UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

INTELIGÊNCIA COMPUTACIONAL APLICADA À DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS - ESTUDO DE CASO EM CONSUMO DE ENERGIA ELÉTRICA

DIEMISOM CARLOS ROMANO DE MELO

DM 30/2015

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2015

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DIEMISOM CARLOS ROMANO DE MELO

INTELIGÊNCIA COMPUTACIONAL APLICADA À DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS - ESTUDO DE CASO EM CONSUMO DE ENERGIA ELÉTRICA

DM 30/2015

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2015

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DIEMISOM CARLOS ROMANO DE MELO

INTELIGÊNCIA COMPUTACIONAL APLICADA À DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS - ESTUDO DE CASO EM CONSUMO DE ENERGIA ELÉTRICA

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Computação Aplicada.

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – PA

2015

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

INTELIGÊNCIA COMPUTACIONAL APLICADA À DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS - ESTUDO DE CASO EM CONSUMO DE ENERGIA ELÉTRICA

AUTOR: DIEMISOM CARLOS ROMANO DE MELO

Este Trabalho foi julgado em ___/___/_____adequado para obtenção do Grau de Mestre em Engenharia Elétrica, e aprovado na sua forma final pela banca examinadora que atribuiu o conceito _____.

_____
Profa. Dra. Adriana Rosa Garcez Castro
ORIENTADORA

_____
Prof. Dr. Ubiratan Holanda Bezerra
MEMBRO DA BANCA EXAMINADORA

_____
Prof. Dr. André Cavalcante do Nascimento
MEMBRO DA BANCA EXAMINADORA

AGRADECIMENTOS

Agradeço, em primeiro lugar, àquele que é o único digno de toda a honra, Deus. Agradeço a minha esposa, Natália, pela dedicação e entrega ao compromisso de ser minha companheira idônea, me ajudando em tudo o que pôde para que eu pudesse concluir esta etapa. À minha princesinha Ana Clara, pelo sorriso mais lindo do mundo que enche meus dias de mais alegria. À minha mãe, Simone, pela educação que me deu e pela ajuda nesses primeiros meses com a Clarinha, a senhora é 10!

À minha orientadora, professora Adriana, principalmente pela confiança em mim depositada, desde o início desse período de mestrado. Obrigado pelo seu esforço em me ajudar a concluir esse trabalho. Espero ter alcançado ao menos a metade da sua expectativa.

Ao amigo que conheci no mestrado, Thiago Nicolau, pela amizade e companheirismo nessa empreitada, sempre transmitindo motivação. Aos amigos Bruno Zefi e Jeff, que com suas palavras me passaram ânimo para continuar até o fim. Ao meu pastor Valério Rendeiro, cujas palavras e amizade sempre apontaram para Cristo e para o seu propósito em minha vida.

Enfim, aos meus irmãos e irmãs, demais parentes, amigos e alunos da UFPA em Belém, em Castanhal e em Cametá, pastores, líderes e amigos da RUG, e todos que contribuíram direta e indiretamente para que eu chegasse até aqui. Muito obrigado mesmo!

Diemisom Carlos Romano de Melo

"Você pode encarar um *outlier* como um erro a ser esquecido, ou como um resultado que aponta uma nova direção."

Steve Jobs - paráfrase

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

RESUMO

A previsão de consumo de energia elétrica é uma tarefa que requer modelos computacionais bastante acurados para que possam influenciar corretamente na tomada de decisão em usinas hidrelétricas e distribuidoras de energia. Estes modelos computacionais são implementados a partir de um conjunto de dados que deve representar fielmente o comportamento das variáveis. Porém, nesses conjuntos de dados é bastante comum a presença de *outliers,* que surgem devido a erros de leitura de sensores, erros no próprio sistema de processamento/armazenamento dos dados ou falhas no sistema de distribuição. Este trabalho propõe então uma nova metodologia baseada em Inteligência Computacional para detecção e correção de *outliers* em séries temporais de consumo de energia elétrica. Uma rede neural artificial auto-associativa é utilizada para detecção de *outliers.* Posteriormente, esta rede neural, em conjunto com um algoritmo genético, é utilizada para a correção dos *outliers* detectados. Esta abordagem foi aplicada a uma série temporal de consumo de Energia Elétrica no Estado do Pará. Os resultados obtidos demonstram a eficiência da metodologia proposta, que identificou e corrigiu todos os *outliers* virtuais introduzidos durante a fase de avaliação da metodologia.

PALAVRAS-CHAVES: Detecção de *Outliers*, Correção de *Outliers,* Redes Neurais Auto-associativas, Algoritmos Genéticos, Séries Temporais.

ABSTRACT

The electric load prediction is a task that requires accurate models, as should properly influence the decision making in hydroelectric plants and power stations. These computer models are implemented from a data set that must faithfully represent the behavior of the variables. However, these data sets are quite common the presence of *outliers*, which arise due to sensor reading errors, errors in the actual processing system / storage of data or faults in the distribution system or power station. This paper proposes a new methodology based on Computational Intelligence for detection and treatment of outliers in time series of electric power load. An auto associative artificial neural network is used for *outlier* detection. Subsequently, it is reused together with a genetic algorithm to correct detected *outliers*. This approach was applied to a time series of electrical power load in the State of Pará. The computational experiments were performed using the MATLAB tool and the results demonstrate the efficiency of the proposal, which identified and corrected all virtual *outliers* introduced during the evaluation phase of the methodology.

KEYWORDS: *Outliers* Detection, *Outliers* Correction, Auto-associative Neural Networks, Genetic Algorithms, Time Series.

1 INTRODUÇÃO

Outliers são objetos em um conjunto de dados que apresentam características diferentes dos demais objetos, parecendo ter sido gerados por um sistema diferente. Detecção de *Outliers* é a tarefa de reconhecer os *outliers* e pode ser tão importante quanto reconhecer os objetos normais, pois *outliers* podem representar informações valiosas em determinados cenários. Certas situações requerem ainda que os *outliers* detectados sejam tratados, geralmente sendo corrigidos para valores mais próximos aos normais.

Muitos trabalhos vêm sendo descritos na literatura na área de detecção e correção de *outliers* (MARKOU e SINGH, 2003, HODGE e AUSTIN, 2004, CHANDOLA, BARNEJEE e KUMAR, 2009, SINGH e UPADHYAYA, 2012 e PIMENTEL et al., 2014). Os primeiros trabalhos nessa área utilizaram técnicas de análise estatísticas, implementando modelos paramétricos e não paramétricos, ou mesmo partindo da análise de características mais triviais dos dados. Tais técnicas têm sido aprimoradas com o passar dos tempos e continuam obtendo bons resultados.

Com os avanços da área de Inteligência Computacional (IC), novas técnicas vêm sendo desenvolvidas e aplicadas ao problema de detecção de *outliers*. Redes neurais artificiais, máquinas de vetores de suporte, análise de vizinhos mais próximos, sistemas baseados em regras de decisão, formação de agrupamentos e redes bayesianas, são algumas das técnicas de IC comumente utilizadas.

Apesar dos bons resultados das metodologias já apresentadas na literatura, muitas pesquisas nesta área ainda vêm sendo realizadas com o objetivo de se encontrar técnicas mais avançadas e confiáveis.

Uma das áreas de aplicação na qual se busca cada vez mais metodologias de detecção e correção de *outliers* mais eficientes é a área de sistemas de previsão. Estes sistemas são desenvolvidos com o objetivo de auxiliar as organizações a tomarem decisões de maneira mais rápida e melhor. No cenário atual, não basta que as organizações esperem pelas mudanças para então tomarem suas decisões. Deve-se tentar adiantar as mudanças e agir antes que elas aconteçam (HARVARD BUSINESS SCHOOL, 2012).

Em usinas hidrelétricas e distribuidoras de energia, os sistemas de previsão de carga realizam a importante tarefa de predizer a demanda futura de energia. Estes sistemas são implementados através da análise de uma série temporal formada por um conjunto de dados da demanda de consumo. Entretanto, devido a erros de leitura de sensores, erros no próprio sistema de processamento/armazenamento dos dados ou falhas no sistema de distribuição,

estes conjuntos de dados podem apresentar *outliers*, necessitando assim de um pré-processamento antes de serem utilizados no desenvolvimento dos modelos de previsão. Parte do pré-processamento consiste exatamente na detecção e correção dos *outliers* presentes na série temporal.

Uma série temporal de consumo de energia livre de *outliers* poderá gerar modelos de previsão mais confiáveis, modelos estes que poderão ser utilizados nas usinas hidrelétricas e distribuidoras de energia para tomadas de decisão cada vez mais acertadas.

1.1 **OBJETIVOS**

Considerando a importância do desenvolvimento de sistemas de detecção e correção de *outliers* para séries temporais de consumo de energia, este trabalho tem como objetivo apresentar uma nova abordagem baseada em Inteligência Computacional para detecção e correção de *outliers*. A metodologia proposta é baseada em uma Rede Neural Auto-associativa para detecção dos *outliers* e trabalha em conjunto com um algoritmo genético para a correção dos *outliers* detectados.

A metodologia proposta será aplicada a uma série temporal de consumo de energia elétrica com o objetivo de mostrar a aplicabilidade da metodologia na área de Sistemas de Energia, mais especificamente na área de previsão de carga. A detecção e correção de *outliers* em uma série temporal de carga podem gerar um conjunto de dados com qualidade para a geração de um modelo de previsão de cargas mais confiável.

1.2 **ESTRUTURA DO TRABALHO**

Este trabalho está dividido em 6 capítulos, organizados da seguinte forma: O capítulo 2 apresenta a fundamentação teórica da tarefa de detecção e correção de *outliers*, desde a definição de *outliers*, os desafios envolvidos na tarefa de detectá-los, uma listagem de aplicações e métodos utilizáveis; O capítulo 3 apresenta a técnica de redes neurais artificiais, incluindo as redes perceptron de múltiplas camadas e as redes auto-associativas. Apresenta também a técnica de algoritmos genéticos, detalhando os operadores de cruzamento e mutação e o processo de seleção natural. O capítulo 4 apresenta o método proposto para detecção e correção de *outliers*. No capítulo 5 são abordados os resultados obtidos pelo método em um estudo de caso, em que é aplicado a uma série temporal de consumo de

energia elétrica. O capítulo 6 apresenta as conclusões do trabalho e algumas possibilidades de trabalhos futuros.

2 DETECÇÃO DE OUTLIERS

2.1 INTRODUÇÃO

A detecção de *outliers* é caracterizada pela busca por amostras de dados que apresentem características diferentes das demais amostras. Estes *outliers* também podem ser chamados de amostras anormais, anomalias, novidades, exceções, faltas, defeito ou contaminantes, dependendo da área de aplicação. Chama-se Detecção de *Outliers* a tarefa de encontrar *outliers* em um conjunto de dados e Tratamento de *Outliers* ao tratamento que estes recebem, uma vez detectados.

Outliers representam informações importantes em diversas áreas de aplicação, por exemplo:

- *Outliers* em dados de transações de cartão de crédito podem indicar que o cartão foi roubado e outra pessoa o está usando;
- *Outliers* em padrões de tráfego em uma rede de computadores podem indicar que a rede foi invadida e está enviando dados sigilosos para fora da rede;
- Em vigilância militar, a presença de regiões não usuais em imagens de satélite pode indicar a movimentação de tropas inimigas;
- *Outliers* em dados de sintomas de doenças podem ser utilizados pelo sistema de saúde pública para verificar a ocorrência de uma nova doença;
- Em séries temporais de registros da bolsa de valores, *outliers* podem indicar mudanças no cenário econômico, ocasionando grandes lucros ou prejuízos.

2.2 DEFINIÇÃO DE *OUTLIER*

A literatura é vasta no estudo de *outliers*. Diversos autores de diversas áreas de estudo já se propuseram a estudá-los. Dessa forma, uma definição formal do termo *outliers* é necessária para que haja uma delimitação do tema em estudo.

BARNETT e LEWIS (1978) propõem a seguinte definição formal para *outliers*:

"*Deve-se definir um outlier em um conjunto de dados como uma observação (ou subconjunto de observações), que parece ser incompatível com o resto desse conjunto de dados*"

HAWKINS (1980) define formalmente o conceito de um *outlier* da seguinte maneira:

"*Um outlier é uma observação que desvia muito de outras observações, despertando suspeitas de que são geradas por um mecanismo diferente*".

A Figura 2. 1 apresenta três subconjuntos de amostras normais e alguns *outliers*.

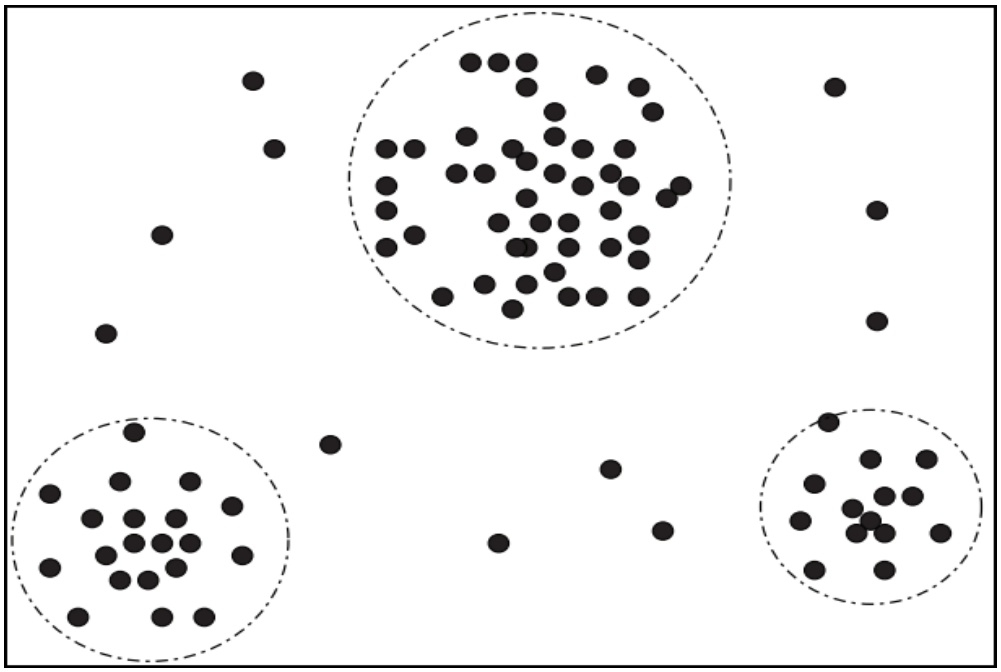

Figura 2.1 - Exemplo de *outliers* em um conjunto de dados.

É importante ressaltar que *outliers* são diferentes de ruído, pois este consiste em um erro aleatório ou variância em uma variável de estudo e deve ser removido antes da análise de *outliers*.

2.3 ASPECTOS DO PROBLEMA DE DETECÇÃO E TRATAMENTO DE OUTLIERS

O problema de detecção e tratamento de *outliers* possui aspectos que devem ser analisados para melhor formulação do problema e escolha das técnicas e métodos de supervisão de forma a potencializar os resultados obtidos.

2.3.1 NATUREZA DOS DADOS DE ENTRADA

Em um conjunto de dados, as amostras de dados, ou objetos de dados, podem ser representados por um vetor de características, também chamado vetor de atributos, em que cada dimensão do vetor representa uma característica ou um atributo desta amostra.

Uma amostra pode possuir apenas um atributo ou múltiplos atributos. Estas características podem ser quantitativas ou qualitativas, de acordo com a seguinte divisão (HAN, KAMBER e PEI, 2012):

Características quantitativas:

(a) valores contínuos, por exemplo, peso;

(b) valores discretos, por exemplo, número de carros;

(c) valores intervalares, por exemplo, duração de um evento ou faixa etária;

Características qualitativas:

(a) nominal ou não-ordenadas, por exemplo, cor;

(b) ordinal, por exemplo, patente militar ou avaliações qualitativas de temperatura, frio ou quente ou ainda intensidade de som, quieto ou rumoroso;

Técnicas estatísticas comumente usadas para detecção de *outliers* e técnicas baseadas em redes neurais artificiais aceitam apenas dados numéricos, portanto dados qualitativos precisam ser mapeados para valores numéricos para que se possa utilizar estas técnicas (CHANDOLA, BARNEJEE e KUMAR, 2009, SINGH e UPADHYAYA, 2012). Técnicas de aprendizado de máquina, como regras de decisão ou sistemas baseados em árvores, aceitam dados quantitativos e qualitativos (PIMENTEL et al., 2014).

Conjuntos de dados mais complexos podem ser formados a partir dos tipos de dados apresentados. Essa complexidade se caracteriza pela existência de relações entre as amostras. Existem basicamente três relações que podem ser encontradas em um conjunto de dados: relações espaciais, relações temporais e relações em grafo.

O conjunto de dados determina a melhor escolha do modelo a ser implementado, por isso é importante conhecer a natureza do conjunto de dados, as variáveis que constituem as amostras, e principalmente como se comportam e como se relacionam as amostras.

Não necessariamente as amostras utilizadas para realizar a detecção de *outliers* são as amostras observáveis no conjunto de dados original. Muitas técnicas utilizam esquemas de pré-processamento de dados, que podem extrair características das amostras ou criar novas dimensões no vetor de características, de forma a encontrar um novo conjunto de dados em que seja mais fácil distinguir entre amostras normais e *outliers*, ou tornando o método computacionalmente mais eficiente (CHANDOLA, BARNEJEE e KUMAR, 2009).

2.3.2 TIPOS DE *OUTLIER*

Outliers podem ser divididos em três categorias e a escolha da técnica de detecção também depende das características do *outlier* que se deseja identificar e de como este se comporta em comparação com o resto do conjunto de dados.

2.3.2.1 *OUTLIER* GLOBAL

Este é o tipo de *outlier* mais comum e o mais estudado, que consiste em uma única amostra que é considerada um *outlier*, devido aos seus atributos possuírem valores discrepantes em relação às demais amostras. A Figura 2.2 ilustra um exemplo de *outlier* global.

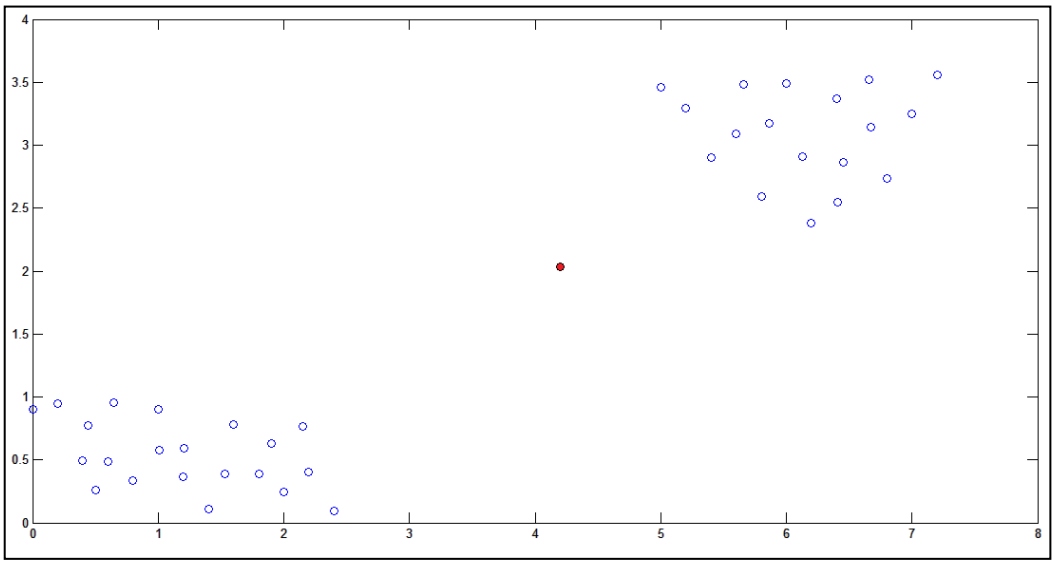

Figura 2.2 - Exemplo de *outlier* global, o círculo vermelho no centro da imagem.

Para se detectar este tipo de *outlier* costuma-se utilizar técnicas de detecção baseadas em medidas de proximidade, sendo necessário encontrar uma métrica de semelhança apropriada aos tipos de dados que formam as amostras. Desta forma, uma amostra é considerada um *outlier* se sua vizinhança é esparsamente habitada por outras amostras (AGGARWAL, 2013, HAN, KAMBER e PEI, 2012).

Este tipo de *outlier* é importante para diversas aplicações, por exemplo, em detecção de intrusão de computadores. Se um computador apresenta um padrão de comunicação diferente dos demais, com altas taxas de broadcast de mensagens, este comportamento pode

ser considerado como um *outlier* global e o computador pode ter sido invadido (HAN, KAMBER e PEI, 2012).

2.3.2.2 *OUTLIER* CONDICIONAL

Este tipo de *outlier* é semelhante ao primeiro tipo, porém ocorre em um conjunto de dados em que existe uma organização natural das amostras, ou uma noção de dependência entre elas, também interpretada como um contexto em que os valores se apresentam. Devido a necessidade desse contexto, este tipo também é chamado de *outlier* contextual (AGGARWAL, 2013, HAN, KAMBER e PEI, 2012). Deve-se ressaltar que a ocorrência desta amostra em outro contexto não seria caracterizado como *outlier*, daí sua caracterização como condicional, já que a mesma amostra, sob outras condições (outro contexto), não caracterizaria um *outlier*.

Outliers deste tipo são comumente encontrados em conjuntos de dados espaço temporais, como séries temporais, e podem ser encontrados em conjuntos de dados representados por grafos, ou mesmo em imagens. Consideremos o exemplo de uma série temporal de temperaturas médias na cidade de Belém, um valor de 0º C é muito provavelmente um *outlier*. A Figura 2.3 ilustra um exemplo de *outlier* contextual (o círculo vermelho no centro) em uma série temporal univariada e exibe a mesma amostra em um contexto em que não é considerada *outlier* (o círculo azul no início).

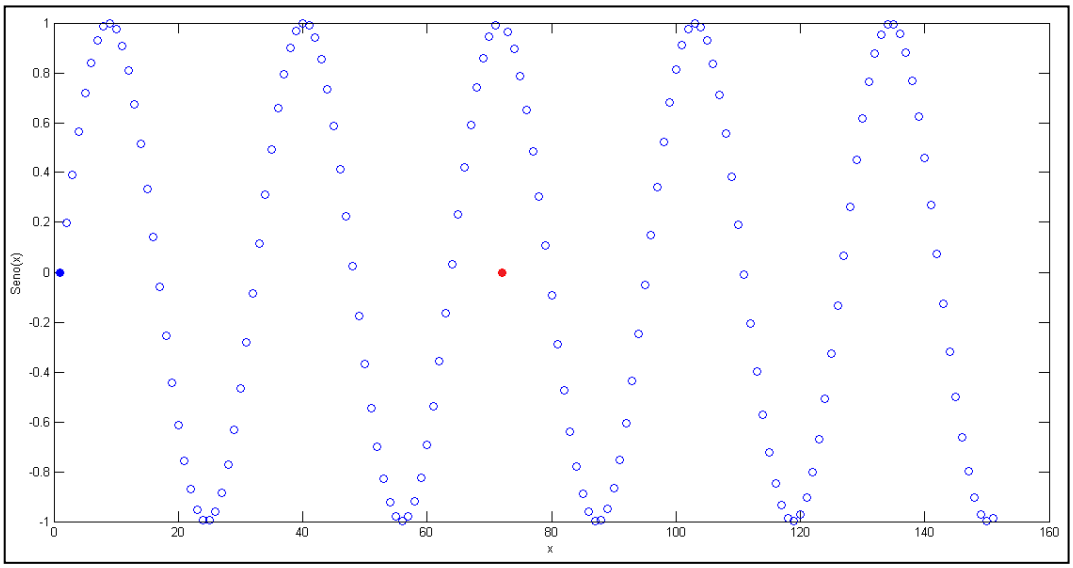

Figura 2.3 - Exemplo de *outlier* condicional.

A detecção de *outliers* globais pode ser considerada uma generalização da detecção de *outliers* contextuais, em que o conjunto de variáveis contextuais é nulo, ou seja, não condiciona a análise a um contexto específico do conjunto de dados, mas o contexto é conjunto completo.

2.3.2.3 *OUTLIER* COLETIVO

O terceiro tipo de *outlier* ocorre quando um subconjunto de amostras apresenta um comportamento diferente do esperado para o conjunto de dados, podendo também ser chamado de *outlier* coletivo (AGGARWAL, 2013). Este tipo de *outlier* também depende de uma estrutura espaço temporal presente no conjunto de dados, de forma que os valores das amostras em si não constituem *outliers*, mas a ocorrência desse conjunto de valores forma uma subestrutura discrepante.

Um exemplo de *outlier* coletivo pode ser uma oscilação rápida e inesperada no consumo de energia de uma determinada cidade, como mostra a Figura 2.4. Para detectar este tipo de *outlier* não basta avaliar as amostras individualmente, deve-se conhecer as relações que cada amostra possui com as demais e analisá-las em grupo.

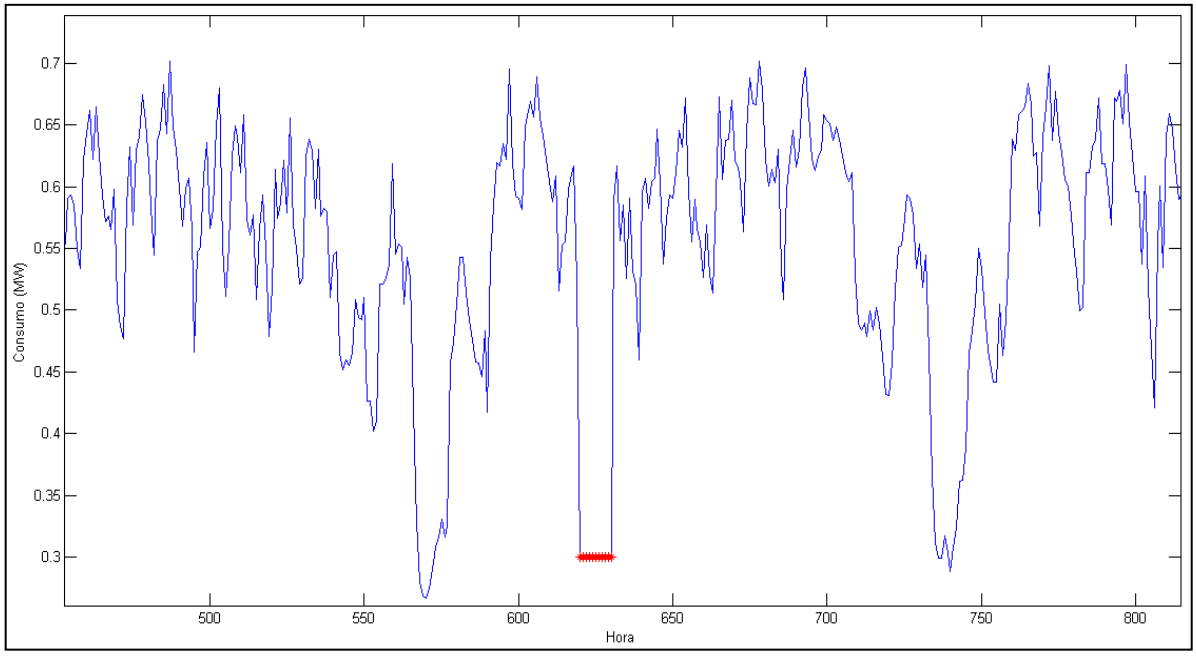

Figura 2.4 - *Outlier* coletivo em uma série temporal de consumo de energia.

2.3.3 RÓTULO DOS DADOS E TIPOS DE SUPERVISÃO

Um fator primordial em detecção de *outliers* é a presença ou não de rótulos no conjunto de dados. Um rótulo é uma informação associada a cada amostra do conjunto de dados, indicando se esta amostra é normal ou se é um *outlier*. Estes rótulos podem ser obtidos através de um especialista ou podem ser gerados juntamente com o conjunto de dados, quando se tratar de um conjunto de dados virtual.

Em geral, é bastante custoso obter um conjunto de dados que contenha todos os possíveis tipos de *outliers* rotulados, inclusive porque novos tipos de *outliers* podem surgir dentro de uma determinada aplicação, de forma que é mais comum obter conjuntos de dados que contenham apenas amostras rotuladas como normais.

Um exemplo que ilustra uma das diversas dificuldades de se obter conjuntos de dados com dados rotulados como *outliers* é o caso de sistemas de segurança aérea, em que um *outlier* pode estar relacionado a um fato extremamente raro, como a colisão de duas aeronaves, ou mesmo a queda de uma aeronave.

De acordo com a presença ou não destes rótulos no conjunto de dados, as técnicas de detecção de *outliers* podem ser divididas em três grupos: técnicas supervisionadas, técnicas não-supervisionadas e técnicas semi-supervisionadas (AGGARWAL, 2013).

2.3.3.1 MÉTODOS SUPERVISIONADOS

Métodos supervisionados podem ser utilizados quando existe um conjunto de dados para treino e existem amostras rotuladas como normais e outras como *outliers*. Dessa forma, estes métodos possuem a noção explícita do comportamento normal e de comportamento anormal, podendo gerar modelos mais exatos.

Métodos supervisionados para detecção de *outliers* dependem da existência de um conjunto de dados pré-rotulado para treinamento, que será o responsável por guiar o aprendizado do sistema. Além disso, é necessária a existência de uma medida de semelhança entre as amostras, para que se possa classificar novas amostras não utilizadas em treinamento, determinando se são mais semelhantes com amostras normais ou com *outliers*.

A existência do conjunto de dados previamente classificado possibilita a construção de modelos mais acurados, uma vez que possuem informação sobre amostras normais e anormais. Todavia, sabe-se que é difícil obter um conjunto de dados que cubra todas as regiões em que possam existir *outliers*, e tratando-se de sistemas dinâmicos é possível que

ocorram novos *outliers* em regiões não conhecidas durante a implementação do modelo de detecção.

Um fator limitante ao uso de técnicas supervisionadas é a dificuldade de se obter conjuntos de dados previamente rotulados que contenham quantidades suficientes de amostras normais e, principalmente, amostras anormais, para que o classificador alcance boa generalização. Em geral, as classes estão desbalanceadas, de forma que a quantidade de amostras normais é muito maior que a quantidade de amostras de *outliers*. Devido a isso, uma forma mais viável e bastante utilizada para obter boas quantidades de amostras anormais, é injetar *outliers* virtuais no conjunto de dados, conforme descrevem SINGH e UPADHYAYA (2012) e CHANDOLA, BARNEJEE e KUMAR (2009).

2.3.3.2 MÉTODOS NÃO SUPERVISIONADOS

Métodos não supervisionados são utilizados em cenários em que não há nenhuma informação sobre as amostras do conjunto de dados, ou seja, não se sabe quais amostras são normais nem quais são *outliers*, inclusive existem métodos desse tipo que sequer necessitam de um conjunto de dados pré-existente para treino (SINGH e UPADHYAYA, 2012).

Estes métodos assumem que as amostras normais estão agrupadas de alguma forma e que são mais frequentes que os *outliers*, logo *outliers* estão mais distantes da maior parte das amostras ou em regiões com baixa densidade de amostras. Caso essa característica não seja verdadeira para o problema em questão, a maioria dos métodos não supervisionados teriam resultados muito abaixo do esperado, gerando muitos "falsos positivos", rotulando muitas amostras normais como *outliers* (HODGE e AUSTIN, 2004, HAN, KAMBER e PEI, 2012).

Muitas técnicas de formação de agrupamentos (*Clustering*) podem ser adaptadas para efetuarem a detecção não supervisionada de *outliers* (HAN, KAMBER e PEI, 2012). A abordagem consiste em, primeiramente, identificar os agrupamentos e depois identificar amostras não agrupadas, rotulando-as como *outliers*. No entanto, tais abordagens precisam processar grandes quantidades de amostra para encontrar as amostras alvo, os *outliers*, tornando-se custosas e não indicadas para conjuntos de dados de grandes dimensões.

2.3.3.3 MÉTODOS SEMI-SUPERVISIONADOS

Métodos semi-supervisionados são utilizados quando apenas amostras de uma das classes estão presentes, ou seja, ou têm-se apenas algumas amostras rotuladas como normais,

ou apenas algumas amostras rotuladas como *outliers*. É necessário cuidado especial ao implementar modelos neste cenário. Um exemplo deste cenário é o caso de um sistema de monitoramento de aeronaves, em que um *outlier* pode representar um dano no motor, o que poderia ocasionar um acidente. Sabotar o motor para que se obtenham amostras *outliers* é uma tarefa cara demais e pode resultar em grandes perdas.

Quando apenas alguns rótulos normais estão disponíveis, é possível utilizar estas amostras rotuladas e amostras semelhantes a elas, que podem ser encontradas utilizando métodos de formação de agrupamentos, por exemplo, ou através de técnicas de densificação, gerando novas amostras semelhantes as normais. Com esse novo conjunto de dados é possível implementar um modelo que reconhece amostras normais, e amostras não reconhecidas pelo modelo são consideradas *outliers*.

Por outro lado, quando existem apenas algumas amostras rotuladas como *outliers*, a implementação de modelos deve ser ainda mais cuidadosa, pois dificilmente estas amostras representam todos os *outliers* possíveis. Neste caso, para melhorar a qualidade do modelo, métodos não supervisionados podem ser utilizados para encontrar amostras normais. Além disso, existem técnicas que modelam apenas o comportamento anormal e amostras não reconhecidas pelo modelo são consideradas normais (SINGH e UPADHYAYA, 2012).

2.3.4 SAÍDA DOS SISTEMAS DE DETECÇÃO DE *OUTLIER*

Outro aspecto importante em problemas de detecção de *outliers* é a saída que é gerada pelo sistema. Existem basicamente duas abordagens: Técnicas de Rotulação e Técnicas de Pontuação.

Técnicas de rotulação rotulam cada amostra como normal ou *outlier*, comportando-se como um sistema de classificação, atuando assim de maneira direta, gerando uma lista de *outliers* com a qual se pode interagir. Em geral, essas técnicas possuem melhor resultados quando os dados de entrada para implementação do modelo estão rotulados, conforme discutido no item 2.2.3.2, sobre métodos supervisionados.

Técnicas de pontuação atribuem um valor, considerando o grau de normalidade da amostra, ou grau de *outlier*. A obtenção desse valor pode ser feita tanto com conjuntos de dados de entrada rotulados ou não rotulados, podendo assim interagir com diversos tipos de conjuntos de dados.

Tais técnicas permitem que o especialista na área de aplicação utilize como limiar um valor que tenha mais significado no domínio, diferentemente das técnicas de rotulação,

que normalmente não permitem esta interação clara, de forma que o especialista precisa manipular parâmetros do modelo para obter resultados diferentes (SINGH e UPADHYAYA, 2012). Técnicas de pontuação também são interessantes em situações em que se queira diferenciar *outliers*, sendo necessário comparar duas amostras, por exemplo, e saber qual é mais anormal.

Uma vez gerado o resultado da técnica escolhida, pode ser necessário avaliar este resultado, para saber qual a acurácea obtida. Em métodos supervisionados, a divisão do conjunto de dados em amostras para treino e amostras para teste é uma abordagem tradicional para calcular a quantidade de acertos do modelo implementado (PIMENTEL et al., 2014). AGGARWAL (2013) explica que avaliar os resultados gerados por técnicas de detecção de *outliers* é uma tarefa difícil, sobretudo quando se trata de métodos não supervisionados ou mesmo semi-supervisionados, devido a falta de técnicas de avaliação precisas que gerem resultados realísticos.

Além destas características próprias do problema de detecção de *outliers*, cada caso específico pode possuir outras restrições, como tempo de processamento, eficiência computacional, acurácea da técnica implementada, entre outras.

2.4 TIPOS DE TÉCNICA DE DETECÇÃO

Técnicas de detecção de *outliers* têm sido implementadas ao longo de décadas por especialistas de diversas áreas, de forma que não é possível apresentar cada técnica, mas sim apresentar os tipos de técnicas existentes, fornecendo uma visão geral do que há disponível na literatura e na prática.

Diversos autores apresentam uma divisão das técnicas de detecção de *outliers* de acordo com suas características, área de conhecimento e intenções. CHANDOLA, BARNEJEE e KUMAR (2009) dividem as técnicas em 7 grupos: baseadas em probabilidade e estatística, baseadas em classificação, baseadas em formação de agrupamentos, baseadas em medidas de semelhança ou proximidade, baseadas em teoria da informação, baseadas em decomposição espectral e técnicas baseadas em visualização da informação.

HAN, KAMBER e PEI (2012) utilizam a seguinte divisão: técnicas estatísticas, técnicas de proximidade, técnicas baseadas em formação de agrupamentos e técnicas de classificação. MARKOU e SINGH (2003) escreveram um *survey* dividido em duas partes, sendo que na primeira abordam métodos estatísticos e na segunda parte métodos baseados em redes neurais artificiais.

HODGE e AUSTIN (2004) observam que as técnicas de detecção de *outliers* têm suas origens em três principais áreas de estudo: Estatística, Redes Neurais e Aprendizado de Máquina, portanto divide as técnicas nestes três grupos e apresenta também técnicas híbridas, que utilizam conceitos de mais de uma área. O *survey* mais recente (PIMENTEL et al., 2014) classifica as técnicas em cinco categorias: probabilística, baseada em distância, baseada em reconstrução, baseada em domínio e técnicas baseadas em teoria da informação.

Abordagens e métodos diferentes podem gerar melhores resultados em conjuntos de dados diferentes, com quantidades de amostras e de características diferentes, densidades diferentes, de acordo com a velocidade que se deseja e com a confiabilidade que se espera, desde que se escolha métodos que possam representar bem as características do conjunto de dados. Assim, a análise dos diversos aspectos do problema e do conjunto de dados é fundamental para decidir qual método utilizar e se ter garantias de que o resultado do sistema como um todo será válido.

2.5 TRATAMENTO DE *OUTLIERS*

Uma vez detectados os *outliers*, deve-se definir que tratamento eles receberão. Existem algumas abordagens neste sentido: aceitar essa amostra e não tratá-la de maneira especial; corrigir essa amostra de forma a suavizar a discrepância; excluir essa amostra; reconstruir esta amostra através de um estimador.

Para correção de amostras pode-se utilizar um algoritmo de interpolação, suavizando-o, ou mesmo substituir esta amostra por outra, dadas algumas restrições específicas do problema. Essa abordagem é comumente utilizada quando se utiliza a técnica de detecção de *outliers* no pré-processamento do conjunto de dados, a fim de utilizar o conjunto de dados processado para implementar um modelo de predição, por exemplo. No caso das séries temporais, caso a amostra no instante *t* seja considerada *outlier*, pode-se excluir essa amostra, e substituí-la pela amostra anterior, no instante *t-1*.

A Reconstrução consiste em utilizar o próprio modelo de detecção para gerar uma nova amostra (reconstruir) para substituir a amostra *outlier*, de forma que a amostra gerada não seja um *outlier*.

A Exclusão consiste em descartar as amostras detectadas como *outliers*, o que elimina o erro, porém perde-se informação, uma vez que não se sabe qual a informação que deveria existir no lugar do *outlier*. Uma vez excluída, pode-se utilizar técnicas de imputação

de valores ausentes para preencher a lacuna deixada, o que caracteriza uma reconstrução ou correção.

A Incorporação consiste em aceitar aquele valor considerado *outlier* como um valor normal, incorporando-o ao conjunto de dados. Essa abordagem também é chamada de acomodação. Quando se utiliza esta abordagem, costuma-se utilizar o termo Detecção de Novidades, uma vez que a amostra considerada *outlier* é interpretada como uma novidade, ou seja, ela não consiste em um erro, mas sim é um reflexo de um novo estado do sistema e costuma ser utilizada para gerar um novo modelo ou para atualizar o modelo existente.

Pode-se também adaptar a técnica utilizada, de forma associar um peso a cada amostra e diminuir o peso das amostras detectadas como *outliers*. No entanto, essa adaptação da técnica pode ser complexa, ou mesmo tornar-se inviável, dada as restrições de tempo do problema.

2.6 APLICAÇÕES

Segue uma lista de aplicações das técnicas de detecção de *outliers*, extraída do trabalho de SINGH e UPADHYAYA (2012), que em alguns casos apresenta o que seria o comportamento *outlier*, os desafios associados e as técnicas já utilizadas para realizar a tarefa.

- Detecção de Intrusão: Consiste em detectar atividades maliciosas em ambientes computacionais. Exemplo: Desafios: grande volume de dados; necessidade de análise online; dificuldade em obter amostras rotuladas; Técnicas: Redes Neurais Artificiais; análise estatísticas; Máquinas de Vetores de Suporte; Regras de Decisão; Redes Bayesianas; Modelos de Mistura.
- Detecção de Fraudes: Consiste em detectar atividades ilícitas de consumo não autorizado de recursos de organizações comerciais, como bancos, empresas de cartão de crédito, operadoras de telefonia, seguradoras, entre outras. Exemplo: Cartões de crédito e débito roubados ou clonados; chips de telefonia clonados; Geralmente o sistema detecta um novo padrão de comportamento do cliente, como total gasto por transação, horário de consumo dos recursos e locais em que os recursos são utilizados. Técnicas: Redes neurais artificiais; Sistemas baseados em regras; Formação de agrupamentos.
- Detecção de *Outliers* em Medicina: Busca identificar comportamentos anormais no organismo, através da análise de resultados de exames médicos,

como eletrocardiogramas e eletroencefalogramas. O maior desafio nessa área é a necessidade de modelos altamente acurados, pois classificar amostras *outliers* como normais pode ser fatal. Técnicas: Modelagem estatística; Redes neurais artificiais; Sistemas baseados em regras; Técnicas baseadas análise de vizinhança;

- Detecção de Danos Industriais: Consiste em identificar danos em componentes de sistemas industriais, que podem ocorre devido ao uso contínuo ou má operação. Estes danos devem ser rapidamente detectados para minimizar as perdas envolvidas. Desafios: Lida com *outliers* contextuais e requer correlação com dados de origens diversas, como informações meteorológicas. Em geral utiliza-se abordagem semi-supervisionada com as seguintes técnicas: Modelagem estatística paramétrica e não-paramétrica; Redes neurais artificiais; Sistemas baseados em regras; Modelos de mistura;
- Detecção de *Outliers* em Dados Textuais: Consiste em detectar novos tópicos de interesse, novas histórias ou eventos repentinos, através da análise de dados textuais gerados em redes sociais, blogs e jornais na internet. Desafios: Os conjuntos de dados relacionados a este tipo de problema são muito grandes, com muitas dimensões e muito esparsos, sendo necessários algoritmos mais eficientes e infra-estrutura adequada. Técnicas: Redes Neurais Artificiais; Formação de Agrupamentos; Modelos de Mistura; Máquinas de Vetores de Suporte.
- Outras aplicações: Detectar amostras com rótulos errados em conjuntos de dados; Detecção de mudanças em imagens; Uso conjunto com sistemas de reconhecimento de fala e reconhecimento de locutor; Detectar acidentes e engarrafamentos em no trânsito; Detecção de falhas e fraudes em aplicações WEB; Detecção de fraudes em Saúde Pública; Detecção de *outliers* em dados de diversas áreas: dados biológicos; dados astronômicos; dados de CRM; dados de censos; entre outras áreas.

Além destas aplicações das técnicas de detecção de *outliers*, pode-se também utilizá-las no pré-processamento de conjuntos de dados para tarefas de mineração de dados. A implementação de modelos para previsão de séries temporais, por exemplo, requer um conjunto de dados tratado, que represente, o mais próximo da realidade possível, as características do fenômeno em estudo.

3 REDES NEURAIS AUTO-ASSOCIATIVAS E ALGORITMOS GENÉTICOS

3.1 INTRODUÇÃO

A Inteligência Computacional (IC) é um ramo da Computação que estuda e desenvolve modelos computacionais que apresentam alguma forma de inteligência, similar à exibida por determinados sistemas biológicos. Alguns dos paradigmas que compõem a IC foram de fato inspirados em sistemas biológicos, como as Redes Neurais Artificiais, enquanto que outros, apesar de não terem inspiração biológica, tentam gerar sistemas que produzam algum tipo de comportamento próximo ao observado em sistemas naturais, como por exemplo, o Algoritmo Genético.

As pesquisas sobre redes neurais iniciaram em 1943, quando Warrem McCulloch e Walter Pitts estabeleceram as bases da neurocomputação, concebendo procedimentos matemáticos análogos ao funcionamento do cérebro biológico. Porém estes procedimentos jamais foram testados pelos seus criadores em situações práticas. No entanto, o passo mais importante quando se trata de estudar redes neurais, foi dado por Donald Hebb em 1949, quando ele propôs um modo de condicionar as redes neurais à capacidade de aprendizado. A partir de então, cada vez mais a comunidade científica vem implementando estruturas de Redes Neurais Artificiais (RNA) mais eficazes.

A área da Computação Evolutiva (CE) é composta por várias técnicas computacionais idealizadas como metáforas da natureza. Uma das primeiras metáforas utilizadas é o Algoritmo Genético (AG), proposto por John Holland, no fim da década de 1950, que combina ideias da Teoria da Evolução das Espécies de Charles Darwin, como a seleção natural, com os princípios da genética, como cruzamento e mutação. Algoritmos genéticos são algoritmos de busca e otimização, também aplicados em aprendizado de máquina, que seleciona as soluções mais aptas, em um conjunto de soluções geradas aleatoriamente, e realiza sobre elas operações individuais ou em conjunto, buscando torná-las melhores, o que nem sempre acontece, devido a natureza aleatória da abordagem (GOLDBERG, 1989).

Neste capítulo, considerando que a metodologia proposta nesta dissertação para detecção e correção de *outliers* em séries temporais é baseada em redes neurais e algoritmos genéticos, serão apresentados então alguns fundamentos básicos teórico.

3.2 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais são sistemas de processamento de informação inspirados nas funções cerebrais biológicas, formados pela interconexão de unidades de processamento, denominados neurônios artificiais. Um neurônio artificial (Figura 3.1) é formado por três elementos básicos (HAYKIN, 2009): um conjunto de sinapses ou elos de conexão, cada uma caracterizada por um peso; um somador para somar os sinais de entrada, ponderado pelas respectivas sinapses (pesos sinápticos) do neurônio e uma função de ativação, para restringir a amplitude da saída do neurônio.

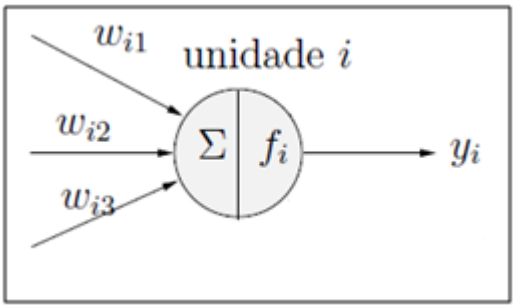

Figura 3.1- Modelo de Neurônio Artificial

Um neurônio artificial (unidade i) computa uma função f_i sobre a soma ponderada das entradas:

$$y_i = f_i(\sum_{j=1}^{n} w_{ij} y_j) \tag{3.1}$$

onde y_i é o sinal de saída da unidade *i*, $f_{i,}$ é a função ativação da unidade *i*, w_{ij} é o peso sináptico que liga a unidade *i* a uma unidade *j* e y_j é a saída da unidade *j* que está interligada a unidade *i*.

As funções de ativação comumente usadas são as funções: logísticas, linear e tangente hiperbólica. A presença da função de ativação é importante, pois pode introduzir o comportamento de não-linearidade inerente aos modelos que a rede neural busca emular (HAYKIN, 2009; AGUIRRE, 2004).

Algumas das vantagens das redes neurais artificiais, conforme HAYKIN (2009), são:

- **Generalização**: capacidade da rede neural de produzir resultados apropriados para entradas não tratadas durante a fase de treinamento, tornando as redes neurais capazes de encontrar soluções aproximadas para problemas complexos de larga escala.

- **Tratamento de problemas não-lineares**: uma rede neural pode ser linear ou não linear, dependendo da função de ativação presente em seus neurônios. Essa possibilidade torna as redes neurais muito úteis em problemas do mundo real, que em geral são não-lineares.
- **Mapeamento entrada-saída**: redes neurais supervisionadas são capazes de aprender o mapeamento entrada-saída de um conjunto de dados utilizado para treinar a rede. Neste cenário os pesos dos neurônios são atualizados de maneira a diminuir a diferença entre a saída desejada e a saída atual gerada pela rede.
- **Adaptabilidade**: através das mudanças nos pesos dos neurônios, as redes neurais podem adaptar-se a mudanças sem perder informações passadas, podendo haver situações em que as redes são implementadas para serem re-treinadas diante de certas condições e mudanças, como quando está operando em ambientes não estacionários.
- **Informação contextual**: redes neurais totalmente conectadas têm a capacidade de lidar com informações contextuais presentes no conjunto de dados, uma vez que cada neurônio de computação é afetado por cada amostra.
- **Tolerância a falhas**: uma rede implementada em hardware, por exemplo, pode sofrer a perda de algum neurônio. Esta perda não afeta o desempenho da rede de maneira drástica, uma vez que o conhecimento da rede está distribuído entre os diversos neurônios e camadas, de forma que a perda deveria se maior para que o desempenho seja degradado seriamente.

3.2.1 **ARQUITETURAS**

Um neurônio pode ser organizado com outros neurônios, de forma a enviar sua saída para a entrada de outro neurônio, formando camadas de neurônios. A arquitetura da rede é definida pela forma na qual os neurônios estão organizados e interconectados, ou seja, o número de camadas, o número de neurônios por camada, tipos de conexão entre os neurônios e a topologia da rede (FACELI et al., 2011; HAYKIN, 2009).

Durante o aprendizado da rede, é possível que a mesma mude sua arquitetura, excluindo ou criando novas conexões entre os neurônios, ou mesmo excluindo neurônios. Isso

acontece motivado em características semelhantes do cérebro humano, em que novas sinapses podem se formar e neurônios podem morrer com o passar do tempo.

Um fator importante na definição da arquitetura da rede é a direção em que os sinais podem fluir pela rede, havendo duas abordagens principais: redes diretas e redes recorrentes. Nas redes diretas, o fluxo dos sinais de entrada percorre a rede da camada de entrada até a camada de saída, e é representada pro um grafo sem ciclos. Redes recorrentes apresentam conexões de realimentação, em que a saída de um neurônio, ou de uma camada inteira de neurônios, realimenta neurônios em camadas anteriores, ou na mesma camada, podendo ocorrer a auto-realimentação, em que a saída de um neurônio realimenta sua própria entrada.

3.2.2 **REDES PERCEPTRON DE MULTIPLAS CAMADAS**

Nas redes Perceptron de Múltiplas Camadas (conhecidas pela sigla MLP, do inglês *Multilayer Perceptron*), além das camadas de entrada e de saída, os neurônios estão organizados também em uma ou mais camadas ocultas, também chamadas de escondidas ou intermediárias, por estarem entre a camada de entrada e a de saída. As camadas ocultas são compostas por neurônios computacionais, e sua função básica é intervir entre a camada de entrada externa e a saída da rede de maneira útil (HAYKIN, 2009).

Cada camada do modelo MLP tem uma função específica:

- Camada de entrada: recebe os sinais de entrada e envia-os para a próxima camada.
- Camadas ocultas: existem uma ou mais camadas ocultas, compostas por nós. São camadas computacionais que efetuam processamento. Nelas são transmitidas as informações por meio das conexões entre as unidades de entrada e saída. Essas conexões guardam os pesos que serão multiplicados pelas entradas, garantindo o conhecimento da rede (FACELI et al., 2011).
- Camada de saída: camada composta por neurônios computacionais que recebem as informações das camadas ocultas fornecendo respostas.

A Figura 3.2 apresenta uma representação em grafo de uma rede do tipo MLP totalmente conectada, onde cada neurônio de uma camada está conectado a todos os neurônios da camada seguinte.

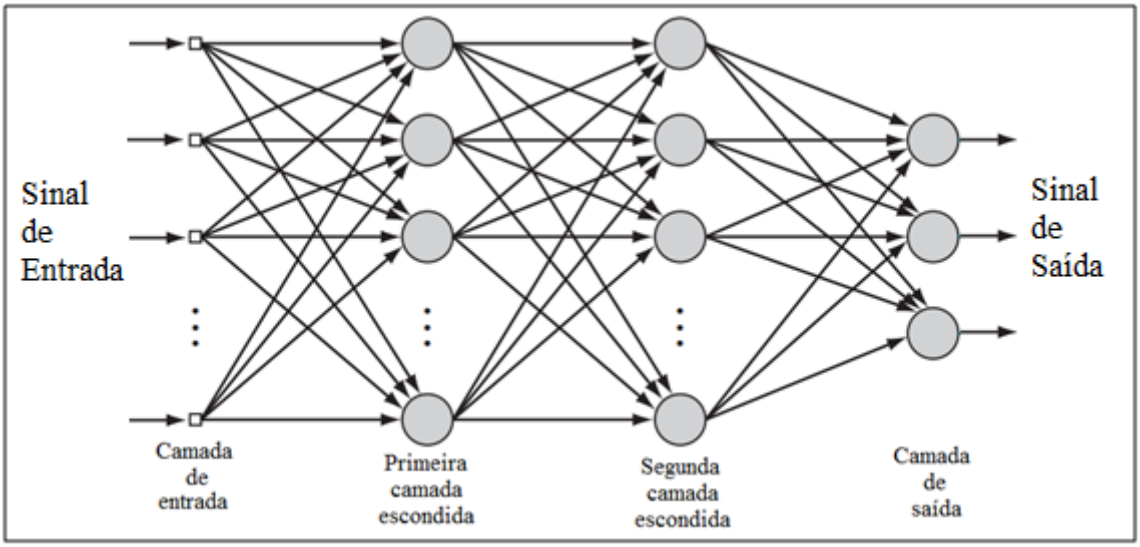

Figura 3.2 - Grafo arquitetural de uma rede MLP com duas camadas escondidas (adaptado de HAYKIN, 2009).

3.2.3 **APRENDIZADO**

Aprendizagem é um processo pelo qual os parâmetros de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (HAYKIN, 2009). Computacionalmente, o aprendizado é implementado na forma de algoritmo de treinamento, que são baseados em técnicas de otimização.

Sabe-se que o aprendizado não é um processo único. Há diferentes processos de aprendizado, isto é, cada aprendizado é adequado a um diferente tipo de rede, sendo que as duas principais classes de aprendizado são: aprendizado supervisionado e aprendizado não-supervisionado.

No aprendizado supervisionado as variáveis de entrada e saída são fornecidas por um supervisor, que tem a função de monitorar as respostas da rede neural artificial para cada valor de entrada. O objetivo é ajustar os parâmetros de tal forma que se possa encontrar uma ligação entre os pares de entrada e de saída da rede. Existem, atualmente, muitos algoritmos de aprendizado, podendo-se citar o algoritmo Backpropagation (RUMELHART, HINTON e WILLIAMS, 1986, apud. HAYKIN 2009), o algoritmo de Levenberg-Marquadt com momento adaptativo (AMPAZIS et al., 2000) , o algoritmo com Regularização Bayesiana (BISHOP, 1995), e algoritmo baseado na otimização com Algoritmos Genéticos (MONTANA e DAVIS, 1989),entre outros.

No aprendizado não supervisionado não existe um supervisor indicando a resposta desejada, de forma que a rede deve aprender a partir das amostras, utilizando regras de aprendizado competitivo, como em técnicas de formação de agrupamentos. Um exemplo de rede neural baseada em aprendizado não supervisionado são os Mapas de Kohonen, também conhecida como *SOM - self-organizing map* (mapa auto-organizável) (HAYKIN, 2009).

O Algoritmo *Backpropagation* é um algoritmo supervisionado que utiliza amostras rotuladas (entrada e saída desejada), para ajustar os pesos dos neurônios, através de um mecanismo de correção de erros (BRAGA, CARVALHO e LUDEMIR, 2007).

O treinamento acontece em duas fases: computação para frente (*forward*) e computação para trás (*backward*). A fase de computação para frente é utilizada para definir a saída da rede utilizando a amostra de entrada, e consiste em realizar o processamento da camada de entrada, passando pelas camadas intermediárias até a camada de saída. A fase de computação para trás utiliza o erro, que é a diferença entre a saída desejada e a saída obtida pela rede, para atualização dos pesos sinápticos, primeiramente na camada de saída, depois nas camadas intermediárias (na camada de entrada não ocorre computação, logo não existem pesos a serem atualizados) (HAYKIN, 2009).

É importante que haja o controle, em abordagens supervisionadas, do fenômeno de super especialização (do inglês *overfitting*, também chamado de *overtraining*). Que consiste em treinar demais a rede, fazendo com que a mesma se especialize nas amostras de treino, mas perca sua capacidade de generalização, gerando resultados ruins para amostras novas.

Uma das formas de evitar o *overfitting* é utilizando a metodologia de treino conhecida como *cross validation* (validação cruzada), em que as amostras do conjunto de dados são divididas em 3 subconjuntos mutuamente exclusivos: treino, validação e teste. O subconjunto de treino é gerado para treinar a rede. A cada época, que consiste em alimentar a rede com todas as amostras do subconjunto de treino, a rede é alimentada com o conjunto de validação e o erro quadrático médio de validação é calculado entre a saída desejada e a obtida pela rede. Enquanto a rede estiver com boa generalização o erro quadrático médio irá diminuir a cada época. Quando este erro começar a aumentar, significa que a rede está entrando em super especialização. O treinamento da rede pode ser interrompido se o erro de validação não voltar a diminuir por algumas épocas, a critério de escolha do projetista. Idealmente, o subconjunto de treino deve ser maior que os outros dois, sendo comumente utilizadas proporções próximas a 70% para treino, 15% para validação e 15% para testes.

3.2.4 **MLP AUTO-ASSOCIATIVAS**

As redes MLP podem ser configuradas de maneira auto-associativa, que consiste em utilizar as amostras de entrada da rede como resultados de saída da mesma, de forma que a rede aprenderá a reproduzir em sua saída os dados de entrada. Dessa forma, uma rede auto associativa aprende as características do conjunto de dados da entrada e se torna capaz de reconhecer amostras pertencentes, ou próximas, a este conjunto, logo, podendo reconhecer também amostras não pertencentes ou distantes do conjunto de treinamento (PIMENTEL et al., 2014, CHANDOLA, BARNEJEE e KUMAR, 2009). A Figura 3.3 apresenta uma rede MLP auto-associativa.

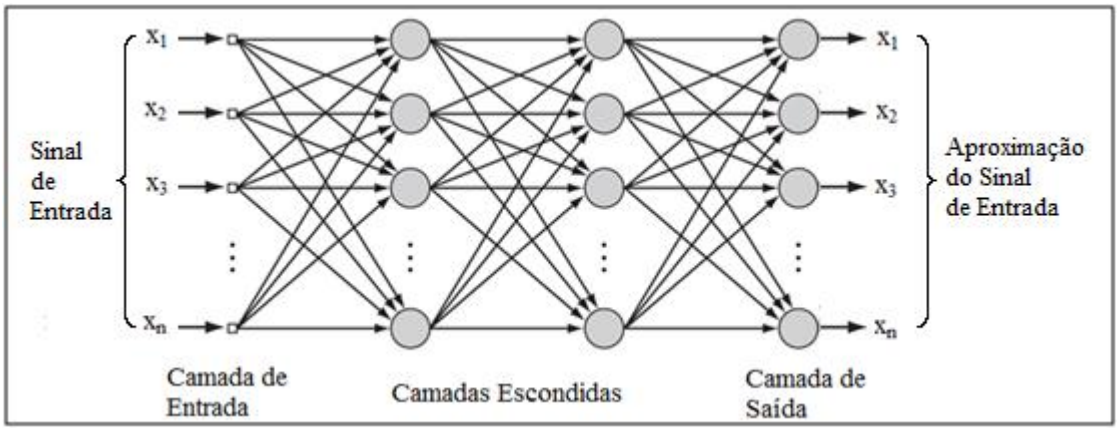

Figura 3.3 - Rede MLP auto-associativa.

Em uma MLP auto-associativa é necessário que o número de neurônios na camada de saída seja igual ao número de neurônios na camada de entrada e a rede pode ter tantas camadas escondidas quanto forem necessárias, cada uma com a quantidade de neurônios suficiente para o melhor desempenho da rede.

Quando alguma camada escondida possuir um número de neurônios menor que a camada de entrada, costuma-se designar a rede de *autoencoder*. A camada escondida funciona como um filtro que permite à rede armazenar conhecimento relacionado apenas às principais características do conjunto de dados de entrada. Esta rede tem sido utilizada amplamente na literatura como uma forma de compressão de dados ou como alternativa a técnicas de Análise de Componentes Principais (MARKOU e SINGH, 2003). Além dessas aplicações, também podem ser utilizadas em tarefas de processamento digital de imagens, reconhecimento de objetos através de dados de múltiplos sensores e restauração de dados ausentes.

3.3 **ALGORITMO GENÉTICO**

Algoritmo Genético (AG) é uma técnica computacional utilizada para resolução de problemas de otimização, baseada na teoria da evolução, criada por John Holland e desenvolvida por ele e seus estudantes na Universidade de Michigan, entre o fim décadas de 1950 até meados de 1970. Apesar de sua popularização na solução de problemas de otimização, AGs também são utilizados em problemas de simulação de comportamento, em que se deseja modelar e descobrir as consequências de uma decisão, dado um conjunto de fatores. AGs têm sido amplamente estudados e utilizados, por se tratar de uma técnica genérica, que pode ser facilmente adaptada a diversos problemas (SIVANANDAM e DEEPA, 2007).

Segundo o processo de seleção natural, proposto por Charles Darwin e Alfred Wallace, os organismos que melhor se adaptam a seu ambiente têm maiores chances de ter suas características reproduzidas em uma nova geração. A genética explica como os filhos herdam características dos pais e como podem acontecer mutações genéticas, alterando características da prole.

A utilização desses conceitos em um algoritmo computacional para solução de um problema real ocorre, de forma simplificada: é gerada uma população inicial, que é um conjunto de possíveis soluções para o problema. Busca-se então, em um processo iterativo, gerar uma boa solução por meio da evolução das melhores soluções da população atual, de acordo com uma função objetivo, que mede a qualidade de cada solução. Para a definição de cada nova população a partir das soluções escolhidas, três operadores genéticos são geralmente aplicados: elitismo (cópia simples das melhores soluções), cruzamento (combinação de partes de pares de soluções) e mutação (altera a composição de algumas soluções, permitindo a criação de soluções que ainda não foram observadas) (FACELI et al., 2011). A Figura 3.4 apresenta um diagrama esquemático pelo qual pode-se compreender estes passos.

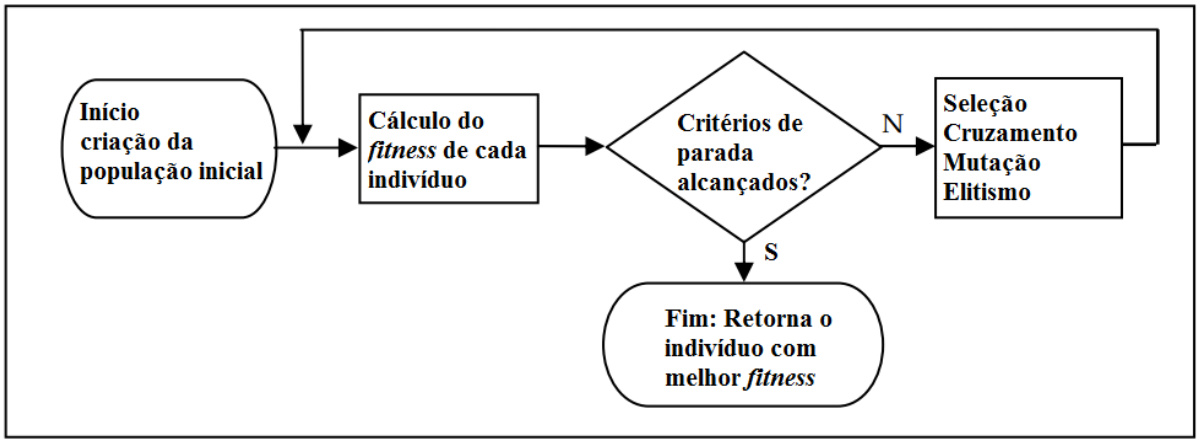

Figura 3.4 - Esquema de execução de um algoritmo genético.

Algumas das aplicações possíveis para os AGs são (SIVANANDAM e DEEPA, 2007):

- Planejamento de trajetórias de robôs;
- Planejamento estratégico;
- Funções para criar imagens ou música;
- Projeto de arquitetura e treinamento de redes neurais artificiais;
- Projeto de filtros de processamento de sinais;
- Encontrar forma de moléculas protéicas;
- Problemas de otimização combinatória, como caixeiro viajante, roteamento, coloração e partição de grafos;
- Problemas de agendamento e alocação de recursos;
- Projeto de aeronaves, dentre outras.

3.3.1 CROMOSSOMO

Em AGs, um cromossomo é uma possível solução para o problema, também pode ser chamado de indivíduo (Figura 3.5). Frequentemente os cromossomos são representados por vetores (*strings*) binários. Nesses vetores, cada elemento, denominado gene, representa a presença (valor 1) ou ausência (valor 0) de alguma característica (FACELI et al., 2011). Valores inteiros e valores reais também têm sido utilizados e a escolha da codificação dos indivíduos depende da natureza do problema tratado.

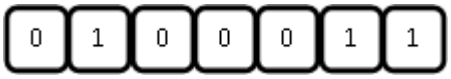

Figura 3.5 - Exemplo de cromossomo formado por uma *string* de bits.

Devido às dificuldades encontradas durante a escolha da codificação dos indivíduos, Linden (2012) propõe que as seguintes regras sejam seguidas, para facilitar a representação e evitar problemas:

- A representação deve ser a mais simples possível;
- Soluções proibidas não devem ser representadas;
- As condições impostas pelo problema devem estar implícitas na representação.

3.3.2 FUNÇÃO OBJETIVO (*FITNESS*)

A função objetivo (também chamada de função de custo ou função *fitness*) é responsável por gerar um valor para cada possível solução, pelo qual a mesmo será avaliada. Este valor é também chamado de *fitness* e funciona como uma pontuação, que deve ser proporcional ao grau de aptidão do indivíduo, ou seja, indivíduos mais aptos devem ser melhor pontuados. Através do *fitness* é possível comparar as soluções, permitindo selecionar as melhores e excluir as piores (MITCHELL, 1999).

A escolha da função objetivo depende da natureza do problema e da codificação escolhida para representar a solução do problema, de forma que quanto mais inserirmos conhecimento sobre o problema na função objetivo, como suas restrições e padrões de qualidade, mais específico se torna o AG (LINDEN, 2012).

Segundo Sivanandam e Deepa (2007), a complexidade ou dificuldade de escolha desta função representa uma das principais desvantagens do uso de algoritmos genéticos. Os autores afirmam que é possível escolher como função objetivo qualquer coisa que possa ser avaliada por um computador, ou mesmo pode-se usar um avaliador humano, que pode estabelecer o *fitness* de cada indivíduo usando critérios complexos ou difíceis de serem programados, como a definição de beleza, por exemplo. Dessa forma, conclui-se que não há limitações matemáticas para a função objetivo.

Para problemas de maximização ou minimização de funções matemáticas complexas, por exemplo, fica claro que a função objetivo deve ser a própria função a ser otimizada e o valor de *fitness* de cada cromossomo deve ser o próprio valor da função (SIVANANDAM e

DEEPA, 2007). Entretanto, isso não é tão claro em problemas de otimização combinatória, por exemplo, em que muitas variáveis devem ser analisadas, como no caso do projeto de uma ponte, em que deve-se otimizar características como: razão força/peso; carga máxima; largura; custo; e tempo de construção.

3.3.3 SELEÇÃO

O processo de seleção tem por objetivo atuar sobre o conjunto de soluções de forma semelhante a atuação do processo de seleção natural, selecionando os indivíduos que devem gerar, passando características para a próxima geração. É importante definir que o processo de seleção também defina de quantos cruzamentos um indivíduo pode participar.

Tal mecanismo deve privilegiar indivíduos mais aptos, sem excluir a possibilidade que indivíduos menos adaptados também possam gerar filhos. Isso é importante, pois soluções que possuem *fitness* baixo podem possuir características importantes de serem passadas para gerações futuras, sendo combinadas com outras soluções (SIVANANDAM e DEEPA, 2007).

O uso da probabilidade faz com que o indivíduo com maior valor de aptidão tenha maiores chances de participar da fase de reprodução. Como resultado, indivíduos cada vez mais aptos são gerados, e indivíduos menos aptos tendem a desaparecer (FACELI et al., 2011). O mecanismo de seleção atua como uma pressão que tende a empurrar a execução do AG para a convergência, de acordo com o método escolhido e a configuração de seus parâmetros. Deve-se encontrar um equilíbrio nessa pressão, pois quanto mais leve ela for, mais tempo tem-se para o algoritmo convergir. Do contrário, se essa força for muito intensa, o algoritmo tende a convergir prematuramente para um ótimo local.

Existem vários métodos para seleção dos pais. Os mais conhecidos são seleção por roleta e seleção por torneio.

3.3.3.1 Seleção por Roleta

Cada indivíduo da população é representado em uma roleta por uma fatia proporcional ao seu *fitness* e a soma das fatias deve ser igual a 100%. Para selecionar *n* indivíduos, a roleta é girada *n* vezes, selecionando a cada rodada o indivíduo cuja fatia é apontada pela agulha. Obviamente, não é possível "girar uma roleta" no computador, assim,

este processamento é feito utilizando técnicas de programação de computadores e geração de números pseudo aleatórios. A Tabela 3.1 apresenta uma possível composição de uma roleta.

Tabela 3.1 - Exemplo de formação de roleta para seleção.

Indivíduo	*Fitness*	Proporção da Roleta	Tamanho da Fatia
0110	4	6,67 %	24 °
1001	8	13,33 %	48 °
1101	16	26,67 %	96 °
1111	32	53,33 %	192 °
Total	60,00	100,00 %	360 °

Este método tende a convergir rapidamente conforme a diferença entre os *fitness* aumenta. Isso pode dar origem a problemas, sendo possível que o melhor indivíduo de uma população ocupe 90% da roleta e os demais indivíduos ocupem apenas 10%, muito provavelmente convergindo prematuramente.

3.3.3.2 Seleção por Torneio

A seleção de *n* indivíduos para a fase de cruzamento se dá pela realização de *n* torneios. Um torneio consiste em selecionar aleatoriamente dois ou mais indivíduos da população atual, e comparar seus valores de *fitness*, sendo vencedor o indivíduo que tiver o maior *fitness*.

Dessa forma, a pressão seletiva é exercida não pelo valor absoluto de *fitness* dos indivíduos, mas pela comparação entre eles, semelhante a um *ranking*, de forma que um indivíduo com *fitness* alto têm vantagem sobre os demais indivíduos ao participar de um torneio, mas tem a mesma chance de ser escolhido para um torneio que os demais indivíduos. Além disso, a quantidade de indivíduos participantes do torneio também aumenta a pressão seletiva, de forma que torneios com 2 indivíduos podem convergir mais lentamente que torneios com 5 indivíduos. Outra forma de diminuir a pressão seletiva é utilizando o torneio com probabilidades, em que o indivíduo com melhor *fitness* é escolhido de acordo com uma probabilidade *p*, sendo $p>0,5$¸ ou o indivíduo com menor fitness é escolhido, com probabilidade $q=1-p$.

3.3.4 CRUZAMENTO

Cruzamento é um outro importante operador genético aplicado sobre dois indivíduos, ou mais, como o objetivo de criar novos indivíduos. Não necessariamente todos os indivíduos selecionados devem realizar o cruzamento. Assim, a taxa de cruzamento é uma variável utilizada para controlar a frequência com que acontecerá o cruzamento. Uma taxa de cruzamento de 100% indica que sempre acontecerá o cruzamento, e uma taxa de 0% indica que nunca acontecerá o cruzamento (SIVANANDAM e DEEPA, 2007).

O operador de cruzamento tradicional produz dois novos indivíduos (filhos) a partir de dois indivíduos selecionados (pais). No cruzamento de um ponto, o mais simples, um ponto de corte divide cada pai em duas partes. Cada filho recebe uma parte de um dois pais. A Figura 3.6 ilustra o funcionamento do operador de cruzamento de um ponto (FACELI et al., 2011).

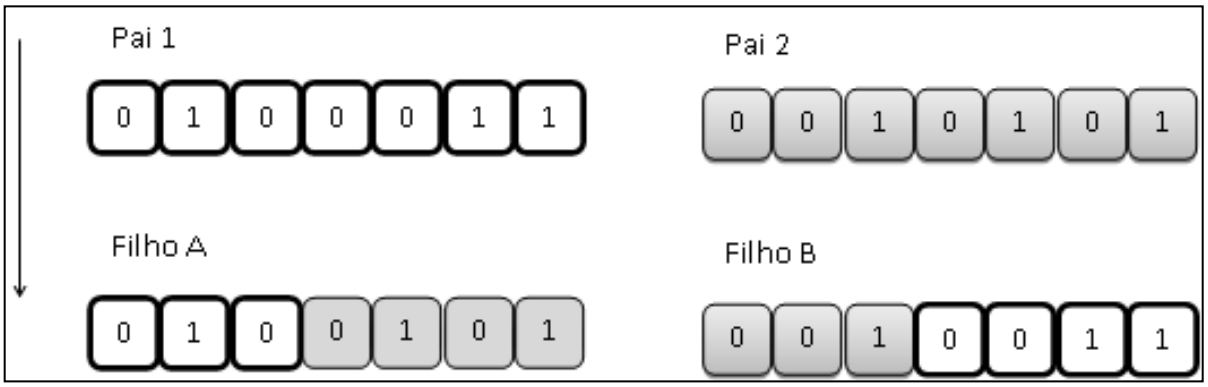

Figura 3.6 - Exemplo de cruzamento por um corte.

O cruzamento de um ponto de corte pode ser generalizado para o chamado cruzamento de *n*-pontos, em que são sorteados *n* pontos de corte e a cada ponto altera-se o indivíduo que fornece as características para os filhos.

Outro tipo de cruzamento é o cruzamento uniforme, em que, a cada gene dos filhos, uma moeda é lançada e caso o resultado da moeda seja "cara" o filho recebe a característica de um pai, caso seja "coroa", recebe a característica do outro pai. Novamente, lançar uma moeda é uma abstração, dado que na realidade é sorteado um número aleatório dentre dois possíveis.

3.3.5 MUTAÇÃO

O operador de mutação evita que o algoritmo fique preso em ótimos locais. Ele faz isso introduzindo diversidade genética na população, permitindo que o AG explore todo o espaço de busca de soluções. Além disso, a mutação possibilita que a população recupere informação genética perdida, que não pode ser recuperada apenas por meio do cruzamento (S e D). Para isso, ele altera aleatoriamente um ou mais genes do cromossomo selecionado, conforme a Figura 3.7.

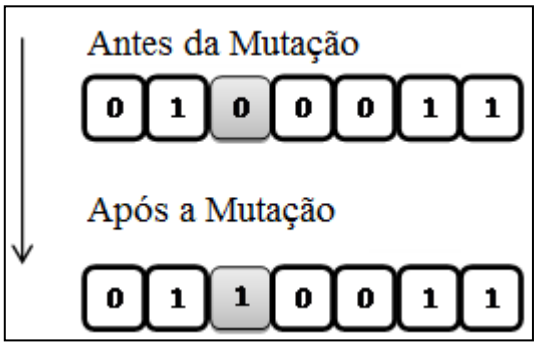

Figura 3.7 - Exemplo de Mutação em um gene de um indivíduo formado por uma *string* de bits.

Sua aplicação é definida de forma probabilística utilizando uma taxa de mutação, que varia de 0% a 100%, e pode ser aplicado da seguinte forma: a cada indivíduo gerado após o cruzamento é sorteado um número de 1 a 100 e se o número gerado for menor ou igual à taxa de mutação, aquele indivíduo deve sofrer mutação em um de seus genes. A escolha de qual gene será afetado também é feita aleatoriamente e quanto mais próximo de 100% mais o algoritmo se parecerá com um "*random walk*", com comportamento extremamente aleatório. Uma possível solução para isso é utilizar uma taxa de mutação que varia de acordo com o comportamento do algoritmo.

Após aplicar os operadores aos indivíduos durante algumas gerações, é necessário estabelecer critérios claros para a parada do algoritmo. Alguns dos critérios de parada utilizados, também chamados de critérios de convergência, são (SIVANANDAM e DEEPA, 2007):

- **Melhor *fitness***: um valor de fitness a ser alcançado é estabelecido e no momento que um indivíduo alcançar este fitness a execução termina.
- **Quantidade máxima de gerações**: uma quantidade máxima de gerações é estabelecida e, caso nenhum outro critério seja satisfeito antes, a execução do algoritmo termina quando esse valor é alcançado.

- **Tempo de execução**: um tempo máximo de execução do algoritmo é fixado no início de sua execução de forma que o algoritmo é interrompido após decorrido este tempo.
- ***Fitness* estável por *n* gerações**: um valor *n* é fixado no início da execução do algoritmo e caso o fitness do melhor indivíduo não melhore por *n* gerações, o algoritmo encerra sua execução.
- **Tempo sem melhora**: semelhante ao critério anterior, uma quantidade de tempo *t*, geralmente em segundos, é determinada e caso o fitness do melhor indivíduo não melhore por *t* segundos, o algoritmo encerra.
- **Mediana**: um valor de convergência é estabelecido e quando metade dos indivíduos possuir fitness maior ou igual a este valor, o algoritmo pára, fornecendo uma quantidade considerável de possíveis soluções.

4 METODOLOGIA PARA DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS

4.1 INTRODUÇÃO

A implementação de modelos de previsão de séries temporais, sejam estatísticos ou de aprendizado de máquina, é uma tarefa de aprendizado supervisionado, ou semi supervisionado em alguns casos, sendo dependente de um conjunto de dados para implementação do modelo. O conjunto de dados deve conter o menor número possível de *outliers* e dessa forma, além de técnicas para detecção dos *outliers*, é necessária também uma técnica para tratamento destes *outliers*, de forma que seja mínimo os seus efeitos sobre o modelo de previsão a ser implementado.

Modelos robustos apresentam bons resultados mesmo na presença de *outliers* e têm sido largamente utilizados na comunidade científica, porém não conseguem superar os resultados de modelos de previsão implementados com conjuntos de dados sem *outliers*. Dessa forma, o estudo e aplicação de técnicas de detecção e tratamento de *outliers* é importante, sobretudo quando voltados ao pré processamento dos dados de uma série temporal.

Considerando a importância do desenvolvimento de sistemas de previsão de séries temporais baseados em conjunto de dados sem a presença de *outliers*, neste capítulo será apresentada uma nova metodologia para detecção e correção de *outliers* em séries temporais. A metodologia é baseada em técnicas de Inteligência Computacional, sendo utilizada uma rede neural auto-associativa em conjunto com algoritmos genéticos. Será apresentada a metodologia em detalhes e no capítulo posterior um estudo de caso de aplicação da metodologia em uma série temporal de consumo de energia.

4.2 SÉRIES TEMPORAIS

Toda atividade realizada ao longo do tempo pode ser medida, dando origem a uma série temporal, que é um conjunto de observações ordenadas no tempo, em que haja uma dependência entre instantes de tempo consecutivos (MORETTIN e TOLÓI, 2006). Geralmente, trabalha-se com séries temporais em que o intervalo de tempo entre as amostras é o mesmo.

Diversas áreas do conhecimento utilizam séries temporais como base de estudo. Estudos relacionados a séries temporais podem ter diversos objetivos: analisar o mecanismo gerador da série; realizar predição de valores futuros das variáveis de estudo; descrever o comportamento da série; verificar a existência de tendências, ciclos e variações sazonais na série e monitorar os valores da série, buscando identificar alterações no mecanismo gerador. Segue uma listagem com algumas séries temporais encontradas em áreas de conhecimento específicas:

- Engenharia: vazão de uma usina hidrelétrica medida a cada hora; energia produzida e consumida por dia; demanda por largura de banda em redes DSL;
- Economia: variação de índices econômicos (IPCA, IPI, etc.) ao longo do tempo; valores das ações de empresas; lucro e faturamento de empresas; taxa de desemprego;
- Medicina: quantidade mensal de casos de uma doença; taxa de batimentos cardíacos por minuto; exames como eletrocardiograma e eletroencefalograma.
- Meteorologia: medição das temperaturas máxima e mínima diária; índice diário de precipitação pluviométrica; medição da velocidade do vento;

Séries temporais podem ser univariadas ou multivariadas. As séries univariadas consistem em medições de uma única variável, como por exemplo a temperatura. Séries multivariadas consistem em duas medições de duas ou mais variáveis no mesmo instante, por exemplo temperatura, consumo de energia e velocidade do vento.

4.3 METODOLOGIA PARA DETECÇÃO E CORREÇÃO DE *OUTLIERS* EM SÉRIES TEMPORAIS

A metodologia proposta para detecção e correção de *outliers* em séries temporais univariadas segue o esquema apresentado na Figura 4.1.

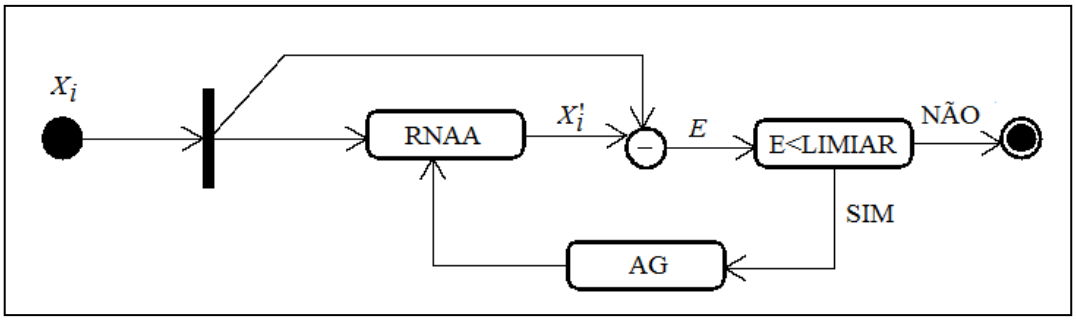

Figura 4.1 - Esquema de detecção e correção de *outliers*.

A metodologia consiste em aplicar um método de detecção de *outliers* baseado em reconstrução utilizando uma rede neural artificial do tipo Perceptron de Múltiplas Camadas auto-associativa (RNAA). Esta rede neural, em conjunto com um algoritmo genético (AG), será utilizada posteriormente para o processo de correção dos *outliers* detectados.

A característica de reconstrução das redes auto-associativas para amostras pertencentes a um processo gerador é especialmente interessante para o problema de detecção de *outliers*. Uma vez treinada a rede auto-associativa, é possível posteriormente apresentar amostras para a RNAA, que terá a capacidade de reconhecer a existência de amostras pertencentes ao processo gerador e descartar amostras que possam possuir valores distantes das amostras do processo gerador. Para avaliar a saída gerada pela RNAA a abordagem mais simples consiste em calcular a somatória dos erros absolutos entre as saídas da rede e as entradas (E). Quanto maior for o valor da somatória dos erros, maior será a probabilidade da amostra possuir algum valor que não pertença ao processo gerador, podendo então ser considerado um *outlier*. Para que possa ser realizada então a detecção de *outliers* um valor limiar para a somatória dos erros na saída da RNAA deve ser definido, sendo que valores da somatória dos erros maiores que o limiar indicarão a presença de *outlier* na amostra analisada.

Os trabalhos de Markou e Singh (2003), Chandola, Barnejee e Kumar (2009) e Pimentel et al. (2014) citam diversos trabalhos que utilizam redes MLP auto associativas em tarefas de detecção de *outliers* em diversas áreas de conhecimento, desde a década de 1990 até a atualidade. Entretanto, não foi possível encontrar na literatura algum trabalho que use as redes MLP auto-associativas para detectar *outliers* em séries temporais.

Considerando o esquema da Figura 4.1, quatro passos são necessários para se obter a série temporal livre de *outliers*:

- **Pré-processamento da série temporal**, formando um conjunto de dados adequado para a tarefa.

- **Treinamento e validação da RNAA**, obtendo um modelo responsável pela reconstrução de um conjunto de dados original.
- **Detecção de *outliers***, através do cálculo do erro de reconstrução E (somatória dos erros absolutos entre as saídas da rede e as entradas) .
- **Correção online dos *outliers* detectados,** utilizando a RNAA em conjunto com um algoritmo genético (AG).

A seguir serão apresentados com detalhes todos os quatro passos utilizados na metodologia proposta.

4.3.1 PRÉ-PROCESSAMENTO DOS DADOS

A série temporal é utilizada para formar um conjunto de dados X que será posteriormente utilizado para treinamento e validação de uma RNAA. Esse conjunto é formado através de uma adaptação da técnica de "*janelamento de sinais*" (HARRIS, 1978). Para isso, deve-se definir o tamanho da janela e o passo do janelamento. O tamanho da janela define quantos valores da série temporal devem compor uma única amostra do conjunto X sendo o valor 5 escolhido para a janela. O valor do passo do janelamento deve ser 1, de forma que todos os valores da série temporal estejam presentes em *X*.

Assim, cada amostra X_i de *X* consiste em um vetor com cinco valores consecutivos da série temporal, e cada valor da série temporal estará presente em até 5 amostras consecutivas do conjunto de dados *X*, uma vez em cada posição. A Figura 4.2 apresenta a técnica de janelamento da série temporal.

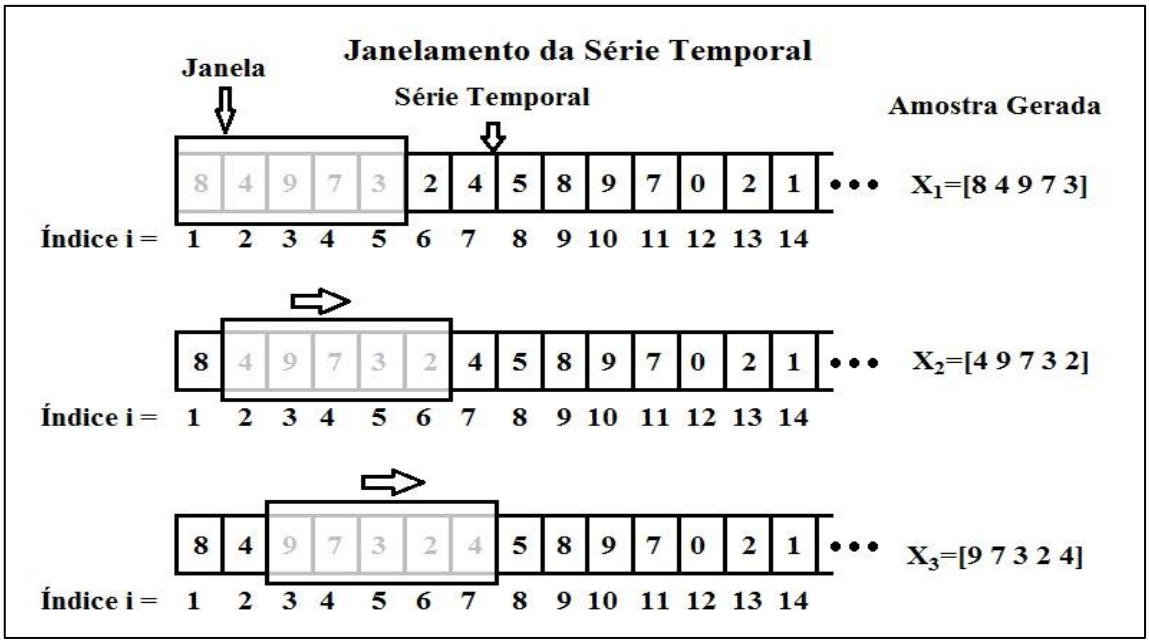

Figura 4.2 - Técnica de janelamento aplicada a uma série temporal univariada.

Tem-se então $X_i = [x_{\mathrm{i}}\ x_{\mathrm{i+1}}\ x_{\mathrm{i+2}}\ x_{\mathrm{i+3}}\ x_{\mathrm{i+4}}]$, onde x_{i} é o valor da série temporal no instante *i*; $x_{\mathrm{i+1}}$ é o valor da série temporal no instante *i+1*; $x_{\mathrm{i+2}}$ é o valor da série temporal no instante *i+2*, $x_{\mathrm{i+3}}$ é o valor da série temporal no instante *i*+3 e $x_{\mathrm{i+4}}$ é o valor da série temporal no instante *i+4*.

Com o conjunto de dados *X* formado, após percorrer toda a série temporal, tem-se então uma base de dados que poderá então ser utilizada para treinamento e validação da RNAA.

4.3.2 TREINAMENTO E VALIDAÇÃO DA RNAA

Com o conjunto de dados *X* pode-se iniciar o treinamento e validação da RNAA. No Capítulo 3 foi mostrado que uma RNAA treinada tem a capacidade de realizar uma associação de forma a reproduzir na saída os valores de entrada. A Figura 4.3 apresenta a estrutura da RNAA que deverá ser treinada para futuramente ser utilizada no processo de detecção de *outliers*. Tem-se como entrada e saída desejada para a RNAA as amostras do conjunto de dados *X*.

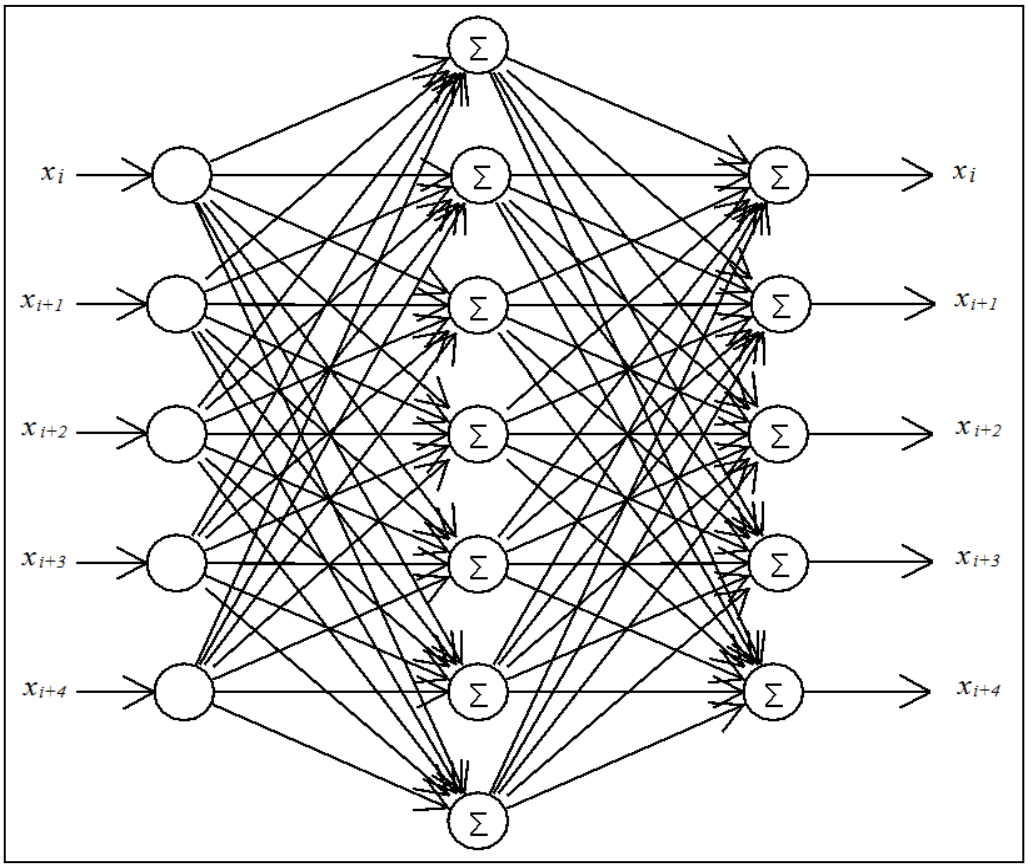

Figura 4.3 - Rede neural utilizada para detecção e correção de *outliers*.

Para realizar o treinamento da RNAA deverá ser utilizado o algoritmo *Backpropagation* com variante *Levenberg-Marquardt*, utilizando o treinamento com validação cruzada, conforme detalhado na seção 3.2.3, onde divide-se o conjunto de dados em 3 três subconjuntos: treino (ajuste dos pesos - 70% das amostras), teste (verificar generalização - 15% das amostras) e validação (evitar *over-fitting* - 15% das amostras). Diversos treinos deverão ser realizados variando-se o número de neurônios na camada escondida. É importante salientar aqui que como o objetivo é apenas a reprodução das amostras da entrada da rede na saída da mesma, não existe então a restrição para a RNAA do número de neurônios na camada escondida ser menor que o número de neurônios na camada de entrada e saída.

Após a realização dos diversos treinamentos, a melhor rede neural, que obteve melhor resultado em relação ao erro médio quadrático na saída tanto para dados de treino quanto para dados de validação será escolhida. Um fator importante para que as redes neurais obtenham bons resultados, o conjunto de dados deve ser suficientemente grande, de forma a representar bem o comportamento da variável de análise.

Com a RNAA já treinada e validada, deve ser iniciado o processo para escolha do valor limiar para o erro de reconstrução E (somatória dos erros absolutos entre as saídas e as entradas da rede) para ser utilizado posteriormente para o processo detecção de *outlier*, conforme a Equação 4.1:

$$E(X_i) = \sum_{j=1}^{5} |x_j - x'_j| \qquad \text{Equação (4.1)}$$

em que X_i é a amostra sendo reconstruída pela rede neural, x_j é *j*-ésima entrada na rede neural e x'_j é a saída reconstruída pela rede neural, correspondendo a *j*-ésima entrada.

Para tanto, deve-se apresentar para a RNAA treinada todas as amostras utilizadas na fase de treinamento e então calcular o erro de reconstrução para cada amostra sendo que o maior valor obtido será considerado como o valor limiar para o erro de reconstrução do sistema de detecção de *outliers*.

4.3.3 DETECÇÃO DE *OUTLIERS*

Com a RNAA devidamente treinada e validada e com valor de limiar de erro de reconstrução definido, tem-se então o sistema para detecção de *outliers* . Para a realização da detecção de *outliers* na série temporal os seguintes passos deverão ser seguidos:

- Passo 1 - **Criar amostra X_i que será apresentada a RNAA treinada**: a partir da série temporal a ser avaliada, série esta que deve ter sido obtida do mesmo processo gerador (mesma distribuição) dos dados utilizados para treinamento da RNAA, formar o primeiro vetor de entradas que será apresentado à rede para detecção de *outliers*, $X_i = [x_{\mathrm{i}}\ x_{\mathrm{i+1}}\ x_{\mathrm{i+2}}\ x_{\mathrm{i+3}}\ x_{\mathrm{i+4}}]$, sendo *i=1*. Para formação deste vetor deverá será utilizada a técnica de janelamento apresentada anteriormente.
- Passo 2 - **Encontrar a primeira amostra de *X* que não contenha *outliers***: deve-se submeter à primeira amostra do conjunto X à RNAA e obter o erro de reconstrução. Se o erro de reconstrução for maior que o limiar, a amostra contém um ou mais *outliers.* Nesse caso, deve-se submeter a segunda amostra à RNAA e calcular o erro de reconstrução, caso o erro seja maior que o limiar, deve-se calcular o erro da próxima amostra, e assim por diante. Este processo iterativo termina quando uma amostra de X obtiver erro de reconstrução menor que o limiar, sendo assim, esta amostra será a primeira

amostra sem *outliers* e a partir desta amostra será então iniciado o processo para a busca de valores que sejam *outliers* na série temporal.

- Passo 3 – **Detectar e Corrigir os *outliers* do início da série temporal**: Considerando a amostra X_i obtida no passo 2 como a primeira amostra sem *outlier*, deve-se antes de prosseguir na série temporal, corrigir os *outliers* contidos nas amostras anteriores a X_i , que foram detectadas no passo 2 como amostras que possuíam erro de reconstrução maior que o limiar, indicando presença de *outlier* na amostra. Para isso uma nova amostra $X_{i-1} = [\, x_{\mathrm{i}-1}\; x_{\mathrm{i}}\; x_{\mathrm{i}+1}\; x_{\mathrm{i}+2}\; x_{\mathrm{i}+3}]$ deve ser criada caminhando agora (janelamento) em direção ao início da série temporal. Apresentar esta amostra a RNAA e calcular o erro de reconstrução:
  1. Caso o erro seja menor que o limiar, o *outlier* não foi detectado e deve-se repetir o Passo 3, decrementando o valor de *i* até que o início da série temporal seja alcançado.
  2. Caso o erro seja maior que o limiar, a presença de um *outlier* é detectada. Neste caso, considerando que X_i não continha *outliers*, pode-se afirmar que o valor $x_{\mathrm{i}+5}$ é o *outlier*, pois é o valor diferente, que não aparece em X_i, que gerou o aumento do erro de reconstrução. Com o *outlier* detectado, realiza-se então a correção deste *outlier*. Após a correção, repetir o Passo 3, até que o início da série temporal seja alcançado.
- Passo 4 - **Detectar e Corrigir os *outliers* seguindo na série temporal a partir de X_i encontrado no passo 2** : Considerando a amostra $X_i = [x_{\mathrm{i}}\; x_{\mathrm{i}+1}\; x_{\mathrm{i}+2}\; x_{\mathrm{i}+3}\; x_{\mathrm{i}+4}]$ como a primeira amostra sem *outlier*, então deve-se incrementar o valor de *i* e formar uma nova amostra $X_{i+1} = [\, x_{\mathrm{i}+1}\; x_{\mathrm{i}+2}\; x_{\mathrm{i}+3}\; x_{\mathrm{i}+4}\; x_{\mathrm{i}+5}]$, seguindo a sequência da série temporal. Apresentar esta amostra a RNAA e calcular o erro de reconstrução:
  1. Caso o erro seja menor que o limiar, deve-se prosseguir na série temporal, incrementando o valor de *i*, repetindo o Passo 4, até chegar ao fim da série temporal.
  2. Caso o erro seja maior que o limiar, a presença de um *outlier* é detectada. Neste caso, considerando que X_i não continha *outliers*, pode-se afirmar que o valor X_{i+5} é o *outlier*, pois é o valor diferente, que não aparece em X_i , que gerou o aumento do erro de reconstrução. Com o *outlier*

detectado, realizar então a correção deste *outlier* para que se possa seguir em frente na série temporal em busca de outros *outliers*.

4.3.4 CORREÇÃO DE *OUTLIERS*

A correção do *outlier* é realizada através do uso de um algoritmo genético em conjunto com a rede auto-associativa, conforme o esquema da Figura 4.1.

Assim que o *outlier* é detectado na amostra $X_{i+1} = [\, x_{i+1}\, x_{i+2}\, x_{i+3}\, x_{i+4}\, x_{i+5}]$, o algoritmo genético deverá ser utilizado para ir em busca do valor da série temporal que deverá substituir o *outlier* encontrado em x_{i+5} . Sendo assim, o objetivo do AG será a busca pelo valor de x_{i+5} que minimizará o erro de reconstrução da Equação 4.1. Ou seja, a Equação 4.1 será a função *fitness* do algoritmo genético.

A amostra, $X_{i+1} = [\, x_{i+1}\, x_{i+2}\, x_{i+3}\, x_{i+4}\, x_{i+5}]$ será o indivíduo utilizado para calcular o valor de *fitness*. Porém, apenas o quinto gene, x_{i+5}, sofrerá mudanças em seu valor, pois este é o *outlier* detectado. A representação do indivíduo do AG é ilustrada na Figura 4.4.

Figura 4.4 - Representação dos indivíduos do AG, em que cada gene é um valor da amostra.

Os critérios de parada do AG podem ser: quantidade máxima de gerações, quantidade de gerações sem alteração no *fitness* do melhor indivíduo, ou pode encerrar assim que obter um erro de reconstrução menor que o limiar. A escolha deste critério deve ser feita de acordo com os resultados das simulações, o que depende de cada conjunto de dados. Para conjunto de dados do estudo de caso deste trabalho, foram utilizados: Máximo de 100 gerações e Máximo de 5 gerações sem melhora no *fitness* do melhor indivíduo.

Após o término da execução do AG, deve-se selecionar o novo valor de x_{i+5}, que será o quinto gene do indivíduo com melhor *fitness*. Este valor deve ser inserido na série temporal, substituindo o valor original de x_{i+5} e então se pode prosseguir a avaliação da próxima amostra para detecção e correção de outros *outliers* que possam ocorrer na série temporal. Sendo que agora o índice *i* corresponde a amostra X_i que teve o *outlier* corrigido, logo é a nova amostra a ser considerada sem *outlier* e *i+1* será a próxima amostra a ser avaliada .

5 ESTUDO DE CASO EM UMA SÉRIE TEMPORAL DE CONSUMO DE ENERGIA

5.1 INTRODUÇÃO

A tomada de decisão no setor de energia e seus desdobramentos para o desenvolvimento da infraestrutura do País dependem fundamentalmente de previsões do consumo de energia elétrica. Uma previsão de qualidade para a demanda futura do consumo de energia elétrica é fundamental para aprimorar a gestão do sistema energético e seus processos operacionais (MINISTÉRIO DE MINAS E ENERGIA, 2009).

Diversos modelos para previsão do consumo de energia têm sido apresentados na literatura, entretanto, para que modelos de previsão confiáveis sejam obtidos, as séries temporais de consumo de energia devem passar por um pré-processamento dos dados antes que possam ser utilizadas na construção do modelo previsor.

Em sistemas de energia, problemas no sistema de medição podem ocorrer, acarretando séries temporais com dados espúrios ou inconsistentes com o resto da série. Para o desenvolvimento de sistemas de previsão de carga é fundamental que esses dados espúrios, geralmente chamados de *outliers*, sejam detectados e corrigidos, visto que a qualidade do histórico de dados pode afetar diretamente a qualidade da previsão.

Considerando então a importância para o desenvolvimento de sistemas de previsão de carga a partir de séries temporais sem a presença de dados espúrios ou inconsistentes (*outliers*), neste capítulo é apresentado um estudo de caso com a aplicação da metodologia proposta nesta dissertação para detecção e correção de *outliers* em uma série temporal de consumo de energia elétrica no estado do Pará entre os anos de 2005 e 2006. Os resultados alcançados demonstram a aplicabilidade da metodologia para o problema de detecção e correção de *outlier* em séries temporais de carga, mostrando-se assim uma ferramenta útil para auxilio ao problema de construção de sistemas de previsão de carga mais confiáveis.

5.2 PROBLEMAS TÍPICOS EM SISTEMAS DE MEDIÇÃO DE CARGA

As séries temporais fornecem em geral informações de fenômenos complexos, e muitas vezes estão sujeitas a eventos inesperados ou incontroláveis. Observações espúrias, que são inconsistentes com as demais observações da série, podem surgir a partir destes eventos inesperados.

Em sistemas de medição de carga alguns problemas podem surgir originando séries temporais com dados que podem ser considerados, segundo a definição da literatura, *outliers*. Dentre os problemas típicos em sistemas de medição de carga destacam-se: ausência de dados, mudanças de nível e picos.

No estudo de caso apresentado neste trabalho será aplicada a metodologia para os casos de *outliers* gerados a partir dos problemas de ausência de dados e picos.

5.2.1 **AUSÊNCIA DE DADOS**

A ausência de dados está geralmente associada a problemas no próprio sistema de medição de dados. Em séries temporais de consumo de energia pode significar problemas no fornecimento de energia, como quedas do sistema. A Figura 5.1 apresenta uma série temporal de consumo de energia com ausência de dados.

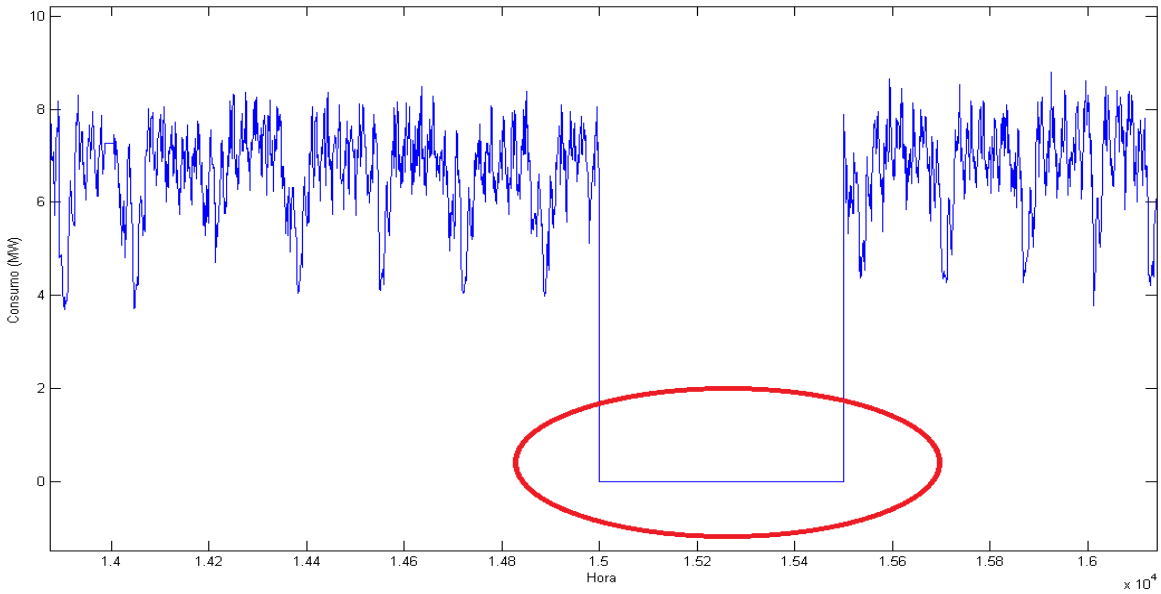

Figura 5.1 - Exemplo de ausência de dados em séries de carga.

5.2.2 MUDANÇA DE NÍVEL

A mudança de nível é identificada como uma mudança no valor médio da série e afeta um conjunto de observações contínuas. Em séries temporais de consumo de energia, podem ocorrer devido a alterações na topologia da rede de distribuição (manobra de chaves, entrada em operação de novos equipamentos, falha de algum dispositivo, etc.) que acaba por redistribuir os fluxos na rede. Em muitos estudos esses dados não necessitam ser corrigidos,

pois não são decorrentes de falhas de medição. A Figura 5.2 apresenta uma série temporal de consumo de energia com mudança de nível.

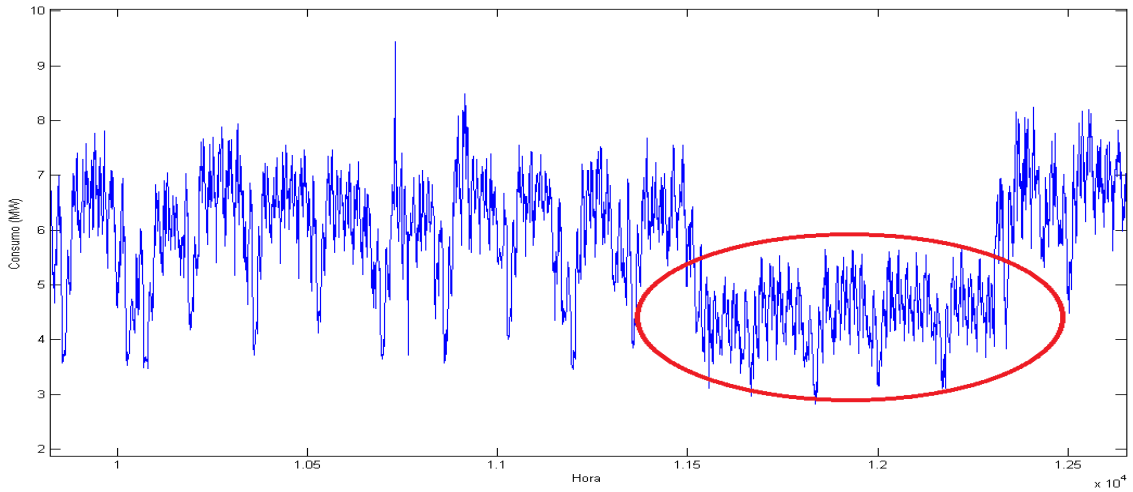

Figura 5.2 - Exemplo de mudança de nível em série temporal de consumo de energia.

5.2.3 PICOS

Os picos são medidas que diferem do valor das demais observações, criando elevações ou abaixamentos fora da sazonalidade própria dos dados. Uma das prováveis causas da ocorrência de *picos* é a instabilidade momentânea durante a transdução da medida. A Figura 5.3 apresenta uma série temporal de consumo de energia com picos.

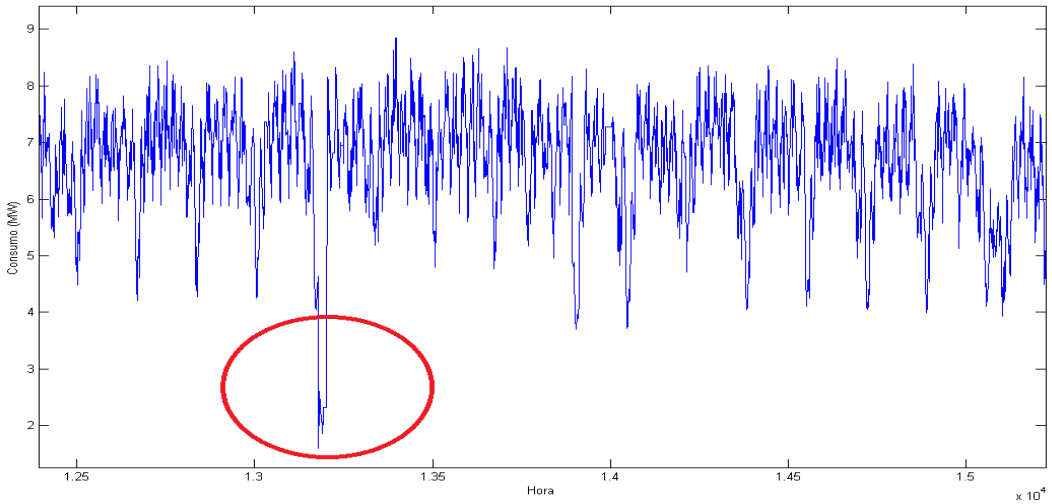

Figura 5.3 - Exemplo de *outliers* do tipo pico em série temporal de consumo de energia.

5.3 ESTUDO DE CASO EM UMA SÉRIE TEMPORAL DE CONSUMO DE ENERGIA

5.3.1 SÉRIE TEMPORAL DE CONSUMO DE ENERGIA

Como estudo de caso, para avaliar a aplicabilidade da metodologia proposta nesta dissertação para detecção e correção de *outliers*, foi utilizado um conjunto de dados de uma série temporal de consumo de energia elétrica no estado do Pará. O estudo abrange dados entre os anos de 2005 e 2006. O intervalo de medida do consumo foi de uma hora e a unidade de consumo Mega Watts. A série de consumo é apresentada na Figura 4.7.

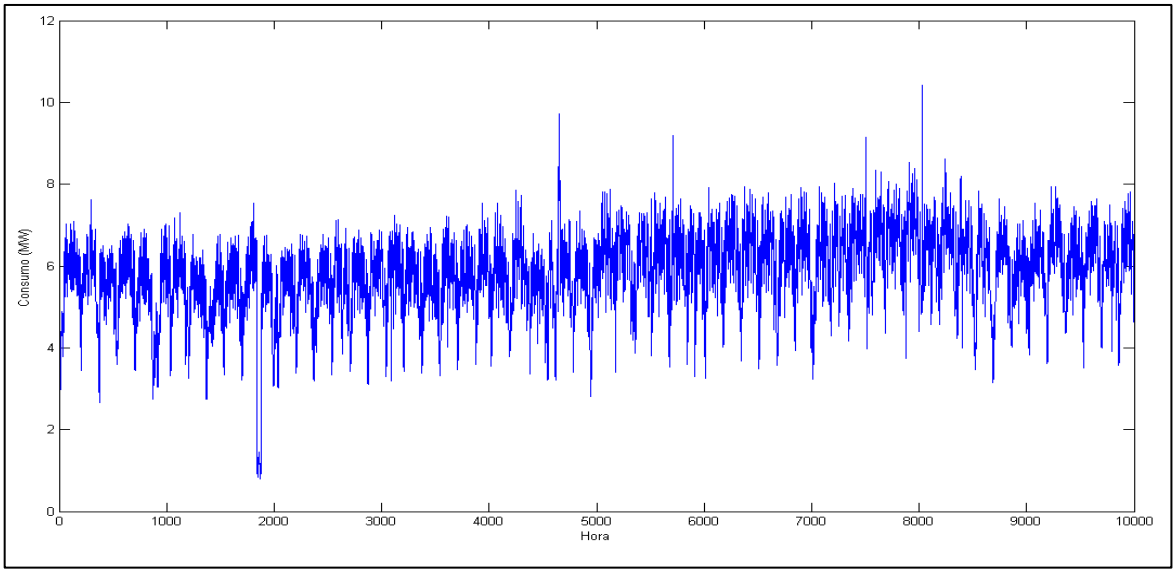

Figura 5.4 - Série temporal de consumo utilizada para estudo de caso

A Figura 5.5 apresenta a série de consumo para apenas 24 horas de um dia, onde pode-se melhor visualizar a presença de um *outlier*, pico de sinal circulado em vermelho.

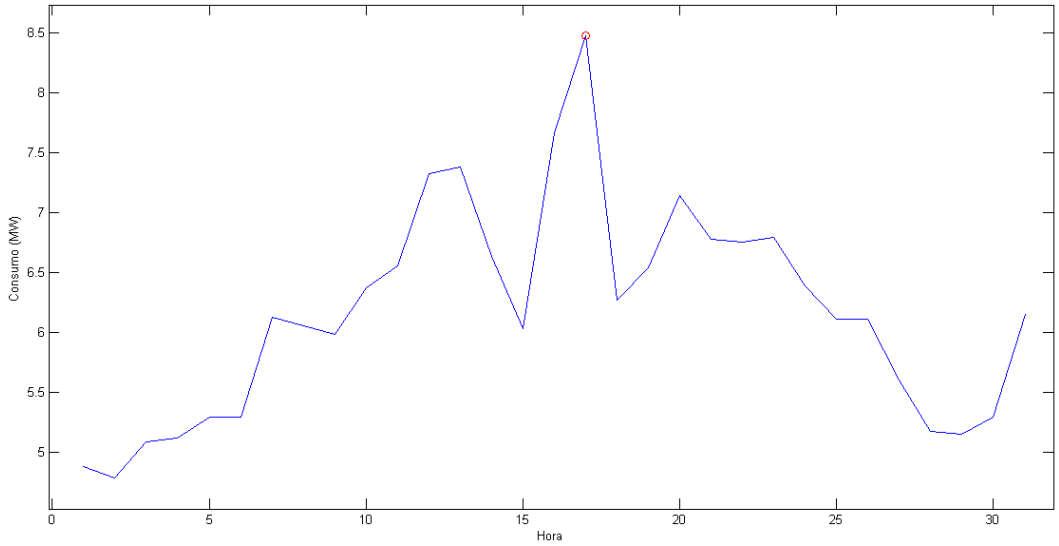

Figura 5.5 - *Outlier* da série de consumo de energia, circulado em vermelho.

5.3.2 PRÉ-PROCESSAMENTO DOS DADOS

Nesta fase, a série temporal é utilizada para formar o conjunto de dados X que será posteriormente utilizado para treinamento e validação da RNAA que será utilizada para reconstrução do sinal e correção de *outlier*.

Visualmente, a partir da Figura 5.4, percebe-se que a série apresenta alguns valores de picos que podem ser considerados *outliers*. Estes valores não serão considerados durante a fase de janelamento e não são copiados para o conjunto de dados X, pois podem influenciar na escolha de um limiar muito alto, gerando muitos resultados falsos negativos.

A série tem um total de 10 mil amostras, sendo que após aplicação do janelamento, construiu-se um banco de dados X com 9996 vetores de 5 amostras cada um. A Tabela 5.1 apresenta alguns dos vetores criados nesta fase de janelamento.

Tabela 5.1 - Exemplos de vetores criados com a técnica de janelamento.

Número da Amostra	Valor de x_1	Valor de x_2	Valor de x_3	Valor de x_4	Valor de x_5
1	4,3031	4,3914	4,3524	4,2831	4,2927
2	4,2953	4,3031	4,3914	4,3524	4,2831
3	4,0173	4,2953	4,3031	4,3914	4,3524
4	3,4657	4,0173	4,2953	4,3031	4,3914
5	3,4163	3,4657	4,0173	4,2953	4,3031
6	3,367	3,4163	3,4657	4,0173	4,2953
7	3,3575	3,367	3,4163	3,4657	4,0173
8	3,2787	3,3575	3,367	3,4163	3,4657
9	3,3375	3,2787	3,3575	3,367	3,4163
10	3,367	3,3375	3,2787	3,3575	3,367
11	3,4847	3,367	3,3375	3,2787	3,3575
12	3,5047	3,4847	3,367	3,3375	3,2787
13	3,4752	3,5047	3,4847	3,367	3,3375
14	3,4259	3,4752	3,5047	3,4847	3,367
15	3,4163	3,4259	3,4752	3,5047	3,4847
16	4,3619	3,4163	3,4259	3,4752	3,5047
17	4,5784	4,3619	3,4163	3,4259	3,4752
18	4,5888	4,5784	4,3619	3,4163	3,4259
19	4,7559	4,5888	4,5784	4,3619	3,4163
20	4,8642	4,7559	4,5888	4,5784	4,3619

5.3.3 TREINAMENTO E VALIDAÇÃO DA RNAA

Com o conjunto de dados *X* formado pode-se iniciar o treinamento e validação da RNAA. Para o treinamento da RNAA utilizou-se 5 (cinco) neurônios nas camadas de entrada e de saída. A RNA auto-associativa foi treinada utilizando a variante de *Levenberg-Marquardt* (HAYKIN, 2009) do algoritmo de retro propagação do erro (*Backpropagation*). Para treinamento da rede utilizou-se o Toolbox de redes neurais do MATLAB.

O conjunto de dados X possui 9996 vetores, sendo que dois terços destes vetores foram utilizados para treinamento e um terço para validação da RNAA. Diversos treinamentos foram realizados variando-se o número de neurônios na camada escondida. A Tabela 5.2 apresenta alguns resultados dos treinamentos realizados.

Tabela 5.2 - Erro quadrático médio para treino e para validação com a variação do numero de neurônios na camada escondida.

Quantidade de Neurônios	Erro de Treinamento	Erro de Validação	Tempo Médio de Treinamento
3	$2{,}321 * 10^{-4}$	$2{,}715 * 10^{-4}$	81 seg.
4	$9{,}552 * 10^{-5}$	$8{,}395 * 10^{-5}$	114 seg.
5	$1{,}801 * 10^{-6}$	$9{,}332 * 10^{-6}$	178 seg.
6	$8{,}569 * 10^{-8}$	$4{,}123 * 10^{-7}$	221 seg.
7	$9{,}01 * 10^{-9}$	$2{,}069 * 10^{-7}$	302 seg.
8	$4{,}403 * 10^{-9}$	$3{,}103 * 10^{-7}$	487 seg.
9	$2{,}29 * 10^{-8}$	$3{,}049 * 10^{-7}$	762 seg.

A partir dos resultados obtidos foi escolhida a RNAA com 7 neurônios na camada escondida. Esta RNAA foi escolhida por ter sido a rede com melhor resultado de generalização, ou seja, foi a rede neural que melhor respondeu para dados que não foram utilizados durante a fase de treinamento da mesma.

A Tabela 5.3 apresenta 10 resultados obtidos pela RNAA escolhida, considerando os dados de validação.

Tabela 5.3 - Exemplos de amostras de validação e os resultados da RNAA treinada.

Nº	E/S	x_1	x_2	x_3	x_4	x_5
1	E	3,41615	3,46541	4,01709	4,30278	4,30278
	S	3,41551	3,46487	4,01648	4,30307	4,30350
2	E	3,36690	3,41615	3,46541	4,01709	4,30278
	S	3,36578	3,41483	3,46466	4,01685	4,30273
3	E	3,35705	3,36690	3,41615	3,46541	4,01709
	S	3,35539	3,36540	3,41561	3,46401	4,01771
4	E	3,27824	3,35705	3,36690	3,41615	3,46541
	S	3,27643	3,35540	3,36534	3,41504	3,46431
5	E	3,33734	3,27824	3,35705	3,36690	3,41615
	S	3,33572	3,27648	3,35520	3,36565	3,41505
6	E	3,36690	3,33734	3,27824	3,35705	3,36690
	S	3,36527	3,33556	3,27645	3,35600	3,36548
7	E	3,48511	3,36690	3,33734	3,27824	3,35705
	S	3,48379	3,36545	3,33575	3,27694	3,35586
8	E	3,50482	3,48511	3,36690	3,33734	3,27824
	S	3,50358	3,48393	3,36540	3,33633	3,27668
9	E	3,47526	3,50482	3,48511	3,36690	3,33734
	S	3,47403	3,50390	3,48379	3,36572	3,33609
10	E	3,42601	3,47526	3,50482	3,48511	3,36690
	S	3,42472	3,47424	3,50327	3,48423	3,36555

(E) Valores de entrada para RNAA treinada

(S) Valores de saída da RNAA treinada

Com a RNAA já treinada e validada, deve ser iniciado o processo para escolha do valor limiar para o erro de reconstrução E (somatória dos erros absolutos entre as saídas e as entradas da rede) para ser utilizado futuramente para o processo detecção de *outlier*.

Para tanto deve-se apresentar para a RNAA treinada todos as amostras utilizadas na fase de treinamento e então calcular o erro de reconstrução para cada amostra sendo que o maior valor obtido será considerado como o valor limiar do sistema de detecção de *outliers*.

Então, com a rede treinada e validada, o vetor de entrada de treinamento foi apresentado para a rede para cálculo dos erros de reconstrução para cada amostra. A Figura 5.6 apresenta os erros de reconstrução obtidos para estes padrões. O erro máximo obtido foi de 0.005982. Assim sendo, este valor foi escolhido como valor limiar do erro de reconstrução para ser utilizado para detecção de *outlier*. Valores acima desse limiar indicarão a presença de *outliers* no vetor de entrada apresentado a RNAA.

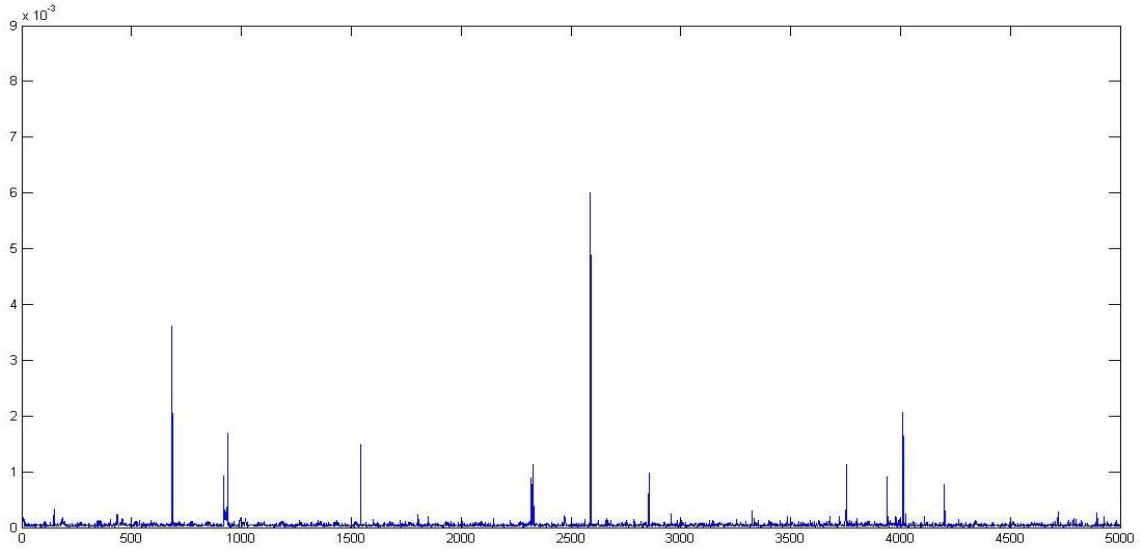

Figura 5.6 - Erros de reconstrução para as amostras de treinamento.

5.3.4 DETECÇÃO E CORREÇÃO DE *OUTLIERS*

Com a RNAA devidamente treinada e validada e com valor de limiar de erro de reconstrução definido, tem-se então o sistema para detecção e correção de *outliers* segundo a Figura 4.1. A RNAA poderá agora trabalhar em conjunto com o AG para corrigir os *outliers* detectados.

A codificação do AG foi feita utilizando a função *gaoptsim* do Matlab. Foram utilizadas até 100 gerações com população inicial contendo 50 indivíduos. Foi utilizado o operador de mutação com taxa de 1% e o operador de elitismo com taxa de 5%. Foi utilizado como critério de parada a ausência de melhora no fitness do melhor indivíduo por 3 gerações.

A série temporal da Figura 5.4 foi utilizada então para o processo de detecção e correção dos *outliers*. A Figura 5.7 apresenta o resultado da correção do *outlier* da Figura 5.5.

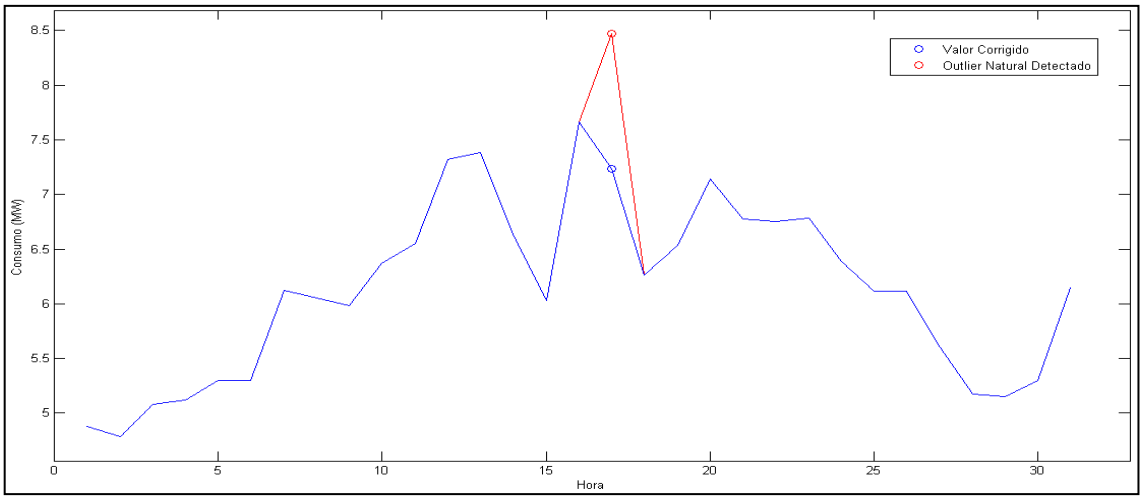

Figura 5.7 - Correção do *outlier* encontrado na série temporal. Valor original 8,4694. Valor corrigido 7,2351.

Para melhor análise da capacidade do sistema de detectar e corrigir *outliers*, alguns *outliers* virtuais foram introduzidos na série temporal. Dois casos de *outliers* virtuais foram trabalhados: para indicar ausência de dados e para indicar presença de picos.

A Figura 5.8 apresenta parte da série temporal em que foram inseridos *outliers* virtuais (picos) e os devidos valores corrigidos.

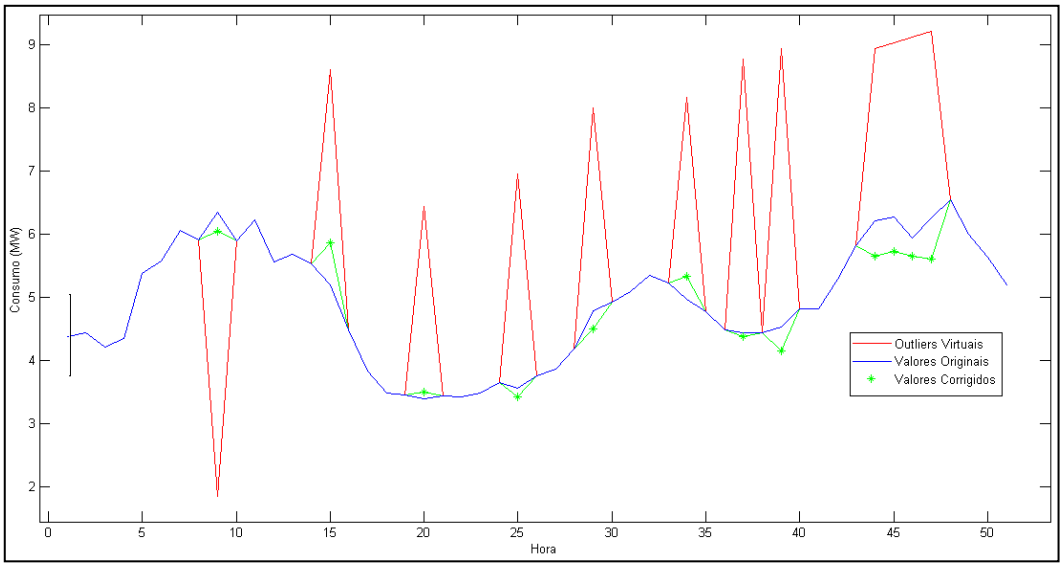

Figura 5.8 - *Outliers* virtuais corrigidos.

Para a Figura 5.8, a Tabela 5.4 apresenta os valores originais da série temporal, os valores virtuais que substituíram os valores originais e os valores corrigidos após sistema para

detectar e corrigir *outliers*. Também é apresentado o valor de erro absoluto entre valor corrigido e valor original para cada caso.

Tabela 5.4 - Apresenta os valores originais, *outliers* virtuais, valores corrigidos e erro.

Valor Original	*Outlier* Virtual	Valor Corrigido	Erro (|Corrigido - Original|)
6,22494	1,84	5,93006	0,29487
5,10168	8,62	5,75514	0,65346
3,32995	6,46	3,43206	0,10211
3,50365	6,98	3,36988	0,13377
4,69639	8,0	4,41478	0,28161
4,88166	8,18	5,23696	0,35530
4,34899	8,7	4,29222	0,05677
4,44163	8,95	4,07245	0,36918
6,08598	8,99	5,53751	0,54847
6,15546	9,0	5,60872	0,54674
5,81964	9,1	5,54403	0,27561
6,13230	9,2	5,50268	0,62961

A partir dos resultados obtidos tem-se então um sistema com a capacidade de correção dos *outliers* para valores próximos do esperado. Além disso, os últimos valores mostram a capacidade do sistema de utilizar valores corrigidos para detectar e corrigir novos *outliers.*

A Figura 5.9 apresenta agora parte da série temporal em que foram inseridos *outliers* virtuais com valores nulos, representando a ausência de dados, e os devidos valores corrigidos.

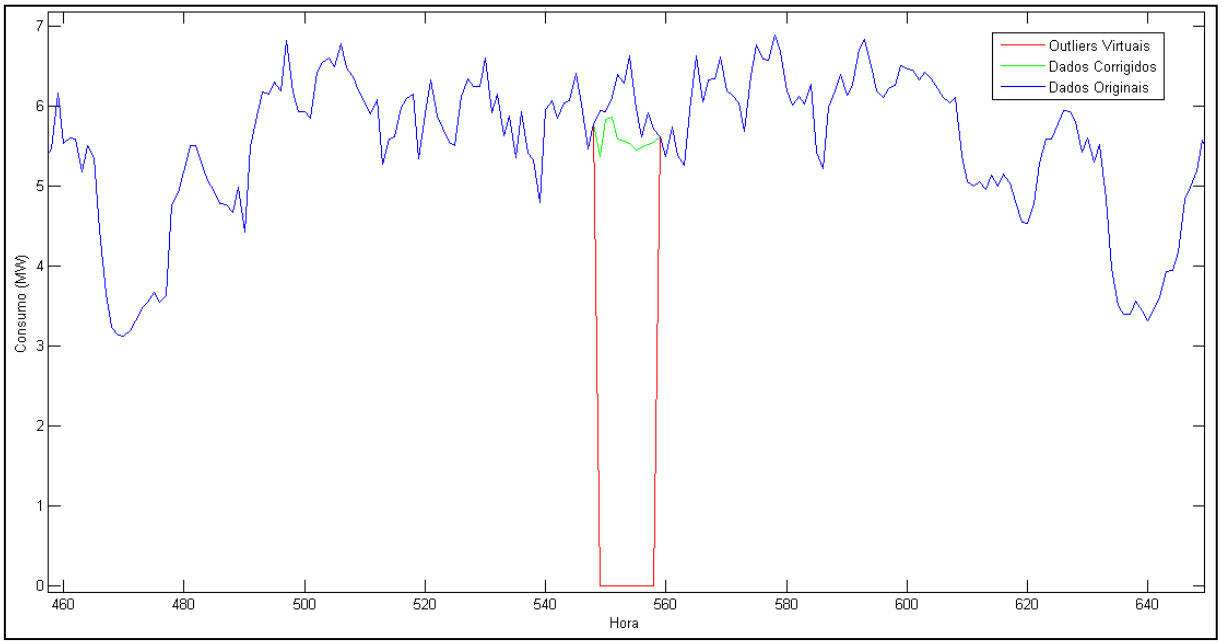

Figura 5.9 - Correção de *outliers* virtuais do tipo valores ausentes.

Para a Figura 5.9, a Tabela 5.5 apresenta os valores originais da série temporal, os valores virtuais que substituíram os valores originais e os valores corrigidos após sistema para detectar e corrigir *outliers*. Também é apresentado o valor de erro absoluto entre valor corrigido e valor original para cada caso.

Tabela 5.5 - Valores corrigidos para dados ausentes em sequência.

Valor Real	Valor Inserido	Valor Corrigido	Erro (|Real-Corrigido|)
5,9467	0	5,3611	0,5855
5,9230	0	5,8308	0,0922
6,1007	0	5,8557	0,2449
6,3968	0	5,5835	0,8134
6,2784	0	5,5543	0,7240
6,6219	0	5,5296	1,0923
5,9703	0	5,4554	0,5149

Em todos os casos apresentados anteriormente, o método corrigiu os *outliers* para valores que minimizaram o erro de reconstrução, sempre gerando valores menores que o limiar de reconstrução.

O tempo de execução total do sistema para detectar e corrigir *outliers* vai depender do tamanho da série a ser analisada e do número de *outliers* na série. O tempo de execução do AG para correção dos *outliers* depende da qualidade da população inicial gerada aleatoriamente, porém nos dados testados variou entre 15 e 50 segundos.

Aqui é importante salientar que o sistema proposto só poderá ser aplicado *off-line*, já com a série temporal a se analisada totalmente obtida a partir do processo gerador. Desta forma o tempo de execução do sistema de detecção e correção de *outlier* não é um fator que possa influenciar nos resultados finais obtidos.

Com uma série temporal de carga então livre de *outliers*, a mesma poderá ser utilizada com mais confiança para o desenvolvimento de um sistema de previsão de carga. Aqui no estudo realizado utilizaram-se apenas dois anos da série temporal de carga como estudo de caso para demonstrar aplicabilidade da metodologia. No entanto, é importante enfatizar que para que um sistema de previsão de carga possa ser desenvolvido, necessita-se de uma série temporal maior, com dados de mais anos, para que se tenha um sistema de previsão mais confiável.

6 CONCLUSÃO

O objetivo deste trabalho foi apresentar uma nova abordagem baseada em Inteligência Computacional para detecção e correção de *outliers* em séries temporais, demonstrando sua aplicabilidade na área de Sistemas de Energia, mais especificamente na área de previsão de carga, gerando um conjunto de dados com qualidade para a implementação de modelos.

A principal motivação para uso das redes neurais artificiais auto-associativas (RNAA) para este problema, é que a literatura apresenta diversos trabalhos em que estas redes alcançam bons resultados na detecção de *outliers*, porém, não foi encontrado seu uso para detecção de *outliers* em séries temporais. O uso dos AG's para otimização em redes neurais é bastante difundido na comunidade científica e tem se mostrado adequado para esta tarefa.

Buscou-se contextualizar o problema tratado, apresentando os principais conceitos relacionados à detecção de *outliers*, suas aplicações nos mais diversos campos de conhecimento e algumas dificuldades encontradas, como escassez de conjuntos de dados rotulados, com amostras normais e anormais.

As técnicas utilizadas também foram apresentadas, de maneira a possibilitar o entendimento do seu funcionamento. Esse entendimento também é facilitado pelas próprias técnicas, que são analogias de teorias conhecidas, como a seleção natural, a mutação e a herança de características genéticas que ocorre entre pais e filhos, além das teorias da neurociência, base da neurocomputação e das redes neurais artificiais.

A abordagem proposta foi aplicada a uma série temporal de consumo de energia elétrica, obtido de uma concessionária do Estado do Pará. Os resultados obtidos foram satisfatórios, já que todos os *outliers* virtuais foram detectados e corrigidos para valores próximos aos originais. Além disso, também foram detectados *outliers* previamente existentes na série temporal.

Os resultados obtidos permitem considerar o método como um importante benefício para o setor de energia e para o problema de previsão de carga, pois possibilita a construção de modelos de previsão de consumo mais eficazes, por se beneficiarem de séries temporais limpas. O método também pode ser aplicado a outros cenários em sistemas de energia, podendo trabalhar sobre outras variáveis do sistema, não apenas o consumo por hora.

No entanto, este trabalho não explorou todo o tema, sendo possível ainda realizar vários experimentos, de maneira a corroborar os resultados já alcançados, utilizando outras

séries temporais, com comportamentos diferentes da utilizada. Além disso, pode-se realizar um estudo comparativo entre técnicas de otimização, além dos AG's, utilizando enxame de partículas ou colônias de formigas, por exemplo. Tais técnicas também podem ser comparadas com um método de busca baseada em força bruta, cobrindo todo o espaço das amostras.

Outra comparação que pode ser ainda realizada é com métodos estatísticos de correção de *outliers* ou com métodos de formação de agrupamentos adaptados para séries temporais. Por fim, uma generalização da abordagem atual pode ser elaborada para séries temporais multivariadas.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AGGARWAL, CHARU C.; *Outlier* Analysis; New York, USA: Springer, 2013.

AGUIRRE, LUIS A.; Introdução à identificação de sistemas - Técnicas lineares e não-lineares aplicadas a sistemas reais. Belo Horizonte, Brasil: Editora UFMG, 2004.

AMPAZIS, N. e PERANTONIS, S. J.; Levenberg-Marquardt algorithm with adaptive momentum for the efficient training of feedforward networks; Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000.

BARNETT, V. e LEWIS, T.; *Outliers* in statistical data; New York; Willey, 1978.

BISHOP, CHRISTOPHER M.; Neural networks for pattern recognition; Oxford university press; 1995.

BRAGA, A. P., CARVALHO, A. C. P. L. F. e LUDEMIR, T. B.; Redes Neurais Artificiais - Teoria e Aplicações, 2ª ed.; Rio de Janeiro: LTC, 2007.

CHANDOLA, VARUN; BARNEJEE, ARINDAM e KUMAR, VIPIN; *Outlier* Detection: A Survey; ACM Computing Surveys (CSUR), 2009.

FACELI, K., LORENA, A. C., GAMA, J. e CARVALHO, A. C. P. L. F.; Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina; Rio de Janeiro: LTC, 2011.

GOLDBERG, DAVID E.; Genetic Algorithms in search, optimization and machine learning; Addison-Wesley, 1989.

HAN, J., KAMBER, M. e PEI, J.; Data Mining – Concepts and Techniques; Morgan Kaufmann Publishers - Elsevier, 2012.

HARRIS, FREDRIC J.; On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform; Proceedings of the IEEE 66. 1978.

HARVARD BUSINESS SCHOOL; The Evolution of Decision Making: How Leading Organizations Are Adopting a Data-Driven Culture. 2012.

HAWKINS, DOUGLAS M.; Identification of *Outliers*; Chapman and Hall. 1980.

HAYKIN, SIMON; Neural Networks and Learning Machines, 3ª ed; Pearson Education, Inc., 2009.

HODGE, VICTORIA J. e AUSTIN, JIM; A Survey of Outleir Detection Methodologies; Artificial Intelligence Review; Springer; 2004.

LINDEN, RICARDO; Algoritmos Genéticos, 3ª ed.; Ciência Moderna, 2012.

MARKOU, MARKOS e SINGH, SAMEER; Novelty detection: a review - part 2: neural network based approaches; Signal Processing 83; Elsevier, 2003.

MINISTÉRIO DAS MINAS E ENERGIA. Plano decenal de energia 2008-2017. 2009.

MITCHELL, MELANIE; An introduction to genetic algorithms; Cambridge, Massachusetts London, Inglaterra; 1999.

MONTANA, DAVID J. e DAVIS, LAWRENCE; Training feedforward neural networks using genetic algorithms; Proceedings of the 11th International Joint Conference on Artificial Intelligence; 1989.

MORETTIN, P. A.; TOLÓI, C. M. C.; Análise de séries temporais; 2. ed.; São Paulo: Edgard Blücher, 2006.

PIMENTEL, MARCO A. F.; CLIFTON, DAVID A.; CLIFTON, LEI e TARASSENKO, LIONEL; A review of novelty detection; Signal Processing 99; Elsevier; 2014.

SINGH, K. e UPADHYAYA, S.; *Outlier* Detection: Applications and Techniques; International Journal of Computer Science Issues; Vol. 9, Issue 1; Janeiro de 2012.

SIVANANDAM, S. N. e DEEPA, S. N.; Introduction to genetic algorithms; Springer Science and Business Media; 2007.