta de Conhecimento em Base de Dados (*Knowledge Discovery in Databases* - KDD), já que direciona a atenção para possíveis problemas que exijam inspeção e correção, além de fornecer informações objetivas para a tomada de decisão. Uma forma automatizada de correção garante um diferencial no desenvolvimento de estratégias de investimento de empresas públicas e privadas.

Considerando os fatos, este trabalho de conclusão de curso foi desenvolvido com o objetivo de propor um modelo computacional para identificação e tratamento de falhas em medições de energia de elétrica e realizar com base em ferramentas específicas.

1.1 Justificativa e Motivação

A previsão de cargas elétricas auxilia no bom desempenho dos sistemas elétricos, pois fornece planejamento e ajuda nas tomadas de decisões, garantindo a qualidade, segurança e confiabilidade do sistema elétrico.

Porém, medições incorretas podem contaminar o conjunto de dados, de modo a prejudicar o aprendizado de um sistema de previsão. Assim, se faz necessária uma filtragem das medições, ou seja, é necessária a utilização de um processo capaz de identificar e corrigir o maior número possível de medições problemáticas (Guirelli, 2006).

Na previsão por barramento, a análise da consistência dos dados e identificação de *outliers* [1] também é muito importante, já que a qualidade dos dados reflete diretamente no resultado da previsão. E devido ao fato da grande quantidade de informação, é essencial a utilização de um sistema computacional que possibilite a manipulação, visualização e levantamento estatístico de dados de forma simples e direta (Salgado, 2009).

Neste contexto, o processo KDD tem um papel importante por estar apto a lidar com grandes volumes de dados, que, sem a ajuda da inteligência computacional, seriam inviáveis se fossem investigados por pessoas, por maior que fosse a equipe de trabalho.

Portanto, devido a grande quantidade de informação produzida e a necessidade da análise dos dados, em busca de problemas que exijam inspeção e correção, além da inviabilidade de serem analisadas e validadas por especialistas humanos, justifica-se o uso de uma ferramenta computacional para a realização desse trabalho. Já que uma ferramenta automatizada auxilia na tomada de decisão de uma empresa, garantindo um diferencial no desenvolvimento de estratégias de investimento, a fim de melhorar a qualidade do serviço e garantir a lucratividade.

xxiixxii———————————

[1] O termo *outlier* representa uma medição de energia elétrica que foge ao padrão usual de comportamento da série. Sua definição será abordada no capítulo 2.

1.2 Problematização

Tento em vista a necessidade de uma ferramenta computacional para análise e correção dos dados de medições de energia elétrica, esta pesquisa se propõe a resolução da seguinte questão:

É possível a identificação e correção de *outliers* em medições de energia elétrica de forma satisfatória?

1.3 Objetivos

1.3.1 Gerais

Analisar um conjunto de dados provenientes de medições de energia elétrica, verificar a consistência através de métodos estatísticos e realizar o tratamento dos dados identificados como inconsistentes com técnicas de mineração de dados.

1.3.2 Específicos

Os objetivos específicos desta monografia são:

- Estudar os métodos de identificação de *outliers* a fim de selecionar os que mais se enquadram no problema;
- Codificar e aplicar os métodos selecionados;
- Estudar as técnicas de mineração de dados para tratamento de *outliers* a fim de selecionar as que mais se enquadram no problema;
- Codificar e aplicar as técnicas selecionadas;

1.4 Organização da Monografia

O desenvolvimento deste trabalho de conclusão de curso encontra-se dividido com a seguinte sequência:

O presente capítulo apresenta a justificativa, a motivação e os objetivos para a realização deste trabalho juntamente com a organização dos capítulos subsequentes.

O Capítulo 2 apresenta a descrição do problema de *outliers* em dados de energia elétrica. É abordado o conceito de *outliers*, as características da carga elétrica e apresentada a base de dados que será utilizada neste trabalho.

No Capítulo 3, por sua vez, é apresentada a fundamentação teórica, abordando os métodos para identificação e tratamento de *outliers*, envolvendo todo o processo KDD.

Já o Capítulo 4 apresenta a metodologia proposta aplicada ao problema de identificação e tratamento de *outliers* dentro do processo KDD.

No Capítulo 5 será realizada uma descrição do funcionamento do software desenvolvido como parte desde trabalho. Serão apresentadas as especificações técnicas, o banco de dados utilizado e as tecnologias envolvidas.

No Capítulo 6 são apresentadas as conclusões do trabalho diante dos resultados obtidos pelo software desenvolvido.

As referências bibliográficas estão disponíveis no Capítulo 7.

No Capítulo 8, são apresentados os anexos desta monografia, estando presentes os valores críticos tabelados para os testes de identificação de *outliers* estudados no capítulo 3.

Por fim, o Capítulo 9, apêndice, é apresentado o resultado da execução do software desenvolvido para alguns dados selecionados.

2
Descrição do Problema

Este capítulo apresenta a descrição do problema de outliers em dados de energia elétrica. É abordado o conceito de outliers, as características da carga elétrica e apresentada a base de dados que será utilizada.

Devido a grande quantidade de informação produzida no setor elétrico, é comum a presença de erros, seja por falhas nos equipamentos utilizados ou por erros humanos. Assim, a análise e o tratamento dos dados em busca desses erros são fundamentais para garantir o sucesso da previsão, por exemplo, como foi abordado no capítulo 1.

Para garantir um tratamento adequado e satisfatório para um dado problemático, é importante levar em consideração o comportamento da carga elétrica em feriados, dias úteis, sábados, domingos, entre outros fatores, que contribuem para um comportamento diferenciado da série.

Assim, este capítulo apresenta na seção 3.1 o problema de medições incorretas em dados elétricos. A seção 3.2 define o conceito de *outliers* e a importância em identificá-los. Já na seção 3.3 é apresentada a base de dados que será utilizada neste trabalho. A seção 3.4, por sua vez, aborda as características da carga elétrica que devem ser levadas em consideração para o tratamento adequado dos dados. Por fim, na seção 3.5 são apresentados alguns cálculos estatísticos como resumo das bases de dados.

2.1 Energia Elétrica

A energia elétrica ocupa um lugar de destaque na matriz energética brasileira, sendo a modalidade de energia atualmente mais consumida no país (Alvarez, 1998). Assim, quanto maior o consumo, maior serão os pontos de medições e consequentemente, maior será a quantidade de informações produzidas.

A medição de qualquer grandeza, além do erro inerente ao medidor, está sujeita a ruídos não estacionários e não gaussianos para os quais não existe modelamento, além de problemas como possíveis falhas dos medidores (Guirelli, 2006).

Nesse sentido, erros de medições, dados incompletos, errados, corrompidos ou distorcidos, falhas humanas ou dos equipamentos utilizados, dentre muitos outros fatores, contribuem para falta de confiabilidade nas informações.

Além disso, conjuntos de dados volumosos apresentam grande probabilidade de apresentarem problemas devido à dificuldade de manipulação e entendimento dos dados por especialistas humanos.

Problemas, como dados incompletos, errados ou corrompidos, distorcem o problema em análise e dificultam ainda mais a indução de hipóteses por algoritmos de Aprendizado de Máquina (Libralon, 2007).

Portanto, a análise dos dados é uma tarefa crítica, principalmente em ambientes que exijam segurança e confiabilidade dos dados, uma vez que a presença de dados inconsistente prejudica na tomada de decisão, além de afetar o desempenho do sistema, a segurança e a confiabilidade das informações.

2.2 Outliers em dados elétricos

Observando o gráfico da figura 1, que representa medições em MW/minuto da carga elétrica ao longo de um dia, pode-se observar que há padrões de comportamento nos horários próximos. Porém há grande variação próxima às 20h, em uma única medição, que caracteriza um possível erro, devido à diferença da carga em relação aos minutos anterior e posterior a essa medição.

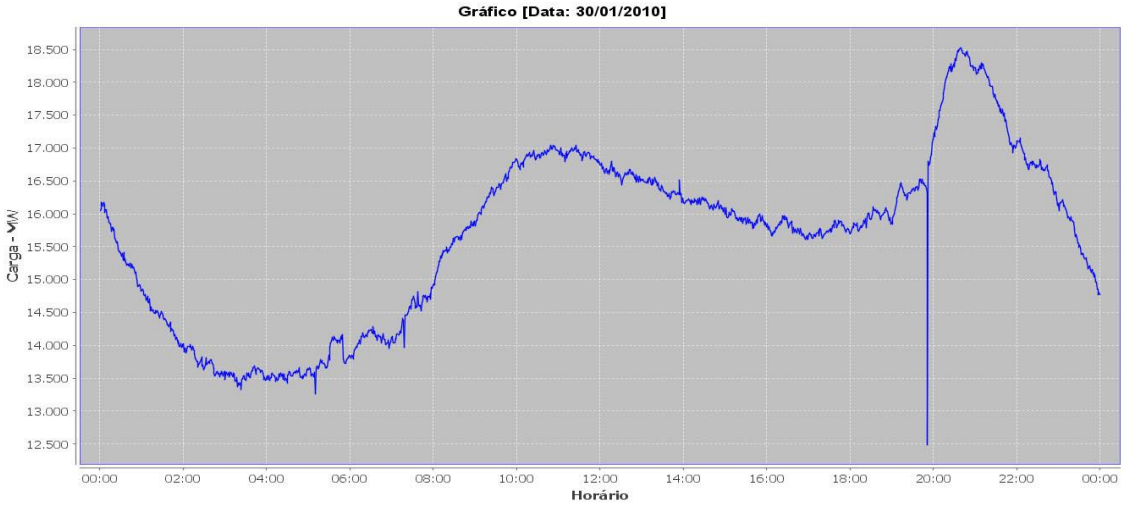


Figura 1 – Gráfico da série A do dia 30/01/2010.

Analisando agora a figura 2, que apresenta o gráfico de medições em MW/minuto da carga elétrica ao longo de um ano num mesmo horário, pode-se observar que há um padrão de comportamento da carga em todos os dias, exceto no dia 11/09/2010, em que a carga elétrica cai expressivamente, não se comportando como o esperado.

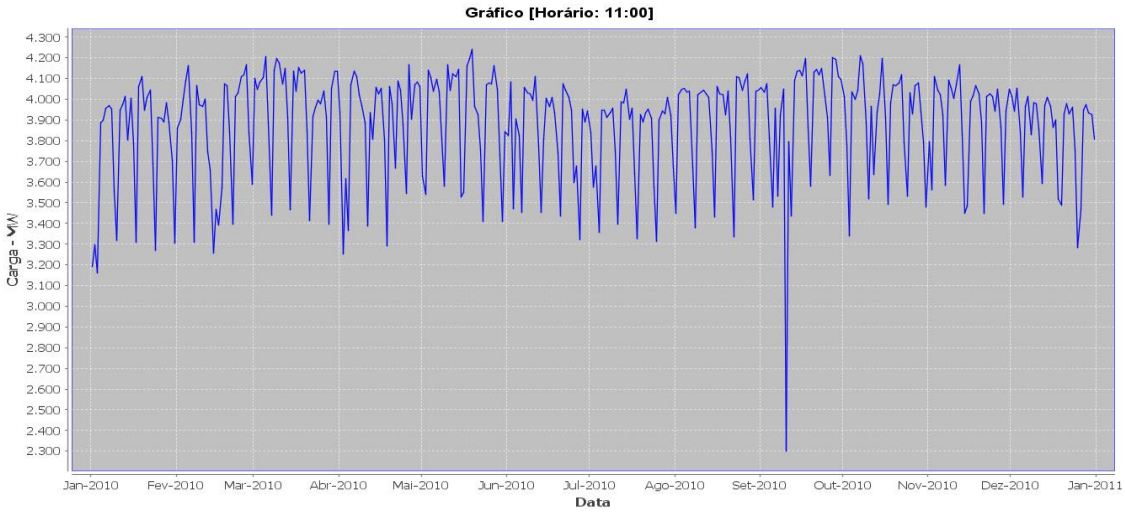


Figura 2 – Gráfico da série B no horário 11h00m.

Essas variações atípicas observadas nas figuras 1 e 2 são denominadas de várias formas: *outliers*, exceções, peculiaridades, anomalias, valores aberrantes, observações aberrantes, dados atípicos, observações discrepantes, observações

discordantes. Entretanto, o uso mais comum e o escolhido para este trabalho é o termo *outliers*.

Segundo Hawkins (1980) um *outlier* é uma observação que se desvia muitos das demais observações, a ponto de suspeitar-se que tenha sido gerada por um mecanismo diferenciado. Já para Barnett & Lewis (1994), um *outlier* é uma observação (ou subconjunto de observações) que parece ser inconsistente em relação ao restante do conjunto de dados.

Há uma variedade de aplicações, nas quais os *outliers* são importantes, entre elas destaca-se: diagnósticos de falhas; detecção de fraudes; detecção de intrusão em sistemas; monitoração de condições médicas. Nessas aplicações, observações que apresentam *outliers* precisam ser detectadas para que possam ser tratadas adequadamente, de acordo com a necessidade da aplicação (Hodge & Austin, 2004).

Dependendo da sua natureza, os *outliers* podem causar um efeito substancial na análise dos dados. Assim, é importante a identificação dos *outliers* por várias razões, entre elas:

- Melhor entendimento da série em estudo: um *outlier* detectado pode ser a evidência da ocorrência de algum fator externo afetando a série. Por exemplo, falha nos equipamentos de medição;
- Melhor modelagem e estimação: eventos desconhecidos podem afetar na modelagem e/ou estimação. Assim, não identificar os *outliers* compromete na estimativa de parâmetros do modelo, tornando pouco confiável;
- Melhor tratamento: a presença de *outliers* influencia no resultado do tratamento, pois a qualidade dos dados reflete diretamente no resultado obtido pela média e Redes Neurais Artificiais (RNAs), por exemplo.

2.3 Base de Dados

A base de dados utilizada neste trabalho corresponde a pontos de medição da carga elétrica, medidos em Megawatts (MW), discretizadas minuto a minuto, no período de 01/01/2010 ás 00h00min a 31/12/2010 às 23h59min.

Serão analisadas duas bases de dados, denominadas como série A e série B, de dados reais fornecidos por órgão do setor elétrico e representam dois pontos de medição distintos.

As figuras 1 e 2 apresentadas na seção anterior são gráficos da série A ao longo do dia 30/01/2010 e da série B, ao longo do ano de 2010 no horário das 011h00m, respectivamente. A figura 3 apresenta um gráfico da série B, ao longo do ano de 2010, no horário das 13h00min.

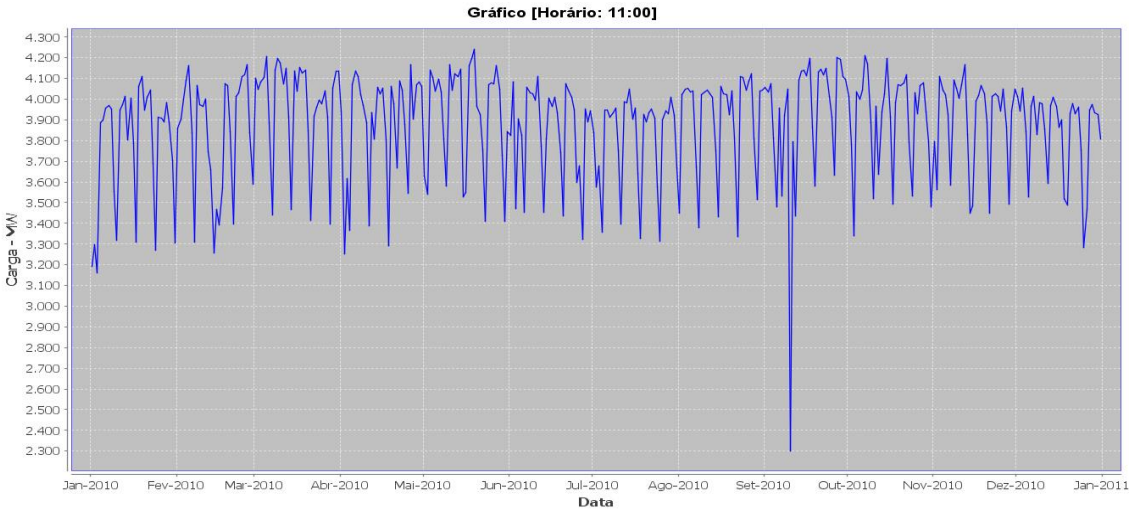


Figura 3 – Gráfico da série B no horário 13h00min.

As séries A e B são constituídas por 365 dias, sendo que cada dia contém 1440 minutos, representando então 525.600 medições de carga elétrica por série. Assim, como serão analisadas duas séries, há um total de 1.061.200 cargas elétricas a serem analisadas.

Foram disponibilizadas também as bases de dados das séries A e B tratadas manualmente por especialista de órgãos elétricos, que serão utilizadas como parâmetros de comparação com os resultados obtidos neste trabalho.

2.4 Características da Carga Elétrica

Há fatores que influenciam o comportamento da carga elétrica, que devem ser analisados com atenção especial para não serem tratados como *outliers*. Pode-se classificar estes fatores nas seguintes categorias:

- Econômicos
- Temporais
- Climáticos
- Aleatórios

Os fatores econômicos influenciam diretamente no comportamento da carga. Fatores como os níveis de atividade industrial, por exemplo, tem impactos significativos sobre o crescimento ou queda da carga. Porém, vale observar que os fatores econômicos operam em uma constante de tempo maior, quando comparadas a outros eventos que influenciam o comportamento da demanda.

Entre os fatores temporais que influenciam a carga estão: efeitos sazonais, feriados, emenda de feriados, dia após feriado e emenda de feriados com fim de semana. Os efeitos sazonais estão diretamente ligados a estações do ano e determinam o aumento da demanda de energia em função do período do ano (verão ou inverno). Já os feriados influenciam na diminuição significativa de carga para níveis bem abaixo do normal. Além disso, nos dias que antecedem ou sucedem os feriados, variações nos padrões também são notadas.

Já os fatores climáticos influenciam devido às condições meteorológicas que são responsáveis por variações significativas nos padrões de carga. Assim, temperatura e umidade, por exemplo, influenciam no valor da carga.

Por fim, os efeitos aleatórios agrupam uma variedade de eventos que causam variações nos padrões de carga que não podem ser explicados em termos dos fatores discutidos anteriormente. Um exemplo é a Copa do Mundo, em que os horários dos jogos do Brasil, influenciam no comportamento da carga.

2.5 Medidas Estatísticas

Nesta seção serão apresentadas algumas medidas estatísticas das séries A e B que descrevem seus comportamentos. Assim, a tabela 1 apresenta a média, a mediana, o desvio padrão (S), a variância (V) e o coeficiente de variação (CV), além dos valores máximo e mínimo das cargas elétricas das duas bases de dados utilizadas neste trabalho.

Tabela 1 – Medidas Estatísticas das séries A e B.

Medidas Estatísticas	*Série A*	*Série B*
Média	15834,37	3847,91
Mediana	16148,06	3872,6
Desvio Padrão (S)	2737,63	254,02
Variância (V)	7494606,20	64527,15
Coeficiente de variação (CV)	17,29 %	6,60 %
Máximo	99999,00	4891,27
Mínimo	0	-3559,82

O desvio padrão apresenta a variação dos dados a partir da média e assim como esta, é influenciado por *outliers* que podem elevar o valor dramaticamente. Assim, a série A apresenta mais *outliers* em relação à série B, já que o valor do desvio padrão é expressamente maior. Esse fato pode ser confirmado com o CV da série B que tem consideravelmente menos variação em comparação com o CV da série A, indicando que série A apresenta mais *outliers* em relação à série B.

Já a mediana de um conjunto de dados é o valor do meio quando os dados originais estão arranjados em ordem crescente (ou decrescente) de magnitude. Assim, ao contrário da média, não é influenciada por valores extremos, oferecendo uma medida mais precisa para os dados.

Outra medida que merece destaque são os valores máximo e mínimo das bases de dados, que como ilustra a tabela, indica a presença de *outliers* para o valor

máximo da série A, que desvia significativamente do comportamento da série, conforme mostra a figura 4, e para o valor mínimo da série B, que caracteriza um erro, já que uma medição de carga elétrica nunca será negativa.

Já o valor mínimo da série A, pode indicar um *outlier* ou que esse determinado momento não apresentou consumo de energia. Porém analisando o gráfico mostrado na figura 5, que corresponde ao dia 12/11/2010 no horário 05h45m, fica confirmado que trata-se de um *outlier*, já que o comportamento dos minutos anteriores e posteriores apresentam consumo de energia.

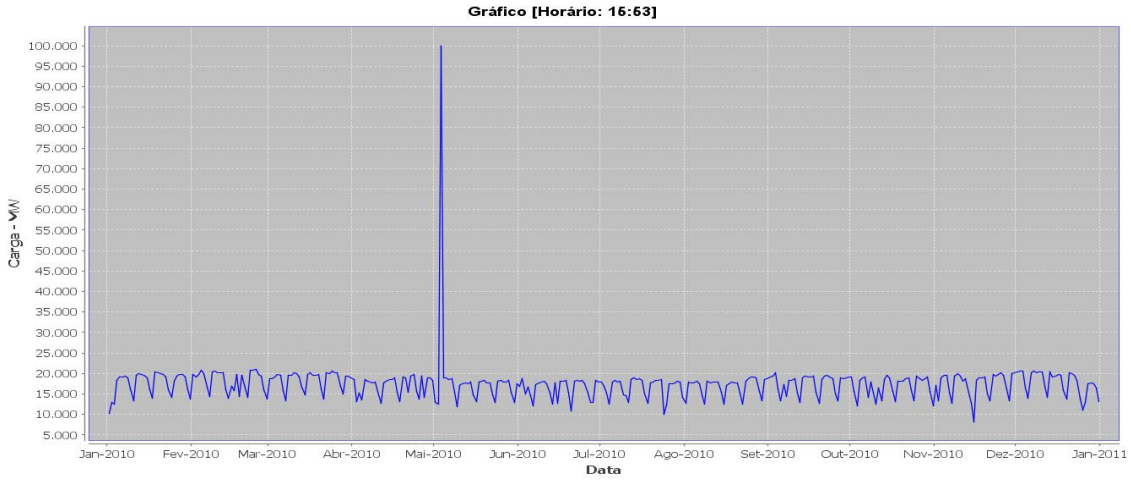


Figura 4 – Gráfico da série A no horário 15h53min.

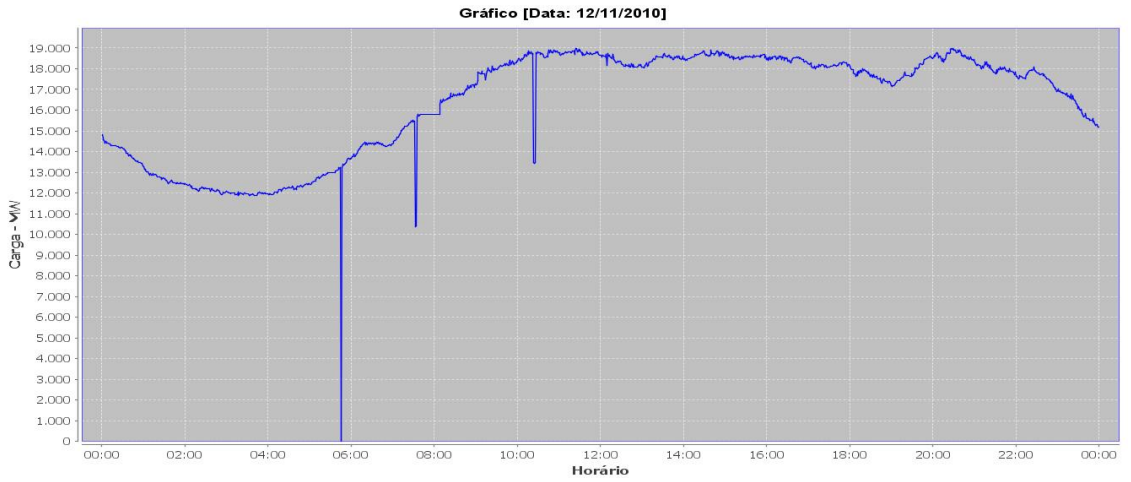


Figura 5 – Gráfico da série A do dia 12/11/2010.

3
Fundamentação Teórica

Este capítulo apresenta a metodologia utilizada neste trabalho. Serão apresentadas todas as fases do processo KDD, dando ênfase as etapas de pré-processamento, que envolve os métodos para identificação de outliers, e Mineração de Dados, em que as técnicas de tratamento de outliers são aplicadas.

Como apresentado nos capítulos anteriores, devido a grande quantidade de informação produzida no setor elétrico, técnicas computacionais devem ser utilizadas devido à inviabilidade de serem analisadas e validadas por especialistas humanos.

Assim, o processo KDD é uma ferramenta que auxilia no setor elétrico desde a verificação da consistência de pontos de medições elétricas, até a etapa de mineração de dados, que busca corrigir valores inconsistentes, seja por erro humano ou falha nos equipamentos de medição.

Este capítulo descreve o processo KDD, apresentando todas as suas fases, dando ênfase às etapas de Pré-processamento, que envolve a aplicação dos métodos de identificação de *outliers*, e Mineração de Dados, que busca corrigir os valores inconsistentes.

3.1 O Processo KDD

O crescimento rápido do volume das bases de dados em tamanho e dimensionalidade criou a necessidade e a oportunidade para extrair conhecimento destas. O processo KDD é um ramo da computação com o objetivo principal de encontrar uma maneira automatizada de explorar essas bases de dados e reconhecer os padrões existentes nesse conjunto de dados.

Em muitos textos, o termo Mineração de Dados é utilizado para designar o processo KDD como todo. Isso se deve ao fato da Mineração de Dados ser uma das principais etapas do KDD. Neste trabalho o termo Mineração de Dados será utilizado apenas para designar uma das etapas do processo KDD.

Uma definição para KDD conforme Rezende (2005, pág. 309) *apud* Fayyad *et al*. "...o processo, não trivial, de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados...".

É muito importante a distinção entre dados, informação e conhecimento nesse contexto. O dado é a estrutura fundamental sobre o qual o processo KDD atua. A partir do conjunto de dados é extraída uma informação, que pode ser entendida como uma representação ordenada e enxuta dos dados. O conhecimento provém da interpretação das informações apresentadas no processo KDD.

O processo KDD tem como objetivo extrair conhecimento de uma base de dados para ser utilizado em um processo de decisão. É uma área multidisciplinar que incorpora técnicas utilizadas em diversas áreas como Banco de Dados, Inteligência Artificial e Estatística (Rezende, 2005, pág. 309).

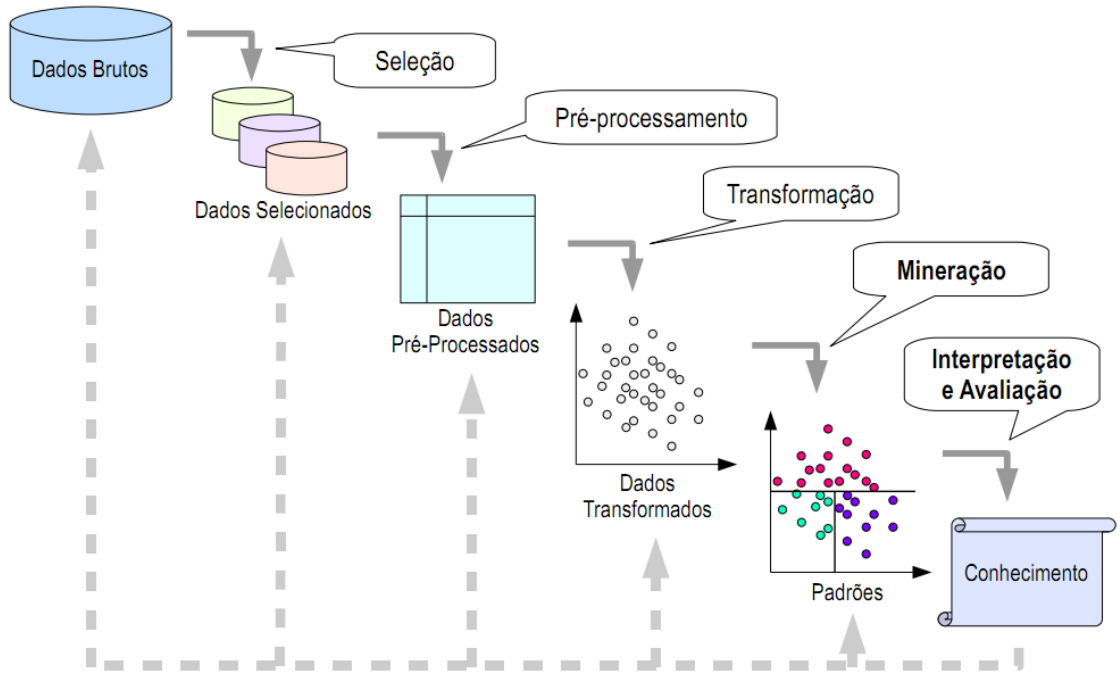


Figura 6 – Fases do processo KDD (Santos, 2009).

Como ilustra a figura 6, o processo KDD é divido em cinco fases: Seleção, Pré-processamento, Transformação, Mineração e Interpretação e Avaliação. É um processo interativo e envolve diversos laços de repetição dentro de uma mesma etapa ou até mesmo entre as fases, até que um resultado satisfatório seja alcançado. Envolve também a aplicação de diferentes tecnologias que devem ser adequadamente escolhidas dependendo do problema em questão (Thomé, 2003).

Na sequencia, são descritas cada uma dessas fases, destacando as principais tecnologias utilizadas para a aplicação do processo KDD.

3.1.1 Seleção

A fase de seleção dos dados é a primeira no KDD. Nesta fase é escolhido o conjunto de dados, pertencente a um domínio que será análisado. Normalmente a escolha dos dados fica a critério de um especialista do domínio.

É um processo complexo, uma vez que os dados podem ter origem de fontes distintas e possuírem formatos diferentes. Este passo possui impacto significante sobre a qualidade do resultado do processo (Prass, 2004).

3.1.2 Pré-processamento

Com os dados selecionados na etapa anterior, o pré-processamento consiste na verificação da consistência dos dados selecionados. É uma parte significativa no processo, pois a qualidade dos dados vai determinar a eficiência dos algoritmos de mineração (Prass, 2004).

Envolve a recuperação de dados ausentes, a identificação de *outliers* e o preenchimento de dados derivados, que são descritos nas próximas subseções.

3.1.2.1 Dados ausentes (*missing values*)

Um problema bastante comum na fase de pré-processamento é a ausência de valores para determinadas variáveis. Podem ocorrer devido a erros humanos, de software ou de hardware, ou porque a informação não estava disponível no

momento do levantamento dos dados. O tratamento é necessário para que os resultados do processo de mineração sejam confiáveis (Prass, 2004).

Uma forma de tratamento é a eliminação do registro que contenha valores faltantes. Apesar de ser uma técnica simples, há perda de informação, podendo gerar cálculos incorretos, ou resultados não satisfatório.

Existem técnicas que podem ser utilizadas para substituição dos valores faltantes. Para variáveis quantitativas, a mais simples, é o uso da média. Para variáveis categorizadas, pode-se utilizar um novo atributo para a variável, como por exemplo, usar a denotação: Desconhecido.

Técnicas mais avançadas, para ambos os tipos de variáveis, como modelos de predição são bastante utilizadas nesta etapa.

3.1.2.2 Outliers

Como definido na seção 2.2, *outliers* são dados que apresentam um comportamento bem distinto dos demais dados da amostra, podendo representar erros, sejam por falha humana ou nos equipamentos, na obtenção dos dados.

Para a identificação de *outliers*, podem ser aplicadas diferentes técnicas, como testes estatísticos, testes baseados em distância e baseados na densidade dos dados. Neste trabalho, será dada ênfase aos testes estatísticos, e também será utilizado o Algoritmo *Nest-Loop* (NL) (Amo, 2010), baseado em distância, porém com uma proposta de cálculo dos parâmetros para esse algoritmo com a utilização de calculos estatísticos.

Assim, a seção 3.2 apresenta os aspectos teóricos dos métodos, enquanto, que na seção 4.2 é apresentada a metodologia proposta dos métodos na identificação de *outliers*.

3.1.2.3 Dados derivados

Muitos dos atributos de um banco de dados apresentam relacionamentos entre eles, sendo possível obtê-los, quando não disponíveis, através da transformação ou combinação de outros atributos. Esses dados são chamados de dados derivados.

Um exemplo de um dado que pode ser calculado a partir de outro é o dia da semana, que pode ser encontrado a partir da data.

3.1.3 Transformação

Após serem selecionados e pré-processados os dados necessitam ser armazenados e formatados adequadamente para que os algoritmos de aprendizado possam ser aplicados.

As seguintes técnicas podem ser aplicadas, segundo Braga (2005):

- Sumarização: reduzir o número de valores mediante de alguma agregação, por exemplo, substituir dados diários por médias semanais.
- Razões: gerar uma nova variável a partir da razão de duas outras.
- Codificação: transformar dados qualitativos em quantitativos.
- Codificação simbólica: intervalos de variação passam a ser associados a uma categoria.
- Redução de variáveis: eliminar variáveis redundantes ou com pouco poder preditivo.
- Parametrização: transformar uma variável em outra cujo domínio de variação seja mais adequado. Por exemplo, a padronização.
- Transformações matemáticas: calcular uma função da variável obtendo-se uma nova variável com propriedades mais convenientes.
- Normalização: converter os valores em uma faixa pré-fixada de valores, por exemplo, no intervalo [0,1].

3.1.4 Mineração

Todas as etapas do processo de KDD possuem grau elevado de importância para o sucesso do mesmo. Entretanto, é a etapa de Mineração de Dados (*Data Mining*) que recebe o maior destaque na literatura, já que nesta fase é realizada a escolha de uma técnica de mineração de dados, com base na tarefa do problema a ser solucionado, com o objetivo de extrair padrões do conjunto de dados.

A escolha da tarefa de mineração de dados é realizada com base nos objetivos do problema a ser solucionado. Como mostra a figura 7, às tarefas são divididas em atividades preditivas e atividades descritivas.

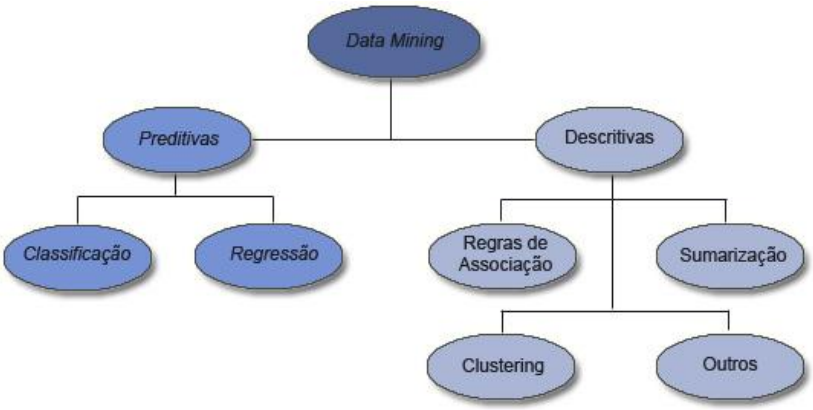


Figura 7 – Tarefas de Mineração de Dados (Rezende, 2005, pág. 318).

Nas Atividades preditivas a aprendizagem do modelo é supervisionada, ou seja, é fornecida uma classe buscando identificar uma nova amostra de dados a partir do conhecimento adquirido de um conjunto de amostras com classes conhecidas. As principais tarefas preditivas são:

- Classificação: consiste na predição de um valor categórico.
- Regressão: esta tarefa é similar à tarefa de classificação, porém consiste na predição de valores contínuos.

Já as atividades descritivas trabalham com um conjunto de dados que não possuem uma classe determinada, buscando identificar padrões de comportamento comuns nestes dados (aprendizado não supervisionado). As principais tarefas descritivas são:

- Regras de Associação: usado para particionar os registros de uma base de dados em subconjuntos. Nesta tarefa, não há classes predefinidas, os registros são agrupados segundo algum critério de semelhança.
- *Clustering*: consiste em identificar um conjunto finito de *clusters*, ou agrupamentos, a partir dos dados. O agrupamento é feito com base no valor de atributos.

- Sumarização: consiste em identificar e apresentar, de forma concisa e compreensível, as principais características dos dados em um conjunto de dados.

Com base na tarefa de mineração de dados é escolhida uma técnica de mineração de dados a ser aplicado ao problema. Não existe uma única técnica que resolva todos os problemas de mineração de dados, pois diferentes técnicas servem para diferentes propósitos, cada uma oferecendo vantagens e desvantagens (Prass, 2004).

Assim, a escolha da técnica está fortemente relacionada com o tipo de conhecimento que se deseja extrair ou com o tipo de dado no qual ela será aplicada. Entre as técnicas utilizadas, destacam-se: Árvores de Decisão, regras de produção, RNAs, Redes *Bayesianas*, entre outras.

Portanto, a etapa de Mineração de Dados compreende a aplicação de uma técnica de mineração de dados escolhida com base na tarefa do problema, com o objetivo de extrair padrões contidos nos dados.

A seção 3.3 apresenta os aspectos teóricos das técnicas utilizadas neste trabalho. Já a seção 4.3 apresenta os detalhes de como essas técnicas foram aplicadas ao problema proposto.

3.1.5 Interpretação e Avaliação

O resultado obtido deve ser interpretado e avaliado para verificar se o objetivo foi alcançado. Caso o resultado não seja satisfatório, o processo pode retornar a qualquer um dos estágios anteriores do processo KDD ou até mesmo ser recomeçado.

Entre as ações mais comuns caso o resultado não seja satisfatório são: modificar o conjunto de dados iniciais, alterar as configurações de entrada do algoritmo aplicado, escolher um novo algoritmo de mineração de dados (Prass, 2004).

3.2 Métodos de identificação de *Outliers*

Durante a etapa de pré-processamento dos dados no processo KDD, uma das tarefas é a análise da consistência dos dados, tendo como objetivo eliminar dados redundantes e inconsistentes, recuperar dados incompletos e avaliar possíveis *outliers*, indicando algum erro no processo de obtenção desses dados.

Nesta seção, serão apresentados os principais testes estatísticos para identificação de *outliers*. Esta é considerada uma das etapas mais importantes do processo, já que direciona a atenção para possíveis problemas que exijam inspeção e correção, além de fornecer informações objetivas para a tomada de medidas corretivas.

3.2.1 Boxplot

O *boxplot*, também conhecido como diagrama em caixa, é um gráfico frequentemente usado para revelar o centro, a dispersão e a distribuição dos dados, além da presença de *outliers* (Triola, 2005, pág. 79).

É construído com base na mediana, no quartil inferior (q1), no quartil superior (q3) e no intervalo interquartil (IQR), que é calculado com base na equação 1.

$$IQR = Q_3 - Q_1 \tag{1}$$

Assim, para a construção do gráfico, traça-se uma linha central marcando a mediana do conjunto de dados. A parte inferior da caixa é delimitada pelo quartil inferior e a parte superior pelo quartil superior. As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior até o limite inferior calculado pela equação 2 e do quartil superior até o limite superior calculado pela equação 3. Os valores inferiores ao limite inferior e superiores ao limite superior são caracterizados como *outliers* (Silva, 2008).

$$\lim ite\ \inf erior = q1 - 1.5 \times IQR \tag{2}$$

$$\lim ite\ \sup erior = q3 - 1.5 \times IQR \tag{3}$$

Os valores limite inferior e o limite superior delimitam, respectivamente, os traços inferiores e superiores e constituem limites para além dos quais, como visto, os dados passam a ser considerados *outliers*. A figura 8 representa um gráfico *boxplot*, indicando as principais medidas estatísticas e exemplos de *outliers*.

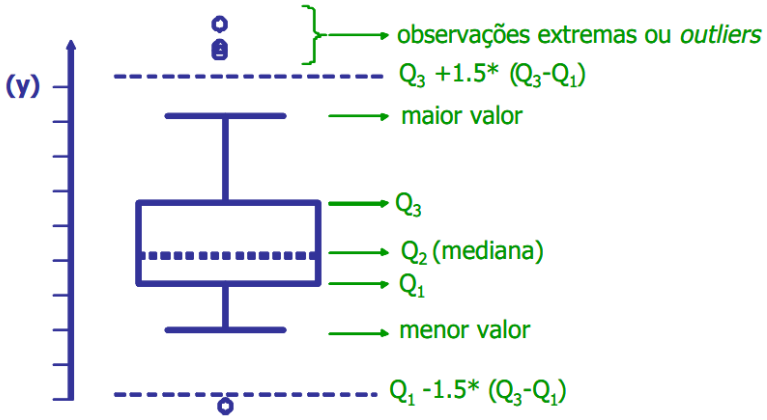


Figura 8– Gráfico de *Boxplot* (Conceição, Alencar & Alencar, 2010).

O *boxplot* é um método simples de ser aplicado para identificação de *outliers*, porém verifica a presença de *outliers* somente nos extremos de um conjunto de dados.

3.2.2 Teste de *Dixon*

O teste de *Dixon* é um dos testes mais citados na literatura para avaliar a presença de *outliers* em um conjunto de dados. É um método simples de ser aplicado, baseado na comparação do valor suspeito com os demais valores do conjunto de dados. As próximas duas subseções apresentam, respectivamente, o teste de *Dixon* para pequenas amostras de dados, também conhecido como teste Q de *Dixon*, e os testes para amostras maiores de dados.

3.2.2.1 Teste Q de *Dixon*

O Teste Q de *Dixon* é apropriado para pequenos conjuntos de dados, geralmente, para até 10 (dez) observações (Ellison, Barwick & Farrant, 2009, pág. 49). Tem como

objetivo verificar a presença de *outliers* nos extremos, ou seja, verifica se o menor e o maior valor do conjunto de dados são *outliers*.

O teste consiste em calcular o valor Q para o menor e maior valor da amostra e compará-los com o valor crítico tabelado para o nível de confiança desejado. Caso os valores calculados excedam o valor crítico tabelado, são considerados como *outliers*.

Para a aplicação do teste, conforme Ellison, Barwick & Farrant (2009, pág. 50) segue-se o algoritmo abaixo:

1. Ordenar os dados amostrais em ordem crescente;
2. Calcular o valor de Q para o menor e maior valor da amostra, conforme as equações 4 e 5, respectivamente;

$$Q_{menor\_valor} = \frac{x_2 - x_1}{x_n - x_1} \tag{4}$$

$$Q_{maior\_valor} = \frac{x_n - x_{n-1}}{x_n - x_1} \tag{5}$$

3. Calculado os valores de Q para o menor e maior valor da amostra, obtém-se o valor crítico da tabela de *Dixon*, e aplica-se o teste abaixo:
    a. Para o menor valor:
        i. Se $Q_{menor\_valor} > Q_{crítico}$ o valor é considerado um *outlier*;
        ii. Se $Q_{menor\_valor} \leq Q_{crítico}$ o valor não é considerado um *outlier*;
    b. Para o maior valor:
        i. Se $Q_{maior\_valor} > Q_{crítico}$ o valor é considerado um *outlier*;

ii. Se $Q_{maior\_valor} \leq Q_{crítico}$ o valor não é considerado um *outlier*;

Os valores críticos tabelados para o teste de *Dixon* são apresentados no Anexo A, e também podem ser obtidos em Kanji (2006, pág. 198) e Ellison, Barwick & Farrant (2009, pág. 207) para vários níveis de significância.

3.2.2.2 Teste de *Dixon* para amostras maiores

O teste de *Dixon* inclui uma variedade de testes diferentes para amostras de dados maiores que dez. Para ser aplicado, segue os mesmos passos do Teste Q de *Dixon*, com a diferença de que as equações para o menor a maior valor do conjunto de dados será escolhida conforme o tamanho da amostra. Assim, a tabela 2, mostrada na página seguinte, apresenta as equações para calcular o valor Q, para o menor e maior valor da amostra, conforme a quantidade de dados.

Usando um teste específico fora da faixa de dados recomendada como, por exemplo, usar o teste Q de *Dixon* recomendado para amostras de tamanho até 10 (dez) para conjuntos de dados maiores corre-se risco de mascaramento de dois valores extremos (Ellison, Barwick & Farrant, 2009, pág. 50). Assim, seguindo as recomendações mantém a probabilidade de detecção de *outliers*.

A aplicação de diferentes testes desta série também não é aconselhável. Deve-se escolher um critério adequado para o tamanho do conjunto de dados, tendo em conta os valores adicionais suspeitos, se necessário (Ellison, Barwick & Farrant, 2009, pág. 50).

Portanto, o teste de Dixon é um método simples de ser aplicado, porém é utilizado para identificação de *outliers* apenas nos extremos de conjunto de dados, ou seja, apenas para o menor e o maior valor da amostra.

Tabela 2 – Fórmulas para o calculo de Q do Teste de *Dixon* (Kanji, 2006, pág. 54).

Tamanho da amostra de Dados	*Teste*	
	Para o menor valor Q_{menor_valor}	*Para o maior valor* Q_{maior_valor}

3 a 7	$\frac{x_2 - x_1}{x_n - x_1}$	$\frac{x_n - x_{n-1}}{x_n - x_1}$
8 a 10	$\frac{x_2 - x_1}{x_{n-1} - x_1}$	$\frac{x_n - x_{n-1}}{x_n - x_2}$
11 a 13	$\frac{x_3 - x_1}{x_{n-1} - x_1}$	$\frac{x_n - x_{n-2}}{x_n - x_2}$
14 a 25	$\frac{x_3 - x_1}{x_{n-2} - x_1}$	$\frac{x_n - x_{n-2}}{x_n - x_3}$

3.2.3 Teste de *Grubbs*

O Teste de *Grubbs* é um teste similar ao teste de *Dixon*, porém em vez de realizar a comparação do valor suspeito com os demais valores do conjunto de dados, é feita a comparação entre o valor suspeito e a média do conjunto de dados. E também é utilizado o desvio padrão como denominador em vez da amplitude, como acontece com o teste de *Dixon* para conjunto de dados de tamanho até dez.

A seguir, são apresentados os passos para a aplicação do teste, conforme Alfassi, Borger & Ronen (2005, pág. 70), para todos os valores suspeitos de serem *outliers*:

1. Ordenar os dados em ordem crescente;
2. Calcular a média ($\bar{x}$) da amostra de dados sendo analisada;
3. Calcular a diferença do valor suspeito de em relação à média, conforme a equação 6;

$$d_i = \left| x_i - \bar{x} \right| \tag{6}$$

4. Calcular o desvio padrão (s).

5. Com base nos resultados dos passos anteriores, calcular o valor de G para os dados suspeito do conjunto, conforme indica a equação 7;

$$G = \frac{d_i}{s} \tag{7}$$

6. Por fim, comparar o valor de G , com o valor crítico tabelado, representado por $G_{crítico}$, e aplicar o teste abaixo:

   a. Se $G > G_{crítico}$ o valor é considerado um *outlier*;

   b. Se $G \leq G_{crítico}$ o valor não é considerado um *outlier*;

Os valores críticos tabelados para o teste de *Grubbs* são apresentados no Anexo A para os níveis de significância de 90%, 95% e 99% e também podem ser obtidos em Ellison, Barwick & Farrant (2009, pág. 208).

3.2.4 Teste do Erro

O Teste do Erro é um teste estatístico similar ao teste de *Grubbs*, mas não utiliza dos valores críticos tabelados. Assim, é utilizado um valor padrão no lugar do valor crítico tabelado.

A equação 8 apresenta como calcular o erro de um valor considerado suspeito de ser *outlier*, sendo que $\bar{x}$ representa a média da amostra de dados, x_i representa o dado sendo analisado e s representa o desvio padrão.

$$Erro = \frac{|x_i - \bar{x}|}{s} \tag{8}$$

Calculado o valor do erro para um dado considerado suspeito é aplicado o teste abaixo para um nível se significância de 2%, conforme da amostra de dados sendo analisada. Vale ressaltar que o teste só pode ser aplicado para conjuntos de dados maiores que cinco.

1. Se $5 < n \leq 8$ então:

a. Se $Erro > 6$ o valor é considerado um *outlier*;

b. Caso contrário, o valor não é considerado um *outlier*;

2. Se $8 < n \leq 14$ então:

a. Se $Erro > 5$ o valor é considerado um *outlier*;

b. Caso contrário, o valor não é considerado um *outlier*;

3. Se $n > 15$ então:

a. Se $Erro > 4$ o valor é considerado um *outlier*;

b. Caso contrário, o valor não é considerado um *outlier*;

Como visto, é um teste simples de ser aplicado, com a vantagem de não depender dos valores críticos tabelados. Porém, os valores críticos tabelados garantem maior precisão na identificação de *outliers.*

3.2.5 Teste *Z-Score*

O *score* z ou *z-score* é uma medida de posição, que descreve a localização de um valor, em termos de desvios padrões, em relação à média. Assim, um *score* z igual a 3, por exemplo, indica que determinado valor está a três desvios padrões acima da média, e um *z-score* de -3, indica três desvios padrões abaixo da média.

Convertendo os dados em seus valores *z-scores* correspondentes, pode-se utilizar destes valores para a identificação de *outliers,* pois um *z-score* muito alto indica que determinado valor está fora do padrão de comportamento do restante do conjunto de dados.

Assim, para a aplicação deste teste, primeiramente, é calculado o *z-scores* do(s) valor(es) suspeito(s) de ser(em) *outlier*(s), conforme a equação 9.

$$z = \frac{x - \bar{x}}{s} \tag{9}$$

Em seguida, é realizada uma comparação do *z-score* calculado com um valor padrão fixado, de acordo com o tamanho da base de dados. Conforme o resultado dessa comparação, o valor é classificado como um *outlier*.

Assim, conforme Sarabando (2010), com n indicando o tamanho da amostra de dados sendo analisada, tem-se:

1. Se $n \leq 50$ então:
    a. Se $-2.5 > zscore > 2.5$ o valor é considerado um *outlier*;
    b. Caso contrário, o valor não é considerado um *outlier*;
2. Se $50 < n < 1000$ então:
    c. Se $-3.3 > zscore > 3.3$ o valor é considerado um *outlier*;
    d. Caso contrário, o valor não é considerado um *outlier*;
3. Se $n \geq 1000$ então:
    c. Se $-3.3 \geq zscore \geq 3.3$ o valor é considerado um *outlier*;
    d. Caso contrário, o valor não é considerado um *outlier*;

3.2.6 Critério de *Peirce*

O critério de *Peirce* é um método mais elaborado para identificação de *outliers*, baseado na teoria de probabilidade. Não é muito utilizado, devido à dificuldade em calculá-lo e ao fato de que *Chauvenet* propôs um critério semelhante, porém mais simples de ser calculado.

Para ser aplicado, seguem-se os passos abaixo, conforme Ross (2003):

1. Calcule a média ($\bar{x}$) e o desvio-padrão (s) da amostra de dados sendo analisada;

2. Para quaisquer medidas de dados suspeitas, obtenha a diferença entre o valor suspeito e a média da amostra de dados, conforme indica a equação 10;

$$| x_i - \bar{x} | \tag{10}$$

3. Em seguida, calcule a distância máxima permitida obtida pela equação 11. Obtenha o valor de R correspondente ao tamanho do conjunto de dados, a partir da tabela de valores críticos disponíveis no Anexo A. Suponha para a primeira aplicação do teste, o caso de um único valor suspeito, mesmo se parece haver mais de um;

$$R = \frac{| x_i - \bar{x} |_{\max}}{s} \rightarrow | x_i - \bar{x} |_{\max} = R \times s \tag{11}$$

4. Considere como *outliers* os valores que forem maiores que a distancia máxima permitida, como mostra a equação 12;

$$| x_i - \bar{x} | > | x_i - \bar{x} |_{\max} \tag{12}$$

5. Se isso resultar na identificação de algum *outlier*, assumir o caso de duas observações suspeitas e reaplicar o teste, mantendo os valores originais da média, desvio padrão e tamanho da amostra de dados. Caso resultar na identificação de dois *outliers*, aplicar novamente, considerando agora três valores suspeitos. Repita os cálculos em sequência crescente conforme o número de possibilidades de valores duvidosos até que não haja mais dados que precisem ser eliminados.

6. Posteriormente, eliminam-se os dados que foram identificados como *outliers*, calcula novamente a média e desvio padrão do novo conjunto de dados reduzido e retorna ao passo 2.

7. A aplicação do método se repete até que não sejam identificados novos *outliers*.

3.2.7 Critério de *Chauvenet*

O critério de *Peirce* é um método não trivial para identificação de *outliers*, de modo que *Chauvenet* apresentou outro critério, aproximado, porém bastante simplificado.

O critério proposto por *Chauvenet* especifica a eliminação de um único valor duvidoso. Para eliminar um segundo valor seria necessário recalcular a média e o desvio padrão para o novo conjunto de dados e só então aplicar novamente o critério. Porém, *Chauvenet* não especifica nenhum limite para a aplicação do método. Entretanto, como a cada novo cálculo o desvio padrão diminui, é muito provável que essa aplicação sucessiva resulte na eliminação de um grande número de dados. Uma vez que não há garantia formal de que mesmo o primeiro dado eliminado seja realmente um *outlier*, é preferível aplicar o critério uma única vez para cada conjunto de dados, eliminando todos os valores que se encontram fora do intervalo estabelecido (Souto, 2009).

O método especifica que um valor medido será considerado um *outlier* se a probabilidade do desvio padrão em relação à média é menor que $\frac{1}{2n}$, sendo n o tamanho da amostra de dados sendo analisada.

A figura 9 apresenta a curva teórica de uma distribuição normal. Pode-se concluir que os resultados "bons" ocupam a faixa central escura, de área igual a $\left(1-\frac{1}{2n}\right)\times 100\%$ da área total sob a curva. Naturalmente, os *outliers* ocupam as áreas extremas sob a curva de acordo com indicação da figura (Soares, 2009).

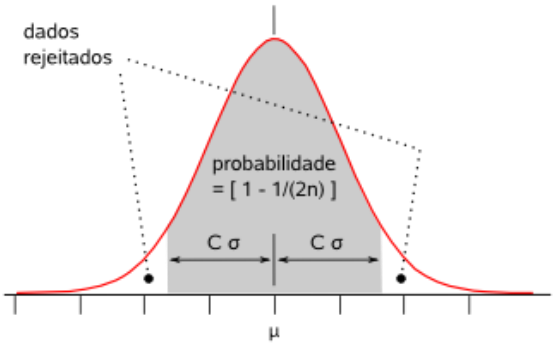


Figura 9 – Curva de distribuição para o critério de *Chauvenet* (Soares, 2009).

Na figura, o coeficiente C correspondente ao número de desvios-padrão para a faixa de valores considerados aceitáveis. Esse valor é tabelado e alguns valores estão disponíveis no Anexo A, e também podem ser obtidos em Gujer (2008, pág. 244), para os níveis de confiança de 90%, 95% e 99%.

Assim, para cada x_i com $0 < i < n$, a faixa de valores aceitáveis para x_i não ser considerado como *outlier*, será dada pela equação 13, em que $\bar{x}$ representa a média da amostra de dados sendo analisada e s o desvio padrão.

$$\bar{x} - C \times s \;<\; x_i \;<\; \bar{x} + C \times s \tag{13}$$

3.2.8 Teste de *Cochran*

O teste de *Cochran* é outro dos testes mais citados na literatura para identificação de *outliers*. Neste teste, comparam-se variâncias, ou seja, verifica se a variância dos resultados obtidos por um grupo é excessiva em relação à dos demais grupos. Tem como limitação o fato de que as amostras são retiradas de k grupos distribuídas normalmente e também à exigência de que cada grupo possua a mesma quantidade de dados (Kanji, 2006, pág. 75).

Assim, para um conjunto de dados de tamanho n, divididos igualmente em k grupos de tamanho m, cada um com desvio-padrão amostral s_i $(i = 1,2,\ldots,n)$, o valor a calcular para o teste de *Cochran* é dado pela equação 14. Na equação s_i^2 representa a estimativa da variância para o grupo i, com $0 < i < k$ e $s_{\max}^2$ representa o maior valor encontrado no conjunto como estimativa da variância.

$$c_{calculado} = \frac{s_{\max}^2}{\sum_{i=1}^{n} s_i^2} \tag{14}$$

Calculado o valor $c_{calculado}$, compara-se com o valor crítico tabelado para k e m adequados. A hipótese de que há grande variação no grupo analisado em

relação aos demais é rejeitada caso o valor observado de $C_{calculado}$ não exceda o valor crítico (Kanji, 2006, pág. 75).

Os valores críticos tabelados para o teste de *Cochran* para os níveis de confiança de 95% e 99% podem ser obtidos em Kanji (2006, pág. 211-212) e Ellison, Barwick & Farrant (2009, pág. 209). Alguns valores estão disponíveis no Anexo A.

3.2.9 Razão Q

O Teste da Razão Q é um método simples de ser aplicado para a verificação de *outliers*, baseado na distância entre o valor suspeito e a amplitude geral do conjunto de dados.

Para a aplicação do teste aplicam-se os seguintes passos, conforme Lopes (2003):

1. Ordenar os dados de modo decrescente;
2. Calcular a diferença entre o valor suspeito (possível *outlier*) e seu vizinho mais próximo (d);
3. Calcular a amplitude dos dados, conforme a equação 15, e aplicar a equação 16, para calcular o valor de Q ;

$$A = x_n - x_1 \tag{15}$$

$$Q_{calculado} = \frac{d}{A} \tag{16}$$

4. Com base no valor de Q calculado e nos valores críticos tabelados, aplicar o seguinte teste:
    - Se $Q_{calculado} \geq Q_{tabelado}$ o valor é considerado um *outlier*;
    - Caso contrário, não é considerado um *outlier*.

Os valores críticos tabelados para o teste da Razão Q para os níveis de confiança de 90%, 95% e 99% estão disponíveis no anexo A e também podem ser obtidos em Lopes (2003, pág. 9).

3.2.10 Algoritmo *Nest-Loop* (NL)

Um dado x de base de dados D é dito um *outlier* se pelo menos uma fração p com $0 < p < 1$ dos valores de D estão a uma distância maior do que d de x (Amo, 2008). Por exemplo, na figura 10 o valor circundado é um *outlier*, para $p = 2/3$ e $d' > d$, ou seja, oito dos doze valores de D estão a uma distância maior do que d deste valor circundado.

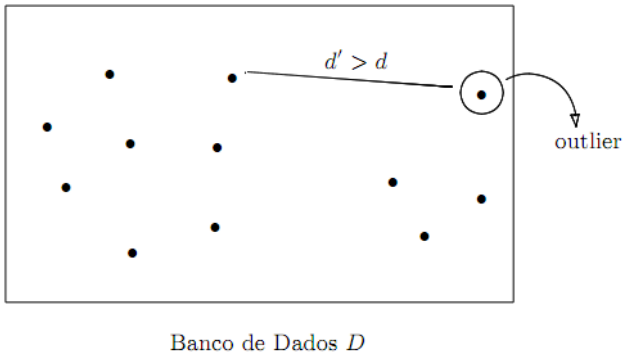


Figura 10 – Exemplo de identificação de *outliers* (Amo, 2008).

Assim, um dado x de uma base de dados D é identificado como um *outlier* se pelo menos uma fração p dos dados sendo analisados estão fora de uma vizinhança de raio d de x.

Sejam D a base de dados, n o tamanho de D, p um valor entre 0 e 1 e $d > 0$. Calcula-se o valor de M, que representa o número máximo de dados que estão dentro de uma vizinhança de um *outlier*, pela equação 17.

$$M = n \times (1 - p) \tag{17}$$

Um algoritmo simples para a verificação de *outliers* é mostrado no quadro 1. A função distância calcula a distância entre os dois parâmetros passados. No final da execução desse algoritmo, os dados não marcados como "não-*outlier*" serão considerados *outliers*.

Quadro 1 – Algoritmo NL.

```
//algoritmo NL;

Para cada dado x_i com 0 < i < n de D

    contador = 0;

    Para cada dado y_i com 0 < j < n de D

        se distância(x_i, y_i) ≤ d então contador = contador + 1;

    fim-para

    se contador ≥ M então marca-se x_i como não-outlier;

fim-para
```

Esse algoritmo é chamado de *Nest-Loop* (NL) e foi proposto por Amo (2008). Apresenta como desvantagem o fato do algoritmo não ser inteiramente automático, pois a determinação dos parâmetros p e d são tarefa de um especialista humano.

3.3 Técnicas de Tratamento para *outliers*

As técnicas de Mineração de Dados são usadas para extração de padrões embutidos nos dados. Embora esse tipo de conhecimento exista nas bases de dados, torna-se inviável de ser realizado manualmente por conta da limitação cognitiva do ser humano em avaliar uma grande quantidade de dados. Um *software* que detenha desse conhecimento poderá utilizá-lo de maneira muito mais rápida e eficiente.

Serão apresentadas duas formas de tratamentos para os dados identificados como *outliers* que foram escolhidas para este trabalho: a média e as RNAs. Optou-se pela média devido às características de comportamento da carga, que são semelhantes para os dias em comum, como por exemplo, em dias úteis, sábados, domingos e feriados, que garantem um resultado satisfatório para o valor tratado com base nos dias anteriores. Já as RNAs foram escolhidas devido as suas

características que permitem a sua aplicação na resolução de vários problemas com desempenho satisfatório.

3.3.1 Média

A média é um dos conceitos mais básicos e cotidianos da estatística. É usada para resumir dados quantitativos aproximadamente simétricos, representando uma medida central de um conjunto de dados.

É calculada conforme a equação 18, em que n representa o tamanho da amostra de dados e x_i representa os elementos do conjunto, com $1 < i < n$.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{18}$$

A carga elétrica apresenta comportamento semelhante entre dias úteis, sábados, domingos, feriados, entre outros. Assim, se determinado dia apresentar um *outlier*, é possível corrigi-lo de forma rápida e eficiente com base na média entre os dias próximos.

Porém, a média é muito sensível a valores extremos, por isso, ao utiliza-lá no tratamento de um dado devem-se evitar valores próximos a esse dado que também sejam classificados como *outliers.*

3.3.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são técnicas de inteligência computacional que apresentam um modelo matemático baseado no sistema nervoso humano. Uma definição conforme Haykin (2001, pág. 28):

> *Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torna-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:*
>
> 1. *O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.*

2. *Forças de conexão entre neurônios, conhecidos como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.*

As unidades de processamento, chamados de neurônios, são distribuídas em uma ou mais camadas. Sua estrutura é apresentada na figura 9. Os valores $x_1, x_2, ..., x_m$ constituem as entradas da RNA, os $W_{k1}, W_{k2}, ..., W_{km}$ são os pesos sinápticos, que como visto, são utilizados para armazenar o conhecimento adquirido. Apresenta também um somador responsável por somar os sinais de entrada ponderados pelas respectivas sinapses do neurônio, uma função de ativação para restringir a amplitude da saída de um neurônio e y_k que representa o saída. O modelo também apresenta b_k que representa a função *bias* responsável pelo efeito de aumentar ou diminuir a entrada líquida da função de ativação.

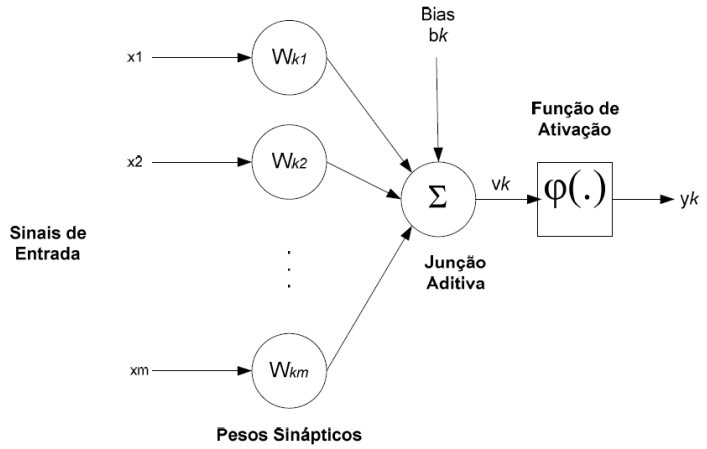


Figura 11 – Estrutura do neurônio de uma RNA.

O processo de aprendizado de uma rede neural artificial está associado à capacidade de as mesmas adaptarem os seus parâmetros como consequência da sua interação com o meio externo (Rezende, 2005, pág. 142).

As RNAs são configuradas para problemas específicos através de um processo denominado aprendizado. Seu uso no tratamento de dados consiste em treinar a RNA com base em um conjunto de dados de entrada e saída.

A arquitetura de uma RNA determina a forma como os neurônios estão organizados. Arquiteturas *Multi Layer Perceptron* (MLP), *Radial Basis Fu*nction (RBF), *Probabilistic Neural Networks* (PNN), *Recurrent Neural Networks* (RNR), são alguns exemplos. Neste trabalho serão utilizadas as MLP que são compostas por

uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. Também será utilizada a técnica *Levenberg-Marquardt* (LMA) que consiste em um aperfeiçoamento do método de *Gauss-Newton* que, por sua vez, é uma variante do método de *Newton*.

Quanto ao método de treinamento, o mais utilizado é o algoritmo de retropropagação do erro (*BackPropagation*) que se baseia na regra de aprendizagem por correção de erro.

4
Metodologia Proposta

Este capítulo apresenta a metodologia proposta pelo trabalho. Serão abordados os métodos de identificação de outliers e as técnicas de tratamentos, passando pelas etapas do processo KDD.

O capítulo 3 apresentou a fundamentação teórica que será utilizada neste trabalho. Foram apresentadas todas as fases do processo KDD, dando destaque para os métodos de identificação de *outliers* e as técnicas de tratamento. Neste capítulo será abordada a metodologia proposta pelo trabalho, passando por todas as etapas do processo KDD, enfocando na aplicação e resolução do problema de identificação e tratamento de *outliers*.

Assim, a seção 4.1 descreve o processo KDD aplicado ao problema proposto. Na seção 4.2 são apresentados os métodos de identificação de *outliers* aplicados na identificação de cargas elétricas problemáticas. Já na seção 4.3, por sua vez, são mostrados as técnicas de tratamento utilizadas para a correção dos *outliers* identificados.

4.1 O Processo KDD na prática

Na primeira fase do processo KDD, a seleção dos dados é realizada com base em dias ou horários específicos. Assim, a análise dos dados poderá ser realizada com base em um dia, que como visto contém 1440 medições correspondente aos minutos do dia ou com base em um horário, em que serão analisadas as 365 cargas elétricas, correspondentes ao mesmo horário em todos os dias do ano sendo analisado.

Realizada a seleção dos dados, e partindo para a próxima etapa do processo KDD, tem-se no pré-processamento dos dados uma das etapas mais importantes do processo, já que direciona a atenção para possíveis problemas que exijam

inspeção e correção. Nesta fase, são aplicados os dez métodos para identificação de *outliers* vistos na seção 3.2, além de uma modificação proposta para o algoritmo NL.

Optou-se pela aplicação dos vários métodos, pois não há um único critério uniforme que possa ser usado para determinar que um valor suspeito seja realmente um *outlier* (Lopes, 2003).

Uma vez que diferentes algoritmos de detecção de *outliers* são baseados em conjuntos de dados diferentes, uma comparação direta entre os métodos nem sempre é possível. Em muitos casos, a estrutura de dados e o mecanismo de geração de *outliers* em que o estudo é baseado é que define qual o melhor método a ser aplicado. Assim, conforme Ben-Gal (2005) há poucos trabalhos que comparam diferentes classes de métodos de identificação de *outliers*. Ele cita em seu livro, diversos trabalhos relacionados entre eles do Williams *et al.* (2002), que sugere a utilização de testes estatísticos para identificação de *outliers* para grandes volume de dados.

Bem-Gal (2005) também cita o trabalho de Penny e Jolliffe (2001) que realiza uma comparação entre seis métodos de detecção de *outliers* multivariados. Os métodos são investigados por meio de um estudo de simulação e os resultados indicam que nenhuma técnica é superior às outras. Os autores indicam vários fatores que afetam a eficiência dos métodos analisados, entre eles: a dimensão do conjunto de dados, o tipo de *outliers* e a proporção de *outliers* no conjunto de dados. O estudo motivou os autores a recomendarem o uso de uma "bateria de métodos multivariados" no conjunto de dados a fim de detectar possíveis *outliers*.

Oliveira (2008) propôs uma comparação entre os testes de *Dixon*, *Chauvenet* e de *Grubbs* para identificação de *outliers*. O teste de *Grubbs* se mostrou mais robusto do que os demais, por identificar dois *outliers*, já o teste de *Chauvenet* identificou apenas um, enquanto que o método de *Dixon*, não detectou nenhum valor suspeito.

Assim, para este trabalho adotou-se que uma carga elétrica será definida como *outlier* quando pelo menos quatro testes a identificar. Essa abordagem pode ser vista no fluxograma mostrado na figura 12. Optou-se por esta técnica para uma analise mais detalhada dos dados, pois os testes apresentam abordagens diferentes, como por exemplo, desvio padrão, amplitude, variância, entre outros cálculos

estatísticos, garantindo assim uma avaliação mais crítica dos dados. Ao mesmo tempo, exigir a identificação de *outliers* por muitos testes restringiria muito a identificação, pois há métodos como o teste de Dixon que avalia somente os extremos do conjunto de dados.

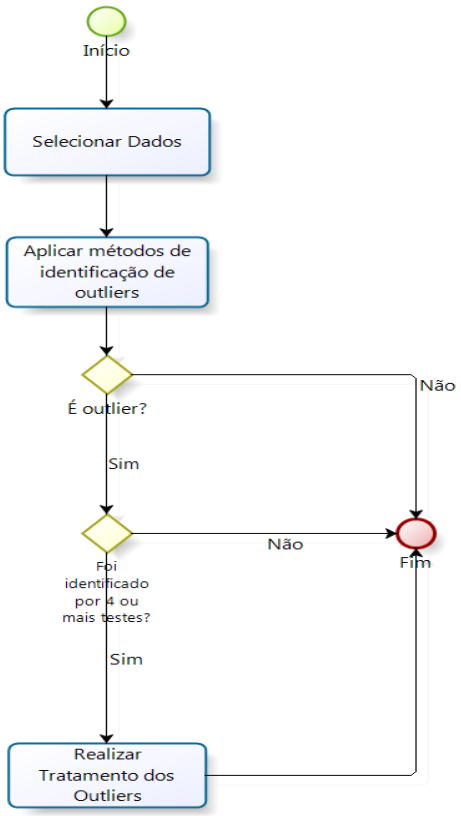


Figura 12 – Fluxograma da metodologia proposta.

Mais informações de como os métodos de identificação de *outliers* foram aplicados ao problema de identificação de cargas elétricas com possíveis erros, e os detalhes de codificação serão discutidos na próxima seção.

Avançando nas fases do processo KDD, na etapa de transformação, os dados serão normalizados em uma escala de zero até um. Essa operação é necessária, pois os dados possuem intervalos de variação diferentes, exigindo assim a normalização para que a rede neural possa tratá-los com a mesma ênfase durante o treinamento.

A fase de Mineração de Dados é considerada uma das etapas mais importantes do processo, pois são processados os algoritmos de aprendizado de

máquina e de reconhecimento de padrões. Serão utilizados a média e as RNAs como ferramentas nesta etapa, que serão descritas na seção 4.3.

Um dos poucos trabalhos encontrados que se aplicam aos dados de energia elétrica é apresentado por Filho (2008) que aborda metodologias computacionais aplicáveis ao problema de validação diária da medição de energia elétrica nos pontos de conexão possibilitando ao agente distribuidor de energia elétrica prevenir erros e/ou problemas no processo de contabilização da energia comercializada. Foi feito um estudo comparativo de desempenho dos métodos baseados em redes neurais com base radial (RN-RBF) e auto-regressivo usando mínimos quadrados (AR MQ).

Por fim, serão comparados os resultados obtidos com os resultados fornecidos por especialistas do setor elétrico, em que os mesmos dados foram tratados de forma manual. Os resultados obtidos nessa etapa do KDD serão apresentados no capítulo 6.

4.2 Métodos de Identificação de *Outliers*

Nesta seção, serão apresentados como os métodos de identificação de *outliers* foram aplicados, além dos detalhes de codificação. Não serão abordados detalhes do funcionamento dos métodos, já que foram discutidos no capitulo anterior. Assim, o objetivo deste capítulo dar uma visão de como os métodos foram implementados no software desenvolvido para identificação de *outliers*.

4.2.1 BoxPlot

Como foi apresentado na seção 3.2.1, o *BoxPlot* é um gráfico que possibilita representar a distribuição de um conjunto de dados e revelar a presença de *outliers*. Assim, ele utilizado como uma ferramenta de identificação de *outliers* para este trabalho.

O trecho de código mostrado no quadro 2 realiza os cálculos das principais medidas utilizadas para este teste. Na identificação de *outliers*, para cada valor

representando uma medição de carga elétrica, é verificado se seu valor é inferior a variável **limite_inferior** ou superior a variável **limite_superior**, o que caracteriza um valor suspeito em caso verdadeiro, como pode ser visto no quadro 3.

Quadro 2 – Cálculo do IQR e limites do *boxplot*.

```
//cálculo do IQR e limites do boxplot;
IQR = quartil75 - quartil25;
limite_inferior = quartil25 - 1.5 * IQR;
limite_superior = quartil75 + 1.5 * IQR;
```

Quadro 3 – Identificando *outliers* pelo *boxplot*.

```
//estrutura de repetição para analisar todos os dados;
for (int i=0; i < dados.size(); i++) {
   //compara se a carga elétrica é inferior ou superior aos
limites inferior e superior, respectivamente;
   if(dados.get(i).getCargaEletrica()<limite_inferior||
dados.get(i).getCargaEletrica()>limite_superior) {
     //em caso verdadeiro, o valor é adicionado aos suspeitos;
     this.suspeitos.add(dados.get(i));
   }
}
```

4.2.2 Teste de *Dixon*

O teste de *Dixon* foi apresentado na seção 3.2.2, e como visto é um teste simples de ser calculado, porém verifica a presença de *outliers* só nos extremos do conjunto de dados, ou seja, só verifica se o menor e maior valor são *outliers*.

Assim, foi implementado como uma ferramenta de auxílio para identificação de *outliers*, e um trecho do código pode ser visto no quadro 4. No código, depois da ordenação dos dados é feita uma chamada ao método **calculaQ()**, que calcula os valores para o menor e maior valor do conjunto de dados, com base nas fórmula apresentadas na tabela 2 apresentado na seção 3.2.2.

Posteriormente, é feito uma comparação dos valores calculados, com o valor crítico tabelado, obtido pelo método **valorTabelado()**. Caso algum dos valores calculados exceda o valor crítico tabelado, este será classificado como suspeitos.

Quadro 4 – Identificando *outliers* pelo teste de Dixon.

```
//ordenando os dados pelo método sort() da classe Collections;
Collections.sort(dados, ordena);
//faz uma chamada a método calculaQ();
this.calculaQ(dados);
//faz uma chamada ao método valorTabelado(), que retorna o
valor crítico do teste de Dixon, e compara com o valor
calculado para a menor carga elétrica do conjunto de dados;
if(this.q_menor > this.valorTabelado())
     this.suspeitos.add(dados.get(0));
//faz uma chamada ao método valorTabelado(), que retorna o
valor crítico do teste de Dixon, e compara com o valor
calculado para a maior carga elétrica do conjunto de dados;
if(this.q_maior > this.valorTabelado())
     this.suspeitos.add(dados.get(dados.size()-1));
```

4.2.3 Teste de Grubbs

No quadro 5, tem-se um trecho de código do teste de *Dixon,* que foi apresentado na seção 3.2.3. As variáveis globais **media** e **desvio padrão** são calculadas para a amostra de dados sendo analisada. Em seguida para cada dado do conjunto, é calculado a sua distância em relação à média. Por fim, para o cálculo da variável **G**, a distância é dividida pelo desvio padrão do conjunto. Após esses cálculos, é feito

um teste, para determinar se o dado sendo analisado será classificado como suspeito ou não.

Quadro 5 – Identificando *outliers* pelo teste de Grubbs.

```
//analisando todos os dados da amostra;
for(int i=0; i < dados.size(); i++){
  //calculando a distância entre a carga e a média dos dados;
  dist = Math.abs(dados.get(i).getCargaEletrica()-this.media);
  //calculando o valor de G;
  G = (double) dist / this.desvioPadrao;
  //faz uma chamada ao método valorTabelado(), que retorna o valor crítico do teste de Grubbs, e compara com o valor da variável G calculada;
  if(G > this.valorTabelado())
      this.suspeitos.add(dados.get(i));
}
```

4.2.4 Teste do Erro

O Teste do Erro como exposto na seção 3.2.4 é um teste similar ao de *Grubbs*, porém não depende dos valores críticos tabelados. Sua implementação consiste em verificar para todos os dados do conjunto, se o valor calculado para o erro ultrapassa aos limites permitidos.

Quadro 6 – Identificando *outliers* pelo teste do Erro.

```
for(int i=0; i<dados.size(); i++){
     //calculando o erro, com base na média e desvio padrão;
     erro = Math.abs(dados.get(i).getCargaEletrica()-this.media)/this.desvioPadrao;
     //chamando o método teste para verificar se é um suspeito;
     if(this.teste(erro, dados.size()))
```

```
            this.suspeitos.add(dados.get(i));
}
```

Assim, o quadro 6 mostra um trecho de código, em que é calculado o valor do erro para todos os dados do conjunto, com base na média e desvio padrão. Posteriormente, é feita a chamada ao método **teste(),** que é mostrado no quadro 7. Este método tem por finalidade verificar se o tamanho do erro calculado é maior que o permitido, com base na quantidade de dados da amostra em análise.

Quadro 7 – Método teste() do teste do Erro.

```
//método para verificar se o erro calculado é maior que o erro
permitido, com base no tamanho do conjunto de dados;
private boolean teste(double erro, int tamanho){
     if(tamanho <= 8){
          if(erro > 6) return true;
           else return false;
     } else if(tamanho <= 14){
           if(erro > 5) return true;
           else return false;
      } else {
           if(erro > 4) return true;
           else return false;
      }
}
```

4.2.5 Teste Z-Score

O teste do *z-score*, como foi apresentado na seção 3.2.5, é calculado com base no *z-score* e no tamanho do conjunto de dados.

Um trecho de código é apresentado no quadro 8. Para cada dado da amostra de dados selecionada, é calculado o valor em *z-scores* correspondente e em seguida, esse valor é passado para um método, identificado como **testeZScore()**, cujo código

é mostrado no quadro 9. Este método recebe como parâmetro, o *zscore* do dado sendo analisado e o tamanho da amostra. Com base nesses dois parâmetros é aplicado um teste para saber se o dado será ou não definido como suspeito de ser um *outlier*.

Quadro 8 – Identificando *outliers* pelo teste de *z-score*.

```
//estrutura de repetição para analisar todos os dados;
for(int i=0; i < dados.size(); i++){
    //calculando o z-score da carga elétrica;
     zscore = (dados.get(i).getCargaEletrica()-media) /
desvio_padrao;
   //faz uma chamada ao método testeZScore(), que retorna um
valor booleano indicando se o dado é ou não um outlier;
   if(this.testeZScore(zscore, dados.size()))
        this.suspeitos.add(dados.get(i));
}
```

Quadro 9 – Identificando *outliers* pelo teste de Grubbs.

```
//método para verificar se o z-score calculado é maior que o
limite permitido, com base no tamanho do conjunto de dados;
private boolean testeZScore(double zscore, int tamanho) {
     if(tamanho < 50){
           if(zscore < -2.5 || zscore > 2.5)
               return true;
           else return false;
     } else if(tamanho < 1000){
           if(z_calculado < -3.3 || z_calculado > 3.3)
               return true;
           else return false;
     } else {
```

```
            if(z_calculado <= -3.3 || z_calculado >= 3.3)
                return true;
            else return false;
      }
}
```

4.2.6 Critério de *Peirce*

No quadro 10, observa-se um trecho da implementação deste método. O método consiste em supor inicialmente que há um único *outlier* na base de dados. Assim, é calculada a distância máxima permitida, com base no desvio padrão e no valor crítico tabelado.

Quadro 10 – Identificando *outliers* pelo critério de *Peirce*.

```
//para cada iteração do teste, é chamado este método. Mx
representa a distância máxima permitida;
max = this.desvioPadrao * this.valorTabelado();
for (int i = 0; i < dados.size(); i++) {
     distancia = (Math.abs(dados.get(i).getCargaEletrica() -
this.media));
     if (distancia > max)
          this.suspeitos.add(dados.get(i));
}
```

Na sequência, para todos os dados é realizado um teste com base na distância máxima permitida já calculada. Caso seja maior, o valor é considerado suspeito. A aplicação do código abaixo envolve vários laços de repetição enquanto novos *outliers* forem sendo identificados, como foi discutido na seção 3.2.6.

4.2.7 Cretério de *Chauvenet*

Como abordado na seção 3.2.7 o critério de *Chauvenet* é uma alternativa simplificada do critério de *Peirce*. O quadro 11 apresenta a implementação

proposta, em que são calculados os limites para além dos quais os dados passam a ser considerados como suspeitos de serem *outliers*. Assim, o laço de repetição analisa todos os dados da amostra, de modo que os valores que ultrapassam os limites inferior e superior são marcados como suspeitos.

Quadro 11 – Identificando *outliers* pelo critério de *Chauvenet*.

```
//calculando os limites para além dos quais os dados passam a
ser considerados outliers;
limite = this.desvioPadrao * this.valorTabelado();
lim_inferior = this.media - limite;
lim_superior = this.media + limite;
for(int i=0; i<dados.size(); i++){
      //comparando cada dado da amostra com os limites;
      if((dados.get(i).getCargaEletrica()<lim_inferior)       ||
(dados.get(i).getCargaEletrica()>lim_superior))
          this.suspeitos.add(dados.get(i))
}
```

4.2.8 Teste de Cochran

O quadro 12 apresenta um trecho da implementação do critério de *Cochran* que foi apresentado na seção 3.2.8, e como pode ser visto, primeiramente é calculado dentro do *loop for*, a variância para cada grupo k e armazenando esse valor para uso posterior, além de calcular também a soma das variâncias de todos os k grupos.

Posteriormente é identificado o grupo que apresentou a maior variância, sendo este, suspeito de conter algum(s) *outlier(s)*. A variância do grupo suspeito é dividida pela soma das variâncias dos k grupos, calculada anteriormente, e armazenado na variável **valor_calculado**.

Por fim, é realizado uma comparação entre o a variável **valor_calculado** e o valor crítico tabelado para o teste, que é calculado pelo método **valorTabelado()**. Se

o **valor_calculado** exceder o valor crítico tabelado, então o grupo é marcado como suspeito de possuir algum(s) *outlier*(s) em relação aos demais grupos.

Quadro 12 – Identificando *outliers* pelo teste de Cochran.

```
//calculando as variâncias dos grupos e soma das variâncias;
for (int i = 0; i < dados.size(); i++) {
      variancia = super.variancia(dados.get(i));
      soma_variancias += variancia;
      variancias.add(variancia);
}
//calculando o grupo com maior variância;
valor_calculado = Collections.max(variancias)/soma_variancias;
int indice = 0;
//localizando o grupo com maior variância;
for(int  i=0;variancias.get(i)!=Collections.max(variancias);i++)
{     indice++;
}
//teste para verificar se o grupo com maior variância
ultrapassa o valor crítico tabelado;
if (valor_calculado > this.valorTabelado()) {
        this.suspeitos.addAll(dados.get(indice));
}
```

4.2.9 Teste da Razão Q

O teste da Razão Q, como apresentado na seção 3.2.9, é um teste baseado na amplitude dos dados. Na implementação, como se verifica no quadro 13, para cada valor de medição elétrica é calculado o valor da variável **Q** com base na diferença entre o valor da carga e de seu vizinho mais próximo, dividido pela amplitude do conjunto de dados. Caso esse valor exceda o valor crítico tabelado, ele será marcado como *outlier*.

Quadro 13 – Identificando *outliers* pelo teste da Razão Q.

```
//ordena os dados;
Collections.sort(dados, ordena);
//calcula a amplitude;
this.amplitude = dados.get(dados.size()-1).getCargaEletrica() -
dados.get(0).getCargaEletrica();
//para todos os dados da amostra é calculado e valor de Q e
comparado com o valor crítico tabelado;
for(int i=0; i<dados.size(); i++){
     Q        =        (dados.get(i).getCargaEletrica()        -
this.vizinhoMaisProximo(dados.get(i),
dados).getCargaEletrica())/this.amplitude;
      if(Math.abs(Q) > this.valorTabelado())
            this.suspeitos.add(dados.get(i));
}
```

4.2.10 Algoritmo NL e NL Modificado

O algoritmo NL é um teste baseado na distância dos dados, e como visto na seção 3.2.10, ele apresenta a desvantagem do fato do algoritmo não ser inteiramente automático, pois a determinação dos parâmetros p e d são tarefa de um especialista humano.

Assim, para contornar esse problema, e com base em testes realizados, foi proposto os valores $p = 0.90$ e d como sendo duas vezes o desvio padrão da amostra de dados sendo analisada. O quadro 14 apresenta o algoritmo utilizado na identificação de *outliers*.

Como visto, o método exige muitas comparações entre os dados, sendo classificado como da $O(n^2)$ em complexidade, já que para cada dado, tem-se que comparar a distância com todos os demais dados do conjunto.

Quadro 14 – Identificando *outliers* pelo algoritmo NL.

```
//para todos os dados do conjunto é verificada a distância em
relação aos demais dados;
for (int i = 0; i < tamanhoAmostra; i++) {
     contador = 0;
     for (int j = 0; j < tamanhoAmostra; j++) {
         if(this.distancia(dados.get(i),dados.get(j))<= this.D)
                contador++;
     }
     //se o contador for menor ou igual a M, o valor será
considerado suspeito;
     if(contador <= this.M)
         this.suspeitos.add(dados.get(i));
}
```

Assim, levando em consideração este fato, foi proposto um algoritmo simples, em que são comparadas apenas as cargas anterior e posterior para todos os dados do conjunto. Caso, as cargas anterior e posterior diferenciem da carga sendo analisada em um limite definido como $2 \times s$, sendo s o desvio padrão dos dados, o valor será marcado como suspeito. Há exceção para o primeiro dado do conjunto que será verificado somente a carga posterior e para o último, em que só será verificada a carga anterior. O código é mostrado no quadro 15, e apresentado no quadro 16 o método para calcular a diferença entre as cargas elétricas.

Quadro 15 – Identificando *outliers* pelo algoritmo proposto.

```
//calcula o limite para os dados não serem considerados como
outliers;
limite = 2*super.desviopadrao(dados);
for (int i = 0; i < tamanhoAmostra; i++) {
     atual = dados.get(i);
     //caso seja o primeiro elemento do conjunto;
     if(i==0) anterior = null;
```

```
      else anterior = dados.get(i - 1);
      //caso seja o último elemento do conjunto;
      if(i==tamanhoAmostra-1) proximo = null;
      else proximo = dados.get(i + 1);
      //calcula a distância e compara com o limite;
      if((this.distancia(atual, anterior) > limite) &&
(this.distancia(atual, proximo) > limite)){
            this.suspeitos.add(atual);
       }
}
```

Quadro 16 – Método distancia do algoritmo proposto.

```
//método distancia() utilizado no algoritmo;
public double distancia(Dados dado1, Dados dado2) {
      double distancia;
      if(dado1 == null || dado2 == null)
            distancia = 0;
      else {
            distancia = Math.abs(dado1.getCargaEletrica() -
dado2.getCargaEletrica());
         return distancia;
     }
```

4.3 Técnicas de Tratamento Utilizadas

Como já discutido neste trabalho, foram utilizados o operador Média Aritmética e modelos baseados em Redes Neurais Artificiais como ferramentas para tratamento

dos dados inconsistentes identificados na fase anterior. Na sequência são abordados como estas foram aplicadas ao problema de tratamento de dados.

4.3.1 Média

O valor encontrado pelo operador média aritmética foi estimado com base em três semanas anteriores ao *outlier* identificado. Assim, a média é calculada obtendo três dias anteriores correspondentes ao mesmo dia da semana do *outlier*.

Contudo, foi acrescentando um teste para verificar se algum dos dias anteriores não são *outliers*, já que um possível *outlier* seria considerado um valor extremo e afetaria significativamente o valor da média.

4.3.2 Redes Neurais Artificiais

Para o tratamento dos *outliers,* foram utilizadas as três implementações de RNAs existentes na literatura: *Neuroph* (Neuroph, 2008), *FeedForward* (Heaton, 2008) e *Encog* (Encog, 2008).

A base de treinamento para as RNAs utilizadas neste trabalho consiste de dez dias próximos ao *outlier* identificado. Optou-se por trabalhar com dados mais recentes, pois estes trazem informações importantes sobre as tendências da carga, fortemente relacionadas com as variações nas condições climáticas (Salgado, 2009).

Os testes de treinamento indicaram uma configuração de três neurônios de entrada, uma camada interna com três neurônios e uma saída, como adequada para todas as redes neurais utilizadas. As demais configurações e parâmetros serão especificadas nas próximas seções para cada RNA.

4.3.2.1 Neuroph

O *Neuroph* é um *framework* gratuito desenvolvido em *Java* muito utilizado para a criação e treinamento de RNAs. Apresenta um pequeno número de classes em código aberto que correspondem aos conceitos básicos, tornando-se intuitiva e fácil de aprender. Também dispõe de uma interface gráfica que facilita a tarefa de

criação e treinamento de uma RNA. Mais informações sobre o projeto *Neuroph* podem ser obtidas em *Neuroph* (2008).

Neste trabalho, optou pela utilização do *Neuroph* com a criação de uma RNA MLP, com três neurônicos de entrada, três neurônios na camada intermediária e um neurônio de saída, como ilustra a figura 13, além do *bias*. A função de ativação escolhida foi a tangente hiperbólica.

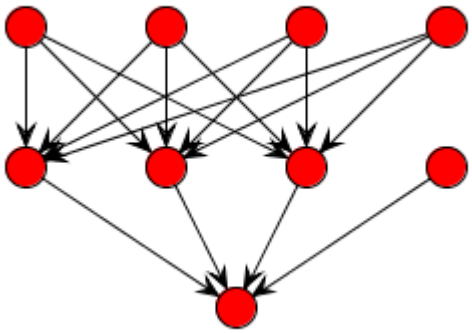


Figura 13 – Estrutura da RNA.

Foi empregada a técnica de treinamento supervisionado *BackPropagation*, que como apresentado no capítulo 4, é uma técnica de propagação do erro. Foi utilizado para o treinamento, 0,0001% de erro, ou em caso de convergência, um limite máximo de cinco mil iterações como critério de parada. O quadro 16 apresenta um fragmento de código, definindo as principais configurações da RNA.

Quadro 17 – Código *Neuroph*.

```
//criando uma RNA MLP com 3 neurônios de entrada, uma camada interna de 3 neurônios e uma saída;
MultiLayerPerceptron redeneural = new MultiLayerPerceptron(TransferFunctionType.TANH, 3, 3, 1);
//criando arquivo de treinamento, com 3 entradas e uma saída;
TrainingSet trainingSet = trainingSet = new TrainingSet(3, 1);
//A técnica de treinamento é uma variação Neuroph para o Backpropagation;
```

```
DynamicBackPropagation train = new DynamicBackPropagation();
train.setNeuralNetwork(redeneural);
redeneural.setLearningRule(train);
```

4.3.2.2 *Encog*

O *Encog* é uma *framework* gratuito disponível em *Java*, *.Net* e *Silverlight*, para a criação de RNAs. É um framework mais completo, com várias técnicas mais avançadas de treinamento em relação ao *Neuroph*, além de ser bem mais documentado.

Foi criada uma RNA com três neurônicos de entrada, três neurônios na camada intermediária e um neurônio de saída. A função de ativação escolhida foi a tangente hiperbólica.

Possui uma grande variedade de técnicas de treinamento para a rede, na qual se optou pela técnica LMA, que como visto, é um aperfeiçoamento do método de *Gauss-Newton*. O treinamento foi realizado com 0,0001% de erro, ou em caso de convergência, um limite máximo de cinco mil iterações como critério de parada. O quadro 18 representa um fragmento de código, com as configurações da RNA.

Quadro 18 - Código *Encog*.

```
//criando uma RNA com 3 neurônios de entrada, uma camada interna de 3 neurônios e uma saída;
BasicNetwork redeneural = new BasicNetwork();
redeneural.addLayer(new BasicLayer(new ActivationTANH(), true, 3));
redeneural.addLayer(new BasicLayer(new ActivationTANH, true, 3));
redeneural.addLayer(new BasicLayer(new ActivationTANH, true, 1));
//utiliza a lógica feedforward;
redeneural.setLogic(new FeedforwardLogic());
```

```
//Finaliza a estrutura e reseta os pesos de forma aleatória;
redeneural.getStructure().finalizeStructure();
redeneural.reset();
//criando o arquivo de treinamento;
NeuralDataSet trainingSet = new BasicNeuralDataSet(entradas,
saida);
// A técnica de treinamento é a LMA;
Train train = new LevenbergMarquardtTraining(redeneural,
trainingSet);
```

4.3.2.3 *FeedForward*

Uma rede neural *feedforward* é uma rede em que os neurônios estão ligados somente à camada seguinte. Não há conexões entre os neurônios nas camadas anteriores ou entre os neurônios e eles próprios. Além disso, os neurônios não estão conectados aos neurônios para além da camada seguinte. Como um padrão é processado por um design *feedforward*, os limiares e pesos de conexão serão aplicados.

Neste trabalho, optou pela utilização da *FeedForward* com a criação de uma RNA com múltiplas camadas, com três neurônicos de entrada, três neurônios na camada intermediária e um neurônio de saída. A função de ativação escolhida foi a tangente hiperbólica.

Foi empregada a técnica de treinamento supervisionado *BackPropagation* e para o treinamento, 0,0001% de erro, ou em caso de convergência, um limite máximo de cinco mil iterações como critério de parada. O quadro 19 apresenta um fragmento de código, definindo as principais configurações da RNA.

Quadro 19 – Código: FeedForward.

```
//criando uma RNA multi camadas com 3 neurônios de entrada, uma
camada interna de 3 neurônios e uma saída;
FeedforwardNetwork redeneural = new FeedforwardNetwork();
redeneural.addLayer(new FeedforwardLayer(new ActivationTANH(),
3));
```

```
redeneural.addLayer(new FeedforwardLayer(new ActivationTANH (), 3));
redeneural.addLayer(new FeedforwardLayer(new ActivationTANH (), 1));
redeneural.reset();
//configurando treinamento para Backpropagation;
Train train = new Backpropagation(redeneural, entradas, saida, 0.7, 0.9);
```

5
Sistema de Suporte

Este capítulo apresenta uma visão do sistema desenvolvido para identificação e tratamento de outliers, como parte deste trabalho, intitulado OUTLES.

Este capítulo apresenta o sistema computacional OUTLES, desenvolvido neste trabalho. Será feita uma descrição de suas funcionalidades, destacando a sua contribuição no processo de identificação e tratamento de *outliers.*

São apresentados, via interface do OUTLES, visualização dos dados, geração de gráficos da base de dados, identificação de *outliers*, gráficos com sinalização dos *outliers* identificados, tratamento dos *outliers*, e por fim, o gráfico com os *outliers* tratados. Vale ressaltar que o objetivo deste capítulo é dar uma visão do software OUTLES na solução do problema de identificação e tratamento de *outliers* em cargas elétrica. Assim, não são realizadas apresentações detalhadas dos botões, menus e outros itens que compõem o sistema.

5.1 Descrição Introdutória

Os procedimentos de análise e tratamento de dados geralmente envolvem diversas variáveis e parâmetros. Uma boa compreensão dos dados depende da disponibilidade de ferramentas e modelos que possibilitem o estudo detalhado da série (Salgado, 2009).

Assim, com o uso de uma ferramenta computacional adequada é possível realizar a análise da consistência dos dados, devido à inviabilidade de serem analisadas e validadas por especialistas humanos, pelo fato da grande quantidade de dados produzidos.

Além disso, com um sistema adequado, é possível ter uma boa visão dos dados direcionando a atenção para possíveis problemas que exijam inspeção e correção, além de fornecer informações objetivas para a tomada de decisão.

Portanto, para auxiliar na análise e tratamento dos dados de origem do setor elétrico, este trabalho propõe um sistema computacional de suporte denominado OUTLES, implementado para facilitar na identificação e tratamento de *outliers*, além de proporcionar a visualização dos dados de forma intuitiva.

5.2 Características Técnicas

O OUTLES foi implementado utilizando a linguagem de programação *Java*. Esta escolha fundamentou-se nas seguintes razões:

- É uma linguagem gratuita;
- Arquitetura orientada a objetos, tornando o código flexível e de fácil manutenção;
- Portabilidade entre diversas plataformas, sem a necessidade de alteração do código da aplicação.
- Reusabilidade: permite reuso de código já produzido, evitando retrabalho e principalmente, dando mais qualidade ao trabalho.

Para o desenvolvimento do sistema, foi utilizado o ambiente de desenvolvimento integrado *NetBeans* IDE. O ambiente *NetBeans* foi escolhido por apresentar inúmeras facilidades integradas, com destaque para a criação de janelas gráficas interativas e bem definidas.

5.3 O Sistema OUTLES

O OUTLES representado pelo logotipo (OUTLES) é um sistema de identificação e tratamento de *outliers* em dados de medição de energia elétrica. O objetivo

principal é dar suporte na realização da análise e tratamento dos dados de demanda elétrica.

Apresenta uma interface de fácil uso, amigável e flexível, permitindo ao usuário comodidade ao utilizá-lo. Um recurso disponível é a geração dos gráficos de forma simples, com diversas opções para visualização, como por dia, horário ou um período especificado pelo próprio usuário.

Os dados a serem analisados pelo OUTLES são armazenados em um banco de dados, garantindo mais confiabilidade aos dados.

5.4 Banco de Dados

A figura 11 apresenta a modelagem das tabelas do banco de dados. Foi utilizado o *software Microsoft SQL Server* como ferramenta de banco de dados do projeto.

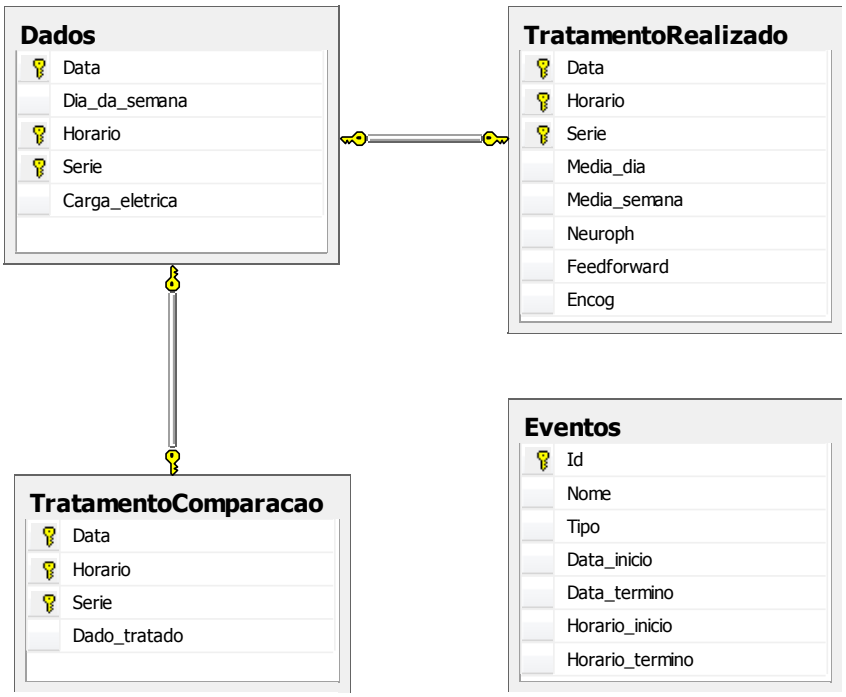


Figura 14 – Modelagem do Banco de Dados.

A seguir são descritas as tabelas do banco de dados:

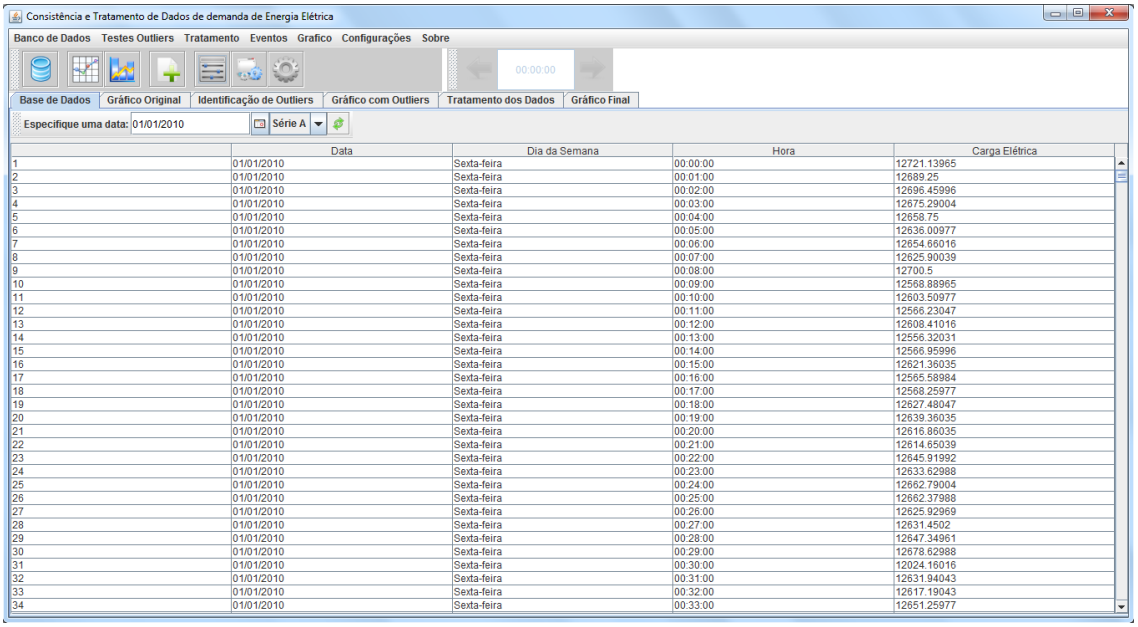
- Dados: armazena os dados a serem analisados pelo software.
- TratamentoRealizado: contém os dados que foram identificados como *outliers* e tratados pelo OUTLES.
- TratamentoComparacao: contém os dados das séries tratados manualmente por especialista do setor elétrico.
- Eventos: contém o cadastro de feriados, dias após feriados, emenda de feriados, dias de eventos especiais, como a copa do mundo e dias de início e fim de horário de verão. Esses dias em específico apresentam um comportamento fora do padrão em relação aos demais dias, que podem levar o software a caracterizá-los como *outliers*, porém são valores reais, que não apresentam erros em suas medições.

Para a comunicação do banco de dados com a aplicação, foi utilizado o *framework JPA EclipseLink* 2.0. Esta escolha baseia-se nas facilidades oferecidas por essa ferramenta.

5.5 Funcionamento do OUTLES

Ao iniciar o OUTLES, deve-se estabelecer a conexão com o banco de dados, que carregará por padrão, os dados do primeiro dia da base de dados, na primeira aba do programa titulada como **Base de Dados**. Para as bases de dados utilizadas neste trabalho, o primeiro dia corresponde a 01/01/2010. Essa etapa é mostrada na figura 15.

Depois de estabelecida a conexão com o banco de dados e carregado os valores iniciais para a tela do sistema, o usuário pode selecionar outro dia qualquer para visualização dos dados, que mostrará informações do valor da carga elétrica, o dia e horário de medição, além do dia da semana correspondente.



	Data	Dia da Semana	Hora	Carga Elétrica
1	01/01/2010	Sexta-feira	00:00:00	12721.13965
2	01/01/2010	Sexta-feira	00:01:00	12689.25
3	01/01/2010	Sexta-feira	00:02:00	12696.45996
4	01/01/2010	Sexta-feira	00:03:00	12675.29004
5	01/01/2010	Sexta-feira	00:04:00	12658.75
6	01/01/2010	Sexta-feira	00:05:00	12636.00977
7	01/01/2010	Sexta-feira	00:06:00	12654.66016
8	01/01/2010	Sexta-feira	00:07:00	12625.90039
9	01/01/2010	Sexta-feira	00:08:00	12700.5
10	01/01/2010	Sexta-feira	00:09:00	12568.88965
11	01/01/2010	Sexta-feira	00:10:00	12603.50977
12	01/01/2010	Sexta-feira	00:11:00	12566.23047
13	01/01/2010	Sexta-feira	00:12:00	12608.41016
14	01/01/2010	Sexta-feira	00:13:00	12556.32031
15	01/01/2010	Sexta-feira	00:14:00	12566.95996
16	01/01/2010	Sexta-feira	00:15:00	12621.36035
17	01/01/2010	Sexta-feira	00:16:00	12565.58984
18	01/01/2010	Sexta-feira	00:17:00	12568.25977
19	01/01/2010	Sexta-feira	00:18:00	12627.48047
20	01/01/2010	Sexta-feira	00:19:00	12639.36035
21	01/01/2010	Sexta-feira	00:20:00	12616.86035
22	01/01/2010	Sexta-feira	00:21:00	12614.65039
23	01/01/2010	Sexta-feira	00:22:00	12645.91992
24	01/01/2010	Sexta-feira	00:23:00	12633.62988
25	01/01/2010	Sexta-feira	00:24:00	12662.79004
26	01/01/2010	Sexta-feira	00:25:00	12662.37988
27	01/01/2010	Sexta-feira	00:26:00	12625.92969
28	01/01/2010	Sexta-feira	00:27:00	12631.4502
29	01/01/2010	Sexta-feira	00:28:00	12647.34961
30	01/01/2010	Sexta-feira	00:29:00	12678.62988
31	01/01/2010	Sexta-feira	00:30:00	12024.16016
32	01/01/2010	Sexta-feira	00:31:00	12631.94043
33	01/01/2010	Sexta-feira	00:32:00	12617.19043
34	01/01/2010	Sexta-feira	00:33:00	12651.25977

Figura 15 – Tela do sistema: Base de Dados.

Na segunda aba do programa, identificada como **Gráfico Original**, o usuário tem a opção de visualizar os gráficos das séries de cargas elétricas armazenadas no banco de dados. Tem-se a opção de visualização por dia, minuto ou período específico. Um exemplo é ilustrado na figura 16, para a opção **Minuto** 05h00m.

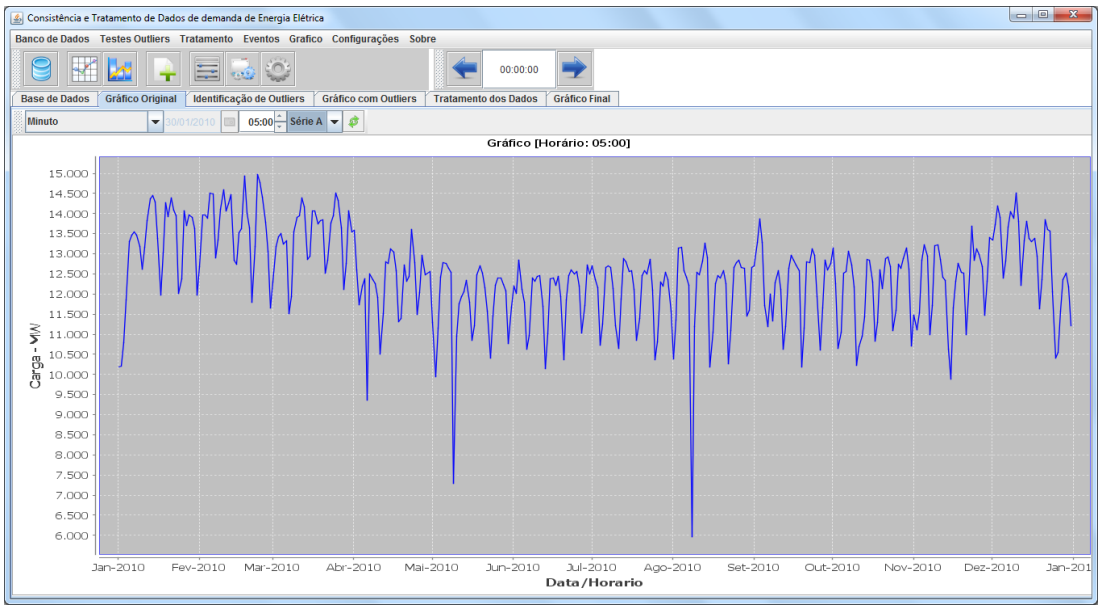


Figura 16 – Tela do sistema: Gráfico Original.

Já na terceira aba do programa, como nome **Identificação de *Outliers*** como mostra a figura 17, o usuário pode aplicar os métodos para analisar a consistência dos dados selecionando os parâmetros desejados. Assim, é possível a identificação de *outliers* por dia, horário ou até mesmo um período específico. É exibido na tela, um relatório, com os valores que foram classificados como *outliers* e também os valores suspeitos de conterem algum erro no processo de medição.

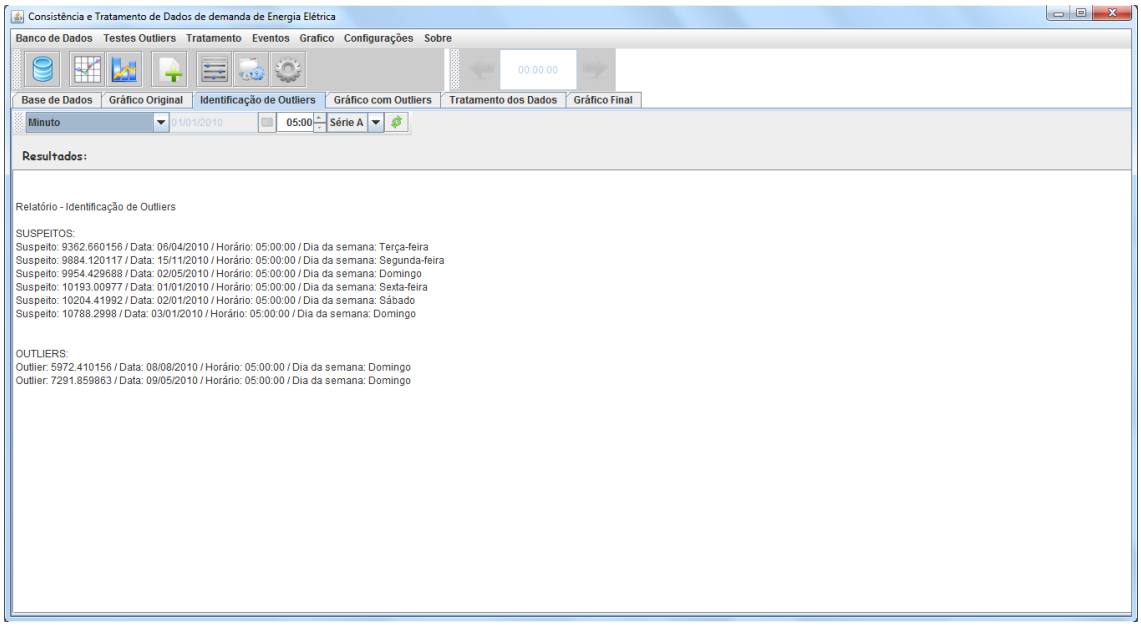


Figura 17 – Tela do sistema: Identificação de *Outliers*.

Executada a análise dos dados, na aba **Gráfico com Outliers**, é mostrado o gráfico com marcações dos valores suspeitos e dos *outliers* identificados na etapa anterior do programa. A figura 18 apresenta o gráfico com os valores suspeitos e *outliers* identificados na etapa anterior.

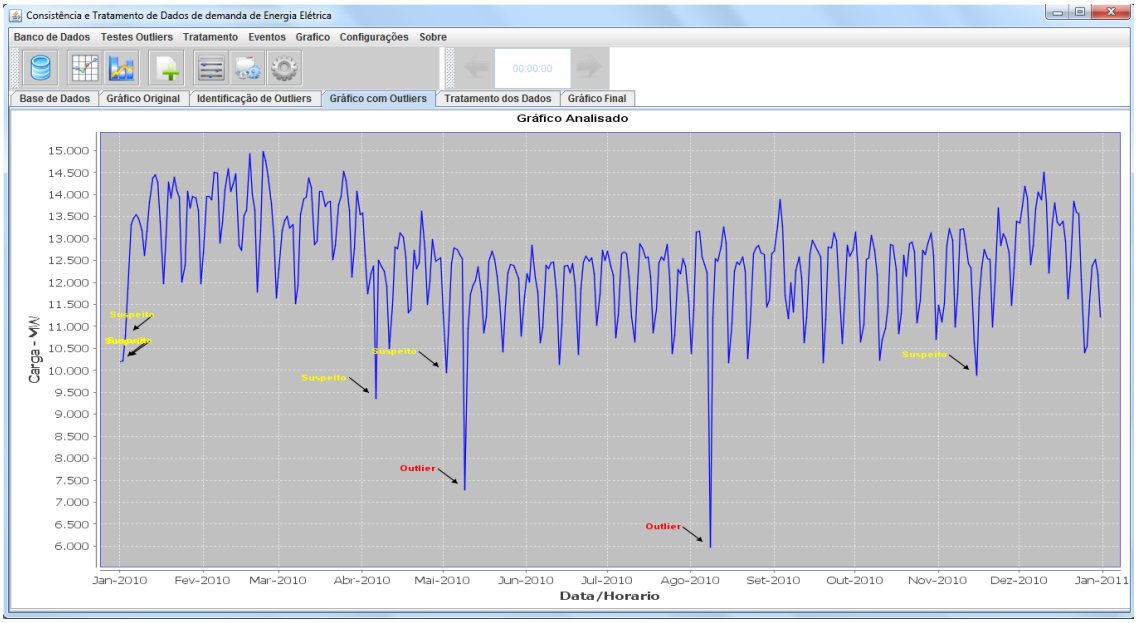


Figura 18 – Tela do sistema: Gráfico Analisado.

Na aba **Tratamento dos Dados**, é mostrada uma tabela com os *outliers* e os valores tratados com os métodos utilizados no software, que foram discutidos nos capitulos 4 e 5. Como pode-se observar na figura 19, são mostrados os valores tratados para a média baseada nas três semanas anteriores e os valores para as três redes neurais utilizadas neste trabalho, que são a *Neuroph*, a *FeedForward* e a *Encog*.

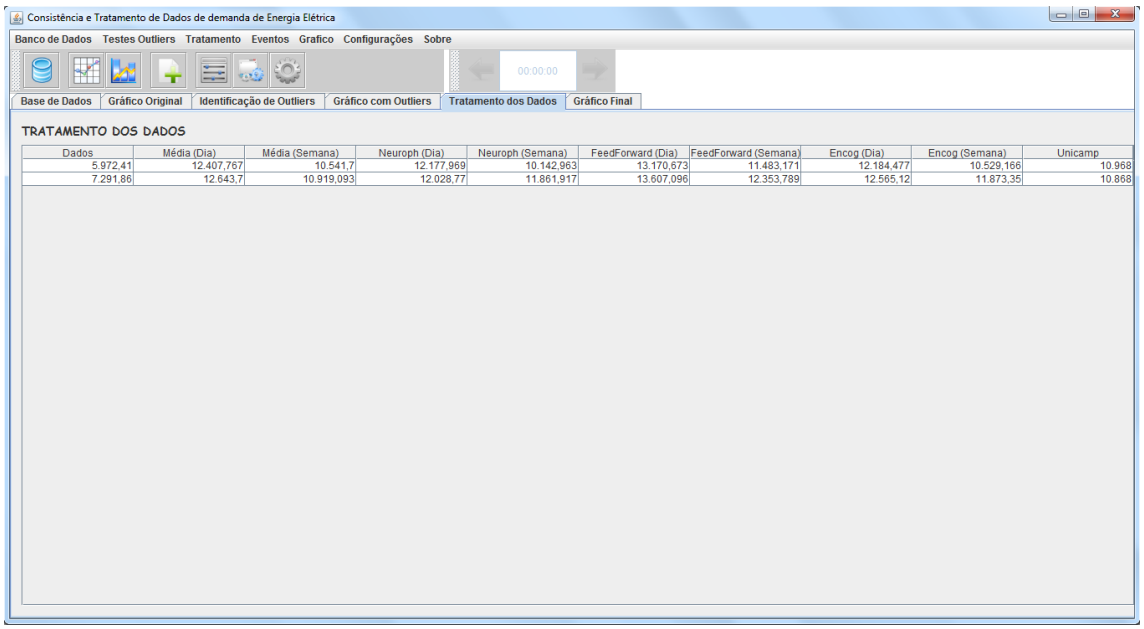


Figura 19 – Tela do sistema: Tratamento dos dados.

Por fim, na aba **Gráfico Final**, é mostrado o gráfico após o tratamento dos valores inconsistentes. A figura 20 apresenta o mesmo gráfico mostrado nas figuras 16 e 17, porém com os *outliers* devidamente corrigidos.

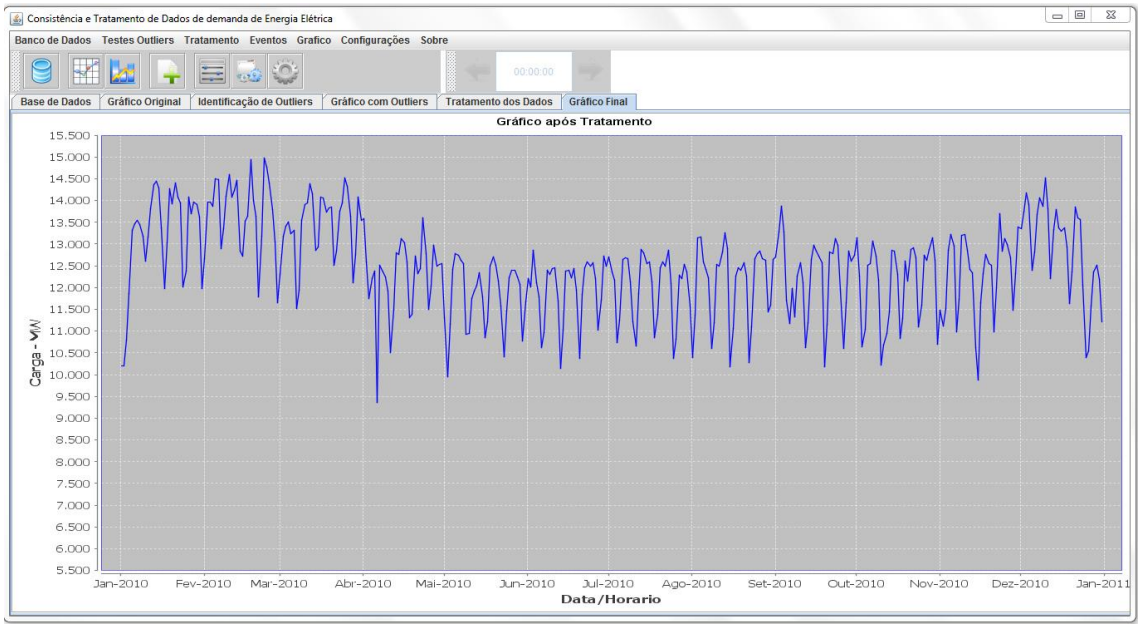


Figura 20 – Tela do sistema: Gráfico Tratado.

Além das funcionalidades apresentadas, o *software* OUTLES também apresenta algumas funcionalidades adicionais, como cadastro de eventos e configurações, que são descritas na próxima seção.

5.6 Funcionalidades Especiais

Como visto no capítulo 3, alguns dos testes para identificação de *outliers* apresentados, envolvem a comparação de um valor calculado com um valor crítico tabelado. Assim, o OUTLES disponibiliza uma janela gráfica onde é possível selecionar o nível de confiança para o valor crítico do teste sendo aplicado. Isso é ilustrado na figura 21.

O teste de *Dixon*, apresentado na seção 3.2.2, é aplicado para amostra de dados de tamanho máximo igual a 25. Assim, para a análise de *outliers* em um dia que contém 1440 minutos, este teste não pode ser aplicado. No entanto, dividindo

os 1440 minutos em 60 conjuntos com 24 minutos cada é possível a aplicação do teste por etapas. Assim, seja por exigência do teste ou mesmo por exigência do usuário em aplicar um determinado teste numa amostra reduzida de dados, o OUTLIES contém uma tela de configurações do tamanho máximo da amostra de dados a ser aplicada para cada teste em específico, conforme ilustra a figura 22.

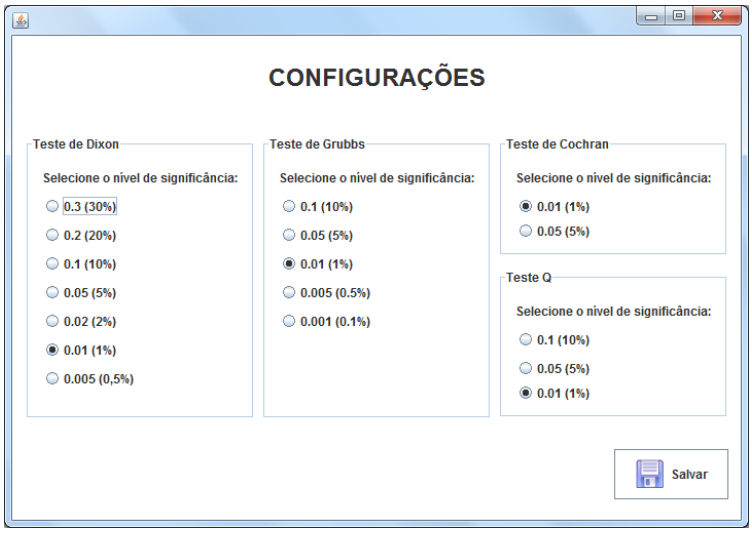


Figura 21 – Tela do sistema: Configurações valores críticos.

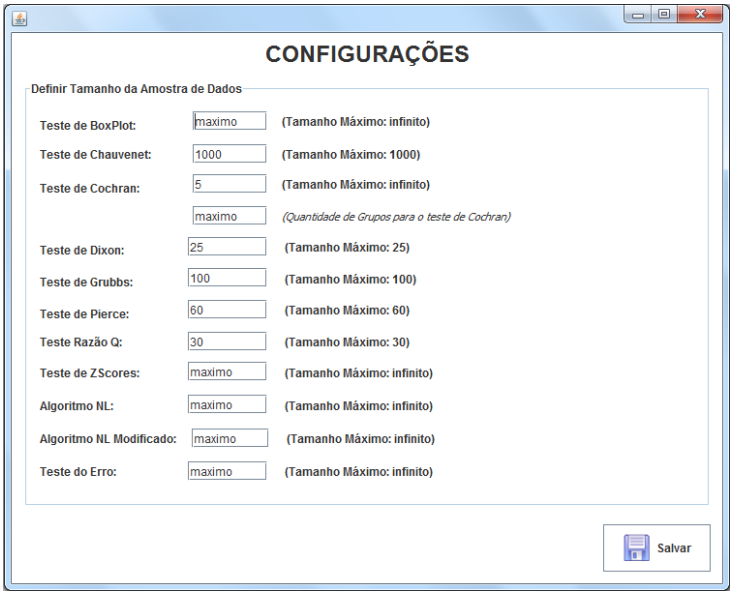


Figura 22 – Tela do sistema: Configurações de tamanho da amostra.

Outra configuração muito útil é permitir ao próprio usuário selecionar o número mínimo de testes para considerar um valor como *outlier*. Lembrando que,

caso o valor escolhido seja 4, como mostra a figura 23, uma determinada carga elétrica será considerada como *outlier*, somente se pelo menos 4 testes a identificar. Caso contrário, será considerada apenas como suspeito.

Assim, quanto maior o número de testes, maior será o nível de exigência, porém não é recomendado o uso de apenas um ou dois testes, pois os testes têm enfoques diferentes sobre os dados. Como exemplo, tem-se o teste de *Dixon* que verifica a presença de *outliers* apenas para o menor e maior valor da amostra de dados sendo analisada.

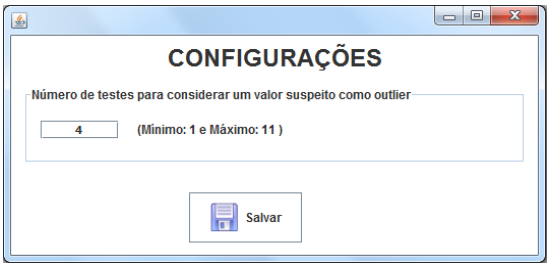


Figura 23 – Tela do sistema: Configurações *outliers*.

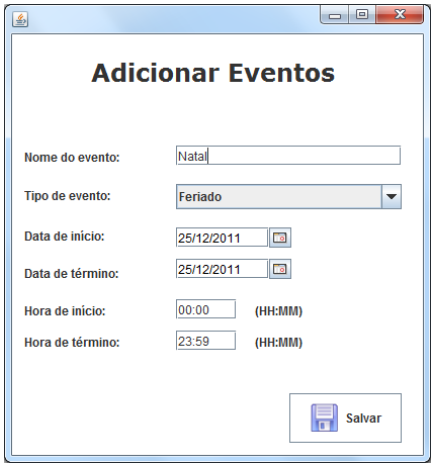


Figura 24 – Tela do sistema: Cadastro de Eventos.

Por fim, a última funcionalidade apresentada é mostrada na figura 24. Trata-se de uma tela de cadastro de eventos, que armazena os valores cadastrados diretamente no banco de dados. É útil para cadastrar feriados, dias após feriados, emendas de feriados, eventos especiais, como a Copa do Mundo, por exemplo, que constituem dias ou horários em que a análise dos dados deve ser diferenciada, pois o valor da medição da carga elétrica terá sofrido um impacto por conta desses eventos.

6
Resultados

Este capítulo apresenta os resultados obtidos pela aplicação dos métodos de identificação e tratamento de outliers.

Este capítulo apresenta os resultados obtidos da aplicação dos métodos de identificação e tratamento de *outliers* discutidos nos capítulos 3 e 4. Assim, na seção 6.1 são descritos os resultados obtidos na identificação de *outliers*, enquanto que na seção 6.2 são apresentados os tratamentos obtidos.

6.1 Resultados obtidos na identificação de Outliers

Nesta seção serão apresentados os resultados obtidos pelos testes de identificação de *outliers*. Devido a grande quantidade de dados, optou-se por analisar a série A por dia e a análise da série B será realizada por minutos.

Os resultados obtidos serão comparados com os resultados dos dados tratados manualmente por especialistas do setor elétrico, verificando quais dados são realmente *outliers*, e quais são falsos positivos.

Assim, a tabela 3 apresenta o total de *outliers* identificados pelos testes, realizando a análise por dia na série A e por minuto na série B.

Tabela 3 – Resultados: total de *outliers* identificados.

	Total de outliers identificados	*Total de falsos positivos identificados*
Série A	130	5
Série B	1562	107

Como se pode observar, apesar das duas séries apresentarem comportamentos diferentes, a análise por minutos é mais rigorosa, pois evita que certos cálculos estatísticos, como a média e o desvio padrão, sejam influenciados por ela.

Com base no total de *outliers* identificados, a tabela 4 mostra a porcentagem da identificação de *outliers* para cada método aplicado, sendo analisados por dia na série A e por minutos na série B.

Tabela 4 – Resultados: métodos de identificação de *outliers*.

Outliers identificados (%)	*Série A*	*Série B*
Algoritmo NL	56%	100%
Algoritmo NL Modificado	25%	78%
BoxPlot	60%	95%
Critério de *Chauvene*t	54%	73%
Critério de *Peirce*	86%	77,%
Teste de *Cochran*	77%	80,%
Teste de *Dixon*	41%	36%
Teste de *Grubbs*	86%	65%
Teste da Razão Q	32%	18%
Teste do Erro	32%	26%
Teste *Z-Score*	55%	59%

Os resultados da tabela estão ilustrados nos gráficos das figuras 25 e 26, para as séries A e B, respectivamente. Como é mostrado nos gráficos, os testes de *Dixon*, Razão Q e Teste do Erro são os que identificaram uma quantidade muito inferior de *outliers* em relação aos demais testes, nas duas bases de dados.

Já o Algoritmo NL teve um acerto de 100% na identificação da série B, enquanto que na série A, apenas 56%, o que caracteriza que a forma de análise influência muito nos resultados. Isso pode ser confirmado com o Algoritmo NL

Modificado e *boxplot* que tiveram um desempenho superior na série B, analisados por minutos.

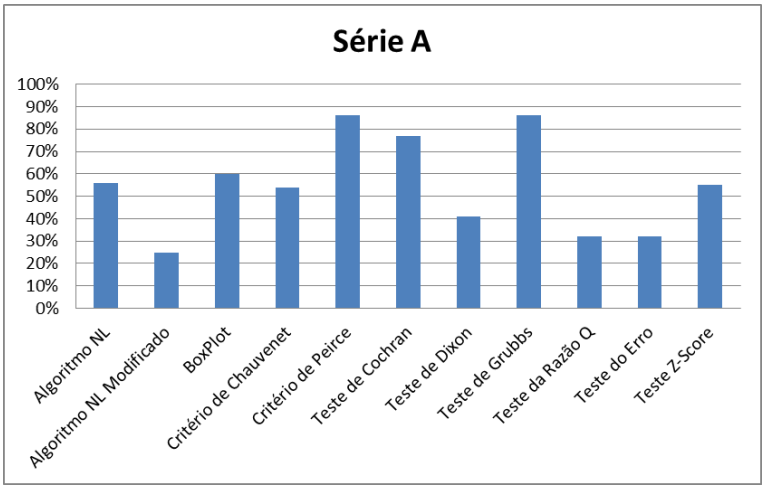


Figura 25 – Gráfico resultados série A.

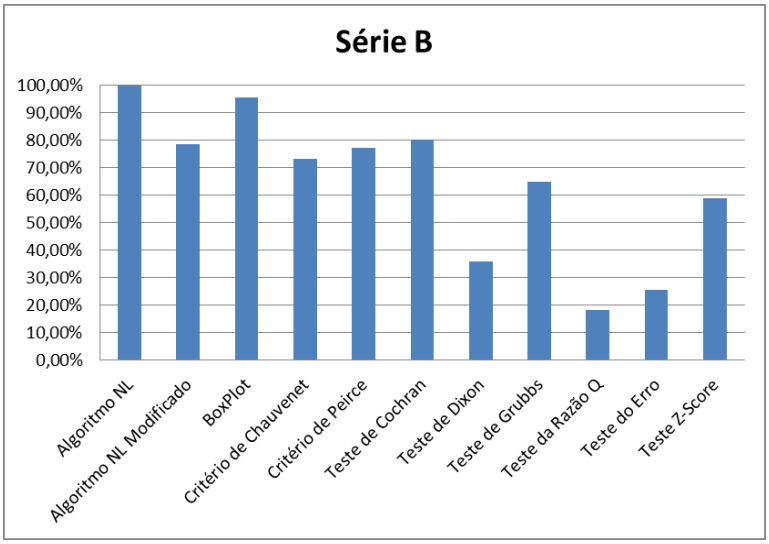


Figura 26 – Gráfico resultados série B.

6.2 Resultados dos tratamentos

Com base nos *outliers* identificados, o tratamento foi realizado com a média e as RNAs, sendo estas a *Neuroph*, *FeedForward* e *Encog*. Para tornar possível e mais compreensível a comparação dos resultados foi adotado o Erro Médio Relativo

(EMR) para avaliar os resultados. O EMR é calculado com base na equação 19, em que x_i representa o valor tratado manualmente por especialistas do setor elétrico e $\widetilde{x}_i$ representa o valor tratado calculado para cada uma das formas de tratamento empregadas neste trabalho.

$$EMR = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \widetilde{x}_i}{x_i}\right| \quad (19)$$

Sendo assim, a tabela 5 apresenta o EMR dos métodos de tratamento utilizados para a série A, dos *outliers* identificados pela análise por dia.

Tabela 5 – Resultados: tratamento por Dia - série A.

Resultados	*Média*	*Neuroph*	*FeedForward*	*Encog*
EMR	5,00%	6,27%	8,59%	7,04%

Já a tabela 6 mostra o EMR dos métodos de tratamento utilizados para a série B, dos *outliers* identificados pela análise por minutos.

Tabela 6 – Resultados: tratamento por Minutos - série B.

Resultados	*Média*	*Neuroph*	*FeedForward*	*Encog*
EMR	1,99%	2,62%	4,19%	8,13%

Como mostra as tabelas, a média teve um desempenho um pouco superior as RNAs. Esse fato é explicado justamente pelas cargas elétricas apresentarem comportamentos bem similares nos dias próximos. Entre as RNAs, a *Neuroph* obteve destaque, porém a *Encog* apresenta uma taxa de treinamento muito superior as demais RNAs utilizadas neste trabalho.

7
Referências Bibliográficas

Alfassi, Z. B., Borger, Z. & Ronen, Y. *Statistical Treatment of Analytical Data*. USA and Canada: CRC Press LLC, 2005. 273 p.

Alvarez, A. L. *Uso Racional de Energia Elétrica: Metodologia para a Detereminação de Potênciais de Conservação dos Usos Finais em Instalações de Ensino e Similares*. Dissertação (Mestrado em Engenharia Elétrica) - EPUSP. São Paulo, SP, 1998.

Amo, S. *Um Curso de Data Mining.* 2008. Disponível em: http://www.deamo.prof.ufu.br/CursoDM2008.html#NotasAula. Acessado em 15-mar-2011.

Barnett, V. & Lewis, T. *Outliers in Statistical Data*. John Wiley & Sons, 3nd edition, 1994.

Ben-Gal, I. *Outlier detection, In: Maimon O. and Rockach L. (Eds.) Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers.* Kluwer Academic Publishers, 2005.

Braga, L. P. V. *Introdução à Mineração de Dados*. 2ª edição revista e ampliada. Rio de Janeiro: *E-Papers* Serviços Editoriais, 2005. 212 p.

Conceição, G. M. S., Alencar, A. P. & Alencar, G.P. *Noções Básicas de Estatística.* Disponível em: http://portal.saude.gov.br/portal/arquivos/pdf/apostila_estatistica.pdf. Acesso em: 30-set-2010.

Ellison, S. L. R., Barwick, V. J. & Farrant, T. J. D. *Practical Statistics for the Analytical Scientist*. A Bench Guide. 2nd Edition, 2009.

Encog, 2008. *Encog Java and DotNet Neural Network Framework.* Disponível em: http://www.heatonresearch.com/encog. Acessado em: 25-abr-2010.

Filho, S. L. C. *Análise de Métodos para validação de medições de energia elétrica*. Dissertação de Mestrado. Universidade Federal do Paraná – UFPR. Curitiba, 2008.

Guirelli, C. R. *Previsão da Carga de Curto Prazo de Áreas Elétricas através de Técnicas de Inteligência Artificial*. Cleber Roberto Guirelli – ed.rev. - São Paulo, 2006. 127p.

Gujer, W. *Systems Analysis for Water Technology*. Springer, 2008. 462 p.
Hawkins, D. *Identification of outliers*. Chapman & HaIl, London, 1980.

Haykin, S. *Redes Neurais: Princípios e Práticas*. 2007. Editora Bookman, 2 ed.

Heaton, J. *Introduction to Neural Networks with Java.* Second Edition, Heaton Research, 2008.

Kanji, G. K. *100 Statistical Test.* 2006. 3 ed. Sage Publications. 527 p.

Libralon, G. L. *Investigação de Combinações de Técnicas de Detecção de Ruído para Dados de Expressão Gênica*. Dissertação de Mestrado. USP – São Carlos, 2007.

Lopes, A. L. *Estatística Aplicada à Análise de Resultados de Ensaios de Proficiência na Avaliação de Laboratórios*. Instituto Adolfo Lutz, Rio de Janeiro. 2003.

Neuroph, 2008. *Java Neural Network Framework.* Disponível em: http://neuroph.sourceforge.net/index.html. Acessado em: 10-abr-2011.

Oliveira, E. C. Comparação de diferentes técnicas para a exclusão de "Outliers". ENQUALAB-2008 – Congresso da Qualidade em Metrologia. 2008.

Prass, F. S. *KKD: Processo de descoberta de conhecimento em bancos de dados*. Grupo de Interesse Em Engenharia de Software, Florianópolis, v. 1, p. 10-14, 2004.

Rezende, S. *Sistemas Inteligentes: Fundamentos e Aplicações*. Manole Editora. 2005. 525p.

Ross, Stephen M. *Peirce's Criterion for the Elimination of Suspect Experimental Data*. J. Engr. Technology. 2003.

Salgado, R. M. *Sistema de Suporte à Decisão para Previsão de Carga por Barramento*. 2009. 207f. Tese de Doutorado – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas (UNICAMP), Campinas, 2009.

Santos, R. *Introdução a Mineração de Dados*. 2011. Disponível em: http://www.lac.inpe.br/~rafael.santos/dmapresentacoes.jsp. Acessado em: 16-maio-2011.

Sarabando, P. *Outliers: Conceitos Básicos*. Disponível em: <http://www.estv.ipv.pt/PaginasPessoais/psarabando/CET%20%20Ambiente%202008-2009/Slides/8.%20Outliers.pdf>. Acessado em: 30-set-2010.

Silva, M. F. *Noções de Estatística com ênfase em Análise Exploratória de Dados*. 2008. Disponível em: <http://marcosfs2006.googlepages.com>. Acesso em: 01-out-2010.

Soares, M., 2009. Critério de Chauvenet. Disponível em: http://www.mspc.eng.br/tecdiv/med200.shtml. Acessado em: 21-fev-2011.

Thomé, A. C. G. *Redes Neurais – Uma Ferramenta para KDD e Data Mining*. 2003. Disponível em: <http://equipe.nce.ufrj.br/thome/grad/nn/curso/mdidatico.htm>. Acesso em: 02-nov-2010.

Triola, M. F. *Introdução à Estatística.* 9 ed. LTC, Rio de Janeiro, 2005. 656p.

Wehenkel, L. *Automatic Learning Techniques in Power Systems*. Kluwer, 1998.

8
Anexo A

Este anexo apresenta alguns dos valores críticos tabelados para os testes estatísticos de identificação de *outliers,* estudados no capítulo 3, para alguns níveis de confiança/significância (α).

8.1 Critério de *Chauvenet*

Tabela 7 – Valores críticos tabelados de *Chauvenet.*

Tamanho do conjunto de dados (n)	*Valor Crítico* $\alpha = 5\%$
3	1,38
4	1,53
5	1,64
6	1,73
8	1,86
10	1,96
15	2,13
20	2,24
25	2,33
50	2,57
100	2,8
250	3,1
500	3,3

1000	3,6

8.2 Critério de *Peirce*

Tabela 8 – Valores críticos tabelados de *Peirce.*

Tamanho do conjunto de dados (n)	*Quantidade de valores suspeitos*			
	1	**2**	**3**	**9**
3	1,196	-	-	-
4	1,383	1,078	-	-
5	1,509	1,200	-	-
10	1,878	1,570	1,380	-
20	2,209	1,914	1,732	1,190
30	2,385	2,103	1,927	1,411
40	2,504	2,230	2,059	1,556
50	2,592	2,326	2,158	1,666
60	2,663	2,401	2,237	1,753

8.3 Teste de *Cochran*

Tabela 9 – Valores críticos tabelados de *Cochran.*

Tamanho do conjunto de dados (n)	*Valor Crítico* $\alpha = 1\%$	*Valor Crítico* $\alpha = 5\%$	*Valor Crítico* $\alpha = 10\%$
3	1,115	1,153	1,148
4	1,492	1,463	1,425
5	1,749	1,672	1,602
6	1,944	1,822	1,729
7	2,097	1,938	1,828
8	2,221	2,032	1,909
9	2,323	2,110	1,977
10	2,410	2,176	2,036
15	2,705	2,409	2,247
20	2,884	2,557	2,385
25	3,009	2,663	2,486
50	3,336	2,956	2,768

100	3,600	3,207	3,017

8.4 Teste de *Dixon*

Tabela 10 – Valores críticos tabelados de *Dixon*.

Tamanho do conjunto de dados (n)	*Valor Crítico* $\alpha = 1\%$	*Valor Crítico* $\alpha = 5\%$	*Valor Crítico* $\alpha = 10\%$
3	0,988	0,941	0,886
4	0,889	0,765	0,679
5	0,780	0,642	0,557
6	0,698	0,560	0,482
7	0,637	0,507	0,434
8	0,683	0,554	0,479
9	0,635	0,512	0,441
10	0,597	0,477	0,409
11	0,679	0,576	0,517
12	0,642	0,546	0,490
13	0,615	0,521	0,467
14	0,641	0,546	0,492
15	0,616	0,525	0,472
16	0,595	0,507	0,454
17	0,577	0,490	0,438
18	0,561	0,475	0,424
19	0,547	0,462	0,412
20	0,535	0,450	0,401
21	0,524	0,440	0,391
22	0,514	0,430	0,382
23	0,505	0,421	0,374
24	0,497	0,413	0,367
25	0,489	0,406	0,360

8.5 Teste de *Grubbs*

Tabela 11 – Valores críticos tabelados de *Grubbs*.

Tamanho do conjunto de dados (n)	*Valor Crítico* $\alpha = 1\%$	*Valor Crítico* $\alpha = 5\%$	*Valor Crítico* $\alpha = 10\%$
3	1,115	1,153	1,148
4	1,492	1,463	1,425
5	1,749	1,672	1,602
6	1,944	1,822	1,729
7	2,097	1,938	1,828
8	2,221	2,032	1,909
9	2,323	2,110	1,977
10	2,410	2,176	2,036
15	2,705	2,409	2,247
20	2,884	2,557	2,385
25	3,009	2,663	2,486
50	3,336	2,956	2,768
100	3,600	3,207	3,017

8.6 Teste da Razão Q

Tabela 12 – Valores críticos tabelados do teste da Razão Q.

Tamanho do conjunto de dados (n)	*Valor Crítico* $\alpha = 1\%$	*Valor Crítico* $\alpha = 5\%$	*Valor Crítico* $\alpha = 10\%$
3	0,941	0,970	0,994
4	0,765	0,829	0,926
5	0,642	0,710	0,821
6	0,560	0,625	0,740
7	0,507	0,568	0,680
8	0,468	0,526	0,634
9	0,437	0,493	0,598
10	0,412	0,466	0,568
15	0,338	0,384	0,475
20	0,300	0,342	0,425
25	0,277	0,317	0,393
30	0,260	0,298	0,372

9
Apêndice A

A seguir serão apresentados os resultados que foram obtidos a partir da aplicação dos testes de identificação de *outliers* e também o tratamento realizado, utilizando à média e as RNAs. Foi aplicado os testes e tratamento para o horário das 15:00 até as 15:59 discretizados em minutos, da série B.

Resultados para a identificação de Outliers:

Total de Outliers identificados: 25

Total de Falsos Ouliers: 02

Total de Outliers identificados pelos testes:

BOXPLOT: 19

CHAUVENET: 15

COCHRAN: 22

DIXON: 18

GRUBBS: 20

NL: 25

NL MODIFICADO: 21

PIERCE: 25

RAZAO Q: 16

ZSCORES: 15

REGRA DO ERRO: 6

Outliers identificados e Tratamento Realizado, com o Erro Relativo em relação aos dados de comparação.

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:00:00

Carga: 2841.310059

Media: 4063.5934243333336

Neuroph: 4159.199124121638

FeedForward: 4300.640800899995

Encog: 3656.790516646966

Unicamp: 4066.0

Erro Relativo Media: 0.05918779308082668

Erro Relativo Neuroph: 2.292157504221306

Erro Relativo FeedForward: 5.770801792916753

Erro Relativo Encog: 10.06417814444255

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:01:00

Carga: 2822.76001

Media: 4081.419921666666

Neuroph: 4178.63510222109

FeedForward: 4275.216816839997

Encog: 3736.5443763154703

Unicamp: 4084.0

Erro Relativo Media: 0.06317527750572621

Erro Relativo Neuroph: 2.317216019125616

Erro Relativo FeedForward: 4.68209639666985

Erro Relativo Encog: 8.507728297858218

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:02:00

Carga: 2819.129883

Media: 4095.9399413333335

Neuroph: 4202.780784769563

FeedForward: 4291.374189139997

Encog: 3754.998730326226

Unicamp: 4076.0

Erro Relativo Media: 0.4892036637226089

Erro Relativo Neuroph: 3.1104216086742658

Erro Relativo FeedForward: 5.2839594980372135

Erro Relativo Encog: 7.8753991578452895

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:03:00

Carga: 2807.780029

Media: 4093.4933266666667

Neuroph: 4174.597217061517

FeedForward: 4280.6887988

Encog: 3754.2374388951184

Unicamp: 4122.0

Erro Relativo Media: 0.6915738314733942

Erro Relativo Neuroph: 1.2760120587461634

Erro Relativo FeedForward: 3.8498010383309036

Erro Relativo Encog: 8.921944713849626

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:04:00

Carga: 2830.22998

Media: 4091.6266276666665

Neuroph: 4192.07148769987

FeedForward: 4292.746894880008

Encog: 3703.753515426646

Unicamp: 4081.0

Erro Relativo Media: 0.26039273870782986

Erro Relativo Neuroph: 2.7216733080095556

Erro Relativo FeedForward: 5.18860315804969

Erro Relativo Encog: 9.243971687658757

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:05:00

Carga: 2832.709961

Media: 4098.249918333334

Neuroph: 4200.401239134821

FeedForward: 4244.513862160007

Encog: 3727.712592561629

Unicamp: 4070.0

Erro Relativo Media: 0.694101187551201

Erro Relativo Neuroph: 3.203961649504191

Erro Relativo FeedForward: 4.287809881081258

Erro Relativo Encog: 8.410010010770781

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:06:00

Carga: 2845.199951

Media: 4106.019938

Neuroph: 4208.375562264829

FeedForward: 4280.823466519994

Encog: 3754.6674751136447

Unicamp: 4086.0

Erro Relativo Media: 0.48996421928537365

Erro Relativo Neuroph: 2.99499662909517

Erro Relativo FeedForward: 4.768073091531922

Erro Relativo Encog: 8.10897026153586

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:07:00

Carga: 2893.719971

Media: 4100.883382

Neuroph: 4186.833396353436

FeedForward: 4255.0338625

Encog: 3764.5440695895068

Unicamp: 4089.0

Erro Relativo Media: 0.2906182929811686

Erro Relativo Neuroph: 2.3925995684381514

Erro Relativo FeedForward: 4.0605004279775105

Erro Relativo Encog: 7.9348478946073175

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:08:00

Carga: 2992.5

Media: 4105.383382

Neuroph: 4200.725742623072

FeedForward: 4265.632036059998

Encog: 3763.5452288972297

Unicamp: 4079.0

Erro Relativo Media: 0.6468100514832062

Erro Relativo Neuroph: 2.9842055068171542

Erro Relativo FeedForward: 4.575436039715578

Erro Relativo Encog: 7.73363008342168

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:09:00

Carga: 2941.47998

Media: 4112.563395

Neuroph: 4212.411863600753

FeedForward: 4265.440072959995

Encog: 3797.227591427758

Unicamp: 4066.0

Erro Relativo Media: 1.1451892523364517

Erro Relativo Neuroph: 3.600882036417938

Erro Relativo FeedForward: 4.905068198720972

Erro Relativo Encog: 6.610241233945943

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:10:00

Carga: 2887.810059

Media: 4107.683268

Neuroph: 4208.805486954685

FeedForward: 4280.594570519996

Encog: 3749.0770833189013

Unicamp: 4079.0

Erro Relativo Media: 0.7031936258886926

Erro Relativo Neuroph: 3.1822870055083343

Erro Relativo FeedForward: 4.94225473204206

Erro Relativo Encog: 8.088328430524609

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:11:00

Carga: 2734.409912

Media: 4103.563395333334

Neuroph: 4194.650723648732

FeedForward: 4282.535019200008

Encog: 3757.416444138513

Unicamp: 4068.0

Erro Relativo Media: 0.8742230907899109

Erro Relativo Neuroph: 3.1133412893002044

Erro Relativo FeedForward: 5.273722202556731

Erro Relativo Encog: 7.634797341727808

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:12:00

Carga: 2637.780029

Media: 4102.6898599999995

Neuroph: 4192.523341554349

FeedForward: 4282.029785000006

Encog: 3774.4252878793473

Unicamp: 4091.0

Erro Relativo Media: 0.2857457834270227

Erro Relativo Neuroph: 2.481626535183303

Erro Relativo FeedForward: 4.669513199706822

Erro Relativo Encog: 7.738321000260395

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:13:00

Carga: 3005.919922

Media: 4103.843261666666

Neuroph: 4202.413368441241

FeedForward: 4290.701142180004

Encog: 3717.2584170023274

Unicamp: 4094.0

Erro Relativo Media: 0.24043140368016527

Erro Relativo Neuroph: 2.64810377238009

Erro Relativo FeedForward: 4.804619984855993

Erro Relativo Encog: 9.202285857295374

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:14:00

Carga: 2970.629883

Media: 4113.176757666667

Neuroph: 4221.700465597899

FeedForward: 4293.944804680001

Encog: 3762.911957138431

Unicamp: 4078.0

Erro Relativo Media: 0.8625982752983588

Erro Relativo Neuroph: 3.5237975870009564

Erro Relativo FeedForward: 5.295360585581185

Erro Relativo Encog: 7.726533665070358

Data: 23/09/2010 | Dia da Semana: Quinta-feira | Horario: 15:14:00

Carga: 4471.02002

Media: 4228.473307333334

Neuroph: 4241.259763119347

FeedForward: 4241.259765999991

Encog: 3838.8498386496753

Unicamp: 4313.0

Erro Relativo Media: 1.9598120256588536

Erro Relativo Neuroph: 1.6633488727255588

Erro Relativo FeedForward: 1.6633488059357517

Erro Relativo Encog: 10.993511740095634

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:15:00

Carga: 3066.469971

Media: 4124.726643666666

Neuroph: 4201.312420066867

FeedForward: 4287.107148040004

Encog: 3783.826207716313

Unicamp: 4092.0

Erro Relativo Media: 0.7997713506027846

Erro Relativo Neuroph: 2.671369014341817

Erro Relativo FeedForward: 4.768014370479086

Erro Relativo Encog: 7.531128843687364

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:16:00

Carga: 3074.929932

Media: 4108.549967333333

Neuroph: 4301.208173906547

FeedForward: 4281.664961000001

Encog: 3778.7604472874077

Unicamp: 4096.0

Erro Relativo Media: 0.30639568684895924

Erro Relativo Neuroph: 5.009965183265308

Erro Relativo FeedForward: 4.5328359619140945

Erro Relativo Encog: 7.745106267397272

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:17:00

Carga: 3136.449951

Media: 4118.270019333333

Neuroph: 4197.397849174274

FeedForward: 4298.0

Encog: 3796.96562224337

Unicamp: 4095.0

Erro Relativo Media: 0.5682544403744332

Erro Relativo Neuroph: 2.5005579773937407

Erro Relativo FeedForward: 4.957264957264957

Erro Relativo Encog: 7.278006782823687

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:18:00

Carga: 3093.389893

Media: 4117.473388333333

Neuroph: 4192.2392466898045

FeedForward: 4278.13227038001

Encog: 3783.2018933197255

Unicamp: 4094.0

Erro Relativo Media: 0.5733607311512672

Erro Relativo Neuroph: 2.3995907838252206

Erro Relativo FeedForward: 4.497612857352459

Erro Relativo Encog: 7.5915512134898515

Data: 10/02/2010 | Dia da Semana: Quarta-feira | Horario: 15:19:00

Carga: 3160.679932

Media: 4112.600016333334

Neuroph: 4196.818627146897

FeedForward: 4221.8199704000035

Encog: 3742.358719114213

Unicamp: 4113.0

Erro Relativo Media: 0.009724864251549488

Erro Relativo Neuroph: 2.0378951409408383

Erro Relativo FeedForward: 2.6457566350596524

Erro Relativo Encog: 9.01145832447817

Data: 12/04/2010 | Dia da Semana: Segunda-feira | Horario: 15:40:00

Carga: 4842.100098

Media: 4152.283284666667

Neuroph: 4258.2597659996545

FeedForward: 4252.285771859992

Encog: 3790.062913362408

Unicamp: 4117.0

Erro Relativo Media: 0.8570144441745685

Erro Relativo Neuroph: 3.43113349525515

Erro Relativo FeedForward: 3.286027978139219

Erro Relativo Encog: 7.941148570259705

Data: 26/05/2010 | Dia da Semana: Quarta-feira | Horario: 15:42:00

Carga: 4431.970215

Media: 4199.7532550000005

Neuroph: 4287.7797849972885

FeedForward: 4287.779785000006

Encog: 3533.3926465789136

Unicamp: 4194.0

Erro Relativo Media: 0.137178230805926

Erro Relativo Neuroph: 2.236046375710265

Erro Relativo FeedForward: 2.2360463757750617

Erro Relativo Encog: 15.751248293302012

Data: 26/05/2010 | Dia da Semana: Quarta-feira | Horario: 15:43:00

Carga: 4437.379883

Media: 4182.203287666666

Neuroph: 4277.399901993206

FeedForward: 4277.399901999993

Encog: 3536.0195741285625

Unicamp: 4171.0

Erro Relativo Media: 0.26859956045711475

Erro Relativo Neuroph: 2.550944665384948

Erro Relativo FeedForward: 2.5509446655476586

Erro Relativo Encog: 15.223697575436049

Data: 26/05/2010 | Dia da Semana: Quarta-feira | Horario: 15:44:00

Carga: 4495.200195

Media: 4191.029947666667

Neuroph: 4290.379882990669

FeedForward: 4290.379882999991

Encog: 3513.40385736469

Unicamp: 4196.0

Erro Relativo Media: 0.11844738639974059

Erro Relativo Neuroph: 2.2492822447728575

Erro Relativo FeedForward: 2.249282244995029

Erro Relativo Encog: 16.267782236303862

Data: 01/01/2010 | Dia da Semana: Sexta-feira | Horario: 15:48:00

Carga: 3166.939941

Media: 4115.150065333332

Neuroph: 4254.917285569275

FeedForward: 4250.568300935738

Encog: 4020.5882465856807

Unicamp: 3167.0

Erro Relativo Media: 29.938429596884507

Erro Relativo Neuroph: 34.35166673726792

Erro Relativo FeedForward: 34.21434483535643

Erro Relativo Encog: 26.95258119942156

Data: 01/01/2010 | Dia da Semana: Sexta-feira | Horario: 15:51:00

Carga: 3155.820068

Media: 4126.746663333333

Neuroph: 4235.005759778307

FeedForward: 4264.329048100001

Encog: 3958.020097000426

Unicamp: 3151.0

Erro Relativo Media: 30.966253993441235

Erro Relativo Neuroph: 34.401960005658744

Erro Relativo FeedForward: 35.33256261821648

Erro Relativo Encog: 25.61155496669077

Média dos resultados de tratamento obtidos:

Erro Relativo Media: 2.7518389184541805

Erro Relativo Neuroph: 5.086927502776473

Erro Relativo FeedForward: 6.49228379754853

Erro Relativo Encog: 10.433331992377797