



**Camilo Velasco Rueda**

**EsnPredictor: Ferramenta de Previsão de Séries  
Temporais Baseada em Echo State Networks Otimizadas  
por Algoritmos Genéticos e Particle Swarm Optimization.**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial  
para obtenção do título de Mestre pelo Programa de  
Pós-Graduação em Engenharia Elétrica da PUC-  
Rio.

Orientadora: Marley Maria Bernardes Rebuszi Vellasco

Rio de Janeiro

Abril de 2014



**Camilo Velasco Rueda**

**EsnPredictor: Ferramenta de Previsão de Séries Temporais baseada em Echo State Networks Otimizada por Algoritmos Genéticos e Particle Swarm Optimization**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Profa. Marley Maria Bernades Rebuzzi Vellasco**  
**Orientador**  
Departamento de Engenharia Elétrica – PUC-Rio

**André Vargas Abs da Cruz**  
UEZO

**Profa. Karla Tereza Figueiredo Leite**  
Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Jorge Luís Machado do Amaral**  
UERJ

**Prof. Luiz Biondi Neto**  
UERJ

**Prof. José Eugenio Leal**  
Coordenador Setorial do Centro  
Técnico Científico

Rio de Janeiro, 15 de abril de 2014

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, da autora e do orientador.

### **Camilo Velasco Rueda**

Graduou-se em Engenharia Mecatronica pela Universidad Santo Tomas de Aquino - Bucaramanga - Colombia Ustabuca em 2012.

#### Ficha Catalográfica

Velasco Rueda, Camilo

EsnPredictor: ferramenta de previsão de séries temporais baseada em echo state networks otimizada por algoritmos genéticos e particle swarm optimization / Camilo Velasco Rueda ; orientador: Marley Maria Bernades Rebuzzi Vellasco – 2014.

94 f. : il. (color.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2014.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Séries temporais. 3. Previsão. 4. Echo State Networks. 5. Particle Swarm Optimization. 6. Algoritmos genéticos. 7. Interface gráfica. I. Vellasco, Marley Maria Bernades Rebuzzi. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

## Agradecimentos

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

À minha orientadora Professora Marley, pelo apoio e parceria para a realização deste trabalho.

Aos meus pais, Luz Marina Rueda e Jaime Velasco, pela educação, atenção e carinho.

A todos os amigos da PUC-Rio.

A todos os amigos e familiares que de uma forma ou de outra me estimularam e me ajudaram.

Aos professores que participaram da Comissão examinadora.

## Resumo

Velasco, Camilo. Vellasco, Marley Maria Bernardes Rebuzzi. **EsnPredictor: Ferramenta de Previsão de Séries Temporais Baseada em Echo State Networks Otimizadas por Particle Swarm Optimization e Algoritmos Genéticos.** Rio de Janeiro, 2014. 94p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A previsão de séries temporais é fundamental na tomada de decisões de curto, médio e longo prazo, em diversas áreas como o setor elétrico, a bolsa de valores, a meteorologia, entre outros. Tem-se na atualidade uma diversidade de técnicas e modelos para realizar essas previsões, mas as ferramentas estatísticas são as mais utilizadas principalmente por apresentarem um maior grau de interpretabilidade. No entanto, as técnicas de inteligência computacional têm sido cada vez mais aplicadas em previsão de séries temporais, destacando-se as Redes Neurais Artificiais (RNA) e os Sistemas de Inferência *Fuzzy* (SIF). Recentemente foi criado um novo tipo de RNA, denominada *Echo State Networks* (ESN), as quais diferem das RNA clássicas por apresentarem uma camada escondida com conexões aleatórias, denominada de “Reservoir” (Reservatório). Este “Reservoir” é ativado pelas entradas da rede e pelos seus estados anteriores, gerando o efeito de “Echo State” (Eco), fornecendo assim um dinamismo e um desempenho melhor para tarefas de natureza temporal. Uma dificuldade dessas redes ESN é a presença de diversos parâmetros, tais como Raio Espectral, Tamanho do Reservoir e a Percentual de Conexão, que precisam ser calibrados para que a ESN forneça bons resultados. Portanto, este trabalho tem como principal objetivo o desenvolvimento de uma ferramenta computacional capaz de realizar previsões de séries temporais, baseada nas ESN, com ajuste automático de seus parâmetros por *Particle Swarm Optimization* (PSO) e Algoritmos Genéticos (GA), facilitando a sua utilização pelo usuário. A ferramenta computacional desenvolvida oferece uma interface gráfica intuitiva e amigável, tanto em termos da modelagem da ESN, quanto em termos de realização de eventuais pré-processamentos na série a ser prevista.

## Palavras-chave

Séries Temporais; Previsão; Echo State Networks; Particle Swarm Optimization; Algoritmos Genéticos; Interface Gráfica.

## Abstract

Velasco, Camilo. Vellasco, Marley Maria Bernardes Rebuzzi. **EsnPredictor: Time Series Forecasting Application based on Echo State Networks Optimized by Genetics Algorithms and Particle Swarm Optimization.** Rio de Janeiro, 2014. 94p. MSc. Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The time series forecasting is critical to decision making in the short, medium and long term in several areas such as electrical, stock market, weather and industry. Today exist different techniques to model this forecast, but statistics are more used, because they have a bigger interpretability, due by the mathematic models created. However, intelligent techniques are being more applied in time series forecasting, where the principal models are the Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS). A new type of ANN called Echo State Networks (ESN) was created recently, which differs from the classic ANN in a randomly connected hidden layer called ‘Reservoir’. This “Reservoir” is activated by the network inputs, and the historic of the reservoir activations generating so, the ‘Echo State’ and giving to the network more dynamism and a better performance in temporal nature tasks. One problem with these networks is the presence of some parameters as, Spectral Radius, Reservoir Size and Connection Percent, which require calibration to make the network provide positive results. Therefore the aim of this work is to develop a computational application capable to do time series forecasting, based on ESN, with automatic parameters adjustment by Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), facilitating its use by the user. The developed computational tool offers an intuitive and friendly interface, both in terms of modeling the ESN, and in terms of achievement of possible pre-process on the series to be forecasted.

## Keywords

Time Series; Forecasting; Echo State Networks; Particle Swarm Optimization; Genetics Algorithms; GUI.

## Sumário

1. Introdução	14
1.1. Motivação	14
1.2. Objetivos	15
1.3. Descrição do Trabalho	15
1.4. Organização da Dissertação	16
2. Fundamentação Teórica	18
2.1. Introdução	18
2.2. Echo State Networks (ESN)	18
2.2.1. Configuração	19
2.2.2. Reservoir	19
2.2.3. Estados do Reservoir	21
2.2.4. Treinamento	22
2.2.4.1. Treinamento On-Line	22
2.2.4.2. Treinamento Off-Line	23
2.3. Particle Swarm Optimization (PSO)	26
2.3.1. Representação	26
2.3.2. Algoritmo	27
2.3.3. Parâmetros	28
2.4. Algoritmos Genéticos (GA)	29
2.4.1. Representação	30
2.4.2. Decodificação	30
2.4.3. Avaliação	30
2.4.4. Seleção e Reprodução	31
2.4.5. Operadores Genéticos	31
2.4.6. Parâmetros	32
3. Técnicas de Previsão de Séries Temporais	33
3.1. Introdução	33
3.2. Técnicas Estatísticas	34

3.3. Técnicas de Inteligência Computacional	35
3.4. Sistemas de Inteligência Computacional Híbridos	36
3.5. Avaliação do Desempenho de uma Previsão	37
3.6. Tipos de Previsão	38
3.7. Parâmetros	39
4. Modelos Híbridos para Previsão de Séries Temporais	40
4.1. Introdução	40
4.2. Modelo ESN-GA	40
4.2.1. Cromossomo	40
4.2.2. Função de Avaliação	42
4.2.3. Seleção e Reprodução	45
4.2.4. Operadores Genéticos	45
4.3. Modelo ESN-PSO	46
4.3.1. Partícula	46
4.3.2. Função de Avaliação	46
5. EsnPredictor: Ferramenta de Previsão de Séries Temporais	49
5.1. Introdução	49
5.2. Estrutura e Metodologia	52
5.3. Módulo de Importação e Visualização	54
5.4. Módulo de Pré-Processamento	56
5.4.1. Normalização	56
5.4.2. Eliminação de Tendência	59
5.4.3. Interpolação	60
5.5. Módulo de Definição do Modelo	61
5.5.1. Tipo de Previsão	61
5.5.2. Tipo de Modelo	62
5.5.3. Parâmetros a serem Otimizados	62
5.6. Módulo de Execução do Modelo	63
5.7. Módulo de Resultados e Exportação	64
5.8. Funcionalidades Básicas	65
5.9. Extras	66



6. Estudo de Casos	68
6.1. Introdução	68
6.2. Dados Utilizados	68
6.3. Metodologia	68
6.4. Pré-Processamento dos Dados	70
6.5. Resultados Obtidos	75
6.6. Resumo dos Resultados	80
7. Conclusões e Trabalhos Futuros	83
7.1. Conclusões	83
7.2. Trabalhos Futuros	84
Referências Bibliográficas	100
Apêndice A– Arquitetura do EsnPredictor	104

## Lista de Figuras

Figura 1. Topologia Geral de uma ESN. ....	19
Figura 2. Funções de ativação de neurônios típicas .....	20
Figura 3. Movimento de uma Partícula. ....	26
Figura 4. Fluxograma do PSO .....	27
Figura 5 Fluxograma Algoritmo Genético .....	29
Figura 6 Passos Previsão Multistep .....	38
Figura 7 Passos Previsão Singlestep.....	39
Figura 8. Cromossomo do GA.....	41
Figura 9. Janela Otimizada .....	42
Figura 10. Fluxograma Função de Avaliação GA .....	43
Figura 11. Divisões do Cromossomo .....	43
Figura 12. Conversão Binário para Decimal .....	43
Figura 13. Mutação Binária .....	45
Figura 14. Cruzamento de Dois Pontos .....	46
Figura 15. Partícula de PSO .....	46
Figura 16. Fluxograma Função de Avaliação PSO .....	47
Figura 17. Conversão Decimal para Binário .....	47
Figura 18. Módulos de EsnPredictor. ....	52
Figura 19. Tela de Abertura.....	53
Figura 20. Tela Principal. ....	53
Figura 21. Abas da Ferramenta EsnPredictor. ....	54
Figura 22. EsnPredictor Toolbar .....	54
Figura 23. Aba de Importação dos Dados. ....	54
Figura 24. Janela Importação de Arquivo. ....	55
Figura 25. Formatos dos arquivos para importar.....	56
Figura 26. Aba de Pré-Processamento dos Dados.....	56
Figura 27. Normalização Standard. ....	57
Figura 28. Série com Outliers.....	57

Figura 29. Definição da Normalização Linear por Partes. ....	58
Figura 30. Parâmetros da Normalização por Partes. ....	58
Figura 31. Resultado da Normalização por Partes. ....	58
Figura 32. Eliminação da Tendência diretamente. ....	59
Figura 33. Eliminação da Tendência por Diferenciação. ....	60
Figura 34. Interpolação da Serie.....	60
Figura 35. Aba de Definição do Modelo. ....	61
Figura 36. Parâmetros Tipo de Previsão.....	62
Figura 37. Parâmetros Tipo de Modelo. ....	62
Figura 38. Parâmetros da Otimização.....	63
Figura 39. Aba Execução do Modelo. ....	63
Figura 40. Aba Resultados e Exportação. ....	64
Figura 41. Arquivo Padrão de Exportação. ....	65
Figura 42. Pré-processamento da Série NN3_101.....	71
Figura 43. Pré-processamento da Série NN3_102.....	72
Figura 44. Pré-processamento da Série NN3_103.....	72
Figura 45. Pré-processamento da Série NN3_104.....	72
Figura 46. Pré-processamento da Série NN3_105.....	73
Figura 47. Pré-processamento da Série NN3_106.....	73
Figura 48. Pré-processamento da Série NN3_107.....	73
Figura 49. Pré-processamento da Série NN3_108.....	74
Figura 50. Pré-processamento da Série NN3_109.....	74
Figura 51. Pré-processamento da Série NN3_110.....	74
Figura 52. Pré-processamento da Série NN3_111.....	75
Figura 53. Resultados da Série NN3_101. ....	77
Figura 54. Resultados da Série NN3_102. ....	77
Figura 55. Resultados da Série NN3_103. ....	77
Figura 56. Resultados da Série NN3_104. ....	78
Figura 57. Resultados da Série NN3_105. ....	78
Figura 58. Resultados da Série NN3_106. ....	78
Figura 59. Resultados da Série NN3_107. ....	78
Figura 60. Resultados da Série NN3_108. ....	79
Figura 61. Resultados da Série NN3_109. ....	79
Figura 62. Resultados da Série NN3_110. ....	79

Figura 63. Resultados da Série NN3_111. ....	79
Figura 64. Resumo dos Resultados. ....	81
Figura 83. MVC.....	92
Figura 84. Diagrama ESN .....	92
Figura 85. Diagrama de Pré-processamento.....	93
Figura 86. Diagrama de Parâmetros do Modelo.....	93
Figura 87. Diagrama Tela Principal .....	94

## Lista de Tabelas

Tabela 1. Principais tipos de representação do Cromossomo (Goldberg, 1991) (Janikow, 1991). .....	30
Tabela 2. Cromossomo 1.....	41
Tabela 3. Cromossomo 2.....	42
Tabela 4 Botões Extras do ESNPredictor. ....	66
Tabela 5 Erros de Utilização do ESNPredictor.....	67
Tabela 6 Configurações para ESN.....	69
Tabela 7 Configurações para ESN-GA. ....	70
Tabela 8 Configurações para ESN-PSO.....	70
Tabela 9 Pré-processamento das Séries NN3.....	71
Tabela 10. Resultados Série NN3_101.....	75
Tabela 11. Resultados Série NN3_102.....	75
Tabela 12. Resultados Série NN3_103.....	75
Tabela 13. Resultados Série NN3_104.....	76
Tabela 14. Resultados Série NN3_105.....	76
Tabela 15. Resultados Série NN3_106.....	76
Tabela 16. Resultados Série NN3_107.....	76
Tabela 17. Resultados Série NN3_108.....	76
Tabela 18. Resultados Série NN3_109.....	76
Tabela 19. Resultados Série NN3_110.....	76
Tabela 20. Resultados Série NN3_111.....	77
Tabela 21. Resumo dos Resultados. ....	80
Tabela 22. Média dos Tempos de Convergência (Minutos).....	81
Tabela 23. MAPE Resultados com Interpolação (INT). ....	82

# 1. Introdução

## 1.1. Motivação

A previsão de séries temporais tem uma importância vital em diversas áreas, por exemplo, a necessidade de se ter uma previsão de carga confiável no setor elétrico, para uma operação eficiente e segura de todo o sistema elétrico. Na bolsa de valores, prever o valor futuro de ações é de extremo interesse não só dos investidores, que podem ganhar ou perder milhões, como também para pesquisadores que buscam entender o comportamento do mercado de ações. Até mesmo uma boa previsão do clima pode salvar milhares de vidas e, no setor industrial, previsões de demanda fundamentam o planejamento de curto e longo prazo de marketing, logística, produção e finanças (Barboza, 2011).

Existem diferentes técnicas para abordar a previsão de séries temporais: métodos estatísticos, numéricos e os de inteligência computacional. Certamente as técnicas estatísticas são as mais utilizadas, principalmente por apresentarem um maior grau de interpretabilidade, garantido pelos modelos matemáticos gerados. No entanto, as técnicas de inteligência computacional estão sendo cada vez mais aplicadas em previsão de séries temporais, com destaque para as Redes Neurais Artificiais (RNA) do tipo *Multi-Layer Perceptron* (MLP), por serem aproximadores universais, podendo aprender qualquer mapeamento entre entrada e saída (Haykin, 2000). Recentemente foi criado um novo tipo de RNA, denominado *Echo State Networks* (ESN), as quais diferem das RNA clássicas por terem uma camada escondida com conexões aleatórias, chamada “*Reservoir*” (Jaeger, 2010). Este “*Reservoir*” é ativado pelas entradas da rede e pelos estados anteriores do próprio *reservoir*, gerando o efeito de “*Echo State*” (Breve, 2005) e fornecendo assim um dinamismo e um desempenho melhor para tarefas de natureza temporal. Entretanto, essas redes possuem diversos parâmetros, tais como Raio Espectral, Tamanho do *Reservoir* e a porcentagem de Conexão, os quais precisam ser calibrados para que a rede funcione de forma adequada. Para

efetuar o ajuste adequado desses valores, precisa-se de experimentação e conhecimento das características dessas redes. No caso da sua utilização em previsão de séries temporais, parâmetros adicionais devem ser especificados, como o tamanho a janela de entrada e o tipo de previsão (*single-step* ou *multi-step*), dificultando ainda mais a modelagem correta da ESN.

## 1.2. Objetivos

Em função do exposto na seção anterior, este trabalho teve como objetivo desenvolver uma ferramenta computacional amigável, baseada na teoria das ESN, capaz de realizar previsões de séries temporais em diferentes horizontes, com boa exatidão, de forma que os parâmetros das ESN sejam especificados de forma automática para o usuário.

Os objetivos secundários são:

- Oferecer uma interface gráfica amigável e intuitiva, de forma a viabilizar seu uso mesmo por usuários que não tenham conhecimento das teorias aplicadas, necessitando apenas de um conhecimento básico sobre as mesmas.
- Permitir o ajuste automático de uma ESN, por meio de *Particle Swarm Optimization* (PSO) e Algoritmos Genéticos (GA), de forma a determinar automaticamente uma configuração dos parâmetros do sistema que resulte em uma maior exatidão da previsão.
- Fornecer flexibilidade e visualização do pré-processamento das séries temporais.

## 1.3. Descrição do Trabalho

Este trabalho foi elaborado seguindo as seguintes etapas:

- Estudo sobre a área de previsão de séries temporais, e das diferentes técnicas de modelagem.

- Estudo sobre modelos de Inteligência Computacional, mais especificamente: *Echo State Networks* (ESN), *Particle Swarm Optimization* (PSO), Algoritmos Genéticos (GA).
- Definição e desenvolvimento dos modelos híbridos ESN-PSO e ESN-GA.
- Construção da ferramenta computacional denominada EsnPredictor.
- Estudo de casos.

Na primeira etapa do trabalho, foi feito um levantamento bibliográfico referente à área de previsão de séries temporais, analisando as vantagens e desvantagens das diferentes técnicas de modelagem.

Além disso, foi realizado um estudo sobre Redes Neurais Artificiais, principalmente as ESN que é a técnica de Inteligência Computacional utilizada na construção dos modelos híbridos e da ferramenta computacional. Também foi feito um levantamento de técnicas de otimização (PSO e GA), de forma a utilizá-las na calibração das ESN estudadas anteriormente.

A partir dos estudos desses trabalhos foram definidas características importantes a serem levadas em conta no desenvolvimento do sistema de previsão híbrido e no desenvolvimento da ferramenta EsnPredictor.

Na etapa de construção da ferramenta computacional o foco principal foi dar ao usuário uma maior flexibilidade e ampla gama de alternativas de pré-processamento e de configuração do modelo de previsão da série temporal, possibilitando a visualização de um quadro comparativo com diferentes configurações, além de permitir ao modelo se adaptar a diferentes tipos de séries.

Na etapa de estudo de casos foram realizados diversos testes, focando nas 11 séries da competição NN3 (Forecasting Competition 2006/7), de forma a permitir a comparação de desempenho com diferentes trabalhos anteriores realizados no mesmo conjunto de dados.

#### **1.4. Organização da Dissertação**

Esta dissertação está organizada em mais cinco capítulos, descritos a seguir.

O Capítulo 2 descreve as técnicas de inteligência computacional, destacando as utilizadas na ferramenta desenvolvida neste trabalho: ESN, PSO e GA. Todos



os conceitos básicos necessários para o entendimento das técnicas e aqueles aplicados na estrutura matemática da ferramenta desenvolvida são descritos neste capítulo.

No Capítulo 3 são apresentados características e conceitos importantes na previsão de séries temporais, além das principais técnicas quantitativas de previsão de séries temporais, focando na aplicação de modelos inteligentes.

O Capítulo 4 apresenta em detalhes a ferramenta EsnPredictor, listando suas funcionalidades, sua estrutura matemática e descrevendo a sua utilização.

O Capítulo 5 apresenta o estudo de caso, no qual se utiliza o EsnPredictor para realizar previsões em 11 séries temporais obtidas do Forecasting Competition de 2006/07 (NN3). O potencial do módulo de otimização por GA e PSO é analisado neste capítulo.

Finalmente, no Capítulo 6 estão descritas as conclusões e os trabalhos futuros.

## 2. Fundamentação Teórica

### 2.1. Introdução

Neste capítulo são apresentados os conceitos fundamentais sobre Redes Neurais do tipo *Echo State Networks* (ESN) e os fundamentos das técnicas de otimização *Particle Swarm Optimization* (PSO) e Algoritmos Genéticos (GA), que são as técnicas de Inteligência Computacional empregadas na construção do modelo e da ferramenta EsnPredictor.

### 2.2. Echo State Networks (ESN)

As ESN (*Echo State Networks*) foram propostas pelo Dr. Herbert Jaeger em 2001, artigo que foi corrigido no ano 2010 por ele mesmo em (Jaeger, 2010), sendo este um dos métodos principais do “*reservoir computing*” junto com as “*Liquid States Machines*” (Natschläger, 2002). Estas surgiram baseadas em estudos prévios sobre neurociência computacional (Dominey, 1995), e subsequentes ramificações do *machine learning* como a regra de aprendizagem *Backpropagation Decorrelation* (Steil, 2004).

O modelo das ESN foi motivado pela ineficiência dos algoritmos atuais para o treinamento das Redes Neurais Recorrentes (RNR), as quais são caracterizadas pela presença de ao menos um laço de realimentação (“recorrência”) em seus neurônios. Em geral, as RNRs apresentam um comportamento dinâmico não linear, já que elas podem manter a ativação ou saída mesmo na ausência da entrada, sendo esta característica chamada de “Memória Dinâmica”.

Os algoritmos usados para treinar as RNRs, propostos em (Willian & Zipser, 1989; Werbos, 1990), adaptam de maneira incremental os pesos sinápticos. No entanto, estes tem uso limitado devido a três problemas principais:

- Convergência lenta;
- As soluções encontradas, em muitos casos são subótimas;

- A maioria dos modelos recorrentes requer convergência a pontos estáveis (Hertz, 1991) fato que limita a capacidade de aprendizado da rede.

As ESN fornecem uma solução dos problemas mencionados acima no treinamento, pois nesta arquitetura somente são atualizados os pesos da camada de saída, sendo a tarefa de aprendizado da rede mais simples.

### 2.2.1. Configuração

Como ilustra a Figura 1, uma ESN é composta por uma camada de entrada, um reservatório de neurônios interconectados de maneira recorrente e gerados aleatoriamente, e uma camada de saída.

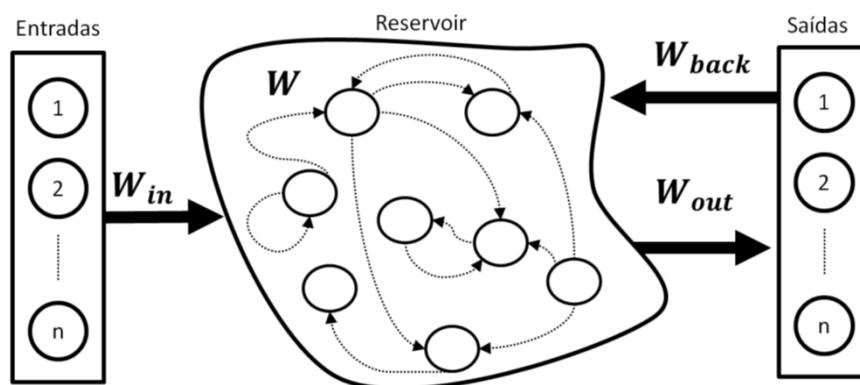


Figura 1. Topologia Geral de uma ESN.

Na Figura 1,  $W$ ,  $W_{in}$ ,  $W_{out}$  e  $W_{back}$  são os pesos do reservatório, das entradas, das saídas e da retroalimentação, respetivamente. A matriz  $W_{out}$  é a matriz de pesos que deve ser atualizada durante o processo de treinamento da rede, enquanto que as outras ( $W$ ,  $W_{in}$  e  $W_{back}$ ) são inicializadas de forma aleatória no momento de criação da rede e não sofrem atualizações (Jaeger, 2010).

### 2.2.2. Reservoir

O componente característico das ESN é o *Reservoir*. Este é definido por uma camada de neurônios com conexões e pesos criados de maneira aleatória.

Essas conexões geradas aleatoriamente podem ser entre um neurônio e outro, ou de um neurônio para ele mesmo. Geralmente uma ESN tem um número maior de neurônios (da ordem de 10 – 1500 ou mais).

Estas interconexões do Reservoir são as que geram a característica do “*echo state*”, dado que para qualquer instante de tempo, o estado de ativação do Reservoir vai depender das entradas da rede, da retroalimentação da rede e do estado no instante anterior. Logo, a rede é suscetível aos estados anteriores do Reservoir, em maior ou menor proporção, de acordo com a matriz de pesos  $W$  do mesmo.

O *reservoir* é gerado de forma aleatória no momento da criação da rede, e suas características dependem de três parâmetros configurados inicialmente:

- Raio Espectral: Valor usado para escalar ou normalizar os pesos, normalmente menor que um para que a rede não apresente um comportamento caótico, e não sejam saturadas as ativações dos neurônios;
- Percentual (%) de Interconexão: Indica a porcentagem de neurônios que vão ser conectados aleatoriamente dentro do Reservoir, normalmente de 60%, para garantir uma dinâmica variada;
- Tamanho do Reservoir: define o Número de Unidades que formará o Reservoir.

Igual aos modelos de redes neurais clássicos, os neurônios do *Reservoir* e da camada de saída das *Echo State Networks* utilizam funções de ativação dos tipos ilustrados na Figura 2.

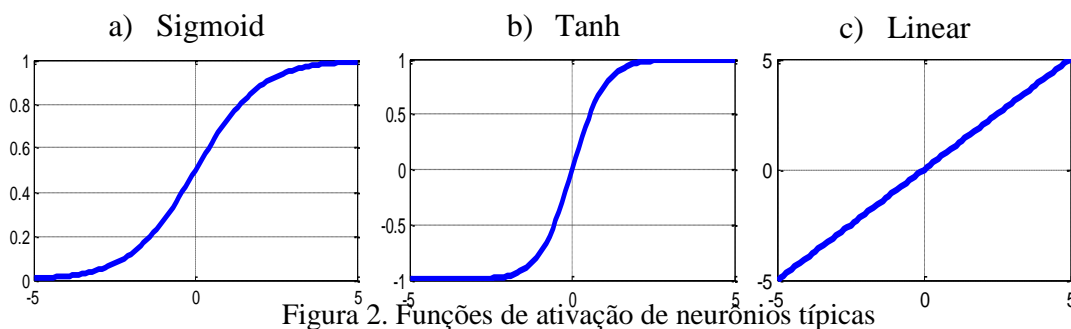


Figura 2. Funções de ativação de neurônios típicas

No *Reservoir*, normalmente, as funções de ativação são do tipo sigmoide ou tangente hiperbólica (imagens a e b da Figura 2), enquanto que a função de ativação do neurônio da saída geralmente é linear (imagem c da Figura 2). Quando se utiliza função Tanh na camada de saída, um pré-processamento dos dados se faz necessário, de modo que os dados apresentados no treinamento estejam escalados entre -1 e 1, e assim os resultados sejam corretos (Jaeger, 2010).

### 2.2.3. Estados do *Reservoir*

O estado do *Reservoir* é um vetor de tamanho  $N$ , onde  $N$  é o número de neurônios do *Reservoir*. Esse vetor contém as ativações de cada neurônio em qualquer instante de tempo, e é calculado a partir da equação (2.1).

$$x(n+1) = f(W_{in}u(n+1) + Wx(n) + W_{back}y(n)) \quad (2.1)$$

Onde:

$x(n+1)$ : Estado do *Reservoir* no instante  $n+1$ .

$x(n)$ : Estado do *Reservoir* no instante  $n$ .

$u(n+1)$ : Entradas da rede no instante  $n+1$ .

$y(n)$ : Saída da rede no instante  $n$ .

$W_{in}$ : Matriz de pesos de entrada da rede.

$W$ : Matriz de pesos do *Reservoir*.

$W_{back}$ : Matriz de pesos de retroalimentação.

$f(\ )$ : Função de ativação.

Para calcular a saída da rede é usada a equação (2.2):

$$y(n) = x(n) W_{out} \quad (2.2)$$

Onde:

$y(n)$ : Saída da rede no instante  $n$ .

$x(n)$ : Estado do *Reservoir* no instante  $n$ .

$W_{out}$ : Matriz de pesos de saída.

A matriz de pesos de saída  $W_{out}$  é modificada durante a etapa de treinamento, enquanto as outras matrizes  $W_{in}, W, W_{back}$  permanecem fixas desde o momento da criação da rede.

#### 2.2.4. Treinamento

As redes ESN têm dois tipos de treinamento, *On-line* e *Off-line*, mas os dois tipos de treinamento são feitos para adaptar os pesos da camada de saída que conecta os neurônios do *Reservoir* aos neurônios da camada de saída da rede.

##### 2.2.4.1. Treinamento *On-Line*

O treinamento *On-line* é realizado em cinco passos:

1. Geram-se as quatro matrizes de pesos  $W_{in}, W, W_{back}$  e  $W_{out}$ , lembrando que a matriz de pesos do *Reservoir*,  $W$ , deve considerar a limitação imposta pela porcentagem de conexões aleatórias permitidas pelo parâmetro “Percentual (%) de Interconexão”, além de utilizar o Raio Espectral para escalar os valores dos pesos.
2. No segundo passo do treinamento, são fornecidas duas condições iniciais: uma vez que no instante inicial a rede precisa de duas informações do tempo anterior para calcular o valor do estado, conforme especificado na Equação (2.1):
  - A primeira condição define o estado inicial do *Reservoir*, que pode ser  $x(0) = 0$ , ou inicializado com valores aleatórios, nesse caso tem que considerar que o eco produzido por valores aleatórios vai demorar mais para desaparecer;
  - A segunda condição é o estado da camada de saída, que pode ser  $y(0) = 0$  ou também iniciado com um valor aleatório; caso  $W_{back} \neq 0$ , deve-se ter a mesma precaução da primeira consideração.
3. Calcula-se a saída da rede  $y(n)$  baseado na equação (2.2).
4. Calcula-se o erro entre a saída da rede  $y(n)$  e a saída desejada  $y_{target}(n)$

$$e_y = y(n) - y_{target}(n) \quad (2.3)$$

5. O quinto e último passo refere-se à atualização da matriz de pesos  $W_{out}$ , baseado no erro calculado no passo anterior. Para o cálculo desses pesos pode ser usado qualquer algoritmo, como *BackPropagation* (BP) (Werbos, 1990) ou Levenberg-Mardquardt (LM) (Levenberg, 1944). Em (Venayagamoorthy, 2007) é proposto um algoritmo adequado para treinamento on-line de ESN, uma vez que considera os estados do *Reservoir* (Equação (2.4)).

$$W_{out}(n+1) = W_{out}(n) + \eta x(n+1)^T e_y(n+1) + \gamma x(n)^T e_y(n) \quad (2.4)$$

Onde:

$W_{out}(n+1)$ :	Matriz de pesos da camada de saída em $n+1$ .
$W_{out}(n)$ :	Matriz de pesos da camada de saída em $n$ .
$x(n+1)$ :	Estado do Reservoir no instante $n+1$ .
$x(n)$ :	Estado do Reservoir no instante $n$ .
$e_y(n+1)$ :	Erro da saída da rede no instante $n+1$ .
$e_y(n)$ :	Erro da saída da rede no instante $n$ .
$\eta$ :	Taxa de aprendizado.
$\gamma$ :	Momentum.

Depois de atualizar a matriz  $W_{out}$ , retorna-se para o passo 3 e é apresentado um novo padrão para a ESN.

#### 2.2.4.2. Treinamento *Off-Line*

O treinamento *Off-line* é também realizado em cinco passos:

1. Geram-se as três matrizes de pesos  $W_{in}$ ,  $W$  e  $W_{back}$  lembrando que a matriz de pesos do Reservoir  $W$  tem uma porcentagem de conexões

aleatórias além de ser escalada pelo Raio Espectral. Note-se que neste tipo de treinamento a matriz de pesos da saída  $W_{out}$  não é gerada, uma vez que será calculada no último passo, depois da apresentação de todos os padrões de treinamento.

2. O segundo passo do treinamento refere-se às definições das duas condições iniciais, similar ao caso do treinamento *On-Line*:
  - Na primeira condição define-se o estado inicial do *Reservoir* como  $x(0) = 0$ , ou com valores aleatórios, nesse caso tem que considerar que o eco produzido por valores aleatórios vai demorar mais para desaparecer;
  - A segunda condição refere-se ao estado inicial da saída, podendo ser  $y(0) = 0$ , ou também um valor aleatório; se  $W_{back} \neq 0$ , deve-se ter a mesma precaução da primeira consideração.
3. Calcula-se o estado do *Reservoir* no tempo  $n + 1$ , usando a equação (2.1), e depois o armazena em uma matriz  $DR$ , além de armazenar todas as saídas desejadas no vetor *Teacher*. Este passo é repetido até que sejam apresentados todos os padrões de treinamento e suas respectivas saídas.
4. Extraí-se da matriz de estados  $DR$  e do vetor *Teacher* o primeiro dado, porque este foi produzido pelas considerações iniciais, gerando assim os elementos  $DR^{forget}$  e  $Teacher^{forget}$ .
5. O quinto e último passo refere-se ao cálculo da matriz de pesos de saída  $W_{out}$ , baseado na equação (2.5).

$$W_{out} = (Pseudoinverse(DR^{forget}) * Teacher^{forget})^T \quad (2.5)$$

A possibilidade das ESN serem treinadas de forma *off-line*, sem precisar modificar os pesos de forma gradual, é sua maior vantagem, uma vez que o treinamento é realizado em um só passo e, dependendo da configuração da rede, pode levar até a um erro de treinamento próximo de zero.

Em suma, os parâmetros configuráveis das ESN que influenciam no desempenho do treinamento *off-line* são: o Raio Espectral, o Tamanho do



*Reservoir* (Número de Neurônios), a Porcentagem de conexões internas do *Reservoir* e o uso do *Feedback*.

## 2.3.

### Particle Swarm Optimization (PSO)

O algoritmo PSO é um algoritmo de otimização estocástico, baseado em populações, que visa à imitação dos movimentos de indivíduos em um enxame durante a procura por um objetivo, como, por exemplo, o alimento (Kennedy e Russell, 1995). Este algoritmo foi inspirado no comportamento de um bando de gaivotas e em cardumes de peixes. No PSO, um enxame de  $N$  indivíduos, chamados de partículas, é usado para buscar a solução mais ótima para determinado problema baseado na avaliação de cada partícula por meio da função objetivo.

No modelo matemático adotado para o PSO, um “enxame de partículas” é gerado aleatoriamente em um determinado espaço de busca. Dentro deste espaço, cada partícula corresponde a uma possível solução ao problema, representada por sua posição no espaço de busca do problema (Vasconcelos, 2007).

#### 2.3.1.

##### Representação

Cada partícula do enxame tem associado um valor de velocidade  $V$  e um valor de posição  $X$  no espaço de busca. O valor de posição  $X$  é deslocado sob a ação de três vetores: inércia, memória e cooperação. A Figura 3 ilustra um exemplo de deslocamento desde uma posição  $x_{i-1}$ , até a posição  $x_i$ , fazendo uso dos vetores mencionados. O primeiro vetor conduz a partícula na direção idêntica à que ela tinha (inércia); o segundo atrai a partícula na direção da melhor posição da mesma partícula até o momento (memória); e o último vetor atrai a partícula na direção da melhor posição descoberta pelo enxame até o momento (cooperação).

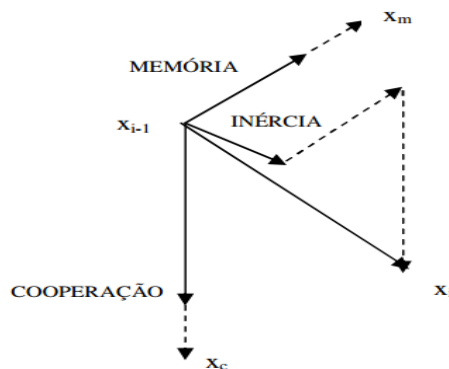


Figura 3. Movimento de uma Partícula.

Quando a partícula atinge a sua nova posição  $X_i$ , esta é avaliada e no caso de superar o seu estado anterior, são atualizados os valores da sua melhor posição e a melhor posição de todo o enxame.

### 2.3.2. Algoritmo

O primeiro passo na implementação do algoritmo é a geração das  $N$  partículas que formarão o Swarm ou o enxame, com suas respectivas posições. Pode-se também, neste momento, arbitrar velocidades iniciais para cada partícula.

Como ilustra o fluxograma da Figura 4, o algoritmo manter-se-á ativo, atualizando os vetores de velocidade e posição de cada partícula até que seja atingido qualquer critério de parada, como o número máximo de iterações ou uma partícula atingir uma posição com uma avaliação desejada.

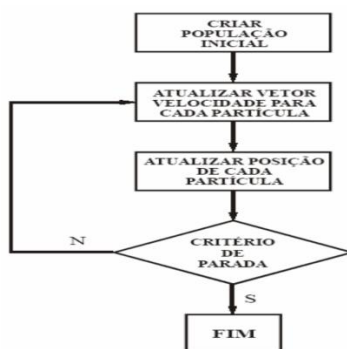


Figura 4. Fluxograma do PSO

O cálculo da velocidade de cada partícula é feito pela equação (2.6)

$$V_{n+1}^i = \omega V_n^i + c_1 r_1 (Xl^i - X_n^i) + c_2 r_2 (Xg^i - X_n^i) \quad (2.6)$$

Onde:

$V_{n+1}^i$ :	Velocidade da partícula $i$ na iteração $n + 1$ .
$V_n^i$ :	Velocidade da partícula $i$ na iteração $n$ .
$X_n^i$ :	Posição da partícula $i$ na iteração $n$ .
$Xl^i$ :	Melhor posição local (histórico da partícula $i$ ).
$Xg^i$ :	Melhor posição global (histórico do enxame).
$r_1, r_2$ :	Valores aleatórios no intervalo $(0,1)$ .
$c_1, c_2$ :	Parâmetros de confiança.
$\omega$ :	Fator de inércia.

O fator de inércia,  $\omega$ , é utilizado para controlar o impacto dos valores anteriores da velocidade na mesma partícula: valores muito elevados aumentam sua capacidade de exploração do espaço de soluções e diminuem sua convergência. O parâmetro de confiança  $c_1$  define o quanto a partícula confia em si e a induz a seguir a direção de sua melhor posição pessoal. O parâmetro de confiança  $c_2$  define o quanto a partícula confia no bando e a induz a seguir a direção da melhor posição global do enxame, geralmente  $c_1 = c_2 = 2$ .

Além de todos os termos citados anteriormente, existem os valores aleatórios no intervalo (0,1)  $r_1$  e  $r_2$  para que os movimentos fiquem menos previsíveis. Assim, duas partículas que tenham posições atuais e melhores posições pessoais parecidas podem se mover em direções diferentes. Finalmente, baseado na velocidade calculada anteriormente, calcula-se a nova posição usando a equação (2.7).

$$X_{n+1}^i = X_n^i + V_{n+1}^i \quad (2.7)$$

Onde:

$X_{n+1}^i$ :	Posição nova da partícula.
$X_n^i$ :	Posição antiga da partícula.
$V_{n+1}^i$ :	Velocidade nova da partícula.

### 2.3.3. Parâmetros

Em suma, os parâmetros configuráveis de PSO que influenciam o desempenho do algoritmo são:

- Tamanho do enxame (Número de partículas);
- Critério de Parada (Gerações ou Erro);
- Parâmetros de confiança  $c_1$  e  $c_2$ ;
- Fator de Inércia  $\omega$ ;

## 2.4. Algoritmos Genéticos (GA)

Os algoritmos Genéticos constituem uma técnica de busca e otimização, inspirada no princípio Darwiniano de seleção natural (Goldberg, 1999) e reprodução genética. Existem diversas técnicas e algoritmos tradicionais de otimização, mas a maioria requer conhecimento do modelo matemático que representa o problema, e na maioria das aplicações reais o modelo matemático não está disponível ou é difícil de obter. Um dos principais diferenciais dos Algoritmos Genéticos está exatamente na dispensa dessa informação, facilitando a otimização de problemas complexos, com grandes espaços de busca e de difícil modelagem. Assim, os algoritmos genéticos têm sido aplicados a diversos problemas de otimização (Barboza, 2011) (Michalewickz, 1994) (Birincie, 2006) (Northern, 2007).

O princípio de seleção da teoria de C. Darwin privilegia os indivíduos mais aptos que têm maior longevidade e, portanto, uma maior probabilidade de reprodução. Estes princípios são imitados nos algoritmos genéticos que buscam uma melhor solução para um determinado problema por meio da evolução de populações de possíveis soluções codificadas em cromossomos artificiais.

Um cromossomo é uma estrutura de dados que representa uma possível solução dentro do espaço de busca de um determinado problema. Cromossomos são então submetidos a um processo evolucionário que envolve avaliação, seleção, recombinação sexual (*crossover*) e mutação. Após vários ciclos de evolução a população deverá conter indivíduos mais aptos, ou seja, soluções melhores avaliadas por uma função de avaliação pré-determinada (Goldberg, 1990). O procedimento básico de um algoritmo genético é resumido no fluxograma da Figura 5.

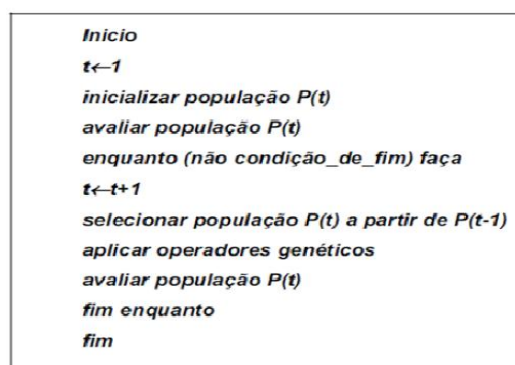


Figura 5 Fluxograma Algoritmo Genético

As seções seguintes apresentam em mais detalhes cada um dos componentes de um algoritmo genético.

#### 2.4.1. Representação

A representação é fundamental na modelagem de um algoritmo genético, pois nesta etapa é definida a estrutura do cromossomo, com os respectivos genes que o compõem, de maneira que este seja capaz de descrever todas as variáveis e condições do problema dentro do espaço de busca (Lazo, 2004). A representação do cromossomo depende diretamente do tipo de problema. Os principais tipos de representação estão contidos na Tabela 1.

Representação	Problemas
<b>Binária</b>	Numéricos, Inteiros
<b>Números Reais</b>	Numéricos
<b>Permutação de Símbolos</b>	Baseados em Ordem
<b>Símbolos Repetidos</b>	Agrupamento

Tabela 1. Principais tipos de representação do Cromossomo (Goldberg, 1991) (Janikow, 1991).

#### 2.4.2. Decodificação

A decodificação consiste em reconstruir a solução real do problema a partir da solução representada pelo cromossomo. O processo de decodificação constrói a solução para que esta seja avaliada pelo problema, e seja comparada com as soluções fornecidas por outros cromossomos.

#### 2.4.3. Avaliação

A avaliação é feita por meio de uma função matemática que pretende ligar o algoritmo genético com o mundo real. A função de avaliação representa o

problema e tem por objetivo fornecer uma medida numérica de aptidão de cada indivíduo da população, que irá dirigir o processo de busca. Funções de avaliação são específicas de cada problema, por isso uma boa função de avaliação, em conjunto com uma boa representação, são elementos vitais na modelagem da solução.

#### **2.4.4. Seleção e Reprodução**

O processo de seleção nos algoritmos genéticos é tipicamente implementado por uma roleta onde cada indivíduo ou cromossomo é representado por uma fatia de tamanho proporcional à sua avaliação relativa; dessa maneira, os indivíduos com as maiores avaliações têm maior probabilidade de serem escolhidos para reprodução.

Existem basicamente quatro técnicas de reprodução que definem a forma pela qual os indivíduos são substituídos entre gerações:

- *Troca de toda a população*: todos os indivíduos da população são trocados a cada geração;
- *Troca de toda a população com elitismo*: todos os indivíduos da população são trocados a cada geração, porém o indivíduo mais apto é sempre copiado para a geração seguinte;
- *Troca parcial da população (steady state)*: substitui-se apenas uma parcela da população a cada geração;
- *Troca parcial da população (steady state) sem duplicados*: técnica semelhante à anterior. No entanto, os indivíduos duplicados são descartados e substituídos;

Para maior informação sobre as características e configurações das técnicas em (Goldberg, 1990).

#### **2.4.5. Operadores Genéticos**

Os indivíduos que foram selecionados para reprodução são recombinados por meio de operadores de *crossover* e de mutação. A realização ou não do

*crossover* e da mutação depende da taxa de *crossover* e da taxa de mutação, que são parâmetros definidos no início da implementação do algoritmo genético.

Nos operadores de *crossover*, os novos indivíduos ou filhos, são criados a partir do cruzamento dos genes dos cromossomos pais que foram selecionados, fornecendo a convergência de boas soluções, enquanto os operadores de mutação trocam de forma aleatória valores de genes dos cromossomos pais, garantindo a diversidade e permitindo que sejam introduzidas informações que não estiveram presentes nos outros indivíduos.

Para uma maior informação sobre os diferentes tipos de operadores de *crossover* e mutação, consulte (Davis, 1990).

#### **2.4.6. Parâmetros**

Em suma, os parâmetros configuráveis dos GA que influenciam o desempenho do algoritmo são:

- Tamanho da População;
- Critério de Parada (Gerações ou Erro);
- Taxa de *Crossover*;
- Taxa de Mutação;
- Cromossomo; e
- Função de Avaliação.



### **3. Técnicas de Previsão de Séries Temporais**

#### **3.1. Introdução**

A previsão de séries temporais é de extrema importância e está presente em diversas áreas: economia, meteorologia, processamento de sinais, controle, setor elétrico, setor financeiro, entre outros (Jenkins, 1976; Jenkins, 2008). Atualmente, todos os tipos de empresas, de pequeno ou grande porte, estatal ou privada, necessitam se planejar de forma a atender, ao menor custo possível, a demanda do mercado, visando a maximizar a sua rentabilidade ou bem-estar da população (Wanke, 2006).

No setor financeiro, a previsão de séries centra-se, principalmente, no mercado de ações na bolsa de valores (Fu-Yuan, 2008; Khoa, 2008; Bajestani, 2009; Hadavandi, 2010). Prever o valor de ações é de extrema relevância para os investidores que buscam ampliar os seus lucros, a partir de valoração das ações, opções ou dividendos. Contudo, na academia, pesquisadores buscam entender o comportamento do mercado de ações (Fang, Luo, Fei e Li, 2010) (Du, Luo, He e Xie, 2010), pois prever o valor das ações é uma questão em aberto, pois estas são afetadas por diversas variáveis econômicas, sociais e políticas, que na maioria dos casos estão inter-relacionadas de forma complexa.

Na economia, bancos centrais precisam de uma perspectiva da inflação futura para controlar os instrumentos que afetam o comportamento da economia e alertar os agentes econômicos e políticos. As decisões sobre política monetária necessitam de diversas informações, tais como: compreender o processo gerador da inflação futura; o mecanismo de transmissão dos juros de curto prazo para os juros de longo prazo; as taxas de câmbio; a atividade econômica real e a dinâmica da inflação (Kooths, Mitze e Ringhut, 2003). Na literatura são encontradas inúmeras técnicas econométricas para calcular a previsão da inflação (Wang, 2010) (Chen, 2001) (Dongdong, 2010) (Chávez, 2013); embora ainda seja uma

tarefa desafiadora, altos investimentos estão sendo feitos para desenvolver modelos com maior exatidão.

Uma das principais aplicações da previsão de séries temporais se dá no setor elétrico onde, normalmente, são feitas: previsão de carga (Dongxiao, 2007; Dongxiao, 2010; Nadimi, 2010; Baczy, 2004), previsão de vazão (Figueiredo, 2007) (Anderson, 2002) (Andreolli, 2003) e previsão dos preços da energia elétrica (Zhou, 2004) (Amjady, 2007; Lei, 2010). As previsões de carga são indispensáveis para a tomada de decisões referentes à compra, geração de energia elétrica, investimentos em infraestrutura e capacidade, perfazendo importantes indicadores para as empresas que atuam no setor elétrico (geração, transmissão, distribuição, financiamento, entre outras) (Campbell, 2006).

As seções seguintes abordam técnicas quantitativas de previsão de séries temporais, mencionando brevemente as técnicas estatísticas e as técnicas de inteligência computacional, dando maior ênfase na aplicação de redes neurais, base da ferramenta desenvolvida neste trabalho.

### **3.2. Técnicas Estatísticas**

Há diversas técnicas estatísticas para previsão, mas em sua maioria são baseadas na identificação de uma ou mais componentes da série temporal (Hanke, 1995), tais como:

*Nível:* Geralmente, é simplesmente o ponto inicial de uma série histórica e representa o comportamento da série caso não existisse nenhuma outra componente.

*Tendência:* Representa o crescimento ou redução da série no médio ou longo prazo.

*Sazonalidade:* Representa um comportamento periódico de curto ou médio prazo da série, por exemplo, sabe-se que a temperatura da terra é menor nos meses do inverno e maior no verão, comportamento que está presente ano a ano.

*Ciclo:* semelhante à sazonalidade, mas reflete flutuações com ocorrência que vão além do período compreendido dentro do ano, como por exemplo, variações econômicas das nações, ciclos solares, etc.

*Aleatoriedade:* São variações imprevisíveis, causadas principalmente por eventos particulares e não recorrentes.

Entre as técnicas mais conhecidas de previsão de séries estão os de amortecimentos exponenciais (simples, duplo e triplo) (Hanke, 1995), as médias móveis (simples e dupla) (Mentzer, 1998), o ARIMA e o SARIMA (Box e Jenkins, 1976). Porém, essas técnicas exigem uma análise prévia da série de forma a identificar seus componentes e, assim, selecionar a técnica mais adequada. Além disso, todas são baseadas na premissa de que a relação entre os valores passados e os futuros é linear, além da presença de alguns pressupostos com respeito ao comportamento da série. E no caso de séries com comportamentos mais complexos elas não conseguem prever de forma correta.

### **3.3. Técnicas de Inteligência Computacional**

A inteligência computacional vem sendo aplicada cada vez mais na previsão de séries temporais, graças à sua flexibilidade, abrangência e facilidade de implementação. Na maioria das técnicas estatísticas é necessário prévio conhecimento sobre análises e estudos das séries temporais, enquanto que uma das principais características das técnicas inteligentes é a dispensabilidade dessa informação.

As principais técnicas inteligentes que são usadas em previsão de séries temporais são as Redes Neurais Artificiais (RNA) (Zhang, 2007; Qingle, 2010; Chai, 2010) e os Sistemas de Inferência Fuzzy (SIF) (Chen, 2011; Chen, 2010; Wong, 2009); porém, esses modelos inteligentes têm parâmetros configuráveis que afetam diretamente o desempenho, como a quantidade de neurônios de uma camada e a taxa de aprendizado de uma Rede Neural, ou os parâmetros das ESN descritos ao longo do Capítulo 2. Atualmente existem diversos meios para encontrar os valores ótimos destes parâmetros, e um deles é a mistura de duas ou mais técnicas inteligentes, gerando sistemas de inteligência computacional híbridos.

### 3.4. Sistemas de Inteligência Computacional Híbridos

Os sistemas híbridos surgem da necessidade de configurar sistemas inteligentes, e existem três tipos básicos (De Souza, 1999):

- *Híbrido sequencial*: Onde um subsistema A atua como entrada de outro subsistema B.
- *Híbrido auxiliar*: Onde um subsistema A invoca um subsistema B para executar uma tarefa.
- *Híbrido incorporado*: Onde não há uma separação visível entre os dois subsistemas.

Os sistemas híbridos com maior aplicação na previsão de series são:

- *Sistemas Neuro-Fuzzy* (Yue, 2008; Yihang, 2009): do tipo híbrido incorporado, consiste em implementar um SIF por meio da arquitetura de uma RNA, utilizando os algoritmos de aprendizado das Redes Neurais para ajustar um Sistema Fuzzy.
- *Sistemas Neuro-Evolutivos* (Li, 2010; Pacífico e Ludermir, 2013): do tipo híbrido auxiliar, Onde uma Rede Neural invoca um algoritmo de otimização para o ajuste dos parâmetros de configuração e aprendizado da rede.
- *Sistemas Fuzzy-Genéticos* (Ding, 2008): do tipo híbrido auxiliar, Onde um Sistema Fuzzy invoca um algoritmo de otimização para o ajuste dos parâmetros de configuração do sistema.
- *Sistemas Neuro-Fuzzy-Evolutivos* (Li, 2006): do tipo híbrido auxiliar, Combinação das técnicas de RNA, SIF e um algoritmo de otimização. São sistemas neuro-fuzzy com otimização de parâmetros.

A ferramenta de previsão desenvolvida nesta dissertação envolve dois modelos de topologia híbrido-auxiliar do tipo Neuro-Evolutivo, no qual os

parâmetros de uma Echo State Network são ajustados automaticamente por *Particle Swarm Optimization* e Algoritmos Genéticos.

### 3.5. Avaliação do Desempenho de uma Previsão

O desempenho de um modelo de previsão é calculado a partir da comparação dos valores previstos com os valores reais (erro). Existem diferentes métricas de erro que podem ser utilizadas em uma previsão. Todas descrevem de alguma forma a diferença entre os valores reais e os valores previstos. As mais utilizadas são *Mean Percentual Error* (MPE) (Equação (3.1)), *Mean Absolute Percentual Error* (MAPE) (Equação (3.2)), *Rooted Mean Squared Error* (RMSE) (Equação (3.3)) e *Symmetric Mean Absolute Percentual Error* (SMAPE) (Equação (3.4)).

$$MPE = \frac{1}{n} \sum_{t=1}^n \frac{R_t - P_t}{R_t} * 100 \quad (3.1)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{R_t - P_t}{R_t} \right| * 100 \quad (3.2)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (R_t - P_t)^2} \quad (3.3)$$

$$SMAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{R_t - P_t}{(R_t + P_t)/2} \right| * 100 \quad (3.4)$$

Onde:

$R_t$ : Valor real da série no instante  $t$ .

$P_t$ : Valor previsto da série no instante  $t$ .

$n$ : Quantidade total de instantes de tempo.

### 3.6. Tipos de Previsão

Existem dois tipos fundamentais de previsão, o “*Multistep*” e “*Singlestep*”, sendo a principal diferença é o fato de que no *Singlestep* os valores previstos nunca são utilizados na janela de entrada da rede. A seguir explica-se a diferença para uma previsão com um horizonte de três passos à frente e uma janela de cinco valores passados consecutivos:

- *Previsão Multistep*: Como ilustra a Figura 6, na previsão *Multistep*, as variáveis de entrada do modelo são os valores imediatamente anteriores do valor que vai ser previsto, exemplo no instante  $k + 1$  foram usados os cinco valores anteriores como entrada, e no seguinte passo para prever  $k + 2$  o valor previsto em  $k + 1$  faz parte da janela de entrada, e no seguinte para prever o  $k + 3$ , os valores de  $k + 1$  e  $k + 2$  fazem parte da janela de entrada.

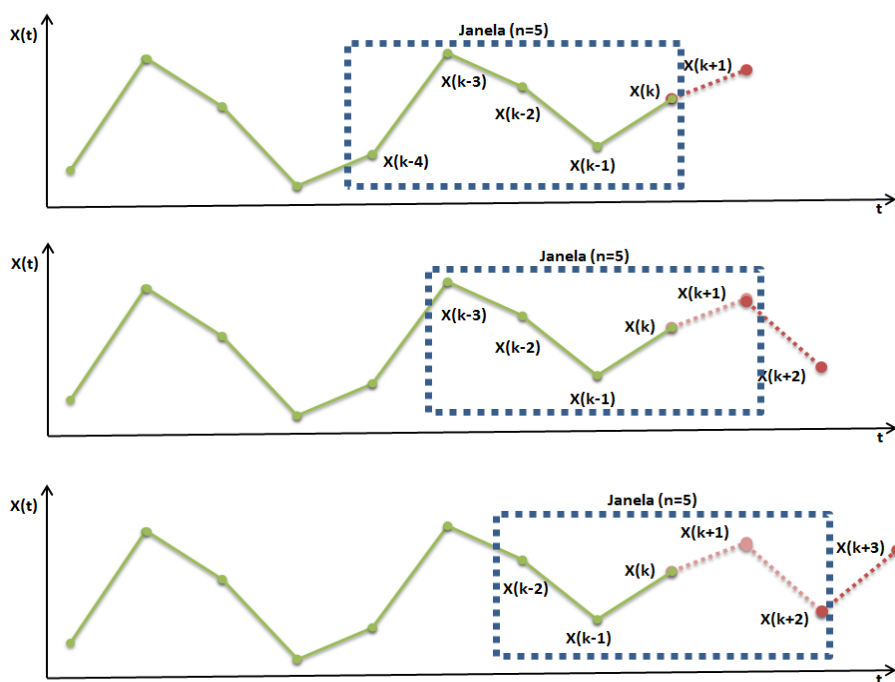


Figura 6 Passos Previsão Multistep

- *Previsão Singlestep*: Como ilustra a Figura 7, na previsão *Singlestep*, são realizadas  $h$  previsões do  $h$ -ésimo valor à frente, onde  $h$  é o horizonte de previsão, por isso na previsão *Singlestep*, os valores previstos nunca são usados como entrada para calcular um próximo valor.

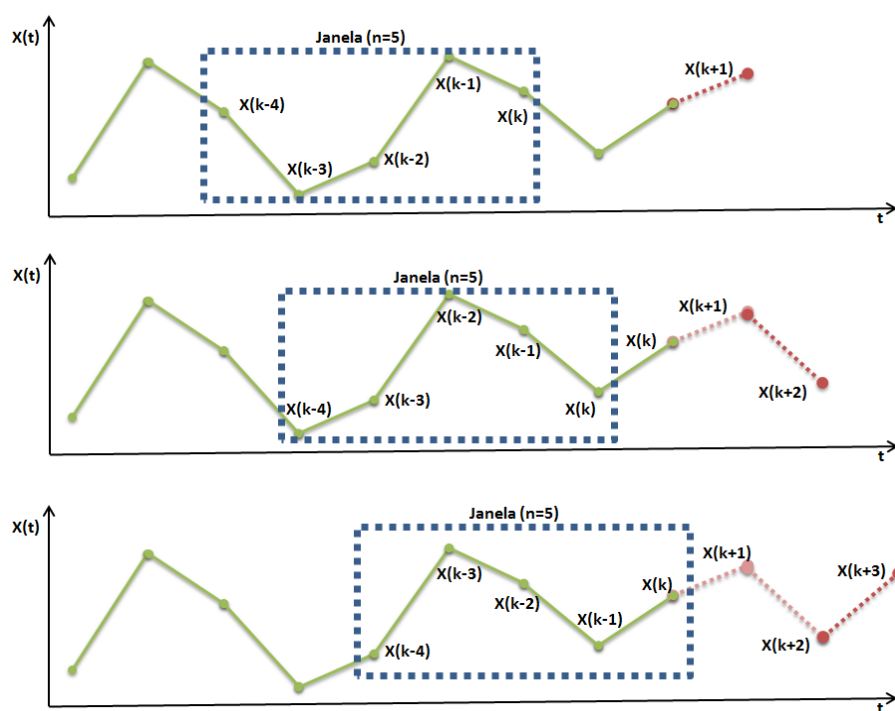


Figura 7 Passos Previsão *Singlestep*

Essa diferença entre os dois tipos de previsão é fundamental e tem grande impacto no desempenho do modelo predictor. Espera-se que para determinadas séries a previsão *Singlestep* gere melhores resultados, enquanto que com outras séries os melhores resultados sejam calculados utilizando o *Multistep*.

### 3.7. Parâmetros

Resumindo, os parâmetros configuráveis de uma previsão que influenciam no desempenho são:

- Tamanho da Janela;
- Tipo de Previsão.

O capítulo seguinte apresenta detalhadamente a ferramenta computacional de previsão de séries desenvolvida nesta dissertação, baseada na teoria das *Echo State Networks*, em conjunto com a otimização de parâmetros por *Particle Swarm Optimization* e Algoritmos Genéticos.

## 4. Modelos Híbridos para Previsão de Séries Temporais

### 4.1. Introdução

A definição dos parâmetros das *Echo State Networks* é de vital importância e tem um impacto direto no desempenho da previsão. Apesar de ser possível realizar ajustes manuais para cada um destes parâmetros até alcançar um desempenho satisfatório, é mais interessante empregar métodos de ajuste automáticos. Por isso, nesta dissertação, foram desenvolvidos dois modelos híbridos do tipo “híbrido auxiliar”, onde os parâmetros de uma *Echo State Network* são ajustados usando Algoritmos Genéticos e *Particle Swarm Optimization*. Além desses dois algoritmos de otimização, testou-se também, numa etapa anterior do projeto, o algoritmo *Simulated Annealing*, entretanto, este foi descartado por ter apresentado um desempenho inferior aos Algoritmos Genéticos e ao *Particle Swarm Optimization*.

### 4.2. Modelo ESN-GA

O primeiro modelo ajusta os parâmetros da *Echo State Network* através de Algoritmos Genéticos. A seguir são descritos os principais componentes da modelagem do algoritmo que compõe este modelo híbrido.

#### 4.2.1. Cromossomo

Devido a limitações da biblioteca usada, foi considerada uma representação binária dividida em seções como ilustra a Figura 8, onde o tamanho do cromossomo é definido pela soma do número de bits que representam os parâmetros a serem otimizados.





Figura 8. Cromossomo do GA

Para o cálculo da quantidade de bits necessários para representar cada parâmetro, primeiro foi calculada a quantidade de valores que o parâmetro pode ter, a qual depende dos valores máximo, mínimo e um valor de precisão, como ilustra a equação (4.1), seguido da aplicação do logaritmo em base dois da equação (4.2), para obter a quantidade de bits. No caso do resultado ser um número decimal, realiza-se um arredondamento para cima. A Tabela 2 contém os resultados do cálculo da quantidade de bits que vão representar cada parâmetro.

$$Quantidade\ de\ Valores = \frac{|max - min|}{Precisão} + 1 \quad (4.1)$$

$$Quantidade\ de\ Bits = \log_2(Quantidade\ de\ Valores) \quad (4.2)$$

Parâmetro	Max	Min	Precisão	Valores	Log <sub>2</sub>	Bits
Tamanho Reservoir	200	1	1	200	7,64	8
% de Conexão	100	0	1	101	6,65	7
Raio Espectral	0,999	0	0,001	1000	9,96	10
FeedBack	1	0	1	2	1	1
Tamanho janela	100	1	1	100	6,64	7
<b>Bits Totais do Cromossomo</b>						<b>33</b>

Tabela 2. Cromossomo 1.

Note-se que no caso do parâmetro *FeedBack*, se deseja otimizar a presença ou ausência do mesmo; por isso é usado só um bit (1=Com *Feedback*, 0=Sem *Feedback*). Além disso, o tamanho máximo do *reservoir* foi limitado em 200 unidades, com o objetivo de minimizar o custo computacional no momento de calcular a pseudoinversa da matriz de estados. Finalmente, no caso do raio espectral foram considerados números com até três casas decimais, por isso foi usada uma precisão de 0,001.

O caso do tamanho da janela é especial, podendo ser representada de duas formas: a primeira com sete bits (Tabela 2), que representam um número inteiro que faz referência ao tamanho; e a segunda como uma representação binária fixa de  $m$  bits onde  $m$  é o tamanho da janela, sendo cada bit associado a uma das variáveis presentes na janela. Assim, são escolhidas as variáveis internas da janela

mais pertinentes, pois nem sempre os  $m$  valores da janela são relevantes. Por exemplo, supondo que após uma otimização, onde  $m = 12$ , o resultado da janela foi 100000100100, isto indica que o terceiro, sexto e décimo segundo valor da janela são relevantes e os outros são descartados como entrada da Echo State Network, conforme ilustrado na Figura 9.

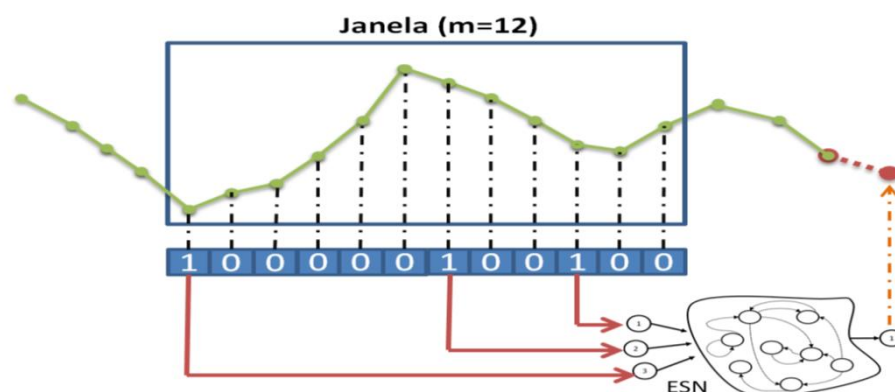


Figura 9. Janela Otimizada

Neste segundo tipo de representação da janela, o limite máximo foi limitado em 100 bits, igual ao valor numérico máximo permitido para o tamanho da janela no caso da primeira forma de representação. A Tabela 3 resume as informações do cromossomo no caso de representar a janela através do segundo formato.

Parâmetro	Max	Min	Precisão	Valores	Log2	Bits
Tamanho Reservoir	200	1	1	200	7,64	8
% de Conexão	100	0	1	101	6,65	7
Raio Espectral	0,999	0	0,001	1000	9,96	10
FeedBack	1	0	1	2	1	1
Janela	Até					100
Bits Totais do Cromossomo Até						126

Tabela 3. Cromossomo 2.

#### 4.2.2. Função de Avaliação

Esta função é a encarregada de medir e avaliar o desempenho de cada cromossomo, seguindo os passos do fluxograma da Figura 10.



Figura 10. Fluxograma Função de Avaliação GA

Decodificação do cromossomo: consiste na transformação da informação contida nos bits do cromossoma. Para isso, o cromossomo é dividido em seções, como ilustra a Figura 11, e descrito na seção 4.2.1.

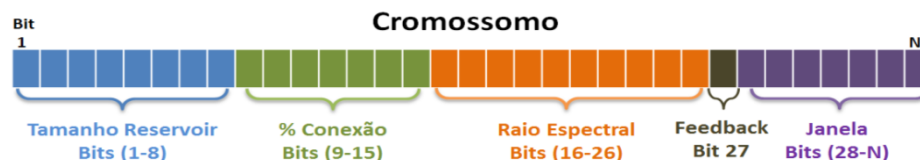


Figura 11. Divisões do Cromossomo

Cada seção é um número binário de tamanho  $m$ , o qual é convertido no número decimal correspondente, conforme exemplificado na Figura 12.

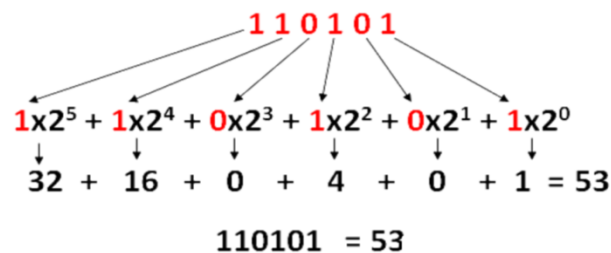


Figura 12. Conversão Binário para Decimal

Após a conversão para decimal, cada resultado tem um tratamento independente:

- O valor real da variável Tamanho do *Reservoir* é igual ao valor da conversão binário-decimal, como indica a equação (4.3).

$$Tamanho\ Reservoir = Tamanho\ Reservoir\ Decimal \quad (4.3)$$

- No caso da porcentagem de conexão, após a conversão binário-decimal tem-se um número inteiro entre 0 e 127; como a porcentagem de conexão real é um valor percentual ou seja entre (0%-100%), para se obter o valor correto deve-se dividir o resultado da conversão por 128 seguido da

multiplicação por 100 e um arredondamento final, como ilustra a equação (4.4).

$$\% \text{ Conexão} = \left\| \frac{\% \text{ Conexão Decimal}}{2^7} * 100 \right\| \quad (4.4)$$

- No caso do Raio Espectral, após a conversão binário-decimal tem-se um número inteiro entre 0 e 1024; como o Raio Espectral real é um número na faixa (0-0.999), para se obter o valor correto deve-se dividir o resultado da conversão por  $2^{10}$ , como ilustra a equação (4.5).

$$\text{Raio Espectral} = \frac{\text{Raio Espectral Decimal}}{2^{10}} \quad (4.5)$$

No caso especial do Tamanho da Janela, o tratamento depende do tipo de representação. Na primeira representação, onde o valor a ser otimizado define o tamanho da janela, o cálculo usado encontra-se na equação (4.6). Já no caso da segunda representação, onde é fixado o tamanho da janela e se otimizam as variáveis internas da mesma, os valores correspondentes aos bits em 0 são retirados das entradas da ESN.

$$\text{Tamanho Janela} = \text{Tamanho Janela Decimal} \quad (4.6)$$

Criação da ESN: Após realizar a decodificação do cromossomo, o passo seguinte é a criação da *Echo State Network*, baseado nas informações tratadas e extraídas do cromossomo (Tamanho do *Reservoir*, % Conexão, Raio Espectral, Janela), além da presença ou ausência do *Feedback*.

Além da criação da rede, neste passo também é realizada a divisão dos dados da série em conjuntos de treinamento e teste, onde o tamanho do conjunto de teste é igual ao horizonte da previsão que será realizada; por exemplo, para uma série composta de 200 pontos onde se deseja efetuar uma previsão de 18 passos à frente, o conjunto de treinamento terá 182 dados e o conjunto de teste conterá os últimos 18 valores.

Treinamento da ESN: Uma vez criada a rede, passa-se ao treinamento da mesma, apresentando todos os padrões do conjunto de treinamento e calculando a matriz de pesos de saída  $W_{out}$ .

Teste da ESN: O último passo para efetuar a avaliação é o teste da ESN, calculando uma métrica de erro (MAPE, RMSE, SMAPE) entre a saída da rede e o conjunto de teste. Esse valor de erro de teste é a medida final de avaliação, a qual vai medir o desempenho do cromossomo.

#### 4.2.3. Seleção e Reprodução

O processo de seleção do algoritmo foi implementado por uma roleta onde cada indivíduo ou cromossomo é representado por uma fatia de tamanho proporcional à sua avaliação calculada na seção 4.2.2; dessa maneira, os indivíduos com as maiores avaliações têm maior probabilidade de serem escolhidos para reprodução.

Com relação à reprodução, implementou-se a técnica que troca a população parcialmente (*Steady State*), mantendo o 5% dos cromossomos da população com maior avaliação na seguinte geração.

#### 4.2.4. Operadores Genéticos

No modelo ESN-GA foram implementados dois operadores genéricos: a mutação binária, a qual realiza a troca do valor de um ou mais bits de 0 por 1, como ilustra a Figura 13.

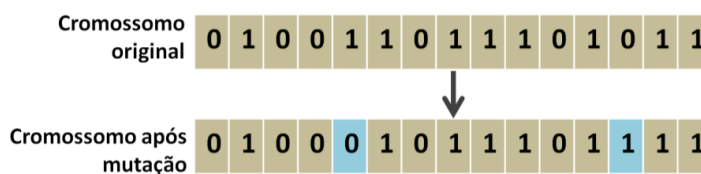


Figura 13. Mutação Binária

O segundo operador usado foi o cruzamento de dois pontos, onde dois cromossomos (genitores) são cortados e recombinaados em dois pontos aleatórios para gerar dois cromossomos novos (filhos), como ilustra a Figura 14.

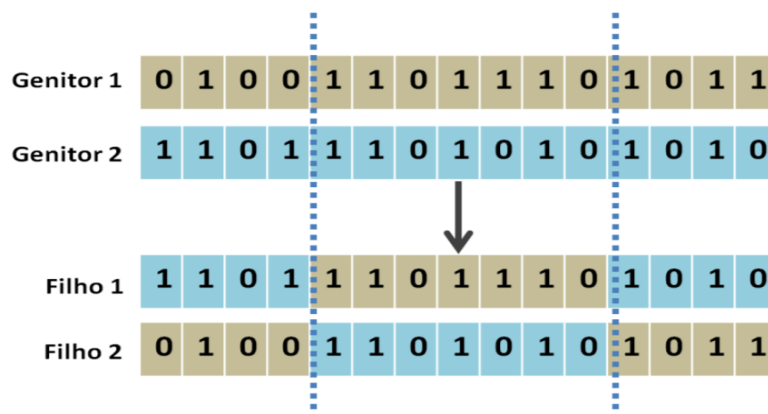


Figura 14. Cruzamento de Dois Pontos

### 4.3.

#### Modelo ESN-PSO

O segundo modelo ajusta os parâmetros da *Echo State Network* através do algoritmo *Particle Swarm Optimization*. As seções seguintes descrevem os principais componentes da modelagem do algoritmo que compõe este modelo híbrido.

#### 4.3.1.

##### Partícula

Diferente do cromossomo do algoritmo genético, a partícula de *Particle Swarm Optimization* é representada diretamente pelos valores reais dos parâmetros, como ilustrado na Figura 15, devido a que o algoritmo faz uma conta matemática direta para atualizar o valor de cada elemento da partícula, acorde ao descrito na seção 2.3.2.

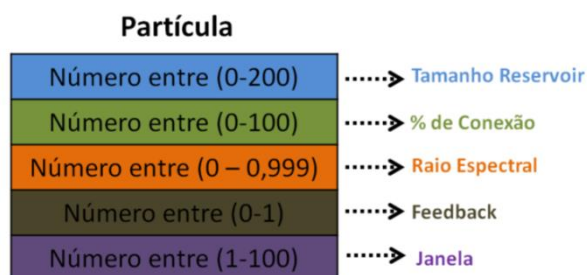


Figura 15. Partícula de PSO

#### 4.3.2.

##### Função de Avaliação

Esta função é a encarregada de medir e avaliar o desempenho de cada partícula, seguindo os passos do fluxograma da Figura 16.

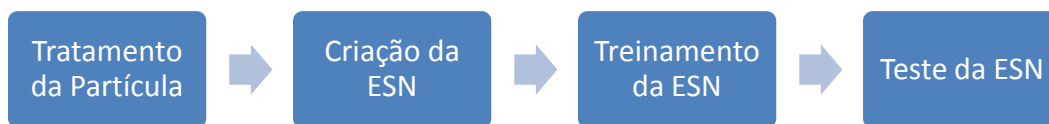


Figura 16. Fluxograma Função de Avaliação PSO

Tratamento da Partícula: nesta etapa são realizados os tratamentos dos valores contidos na partícula. Os três valores da partícula correspondentes ao tamanho do *Reservoir*, à porcentagem de conexão e ao *feedback* são arredondados pois a ESN não pode ter um número decimal de neurônios, assim como a porcentagem de conexão é trabalhada de forma inteira pela ferramenta, e o *feedback* só pode ter dois possíveis valores (0 - sem *feedback* ou 1 - com *feedback*). No caso do terceiro valor da partícula (Raio Espectral), nenhum tratamento é feito, pois ele já é um valor decimal na faixa (0-1).

No caso especial da Janela, o tratamento depende do que será otimizado; na primeira opção, onde se otimiza o tamanho da janela, é realizado o mesmo arredondamento dos valores acima, e, no caso da otimização das variáveis internas da janela, é feito, além do arredondamento, uma conversão decimal para binário baseado na divisão sucessiva por 2 até obter resto 1, como ilustra a Figura 17.

**141/2 = 70 Resta 1**  
**70/2 = 35 resta 0**  
**35/2 = 17 resta 1**  
**17/2 = 8 resta 1**  
**8/2 = 4 resta 0**  
**4/2 = 2 resta 0**  
**2/2 = 1 resta 0**

**141 em binário 10001101**

Figura 17. Conversão Decimal para Binário

Essa conversão é importante para se definir quais variáveis serão eliminadas das entradas da ESN, de acordo com a Figura 9.

Criação da ESN: O passo seguinte é a criação da *Echo State Network*, baseado nas informações obtidas (Tamanho do *Reservoir*, % Conexão, Raio Espectral, Janela), além da presença ou ausência do *Feedback*.

Da mesma forma que no caso do Algoritmo Genético, nesta etapa também se divide a série nos conjuntos de treinamento e teste utilizando o mesmo critério,

onde o tamanho do conjunto de teste é igual ao horizonte da previsão e o restante dos dados é destinado ao treinamento.

Treinamento da ESN: Uma vez que a rede foi criada, o seguinte passo é o treinamento da mesma, apresentando todos os padrões do conjunto de treinamento e calculando a matriz de pesos de saída  $W_{out}$ .

Teste da ESN: Igual ao GA, o último passo é o teste da ESN utilizando uma das métricas de erro existentes na ferramenta (MAPE, RMSE, SMAPE). Esse valor de erro de teste é a medida final de avaliação a qual vai medir o desempenho da partícula.



## 5. EsnPredictor: Ferramenta de Previsão de Séries Temporais

### 5.1. Introdução

O EsnPredictor é uma ferramenta que tem como objetivo principal realizar previsões de séries temporais utilizando um modelo de Redes Neurais do tipo Echo State Network com otimização de parâmetros por *Particle Swarm Optimization* e Algoritmos Genéticos. O EsnPredictor é um aplicativo WPF (*Windows Presentation Foundation*) (WPF, 2014), desenvolvido em duas linguagens de programação:

- C# para os modelos e a lógica da ferramenta;
- Xaml para a parte visual e interação do usuário no ambiente Windows

O EsnPredictor tem dois tipos de otimização:

- *Particle Swarm Optimization*: utilizando a biblioteca SwarmOps, desenvolvida no Hvass-Laboratories, que é desenvolvido pelo Dr. Magnus Erik Hvass Pedersen (Hvasslab, 2014).
- Algoritmos Genéticos: utilizando “The Genetic Algorithm Framework for .Net”, desenvolvido por MSc. John Newcombe (Newcombe, 2014).

Todos os arquivos necessários para o funcionamento do EsnPredictor, encontram-se em uma pasta de nome “EsnPredictor Installer”. A instalação da ferramenta é realizada executando-se o arquivo “setup.exe” presente na pasta. No final da instalação é criado um atalho para a ferramenta EsnPredictor no menu iniciar do Windows e na área de trabalho, assim como um registro na lista de programas instalados. Para removê-lo use o recurso “Adicionar/Remover Programas” do Painel de Controle.

## 5.2. Desenvolvimento

O EsnPredictor foi desenvolvido usando 3 bibliotecas principais para a implementação dos modelos inteligentes propostos nesta dissertação:

- *SwarmOps*: biblioteca desenvolvida no Hvass-Laboratories (Hvasslab, 2014), a qual facilita o uso de *Particle Swarm Optimization*.
- *The Genetic Algorithm Framework for.Net* (Newcombe, 2014): é uma das bibliotecas gratuitas oficiais da Microsoft para implementar algoritmos genéticos.
- *EsnToolbox*: biblioteca de desenvolvimento próprio, a qual facilita o uso e implementação das *Echo State Networks*. Inicialmente foi desenvolvida em MATLAB, e depois migrada para C# visando a uma implementação mais universal que fosse usável em qualquer sistema operativo.

Além das nomeadas acima, foram usadas outras bibliotecas com diferentes objetivos no desenvolvimento da ferramenta:

- *Dynamic Data Display*: biblioteca desenvolvida pelos doutores do grupo oficial de pesquisa da Microsoft, Vassily Lyutsarev e Segey Berezina (DYNAMICDATADISPLAY, 2014), usada para a criação de gráficos e relatórios.
- *DotNetMatrix*: Biblioteca desenvolvida por Paul Selormey (DOTNETMATRX, 2014), usada para a manipulação de matrizes e cálculo da inversa das mesmas.

O EsnPredictor é um aplicativo baseado na arquitetura MVC (Model View Controller), a qual contém diferentes níveis de manipulação (camadas), cada nível com funções específicas: a View centra-se na interação com o usuário e a captura das informações inseridas por ele; o Controller é a camada intermediária encarregada de preparar as informações e padroniza-las para facilitar seu análise e tratamento; e o Model contém todos os cálculos, processos lógicos e tratamentos feitos sobre as informações e os dados do aplicativo.

As principais classes usadas no EsnPredictor são:

- *EchoStateNetworkObject.cs*: Classe com as características principais da configuração da ESN.
- *EchoStateNetworkUtilities.cs*: Classe que contém todos os métodos e funções dedicadas ao treinamento, criação e teste das ESN.
- *GaOptimizationParametersObject.cs*: Contém as informações e a configuração principal do Algoritmo Genético.
- *PsoOptimizationParametersObject.cs*: Contém as informações e a configuração principal do Algoritmo de PSO.
- *SerieTemporal.cs*: Elemento base do EsnPredictor, contém os dados da serie temporal importada, sobre a qual vai ser feita a previsão.
- *PreProcessConfiguraton.cs*: Armazena as características do pré-processamento dos dados.
- *GraficoService.cs*: Classe que contém todos os métodos e funções dedicadas à realização dos gráficos.
- *ImportacaoService.cs*: Contém todos os métodos e funções dedicadas à importação dos dados das series temporais.
- *PreProcesamentoService.cs*: Classe que contém todos os métodos e funções dedicadas à realização do pré-processamento sobre os dados da série temporal, tem ligação com *PreProcessConfiguraton.cs*.
- *ModelParametersObject.cs*: Classe dedicada à junção das informações e características do modelo de otimização que estão contéudas em *GaOptimizationParametersObject.cs* e *PsoOptimizationParametersObject.cs*.
- *ParametersOptimizationObject.cs*: Contém todos os métodos e funções dedicadas à execução da otimização dos parâmetros das ESN, tem ligação direta com *ModelParametersObject.cs* e *EchoStateNetworkObject.cs*.

Em total o EsnPredictor na versão V1.6 (20/07/2014) tem 2425 linhas de código, sem considerar as bibliotecas externas usadas no desenvolvimento. Para maiores informações acerca da arquitetura do software e dos diagramas de classes da ferramenta consultar o Apêndice A.

### 5.3. Estrutura e Metodologia

O EsnPredictor está estruturado em 5 módulos principais, os quais têm um fluxo de utilização ilustrado na Figura 18.

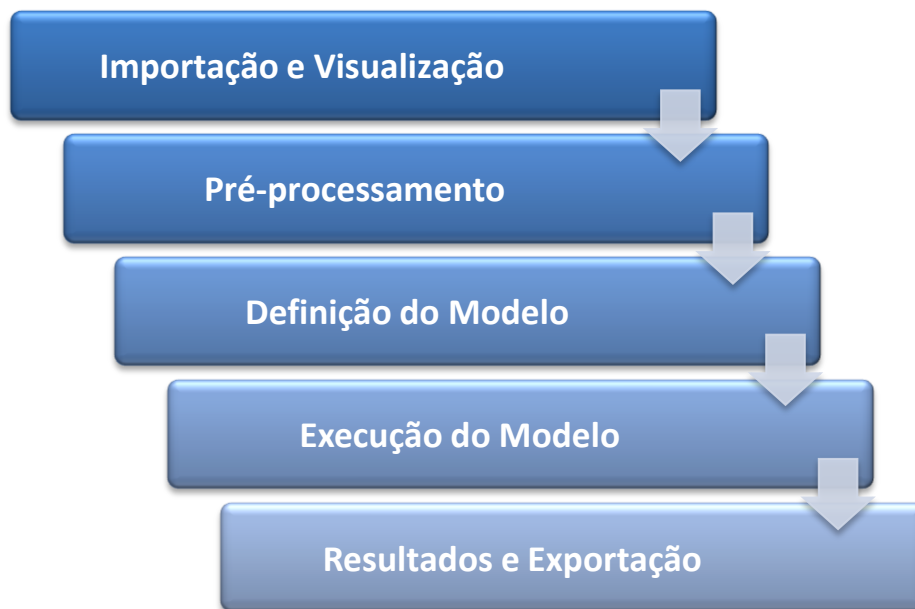


Figura 18. Módulos de EsnPredictor.

No módulo de *Importação e Visualização* é feita a importação da série temporal. Neste módulo é possível visualizar parâmetros e informações relevantes, além do gráfico.

No módulo de *Pré-processamento* é realizado o tratamento dos dados antes de apresentá-los para o modelo preditivo. Note que o pré-processamento dos dados é de vital importância para o desempenho do modelo.

No módulo de *Definição do Modelo* é estabelecida a configuração do modelo, tais como: o tipo de previsão, o horizonte de previsão, o tipo de otimização e os parâmetros que vão ser otimizados.

O módulo de *Execução do Modelo* constrói e executa o modelo que foi definido e configurado previamente, a partir dos dados que foram processados.

O módulo de *Resultados e Exportação* exibe o resultado do modelo e a previsão com possibilidade de exportação dos dados, além da opção de modificar os parâmetros da *Echo State Network* otimizada.

Ao abrir o programa é exibida a tela da Figura 19, informando a versão da ferramenta e os nomes das pessoas envolvidas no projeto.



Figura 19. Tela de Abertura.

Após de alguns poucos segundos a tela de abertura é fechada automaticamente, e a tela principal localizada na aba de importação de dados é aberta (Figura 20). A estrutura do EsnPredictor, tem 3 componentes principais: a toolbar, as abas e o conteúdo de cada aba.

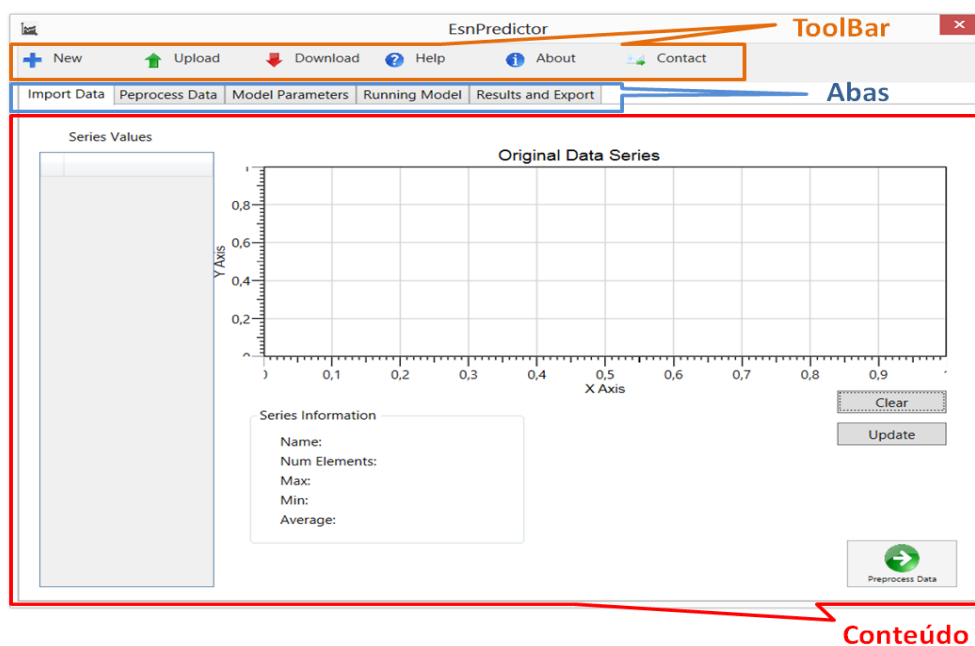


Figura 20. Tela Principal.

Como ilustra a Figura 21, cada módulo tem uma aba associada, situadas na parte superior e que estão desabilitadas inicialmente. As abas podem ser ativadas na medida em que o usuário executa os processos de cada uma, garantindo o fluxo correto da ferramenta.



Figura 21. Abas da Ferramenta ESNPredictor.

Além das abas do ESNPredictor há uma toolbar principal, como é ilustrado na Figura 22, que permite criar um novo predictor, importar a série temporal, exportar os resultados previstos e consultar a ajuda e a descrição da ferramenta, além da informação de contato com os criadores da mesma.



Figura 22. ESNPredictor Toolbar

Com relação ao conteúdo de cada aba, este varia dependendo do módulo. Cada um dos cinco módulos que compõem a ferramenta ESNPredictor é descrito nas seções a seguir.

#### 5.4. Módulo de Importação e Visualização

A primeira aba é destinada à importação e visualização de dados. A Figura 23 ilustra cada um dos componentes da aba.

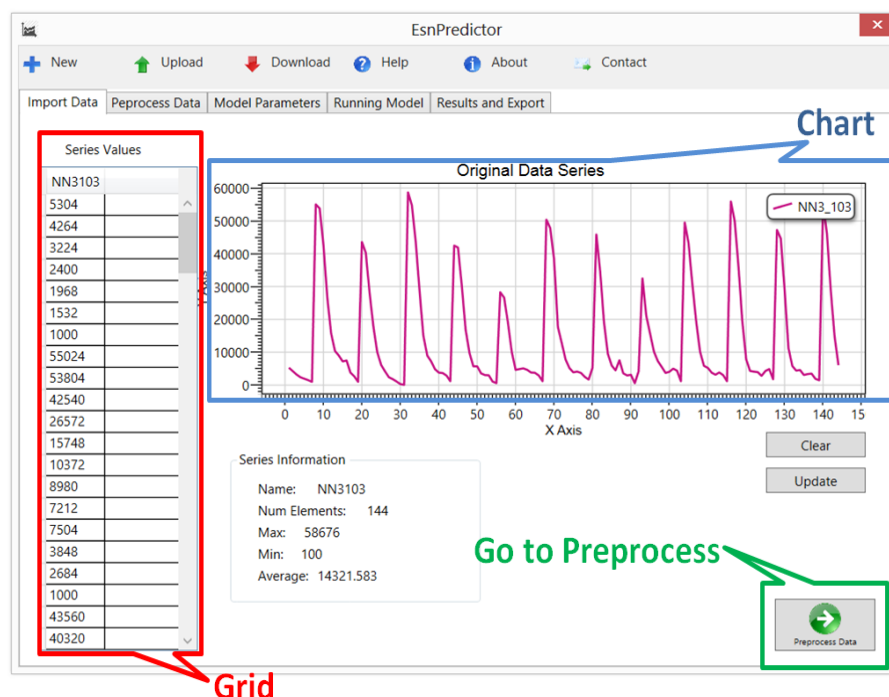


Figura 23. Aba de Importação dos Dados.

Onde:

- *Grid*: Contém o nome e todos os valores numéricos da série temporal;
- *Chart*: Exibe o gráfico da série temporal;
- *Information Box*: Apresenta as informações relevantes da série, como a média, valores mínimo e máximo, o comprimento e o nome;
- *Clear*: Botão que limpa o gráfico;
- *Update*: Botão que atualiza o gráfico com a última série importada;
- *Go Preprocess*: Botão inicialmente desabilitado, que conduz à segunda aba de pré-processamento, que é habilitado após o usuário importar uma série.

Para importar uma série temporal deve-se clicar no botão *Upload* da toolbar (Figura 22), o qual abre a janela de importação (Figura 24) para localizar o arquivo que contém a série temporal. São admitidos arquivos de Excel 97-2003 (.xls), Excel 2010 (.xlsx) e de texto (.txt) como ilustra o filtro na parte inferior direita da Figura 24.

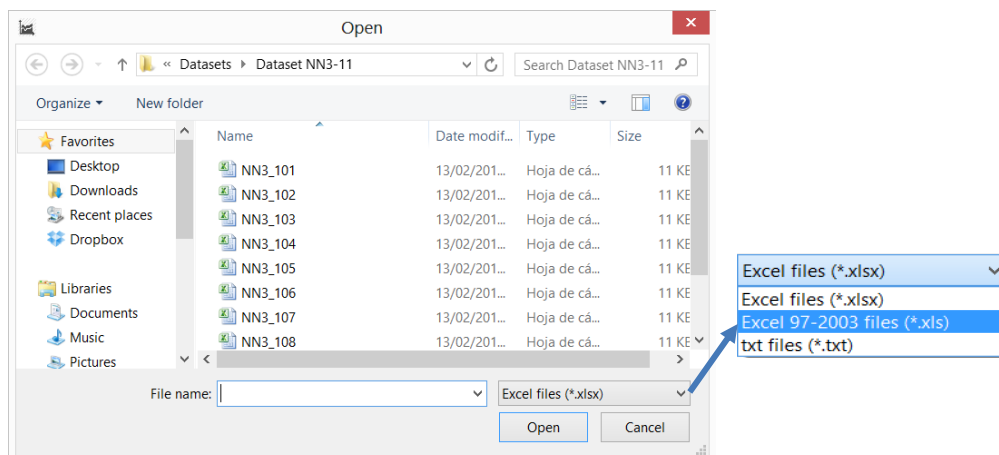


Figura 24. Janela Importação de Arquivo.

Após selecionar o arquivo e clicar no botão *Open*, o conteúdo do grid, o gráfico e as informações da série são atualizadas automaticamente, como ilustra a Figura 23. Neste instante, é habilitada a aba “Preprocess Data”.

Os arquivos de importação devem estar no formato ilustrado na Figura 25. Seja no Excel ou no Bloco de notas, o arquivo deve conter somente uma coluna,

onde o primeiro valor deve ser o nome da série, e do segundo ao último valor da coluna devem conter os valores da série temporal.

A		Serie Temp	
1	Serie Temp	5033	
2	7,636733237	4509,5	
3	5,588205505	3970	
4	1,838429445	3378	
5	4,97948815	2866	
6	5,178456002	2315,5	
7	9,942430106	1895	

b) Excel

a) Texto

Figura 25. Formatos dos arquivos para importar.

## 5.5. Módulo de Pré-Processamento

O objetivo do módulo de pré-processamento é realizar um tratamento prévio sobre os dados originais, facilitando a aprendizagem da rede e visando a um melhor desempenho do modelo. Como ilustrado na Figura 26, há a possibilidade de três tratamentos para realizar sobre a série: Normalização (*Normalization*), Eliminação da Tendência (*Tendency*) e a Interpolação (*Interpolation*).

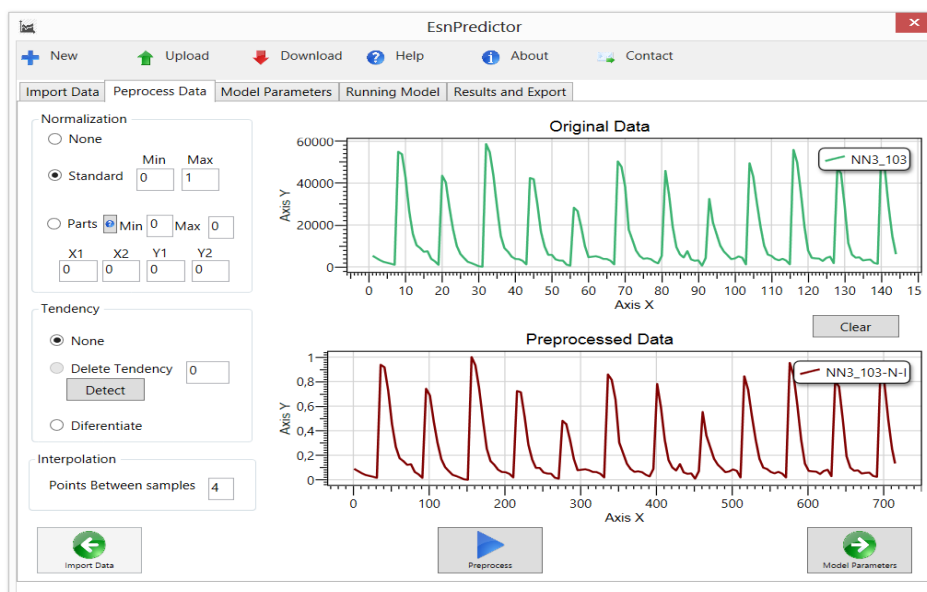


Figura 26. Aba de Pré-Processamento dos Dados.

### 5.5.1. Normalização

O processo de normalização consiste em mapear os valores de uma variável para um novo universo, geralmente  $[0, 1]$ . Isto é feito com o objetivo de facilitar o processo de aprendizado da rede, e visando ao seu pleno funcionamento. O



módulo permite realizar dois tipos de normalização dependendo das características dos dados:

- *Normalização Padrão (Standard)*: É feito para mapear diretamente os valores da série do universo original para um novo universo. No caso da Figura 27, os valores originais estavam entre  $[0, 60000]$  e foram escalados para o universo de  $[0, 1]$ .

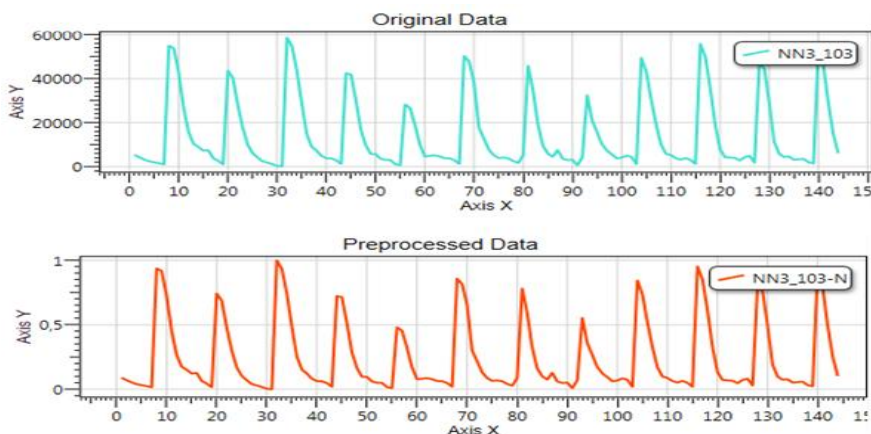


Figura 27. Normalização Standard.

- *Normalização Linear por Partes*: Este tipo particular de normalização é usado quando o valor de uma variável não é uniformemente distribuído no domínio da mesma. Ou seja, quando se tem a maior parte dos dados dentro de uma faixa (geralmente pequena) e uma pequena quantidade de valores com valores distintos, como ilustra a Figura 28.

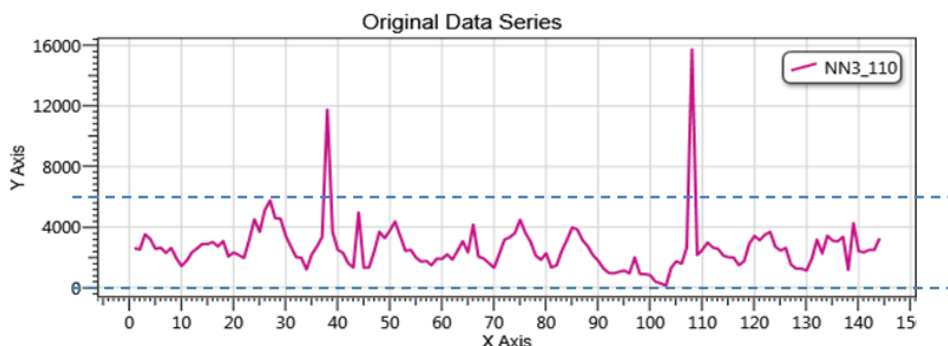


Figura 28. Série com Outliers.

Visando a sanar esse problema, os valores são distribuídos de forma linear em diferentes faixas como dispõe a Figura 29, no qual são designados comumente o percentual dos dados que fica em cada uma destas.

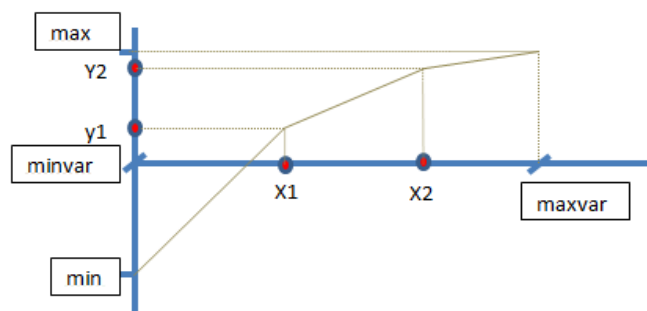


Figura 29. Definição da Normalização Linear por Partes.

A Figura 29 mostra como o universo original (eixo X) deve ser mapeado no novo universo (eixo Y). Este método é baseado em três retas criadas pela Localização dos pontos  $x_1, x_2, y_1$  e  $y_2$ , e a definição dos valores *min* e *max* do novo universo e os valores *minvar* e *maxvar* do universo original da série.

Aplicando a normalização por partes com a configuração da Figura 30 aos dados da

Figura 28, obtém-se o resultado da Figura 31, onde os dois valores que se destacavam na série original (valores próximos às amostras 40 e 110) ainda são os maiores, mas agora sua distância com respeito aos demais é menor.

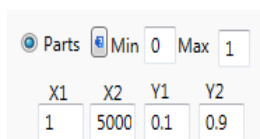


Figura 30. Parâmetros da Normalização por Partes.

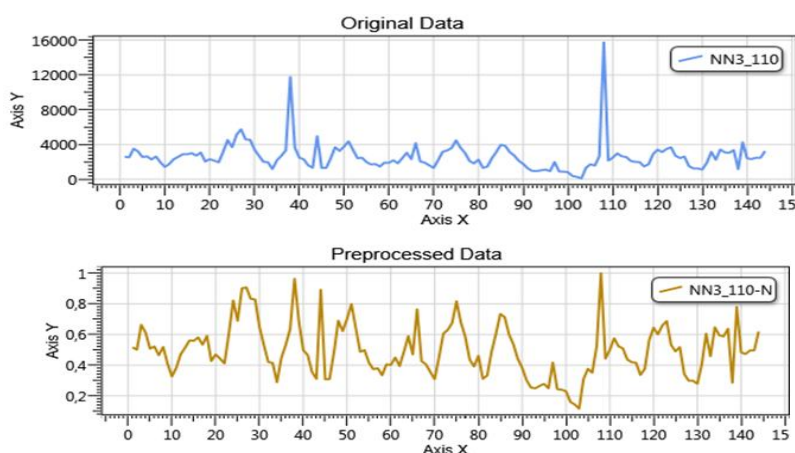


Figura 31. Resultado da Normalização por Partes.

Note que os valores *minvar* e *maxvar* do universo original da série não precisam ser definidos, pois a ferramenta detecta automaticamente. A série resultante da normalização encontra-se na faixa de  $[0, 1]$ , sendo que os valores

que se destacavam anteriormente perderam importância, facilitando assim o treinamento e a aprendizagem da Echo State Network.

### 5.5.2. Eliminação de Tendência

A tendência de uma série é uma componente que pode dificultar o processo de aprendizagem da rede. Por isso, no tratamento dos dados é aconselhável eliminar a tendência, fazendo com que a rede neural foque no comportamento principal da série, sem a tendência. Para isso, a ferramenta fornece dois métodos: eliminar a tendência diretamente ou diferenciar a série.

- *Eliminação Direta:* O primeiro método consiste em criar uma nova série baseada na média móvel dos dados originais, como apresentado no segundo gráfico da Figura 32, o tamanho da media móvel é fixo e corresponde a 10% do tamanho da série. Após subtrair a tendência dos valores da série original, a série resultante tem um comportamento sem tendência, como apresentado no terceiro gráfico da Figura 32. Para o uso deste método é necessário definir graficamente o ponto de corte desde o qual os dados começam a apresentar a tendência. No caso da Figura 32 o ponto de corte foi 60.

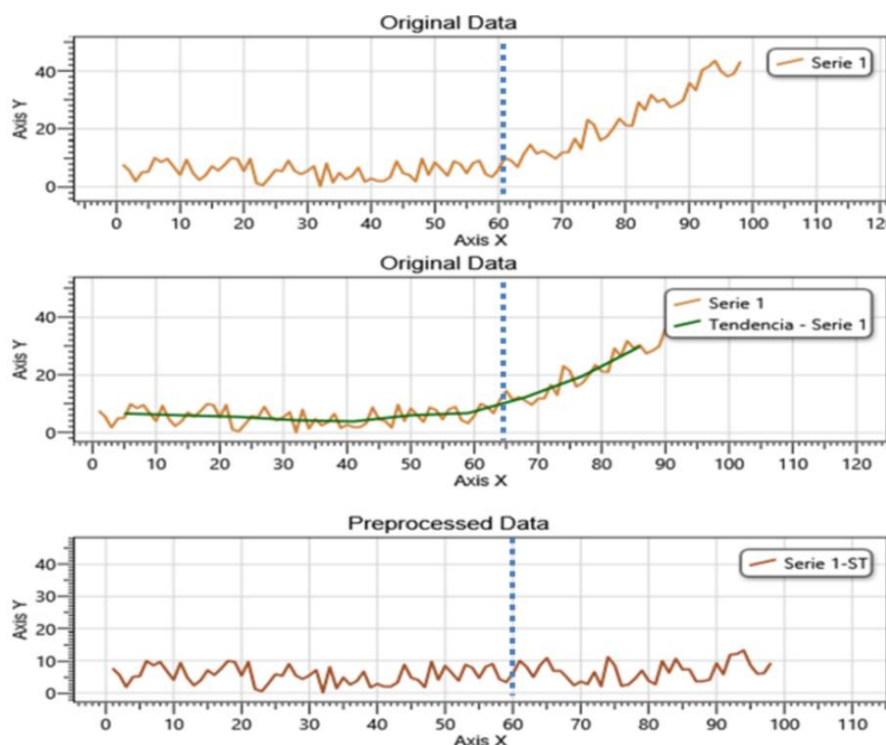


Figura 32. Eliminação da Tendência diretamente.

- **Diferenciação da Série:** O segundo método consiste na criação de uma nova série que contém a diferença entre o valor no instante atual e o valor do instante anterior. Este é calculado pela subtração do valor  $n + 1$  da série com o valor  $n$ , e assim cada valor da nova série mede a variação do valor de um ponto para o seguinte (Figura 33). Quando é usado esse método, a previsão é feita sobre a nova série que contém as variações. Por isso, no final da previsão, a série tem que ser reconstruída para analisar o desempenho de forma correta. Nota-se pela Figura 33 que os valores da nova série oscilam sobre  $y = 0$ .

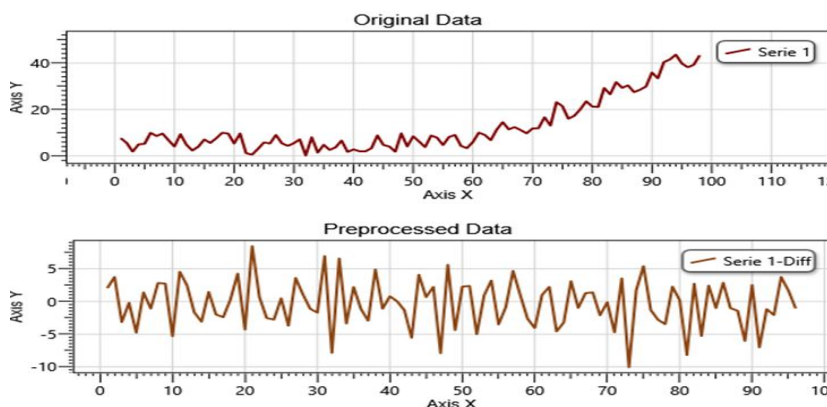


Figura 33. Eliminação da Tendência por Diferenciação.

### 5.5.3. Interpolação

A Interpolação da série consiste em interpolar de forma linear  $k$  novos valores entre dois valores originais, com o objetivo de aumentar o comprimento da série como ilustra a Figura 34, com  $k = 3$ . Esse tratamento é muito útil quando a série é muito pequena ( $n < 100$ ), pois como as *Echo State Networks* trabalham com estados anteriores, estas precisam de um tempo para se estabilizar.

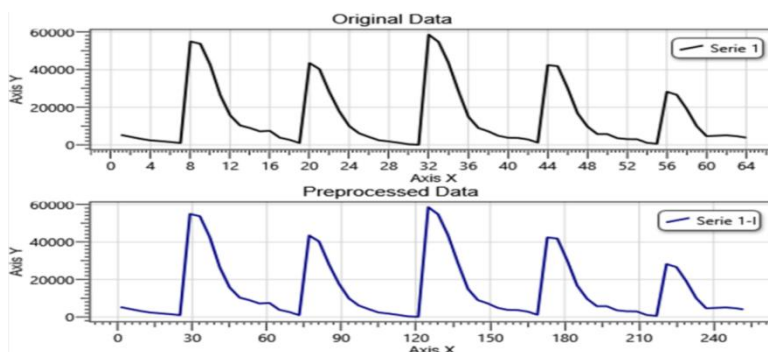


Figura 34. Interpolação da Serie

## 5.6. Módulo de Definição do Modelo

O objetivo deste módulo é configurar todos os parâmetros do modelo que será implementado, além das características gerais da previsão (Figura 35).

The screenshot shows the 'ESNPredictor' application window. The 'Model Parameters' tab is selected. Under 'Prediction Type', 'MultiStep' is chosen with 'Steps' set to 5. Under 'Model Type', 'Single ESN' is selected with parameters: Spectral Radius: 0.8, Connection Percent: 60, Reservoir Units: 20, Window: 1, and Feedback: unchecked. 'GA Optimization' and 'PSO Optimization' are also visible but not selected. The 'Optimization Parameters' section has checkboxes for Spectral Radius, Connection Percent, Reservoir Units, and Feedback, all of which are checked. There are also input fields for 'Fixed Values' and a 'Window Optimization Type' section with 'Window Size' selected.

Figura 35. Aba de Definição do Modelo.

Nota-se, da Figura 35, que o módulo tem três grupos de parâmetros principais, os quais são descritos mais detalhadamente nas seções seguintes:

- *Prediction Type*: Define-se o horizonte de previsão e o tipo de previsão;
- *Model Type*: Estabelece o tipo de modelo que vai ser implementado, seja uma Echo State Network simples ou um modelo híbrido com algum tipo de otimização de parâmetros (GA ou PSO);
- *Optimization Parameters*: É habilitado no caso de ter sido selecionado um modelo híbrido no grupo *Model Type*. Neste grupo são definidos os parâmetros que serão otimizados, tanto da Echo State Network quanto da previsão.

### 5.6.1. Tipo de Previsão

Como ilustra a Figura 36, aqui é possível selecionar o tipo de previsão que será realizado, seja *Multistep* ou *Singlestep*, os quais foram tratados na seção 3.6, além de definir a quantidade de passos ou horizonte da previsão.

Figura 36. Parâmetros Tipo de Previsão.

### 5.6.2. Tipo de Modelo

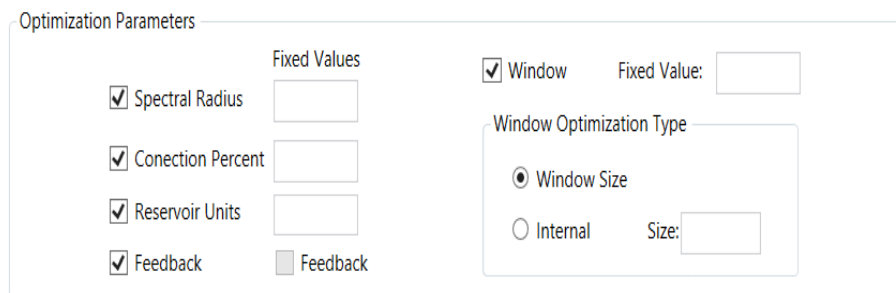
Como ilustra a Figura 37, é possível selecionar o tipo de modelo que será implementado, seja uma ESN simples, ou algum dos dois tipos de modelos híbridos disponíveis (ESN-GA ou ESN-PSO).

Figura 37. Parâmetros Tipo de Modelo.

Após selecionar uma das três possibilidades é habilitado um outro grupo de parâmetros configuráveis, os quais correspondem ao modelo selecionado, como ilustrado a Figura 37. Esses parâmetros correspondem aos descritos nas seções 2.2 (*Echo State Networks*), 2.3 (*Particle Swarm Optimization*) e 2.4 (Algoritmos Genéticos).

### 5.6.3. Parâmetros a serem Otimizados

Se o tipo de modelo selecionado é um dos híbridos, automaticamente é habilitada a configuração ilustrada na Figura 38, onde o *EsnPredictor* abre a possibilidade de subtrair variáveis da otimização, sendo nesse caso necessário definir valores fixos para as variáveis excluídas da otimização. Isso afeta diretamente a representação do modelo, modificando a partícula no caso de *Particle Swarm Optimization* ou o cromossomo no caso dos Algoritmos Genéticos.



Optimization Parameters

Fixed Values

☒ Spectral Radius

☒ Connection Percent

☒ Reservoir Units

☒ Feedback ☐ Feedback

☒ Window Fixed Value:

Window Optimization Type

☒ Window Size

☐ Internal Size:

Figura 38. Parâmetros da Otimização.

Cada variável tem associada uma representação dependendo do modelo híbrido, o qual é descrito ao longo do Capítulo 4.

### 5.7. Módulo de Execução do Modelo

Como ilustra a Figura 39 neste módulo pode-se visualizar um resumo das propriedades do modelo, assim como o resumo dos parâmetros da otimização que foram configurados no módulo anterior. Aqui também é feita a seleção do tipo de erro que será usado como avaliação final do cromossomo ou da partícula no processo de otimização.

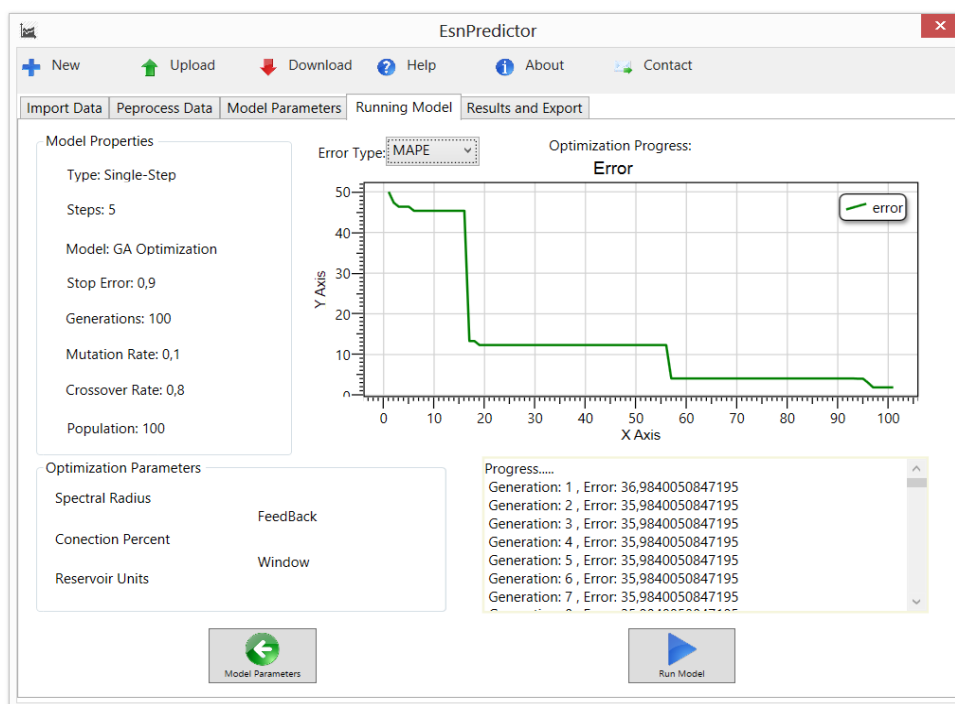


Figura 39. Aba Execução do Modelo.

No caso de ter escolhido uma ESN simples, ao executar o modelo essa rede vai ser criada, treinada e avaliada. No caso de ter sido escolhido algum tipo de otimização, o gráfico da Figura 39 exibirá em tempo real a avaliação do melhor cromossomo ou partícula respectivamente, além de atualizar o registro da geração atual e o valor do erro. O ESNPredictor realiza uma única otimização na execução do modelo, o usuário pode voltar para esse módulo e executar uma nova otimização.

## 5.8. Módulo de Resultados e Exportação

Após o ESNPredictor implementar e executar o modelo, se dirige automaticamente para a última aba de resultados e exportação (Figura 40). Nesta o usuário pode exportar os valores previstos para uma planilha do Excel, além de fazer uma calibração manual do modelo, modificando diretamente os valores dos quadros de texto do grupo *Final Model*.

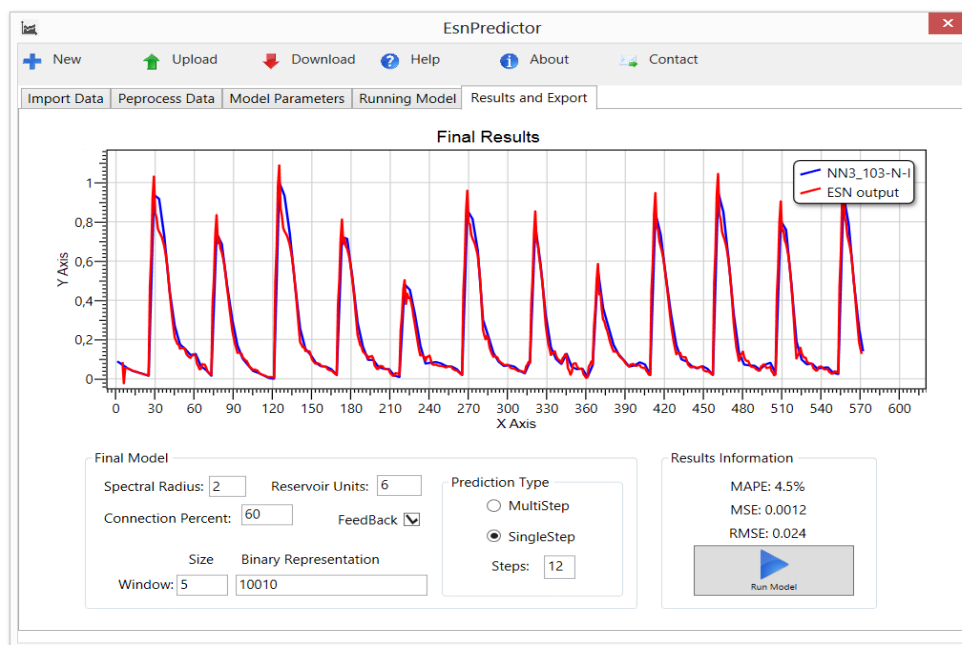


Figura 40. Aba Resultados e Exportação.

Neste módulo a ferramenta ESNPredictor apresenta as características do último modelo - por exemplo, o final da otimização ou a ESN simples que foi



executada - e abre a possibilidade de modificá-las, assim como o horizonte e tipo de previsão, no caso de demandar um ajuste fino dos parâmetros, ou testar configurações novas. Do lado direito são apresentados o MAPE o MSE e o RMSE de teste, além de ter um botão de execução do último modelo.

Após uma calibração manual e o usuário estar satisfeito com os resultados da previsão, este pode exportar as informações para um arquivo em Excel que segue o padrão da Figura 41, clicando no botão Download da Toolbar.

	A	B	C	D	E	F
1	ESNPREDICTOR EXPORT MODEL					
2	ESN	Spectral Radius	Connection %	Reservoir Size	Feedback	
3		0,6	60	50	false	
4						
5	GA	Population	Mutation Rate	Crossover Rate	Generations	
6						
7						
8	PSO	Particles	C1	C2	W	Generations
9		100	1	2	2	150
10						
11	Steps	Prediction Type	MAPE	MSE	RMSE	SMAPE
12	12	Multistep	4,5	0,012	0,02857	11
13						
14	Serie 1	Previsão				
15	7,636733237	7,218074585				
16	5,588205505	5,012404667				
17	1,838429445	1,564094761				

Figura 41. Arquivo Padrão de Exportação.

O arquivo contém informações das características do modelo que foi executado, assim como informações gerais da previsão e o cálculo de quatro métricas de erro básicas. Além disso, o arquivo contém os valores da série original e os valores da previsão.

## 5.9. Funcionalidades Básicas

O EsnPredictor foi criado com foco em uma interface gráfica amigável e na flexibilidade para editar a maior parte do modelo de previsão incorporado. E as funcionalidades básicas que contribuem para uma interface amigável são:

- *Importação da Série:* O EsnPredictor permite a importação de arquivos de Excel ou de bloco de notas sempre que estejam nos formatos da Figura 25.
- *Visualização das Séries:* A ferramenta exibe todas as séries usadas ou criadas como a série original, série com pré-processamento e saída da ESN.

- *Pré-Processamento da Série:* Todas as séries que são importadas ao EsnPredictor, podem ser normalizadas, interpoladas e no caso de ter tendência, esta pode ser eliminada.
- *Definição do Modelo e da Previsão:* Todas as características de uma ESN simples ou dos dois tipos de modelos híbridos podem ser configuradas e modificadas, além das características da previsão como o horizonte e o tipo.
- *Avaliação e Ajuste Manual dos Resultados:* Com ajuda de métricas de erro o EsnPredictor avalia o desempenho da previsão, e após a execução e criação do modelo, este ainda pode ser calibrado manualmente antes de exportar os dados.

A seção seguinte mostra o passo a passo para a configuração e criação de um modelo de previsão, detalhando o uso de cada um dos elementos que estão dentro dos módulos da ferramenta e destacando as funcionalidades básicas apresentadas.

### 5.10. Extras

O EsnPredictor tem uma série de botões que são descritos na Tabela 4.






Botão	Localização	Funcionalidade
 New	ToolBar	Abre um novo EsnPredictor, para criar ou executar um modelo completamente novo.
 Help	ToolBar	Abre uma janela com a ajuda da ferramenta.
 About	ToolBar	Abre uma janela com uma pequena descrição da ferramenta.
 Contact	ToolBar	Abre uma nova janela com os telefones e e-mails para contato dos criadores da ferramenta.
	Preprocess Data	Abre uma nova janela com informações acerca da normalização linear por partes.

Tabela 4 Botões Extras do EsnPredictor.

O EsnPredictor também tem um módulo interno de depuração de erros que exibe mensagens descritivas. As mais comuns são relacionadas na Tabela 5.

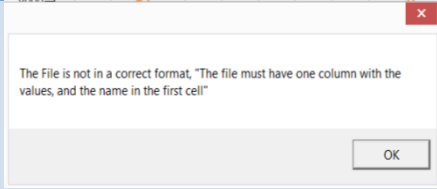
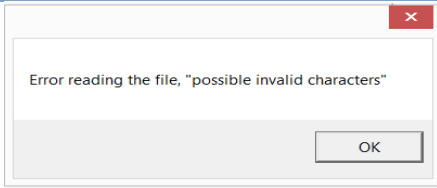
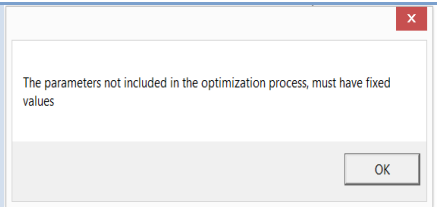
Mensagem	Descrição
	Ocorre quando o arquivo da importação não tem a formatação correta, conforme ilustrada na Figura 25.
	Ocorre quando o arquivo que está se importando, contém caracteres inválidos nos valores da série.
	Ocorre quando é subtraída uma variável da otimização e não se define um valor fixo para a mesma.

Tabela 5 Erros de Utilização do EsnPredictor.

## 6. Estudo de Casos

### 6.1. Introdução

Finalizado o desenvolvimento da ferramenta é indispensável testar todas as funcionalidades, assim como a eficácia dos modelos simples e híbridos que são criados pela ferramenta. Para isto foi selecionado um conjunto de séries temporais e foram realizadas previsões para essas séries utilizando o EsnPredictor.

### 6.2. Dados Utilizados

Foram selecionadas as 11 séries temporais do *Dataset* reduzido da *Forecasting Competition* de 2006/07 (NN3) organizado pela *Artificial Neural Network & Computational Intelligence Forecasting Competition* (NN3, 2013).

O objetivo principal da competição é avaliar o desenvolvimento de modelos de previsão de séries temporais baseados em técnicas de inteligência computacional. O evento foi patrocinado pela empresa SAS e pelo *International Institute of Forecasters* (IIF). Desde sua criação os *datasets* da NN3, têm sido amplamente utilizados para a avaliação e teste de modelos de previsão (Crone, 2007; Safaviieh, 2007; Vasconcelos, 2008; Lendasse, 2008).

Das 11 séries selecionadas, oito apresentam 144 amostras, duas apresentam 133 e uma única apresenta 141. Assim como na competição, foram calculadas as previsões para 18 passos à frente em cada uma das séries. A imagem superior das Figuras (42-53) apresentam graficamente as onze séries selecionadas.

### 6.3. Metodologia

Para cada série temporal foram feitos três conjuntos de testes diferentes (ESN, ESN-GA e ESN-PSO) e no final uma comparação dos melhores modelos de cada conjunto. Em vista do número elevado de testes que poderiam ser feitos

por causa das  $n$  possíveis combinações dos parâmetros configuráveis ( $n > 1000$ ) e da quantidade de séries (11) foram selecionados configurações específicas para analisar o comportamento das ESN. Assim é possível analisar a influência de seus parâmetros, baseando-se na experiência e experimentação em etapas anteriores do projeto e durante o desenvolvimento da ferramenta. As configurações para a criação de modelos baseados em ESN simples estão na Tabela 6.

#	Tamanho da Janela	Tamanho do Reservoir	Porcentagem de Conexão	Raio Espectral	FeedBack	Tipo de Previsão
1	1	50	60	0.6	Sim	SS
2	3	50	60	0.6	Não	MS
3	6	50	60	0.6	Sim	SS
4	12	50	60	0.6	Não	MS
5	18	50	60	0.6	Sim	SS
6	6	1	60	0.6	Não	MS
7	6	20	60	0.6	Sim	SS
8	6	50	60	0.6	Não	MS
9	6	100	60	0.6	Sim	SS
10	6	200	60	0.6	Não	MS
11	6	50	0	0.6	Não	MS
12	6	50	20	0.6	Sim	SS
13	6	50	40	0.6	Não	MS
14	6	50	60	0.6	Sim	SS
15	6	50	80	0.6	Não	MS
16	6	50	100	0.6	Não	MS
17	6	50	60	0.2	Não	MS
18	6	50	60	0.8	Não	SS
19	6	50	60	0.6	Sim	MS
20	6	50	60	0.6	Não	MS
21	6	50	60	0.6	Não	SS
22	6	50	60	0.6	Não	MS

Tabela 6 Configurações para ESN.

Na Tabela 6, observa-se a presença de subgrupos de configurações com cores de fundo diferentes, onde cada grupo varia os valores de um parâmetro dentro do seu domínio, com o objetivo de avaliar a sua influência no desempenho do modelo.

As configurações para a criação de modelos híbridos com otimização por Algoritmos Genéticos estão na Tabela 7, e as configurações dos modelos com otimização por *Particle Swarm Optimization* estão na Tabela 8.

#	Gerações	Taxa de Mutação	Taxa de Cross	População	Tipo de Previsão	Representação da Janela
1	100	0,2	0,7	100	SS	Binária
2	120	0,1	0,7	80	MS	Inteira
3	150	0,1	0,8	50	SS	Binária
4	100	0,2	0,9	80	MS	Inteira
5	120	0,3	0,7	100	SS	Binária
6	150	0,2	0,5	80	MS	Inteira
7	100	0,2	0,8	50	SS	Binária
8	120	0,1	0,7	80	MS	Inteira
9	150	0,1	0,5	100	SS	Binária
10	100	0,2	0,9	80	MS	Inteira
11	120	0,3	0,8	50	MS	Binária
12	100	0,2	0,7	20	SS	Inteira
13	120	0,1	0,6	100	MS	Binária
14	150	0,2	0,8	80	SS	Inteira
15	120	0,3	0,9	50	MS	Binária
16	150	0,2	0,5	20	MS	Inteira

Tabela 7 Configurações para ESN-GA.

#	Gerações	Partículas	C1	C2	W	Tipo de Previsão	Representação da Janela
1	100	100	1	1	1	SS	Binária
2	120	80	1	1	4	MS	Inteira
3	150	50	1	4	1	SS	Binária
4	100	80	4	1	1	MS	Inteira
5	120	100	2	2	1	SS	Binária
6	150	80	1	1	1	MS	Inteira
7	100	50	2	2	2	SS	Binária
8	120	80	1	2	3	MS	Inteira
9	150	100	1	3	2	SS	Binária
10	100	80	3	1	2	MS	Inteira
11	120	50	3	2	1	MS	Binária
12	100	20	2	3	1	SS	Inteira
13	120	100	3	2	3	MS	Binária
14	150	80	1	1	1	SS	Inteira
15	120	50	2	3	2	MS	Binária
16	150	20	1	1	1	MS	Inteira

Tabela 8 Configurações para ESN-PSO.

#### 6.4. Pré-Processamento dos Dados

Cada série temporal foi tratada de acordo com suas características, sendo que na maioria das séries foi aplicada a normalização padrão entre 0-1 com exceção de N3\_108 e N3\_110, nas que foi feita uma normalização linear por

partes. Além da normalização padrão, nas Séries N3\_101, N3\_105 e N3\_109 foi feita a eliminação da tendência.

A Tabela 9 relaciona o pré-processamento feito em cada uma das 11 séries, e as Figuras (42-52) mostram graficamente a série original (parte superior) e a série após do pré-processamento (parte inferior).

Série	Normalização	Eliminação Tendência
NN3_101	Padrão	Diferenciação
NN3_102	Padrão	-
NN3_103	Padrão	-
NN3_104	Padrão	-
NN3_105	Padrão	Diferenciação
NN3_106	Padrão	-
NN3_107	Padrão	-
NN3_108	Por Partes	-
NN3_109	Padrão	Diferenciação
NN3_110	Por Partes	-
NN3_111	Padrão	-

Tabela 9 Pré-processamento das Séries NN3.

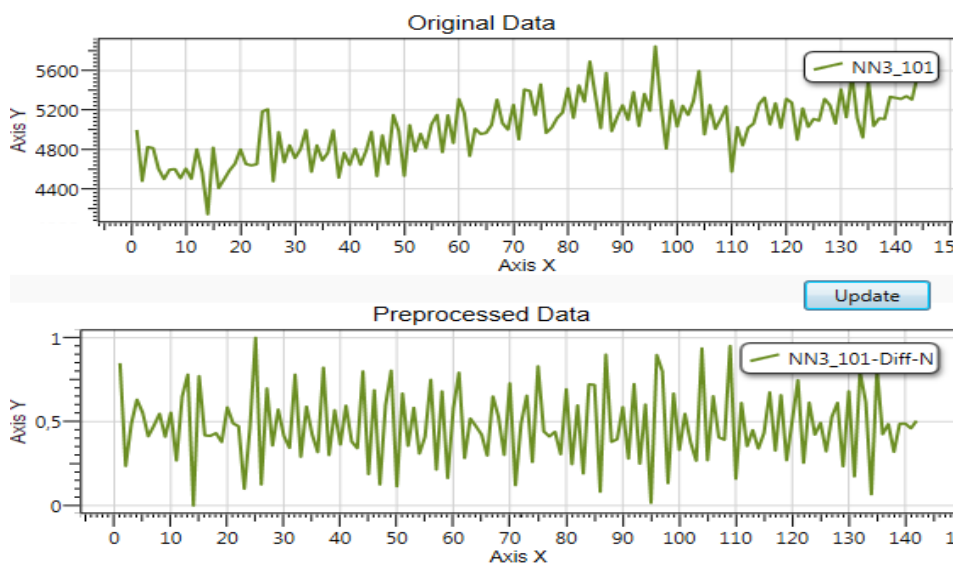


Figura 42. Pré-processamento da Série NN3\_101.

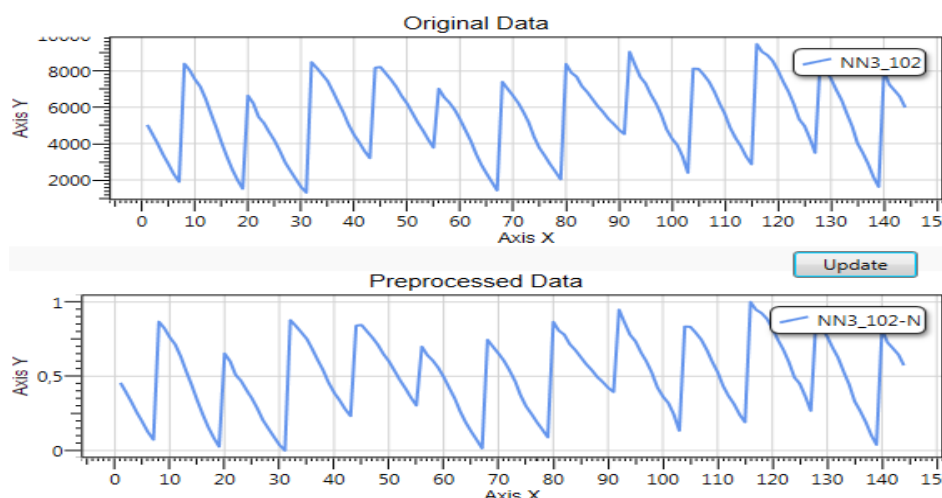


Figura 43. Pré-processamento da Série NN3\_102.

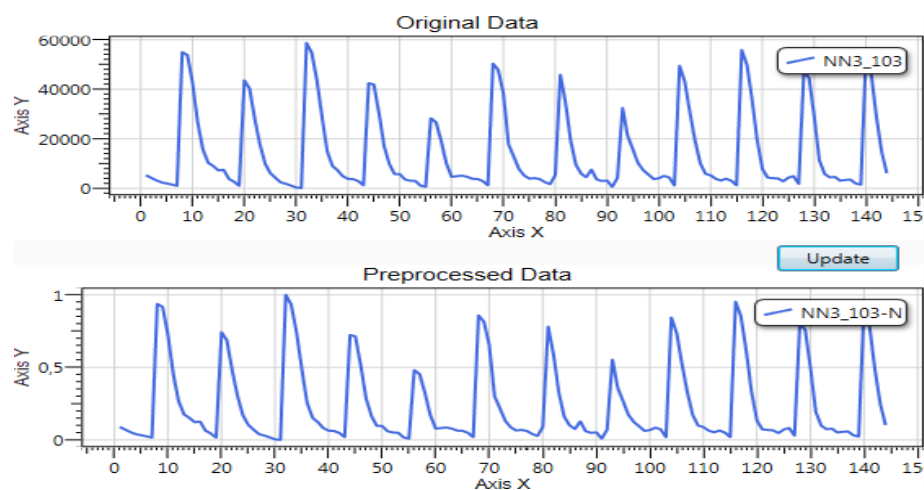


Figura 44. Pré-processamento da Série NN3\_103.

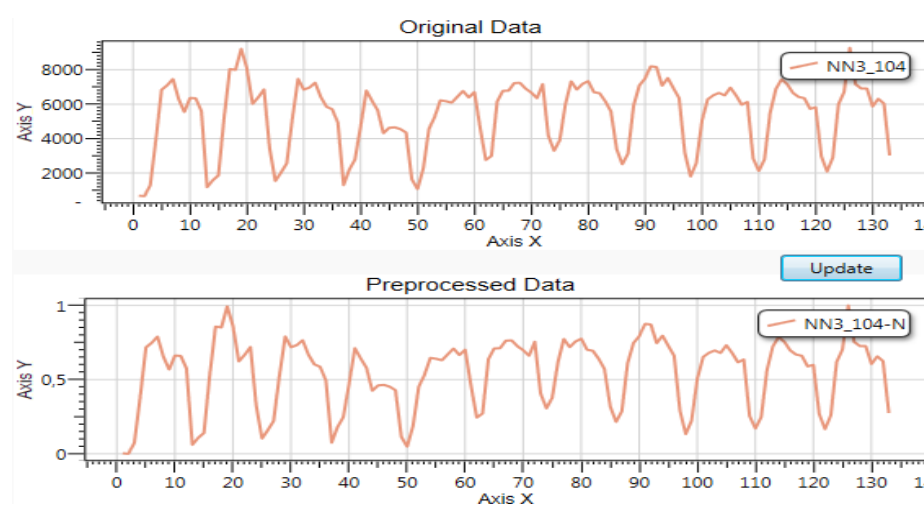


Figura 45. Pré-processamento da Série NN3\_104.



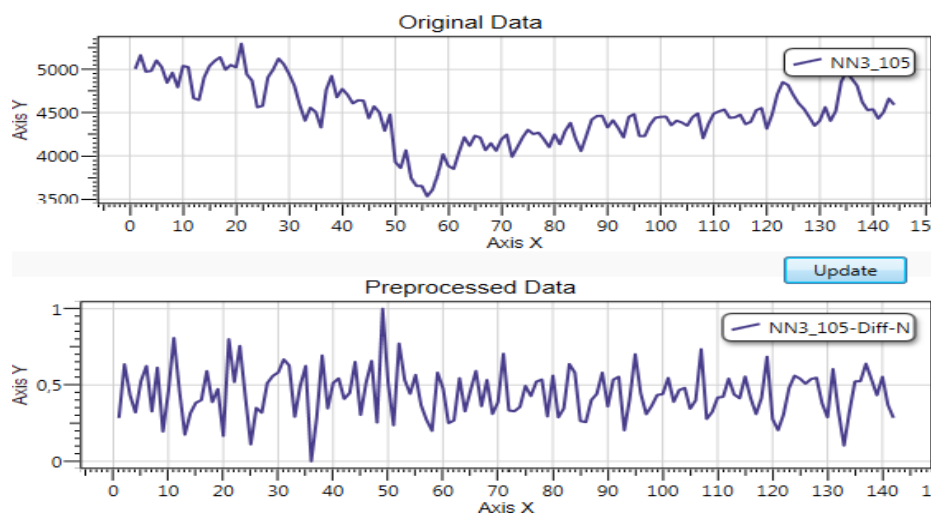


Figura 46. Pré-processamento da Série NN3\_105.

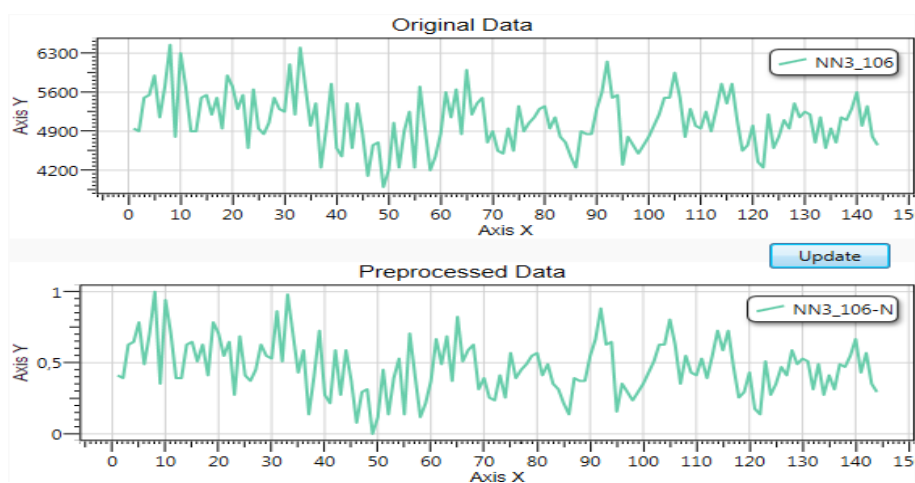


Figura 47. Pré-processamento da Série NN3\_106.

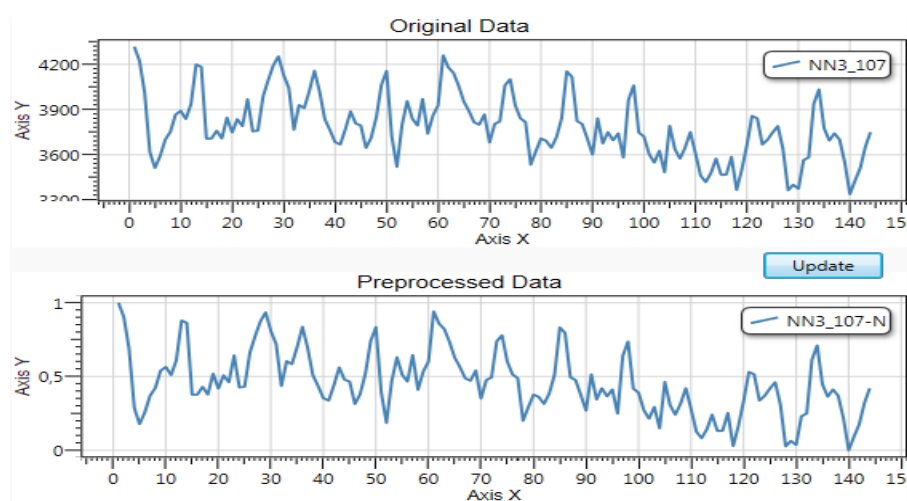


Figura 48. Pré-processamento da Série NN3\_107.

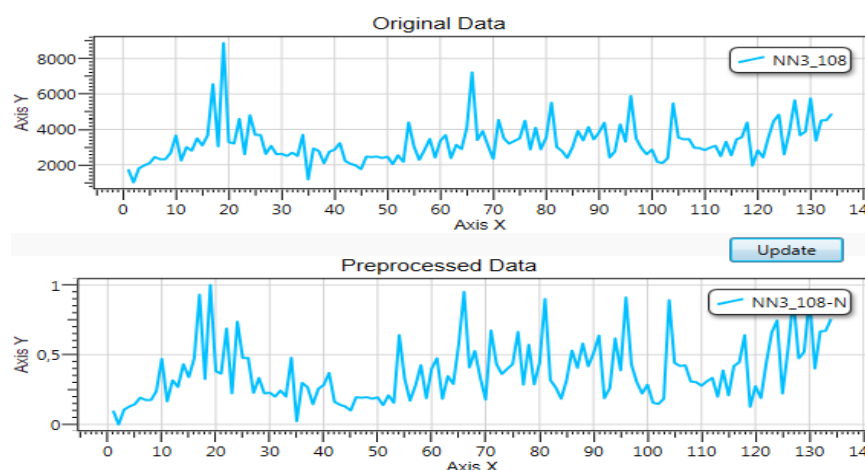


Figura 49. Pré-processamento da Série NN3\_108.

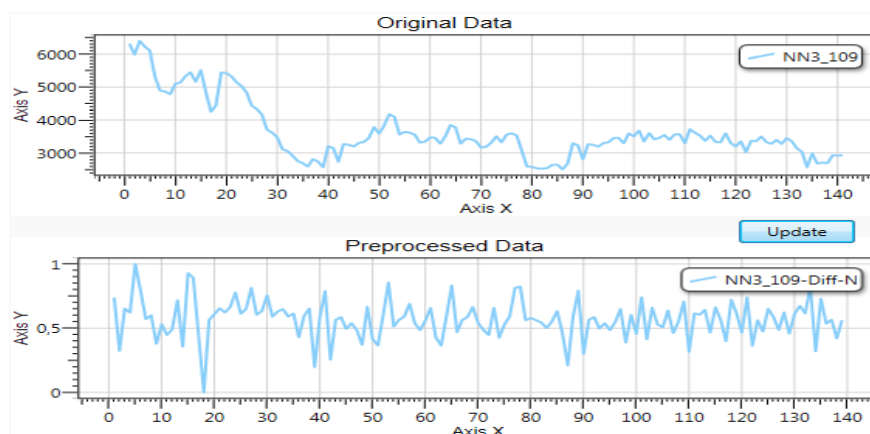


Figura 50. Pré-processamento da Série NN3\_109.

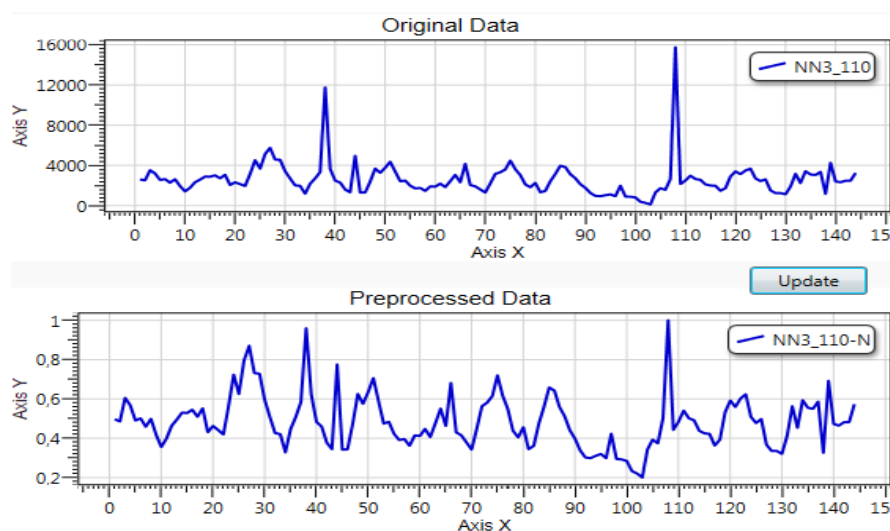


Figura 51. Pré-processamento da Série NN3\_110.

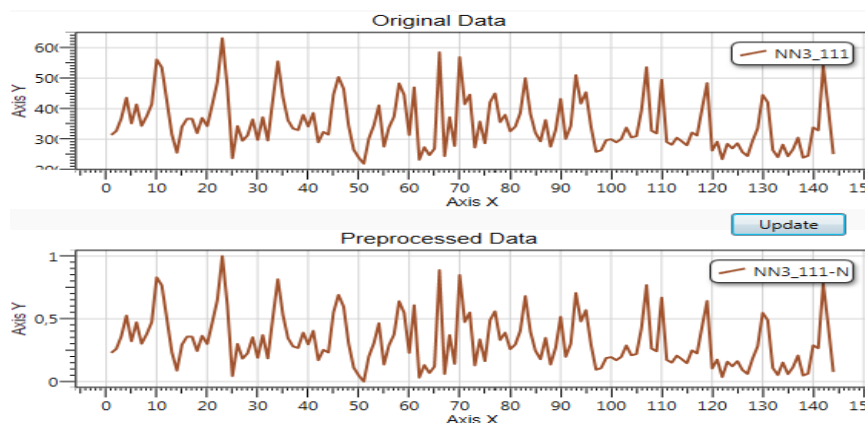


Figura 52. Pré-processamento da Série NN3\_111.

## 6.5. Resultados Obtidos

Todos os experimentos foram realizados com um horizonte de previsão de 18 passos, seguindo as regras da competição NN3, e após aplicar as 54 configurações contidas nas tabelas da seção 6.3, foram selecionadas as melhores configurações de cada grupo. Desse modo, é possível efetuar a comparação entre os três modelos: ESN, ESN-GA e ESN-PSO. As Tabelas (10-20) apresentam os melhores resultados para cada uma das configurações básicas (ESN, ESN-GA e ESN-PSO) em cada uma das 11 séries, seguido das Figuras (53-63), que apresentam os resultados no conjunto de teste para cada série.

NN3_101	# Configuração	MAPE
ESN	1	7,9
ESN-GA	2	4,3
ESN-PSO	7	3

Tabela 10. Resultados Série NN3\_101.

NN3_102	# Configuração	MAPE
ESN	3	9,2
ESN-GA	8	4,9
ESN-PSO	7	4,3

Tabela 11. Resultados Série NN3\_102.

NN3_103	# Configuração	MAPE
ESN	14	5,2
ESN-GA	10	1,3
ESN-PSO	2	2,4

Tabela 12. Resultados Série NN3\_103.

NN3_104	# Configuração	MAPE
ESN	14	7,4
ESN-GA	10	5,4
ESN-PSO	15	4,8

Tabela 13. Resultados Série NN3\_104.

NN3_105	# Configuração	MAPE
ESN	19	10,2
ESN-GA	15	6,4
ESN-PSO	5	4,1

Tabela 14. Resultados Série NN3\_105.

NN3_106	# Configuração	MAPE
ESN	3	7,1
ESN-GA	13	2,1
ESN-PSO	10	3,7

Tabela 15. Resultados Série NN3\_106.

NN3_107	# Configuração	MAPE
ESN	8	9,8
ESN-GA	3	6,2
ESN-PSO	13	5,3

Tabela 16. Resultados Série NN3\_107.

NN3_108	# Configuração	MAPE
ESN	8	11,2
ESN-GA	7	7,2
ESN-PSO	7	5

Tabela 17. Resultados Série NN3\_108.

NN3_109	# Configuração	MAPE
ESN	1	14,2
ESN-GA	7	5,6
ESN-PSO	1	4,2

Tabela 18. Resultados Série NN3\_109.

NN3_110	# Configuração	MAPE
ESN	14	10,5
ESN-GA	5	2,2
ESN-PSO	7	4,2

Tabela 19. Resultados Série NN3\_110.

NN3_111	# Configuração	MAPE
ESN	1	9,4

<b>ESN-GA</b>	<b>8</b>	<b>4,5</b>
<b>ESN-PSO</b>	<b>1</b>	<b>4,2</b>

Tabela 20. Resultados Série NN3\_111.

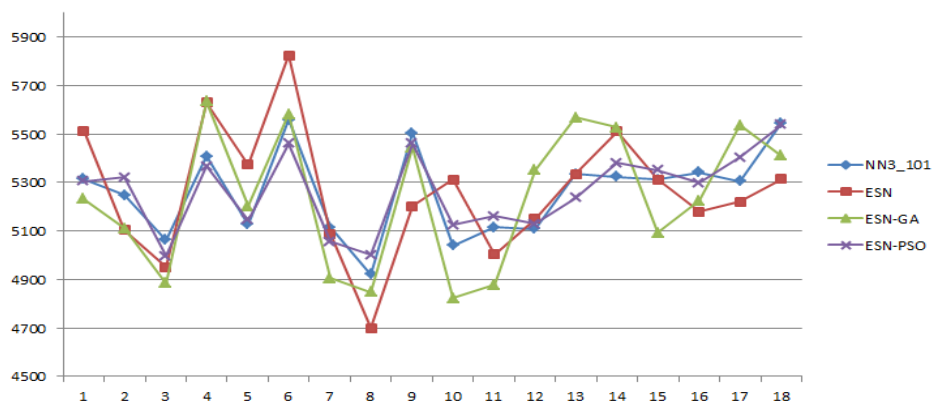


Figura 53. Resultados da Série NN3\_101.

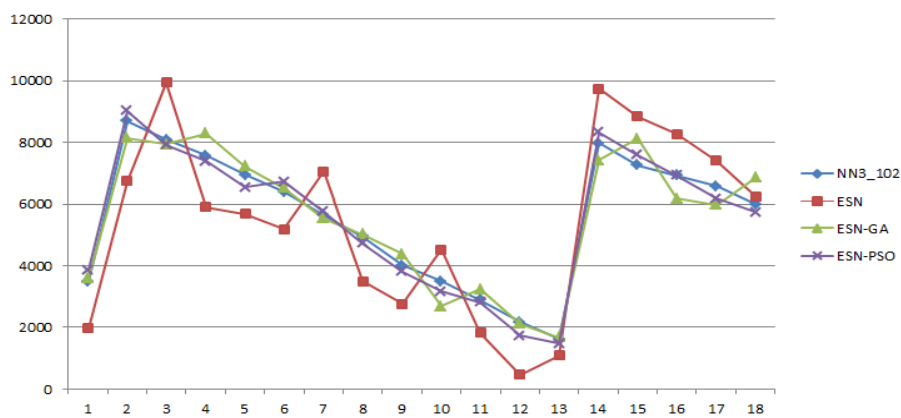


Figura 54. Resultados da Série NN3\_102.

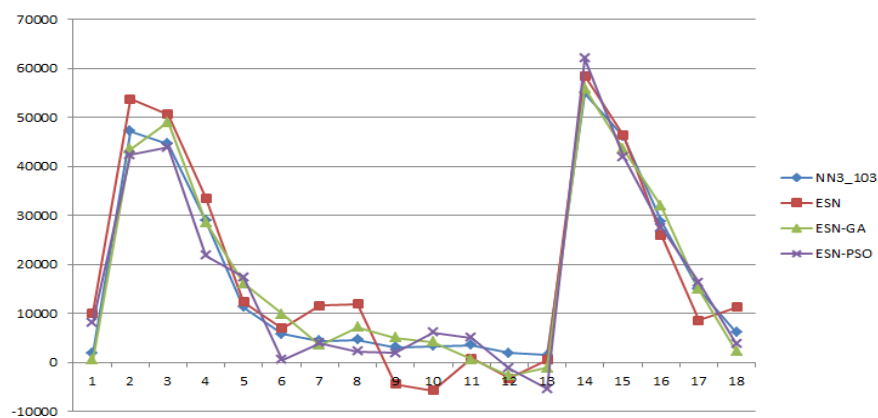


Figura 55. Resultados da Série NN3\_103.

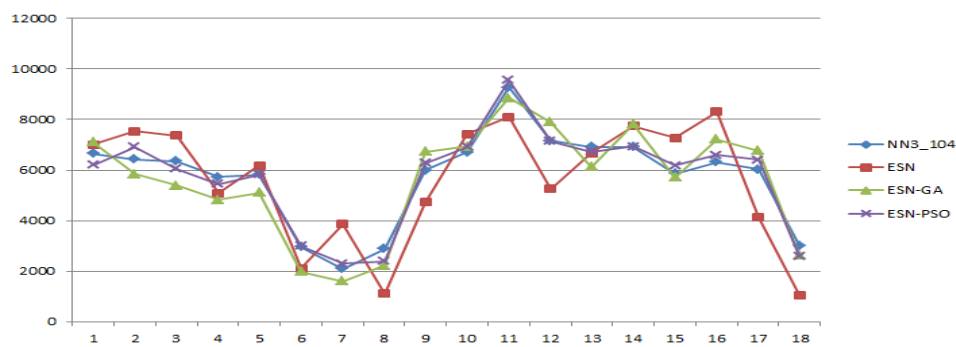


Figura 56. Resultados da Série NN3\_104.

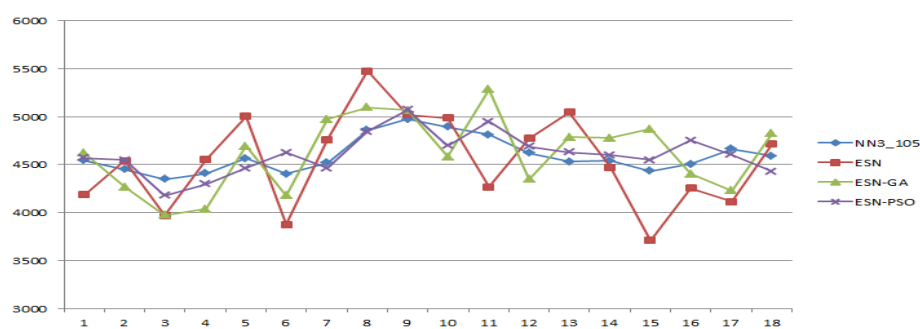


Figura 57. Resultados da Série NN3\_105.

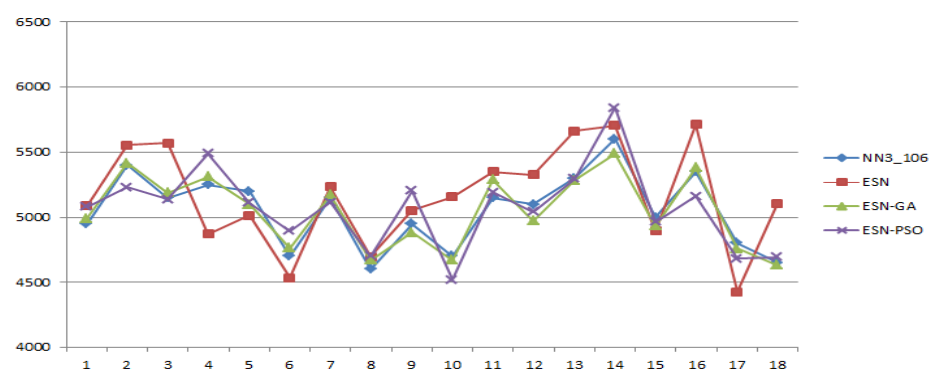


Figura 58. Resultados da Série NN3\_106.

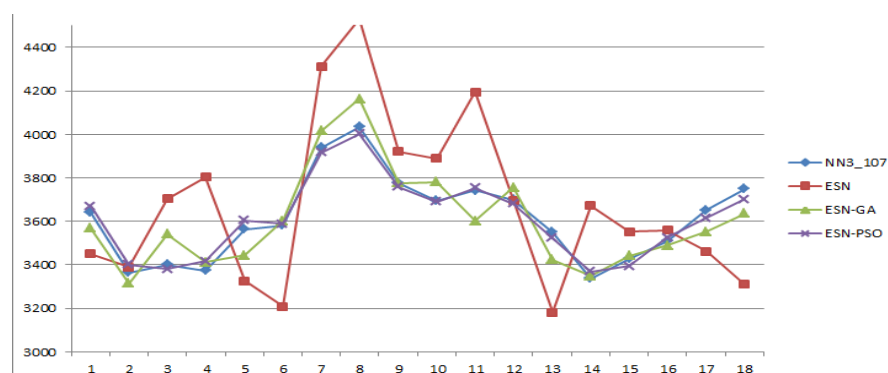


Figura 59. Resultados da Série NN3\_107.

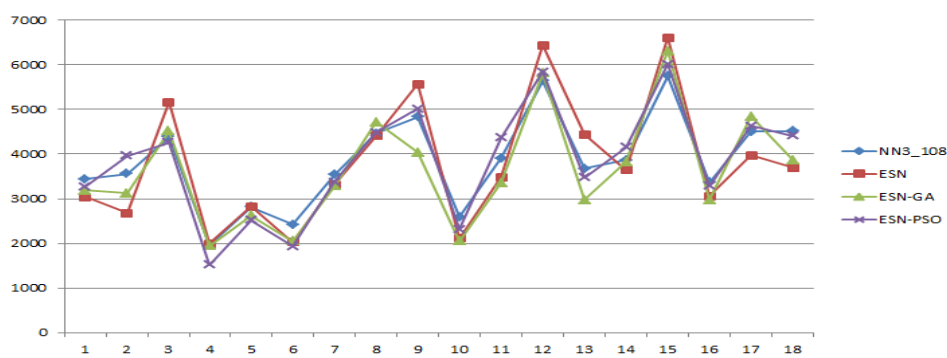


Figura 60. Resultados da Série NN3\_108.

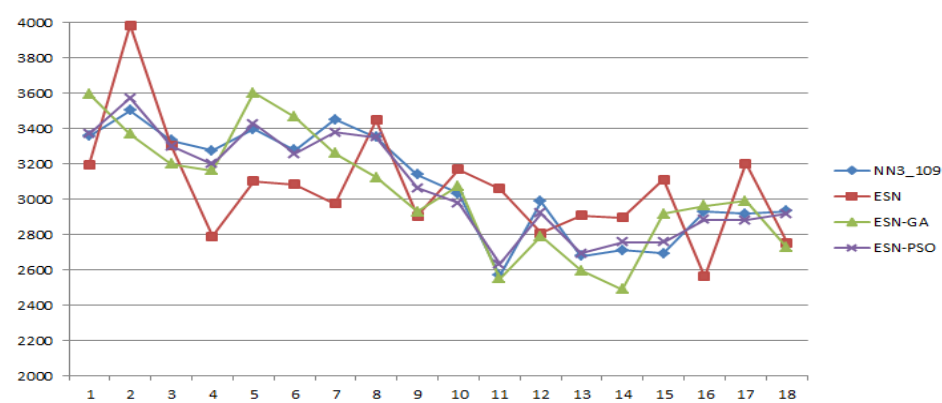


Figura 61. Resultados da Série NN3\_109.

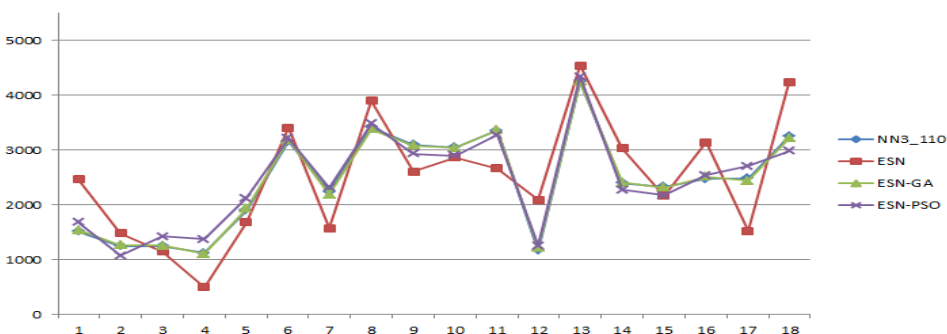


Figura 62. Resultados da Série NN3\_110.

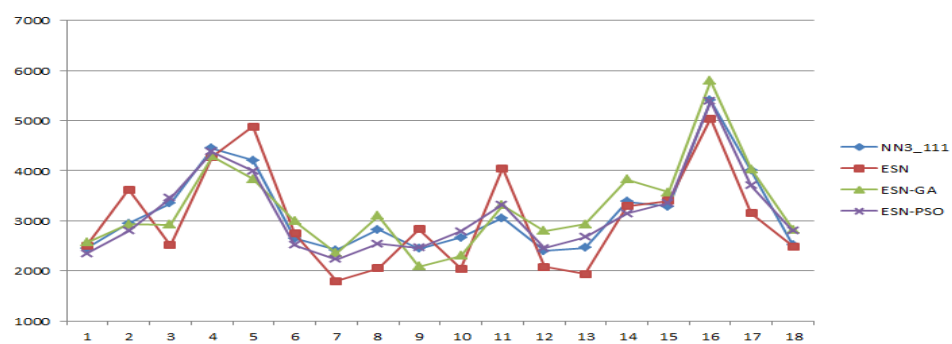


Figura 63. Resultados da Série NN3\_111.

## 6.6. Resumo dos Resultados

A Tabela 21 resume os resultados obtidos com a aplicação da ferramenta nas séries selecionadas.

Série	ESN	ESN-GA	ESN-PSO
NN3_101	7,9	4,3	3
NN3_102	9,2	4,9	4,3
NN3_103	5,2	1,3	2,4
NN3_104	7,4	5,4	4,8
NN3_105	10,2	6,4	4,1
NN3_106	7,1	2,1	3,7
NN3_107	9,8	6,2	5,3
NN3_108	11,2	7,2	5
NN3_109	14,2	5,6	4,2
NN3_110	10,5	2,2	4,2
NN3_111	9,4	4,5	4,2
Média	9,28	4,55	4,1

Tabela 21. Resumo dos Resultados.

Observa-se claramente que as médias dos MAPEs dos modelos híbridos ESN-GA e ESN-PSO proporcionam uma melhora de aproximadamente 50% quando comparado com o melhor modelo de ESN simples, comprovando o potencial que tem a otimização de parâmetros para ajustar os valores dos parâmetros das Echo State Network.

Embora os resultados dos modelos híbridos ESN-GA e ESN-PSO são similares, como é ilustrado na Figura 64, o ESN-PSO é o vencedor, apresentando o melhor desempenho para 8 séries enquanto que o modelo ESN-PSO vence em 3 séries.



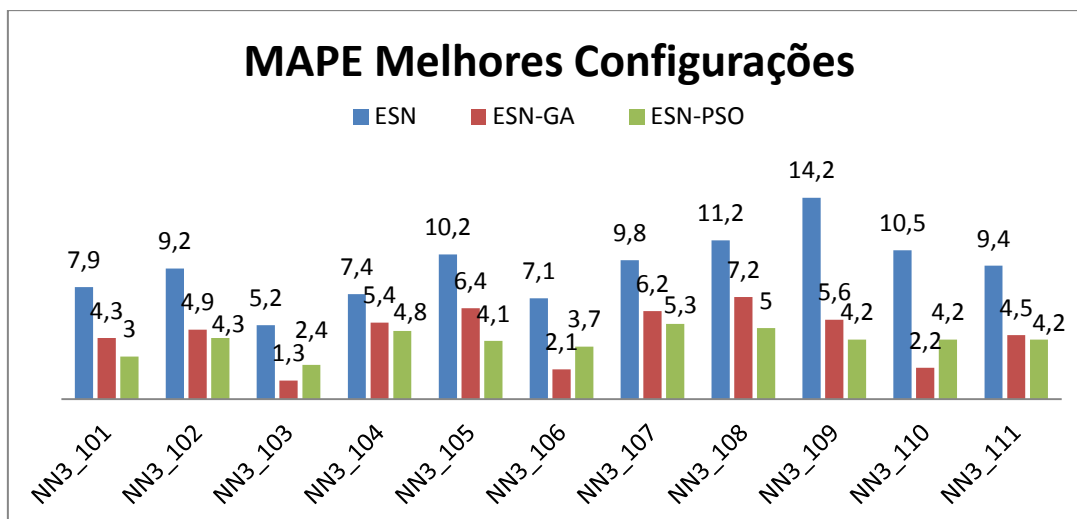


Figura 64. Resumo dos Resultados.

A Tabela 22 mostra a média dos tempos de convergência em minutos das 16 configurações aplicadas nas 11 séries de cada tipo de modelo híbrido ESN-GA e ESN-PSO, demonstrando que para as séries da NN3, o algoritmo PSO converge aproximadamente 50% mais rápido do que GA.

Série	ESN-GA	ESN-PSO
NN3_101	20	10
NN3_102	22	12
NN3_103	18	11
NN3_104	21	10
NN3_105	18	9
NN3_106	17	8
NN3_107	16	11
NN3_108	24	12
NN3_109	22	10
NN3_110	20	12
NN3_111	20	13
Media	19,81	10,72

Tabela 22. Média dos Tempos de Convergência (Minutos).

Além de todos os testes realizados nas seções anteriores, foi feito mais um grupo de experimentos, selecionando cada uma das melhores configurações para cada tipo de modelo, e adicionando, na etapa de pré-processamento, uma interpolação de  $k$  valores novos entre duas amostras, onde  $k = 2$ , aumentando assim o comprimento das séries. A Tabela 23 indica que, para todos os casos, o desempenho foi melhor com o uso da interpolação.

	ESN	ESN-INT	ESN-GA	ESN-GA-INT	ESN-PSO	ESN-PSO-INT
<b>NN3_101</b>	7,9	<b>6,65</b>	4,3	<b>3,62</b>	3	<b>2,08</b>
<b>NN3_102</b>	9,2	<b>7,51</b>	4,9	<b>3,4</b>	4,3	<b>2,80</b>
<b>NN3_103</b>	5,2	<b>4,48</b>	1,3	<b>1,06</b>	2,4	<b>1,7</b>
<b>NN3_104</b>	7,4	<b>6,12</b>	5,4	<b>4,96</b>	4,8	<b>4,3</b>
<b>NN3_105</b>	10,2	<b>10,14</b>	6,4	<b>6,27</b>	4,1	<b>2,56</b>
<b>NN3_106</b>	7,1	<b>5,78</b>	2,1	<b>0,96</b>	3,7	<b>3,09</b>
<b>NN3_107</b>	9,8	<b>8,07</b>	6,2	<b>5,01</b>	5,3	<b>4,16</b>
<b>NN3_108</b>	11,2	<b>9,33</b>	7,2	<b>6,49</b>	5	<b>3,20</b>
<b>NN3_109</b>	14,2	<b>13,5</b>	5,6	<b>3,97</b>	4,2	<b>3,86</b>
<b>NN3_110</b>	10,5	<b>10,1</b>	2,2	<b>2,19</b>	4,2	<b>4,02</b>
<b>NN3_111</b>	9,4	<b>8,5</b>	4,5	<b>2,77</b>	4,2	<b>2,79</b>
<b>Media</b>	9,28	<b>8,22</b>	4,5	<b>3,712</b>	4,1	<b>3,14</b>

Tabela 23. MAPE Resultados com Interpolação (INT).

Verifica-se que o uso de modelos híbridos (ESN-GA, ESN-PSO) fornece um desempenho melhor na previsão de séries temporais, comparado com uma Echo State sem ajuste automático de seus parâmetros. Além disso, no caso das 11 séries da NN3, a interpolação dos dados proporciona uma melhora ainda maior nos resultados obtidos pelos modelos híbridos, devido ao tamanho das séries serem pequenas (entre 130 e 150 dados cada uma).

A ferramenta ESNPredictor proporciona uma relação amigável com o usuário, facilitando seu uso e permitindo a geração, modificação e otimização de modelos de previsão de séries temporais. Destaca-se a ampla gama de possibilidades no pré-processamento dos dados, assim como a facilidade na importação e exportação de informação.

## 7. Conclusões e Trabalhos Futuros

### 7.1. Conclusões

Esta dissertação teve como objetivo principal o desenvolvimento de uma ferramenta computacional amigável e editável capaz de realizar previsões de séries temporais através de redes neurais artificiais do tipo *Echo State Network*. A ferramenta comporta a otimização de parâmetros da *Echo State Network* por *Particle Swarm Optimization* e Algoritmos Genéticos, oferecendo uma interface gráfica simples e intuitiva de forma a facilitar o uso por usuários que não sejam especialistas na matéria.

Para tal, foi concebido o EsnPredictor, ferramenta do tipo WPF compatível com Sistemas Operacionais Windows. Esta é completamente baseada em interface gráfica, onde a partir da janela principal do programa podem-se criar modelos de previsão, alterar facilmente a visualização e tratamento dos dados, além de possibilitar importar e exportar informações no formato Excel e arquivos de texto.

A ferramenta foi utilizada para a previsão das 11 séries temporais do *Dataset* reduzido da NN3, e considerando o SMAPE médio obtido nas séries – a métrica utilizada na NN3 – os resultados colocam o EsnPredictor na 3ª posição da competição (SMAPE médio do conjunto de teste = 14,38%), embaixo do Prof. Dr. Herbert Jaeger, criador das *Echo State Networks*, quem obteve melhores resultados por causa da decomposição das series temporais usando o software X-12-ARIMA.

Como se esperava, observou-se uma melhora de 50% no desempenho das *Echo State Networks* quando os parâmetros do modelo, como Raio Espectral, Porcentagem de Conexão, entre outros, são determinados por um algoritmo de otimização, que no caso são os Algoritmos Genéticos ou *Particle Swarm Optimization*.

Embora os resultados obtidos dos algoritmos de otimização sejam similares, o custo computacional do PSO é consideravelmente menor do que GA, entretanto a escolha correta dos parâmetros  $C1$ ,  $C2$  e  $W$  tornam-se vitais no desempenho deste algoritmo de otimização.

Confirmando o que está na literatura, as *Echo State Networks* são uma ferramenta robusta e eficiente para tarefas de previsão de séries temporais, devido à memória do *reservoir* e o dinamismo gerado pelas conexões aleatórias.

O Raio Espectral e o tamanho do Reservoir são os dois parâmetros mais importantes na configuração da ESN. Um Raio espectral muito alto causa uma saturação nas ativações dos neurônios do Reservoir, e a rede fornece valores errados. Com um tamanho de *Reservoir* muito pequeno, a ESN não consegue aprender os valores que são apresentados, e com um *Reservoir* muito grande apresenta grandes componentes de ruído.

A porcentagem de conexão é um parâmetro de menor importância em comparação com os dois anteriores, mas para valores muito baixos (<10%), o desempenho piora devido a que o estado anterior tem pouca influência no estado seguinte.

Para qualquer configuração dos parâmetros dos algoritmos de otimização (GA ou PSO) obtém-se resultados aceitáveis com relação ao desempenho das previsões; já os parâmetros próprios das ESN, como o tamanho do *Reservoir* e o Raio Espectral, afetam de forma considerável o desempenho da previsão, sendo esses parâmetros das ESN mais críticos do que os parâmetros do algoritmo de otimização.

## 7.2. Trabalhos Futuros

As seguintes oportunidades de trabalhos futuros podem ser destacadas a partir deste trabalho:

- Criação de um módulo de laboratório, possibilitando a execução de vários sistemas ao mesmo tempo e comparar de forma mais eficiente os resultados obtidos pelas configurações.

- Possibilitar a configuração de parâmetros variáveis no tempo para os algoritmos de otimização. No caso dos algoritmos genéticos uma taxa de mutação alta e de *crossover* baixa no início ajudam com uma melhor exploração do espaço de busca, e uma taxa de mutação baixa e *crossover* alta no final, podem auxiliar na fase do ajuste fino. De igual forma no PSO um valor de  $C2$  baixo permite uma melhor exploração das partículas sem convergir rapidamente em um possível mínimo local, e um valor inicial alto para  $W$  permite uma maior exploração do espaço de busca.
- Modificar o sistema para possibilitar a inserção de mais variáveis de entrada além da janela, como outra série temporal, ou informações úteis como a data ou sazonalidade. Como consequência desta modificação, o módulo de pré-processamento também deve ser modificado para inserir um novo tratamento e codificação.
- Programar um operador de *crossover* uniforme com uma máscara fixa, para melhorar a convergência, fazendo cruzamentos de forma dividida em cada uma das seções do cromossomo.

## Referências Bibliográficas

AMJADY, N., **Day-ahead price forecasting of electricity markets by a new fuzzy neural network**. IEEE Transactions on Power Systems, P. 887- 896. Maio 2006.

ANDERSON, M. L., **Coupling HEC-HMS with atmospheric models for prediction of watershed runoff**. Journal of Hydrologic, P. 312-318, 2002.

ANDREOLLI, I., **Previsão de Vazão em Tempo Real no Rio Uruguai com Base na Previsão Meteorológica**. Dissertação de Mestrado, UFRGS. 2003.

BACZY, D., **Influence of artificial neural network structure on quality of short-term electric energy consumption forecast**, IEEE Proc.-Generation, Transmission and Distribution, Vol. 151, No. 2, p. 241-245, March 2004.

BAJESTANI, N. S.; ZARE, A., **Application of optimized Type 2 fuzzy time series to forecast Taiwan stock index**. 2nd International Conference on Computer, Control and Communication. p. 1-6. 2009.

BARBOZA, V., **FuzzyFuture: Ferramenta de Previsão de Séries Temporais Baseada em Sistema Híbrido Fuzzy-Genético**. Dissertação de Mestrado Pontifícia Universidade Católica do Rio de Janeiro - IBICT, Rio de Janeiro, Brasil 2011.

BIRINCI, I. Y.; LEBLEBICIOGLU, K., **Optimum Placement of Microphone Array for Sound Capture Using Genetic Algorithms**. IEEE 14th Signal Processing and Communications Applications, 2006.

BOX, G. E. P.; JENKINS, G. M., **Time Series Analysis: Forecasting and Control**. Oakland: Holden-Day, 1976.

BREVE, F., **Echo State Network Sistemas Inteligentes**. Universidade de São Paulo, São Carlos, Instituto de Ciências Matemáticas e de Computação, 2005

CAMPBELL, P. R. J.; ADAMSON, K., **Methodologies for Load Forecasting**. 3rd International IEEE Conference Intelligent Systems. 2006.

CHAI, M.-L.; SONG, S.; LI, N.-N., **A review of some main improved models for Neural Network forecasting on time series**. IEEE Intelligent Vehicles Symposium, 2005. Proceedings, P. 866-868, 2005.

CHÁVEZ J. L., **Forecasting Mexican Inflation using Neural Networks**. 2013 International Conference on electronics, Communications and Computing (CONIELECOMP), Mexico, p. 32-35, March, 2013, ISBN: 978-1-4673-6156-9/13.

CHEN, X.; RACINE, J.; SWANSON, N. R., **Semiparametric ARX neural network models with an application to forecasting inflation.** IEEE Transactions on Neural Networks, Vol. 12, Nº 4, JULY 2001.

CHEN, S.-M.; WANG, N.-Y., **Fuzzy Forecasting Based on Fuzzy-Trend Logical Relationship Groups.** IEEE Transactions on Systems, Man, and Cybernetics, P. 1343-1358, 2010.

CHEN, S.-M.; CHEN, C.-D. **Forecasting Based on Fuzzy Time Series and Fuzzy Variation Groups.** IEEE Transactions on Fuzzy Systems, P. 1-12, 2011.

CRONE F., **A Naive Support Vector Regression**, Benchmark for the NN3 Forecasting Competition 2007.

DAVIS, L. **Handbook of Genetic Algorithms.** VNR Comp. Library, 1990.

DE SOUZA, F., **Modelos Neuro-Fuzzy Hierárquicos**, Tese de Doutorado, DEE-PUC/RJ, 1999.

DYNAMICDATADISPLAY. **Dynamic Data display.** Disponível em: < <http://research.microsoft.com/en-us/um/cambridge/groups/science/tools/d3/> >. Acesso em 17/10/2013.

DOMINEY, P., **Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning.** Biological Cybernetics, 73:265–274, 1995.

DONGDONG, W., **The Consumer Price Index Forecast Based on ARIMA Model.** International Conference on Information Engineering. p. 307. 2010.

DONGXIAO, N., **Middle – long Electric Power Load Forecasting Based on Co-integration and Support Vector Machine**, ICNC'07 Proceedings of the Third International Conference on Natural Computation, Volume 01, p. 596-600, 2007,

DONGXIAO, N., **Research on short-term power load time series forecasting model based on BP neural network.** 2nd International Conference on Advanced Computer Control. p. 509-512. 2010.

DOTNETMATRX. **Simple Matrix Library for .NET.** Disponível em: < <http://www.codeproject.com/Articles/5835/DotNetMatrix-Simple-Matrix-Library-for-NET> >. Acesso em 13/09/2013.

DU, LUO, XIE., **The Application of Genetic Algorithm-Radial Basis Function (GA-RBF) Neural Network in Stock Forecasting**, Zhejiang University City College Key Laboratory of Intelligent Systems, 2010.

FANG, Z., LUO, G., FEI, F., LI, S., **Stock Forecast Method based on Wavelet Modulus Maxima and Kalman Filter.** 2010 International Conference on Management of e-Commerce and e-Government.

FIGUEIREDO, K., **Modelo de Previsão de Vazão com informação de Precipitação Utilizando Redes Neurais**. Revista Brasileira de Recursos Hídricos, Julho/Setembro 2007. 69-82.

FU-YUAN, H., **Forecasting Stock Price Using a Genetic Fuzzy Neural Network**. International Conference on Computer Science and Information Technology. 2008. p. 549-552.

GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley, 1999.

GOLDBERG, D., **The theory of virtual alphabets, Lecture Notes in Computer Science**, Volume 496, p.13-22, 1991.

HADAVANDI, E.; SHAVANDI, H.; GHANBARI, A., **A genetic fuzzy expert system for stock price forecasting**. Seventh International Conference on Fuzzy Systems and Knowledge Discovery. p. 41-44, 2010.

HANKE, J. E.; REITSCH, A. G., **Business Forecasting**. 5a. Edição. ed. Englewood Cliffs: Prentice Hall, 1995.

HAYKIN, S., **Redes Neurais - Principios e Praticas**. Editora Bookman, 2000.

HVASSLAB. **Hvass Laboratories**. Disponível em: < <http://www.hvass-labs.org/>>. Acesso em 23/01/2014.

HERTZ, J., **Introduction to the theory of neural computation**, Addison-Wesley, 1991.

JAEGER, H., **The “echo state” Approach to Analysing and Training Recurrent Neural Networks – with an Erratum note**. Fraunhofer Institute for Autonomous Intelligent Systems January 26, 2001.

JANIKOIW, C., **An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms**, Proceedings of the Fourth International Conference on Genetic Algorithms: 31–36, 1991.

JOHNNEWCOMBE. **The Genetic Algorithm Framework for .NET**. Disponível em: < <http://johnnewcombe.net/gaf/>>. Acesso em 21/02/2014.

KENNETH L., **A Method for the Solution of Certain Non-Linear Problems in Least Squares**, Quarterly of Applied Mathematics 2: 164–168, 1944.

KENNEDY and RUSSEL., **Particle Swarm Optimization**, in: Proceedings of the IEEE international conference on neural networks (Australia), 1995, 1942-1948, ISBN: 0-7803-2768-3/95.

KHOA, N. L. D.; SAKAKIBARA, K.; NISHIKAWA, I., **Stock Price Forecasting using Back Propagation Neural Networks with Time and Profit**



**Based Adjusted Weight Factors.** International Joint Conference SICE-ICASE, p. 5484-5488, 2006.

KOOTHIS, S.; MITZE, T.; RINGHUT, E., **Inflation forecasting – a comparison between econometric methods and a computational approach based on genetic-neural fuzzy rule-bases.** IEEE International Conference on Computational Intelligence for Financial Engineering. p. 183-190, 2003.

LAZO, J. G., **Determinação do Valor de Opções Reais por Simulação Monte Carlo com Aproximação por Números Fuzzy e Algoritmos Genéticos.** Tese [Doutorado em Engenharia Elétrica] - Pontifícia Universidade Católica do Rio de Janeiro, RJ, 2004.

LEI, M., **Modeling Electricity Price Forecast with Grey and Correlation Method in Competitive Markets.** Asia-Pacific Power and Energy Engineering Conference. 2010.

LENDASSE, A., **Variable Scaling for Time Series rediction: Application to the ESTSP'07 and the NN3 Forecasting Competitions,** International Joint Conference on Neural Networks IJCNN, p. 2812-2816, 2007.

LI, Y.-J.; SUN, X.-Q.; ZHANG, L.-M., **Logistics forecasting technology by RBF neural network trained by genetic algorithm.** International Conference on Computer Engineering and Technology (ICCET). p. V7-513-V7-516, 2010.

LI, Q.; CHEN, S.; WANG, D., **An Intelligent Runoff Forecasting Method Based on Fuzzy sets, Neural network and Genetic Algorithm.** International Conference on Intelligent Systems Design and Applications. p. 948-953, 2006.

LUCIANO D. S. PACIFICO and TERESA B. LUDERMIR., **Evolutionary Extreme Learning Machine Based on Particle Swarm Optimization and Clustering Strategies** Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9, 2013.

MICHALEWICZ, Z. **Algorithms+Data Structures=Evolution Programs.** Springer-Verlag, 1994.

MENTZER, J. T.; BIENSTOCK, C. C., **Sales Forecasting Management.** Thousand Oaks: Sage Publication, 1998.

NADIMI, V., **An Adaptive-Network-Based Fuzzy Inference System for Long-Term Electric Consumption Forecasting (2008-2015): A Case Study of the Group of Seven (G7) Industrialized Nations: U.S.A, Canada, Germany, United Kingdom, Japan, France and Italy,** 2010 Fourth UKSim European Symposium on computer Modeling and Simulation (EMS), p. 301-305, 2010.

NATSCHLÄGER, T., **The liquid computer: a novel strategy for real-time computing on time-series,** Special Issue on Foundations of Information Processing of TELEMATIK, vol. 8, num. 1, p. 39-43, 2002.

NN3. **NN3 Forecast Competition Home**. Disponível em: < <http://www.neural-forecasting-competition.com/NN3/index.htm>>. Acesso em 12/11/2013.

NORTHERN, J.; RIBEIRO, M., **A Genetic Algorithm Model Based Integrated Simulation Framework for Design of Embedded Systems**. IEEE Region 5 Technical Conference. p. 223, 2007.

QINGLE, P.; MIN, Z., **Very Short-Term Load Forecasting Based on Neural Network and Rough Set**. International Conference on Intelligent Computation Technology and Automation (ICICTA). p.1132-1135, 2010.

SAFAVIEH, E., **Forecasting the Unknown Dynamics in NN3 Database Using a Nonlinear Autoregressive Recurrent Neural Network**, International Joint Conference on Neural Networks IJCNN, p. 2105-2109, 2007.

STEIL, J., **Backpropagation-decorrelation: recurrent learning with O(N) complexity**. In Proceedings of the IEEE International Joint Conference on Neural Networks, 2004 (IJCNN 2004), volume 2, pages 843–848, 2004.

VASCONCELOS, A., **Algoritmos Genéticos e Particle Swarm Optimization e suas aplicações problemas de Guerra Eletrônica**, 2007.

VASCONCELOS, G. C., **A Systematic Solution for the NN3 Forecasting Competition Problem Based on an Ensemble of MLP Neural Networks** NeuroTech Ltd. and Center for Informatics, Federal University of Pernambuco, Recife, Brazil, 2008.

VENAYAGAMOORTHY, G. K., **Online design of an echo state network based wide area monitor for a multimachine power system**, The official journal of the international Neural Network Society, April 20, 2007.

WANG, C.; WU, D., **Modeling China's Inflation: Linear versus Nonlinear Method**. International Conference on Computational Intelligence and Software Engineering. 2010.

WANKE, P.; JULIANELI, L.; BRITO, V. **Previsão de Vendas**. Rio de Janeiro Atlas, 2006.

WERBOS, P. J., **Back propagation through time: What it does and how to do it**. Proc. IEEE, vol. 78 No. 10, pp. 1550–1560, 1990.

WILLIAN & ZIPSER., **A learning algorithm for continually running fully recurrent neural networks**. Neural Computation, vol. 1, pp. 270-280, 1989

WONG, H.-L.; TU, Y.-H.; WAN, C.-C., **An Evaluation of Comparison between Multivariate Fuzzy Time Series with Traditional Time Series Model for Forecasting Taiwan Export**. World Congress on Computer Science and Information Engine. p. 462-467, 2009.

WPF. **WPF Home**. Disponível em: <[http://msdn.microsoft.com/en-us/library/ms754130\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms754130(v=vs.110).aspx)>. Acesso em 02/02/2014.

YIHANG, L.; LIELI, L., **Sales Forecasting through Fuzzy Neural Networks**. International Conference on Electronic Computer Technology. p. 511-515, 2009.

YUE, L., **A New Fuzzy Neural Networks Model for Demand Forecasting**. IEEE International Conference on Automation and Logistics. . p. 372-376, 2008.

ZHOU, M., **An ARIMA approach to forecasting electricity price with accuracy improvement by predicted errors**. IEEE Power Engineering Society General Meeting. p. 233-238. 2004.

ZHANG, G. P.; KLINE, D. M., **Quarterly Time-Series Forecasting With Neural Networks**. IEEE transactions on neural networks, P. 1800-1814. 2007.

## Apêndice A– Arquitetura do EsnPredictor

O EsnPredictor é um aplicativo com uma arquitetura por camadas do tipo MVC (Model View Controller), que separa os dados e as entidades da logica do aplicativo e da interação com o usuário, e segue o seguinte fluxograma.

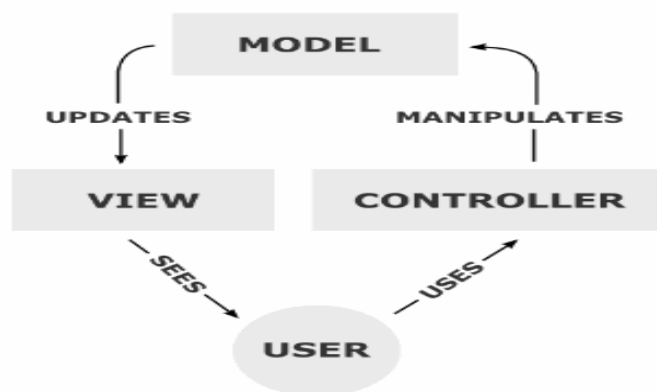


Figura 65. MVC

Os diagramas de classes do aplicativo explicam graficamente a hierarquia e relações dos elementos da ferramenta.

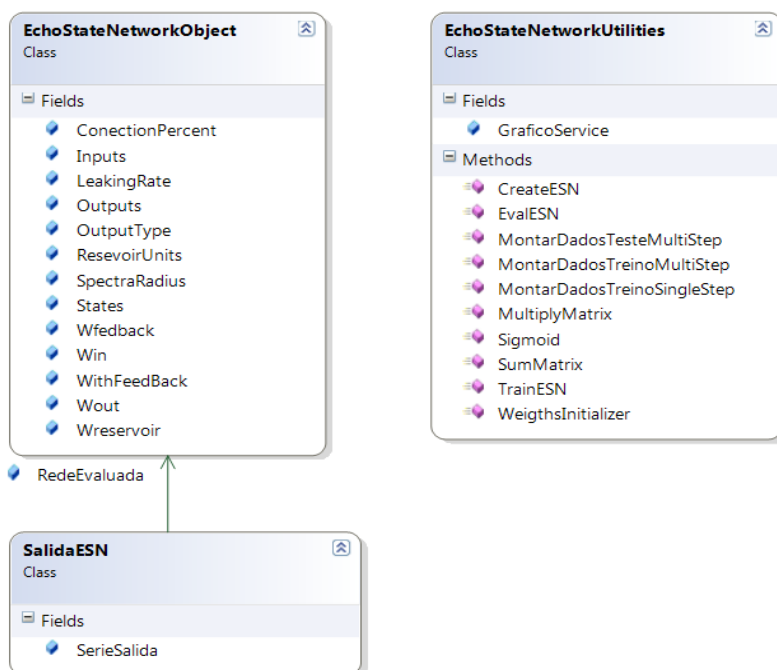


Figura 66. Diagrama ESN

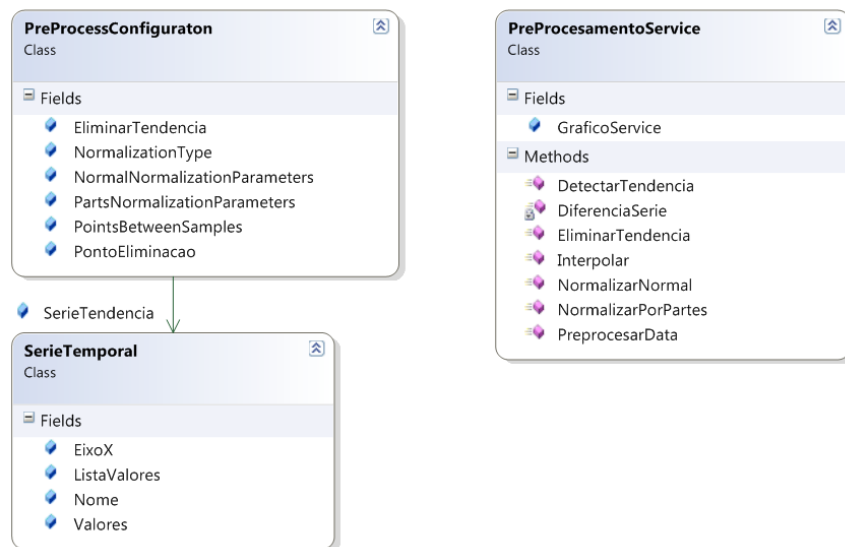


Figura 67. Diagrama de Pré-processamento

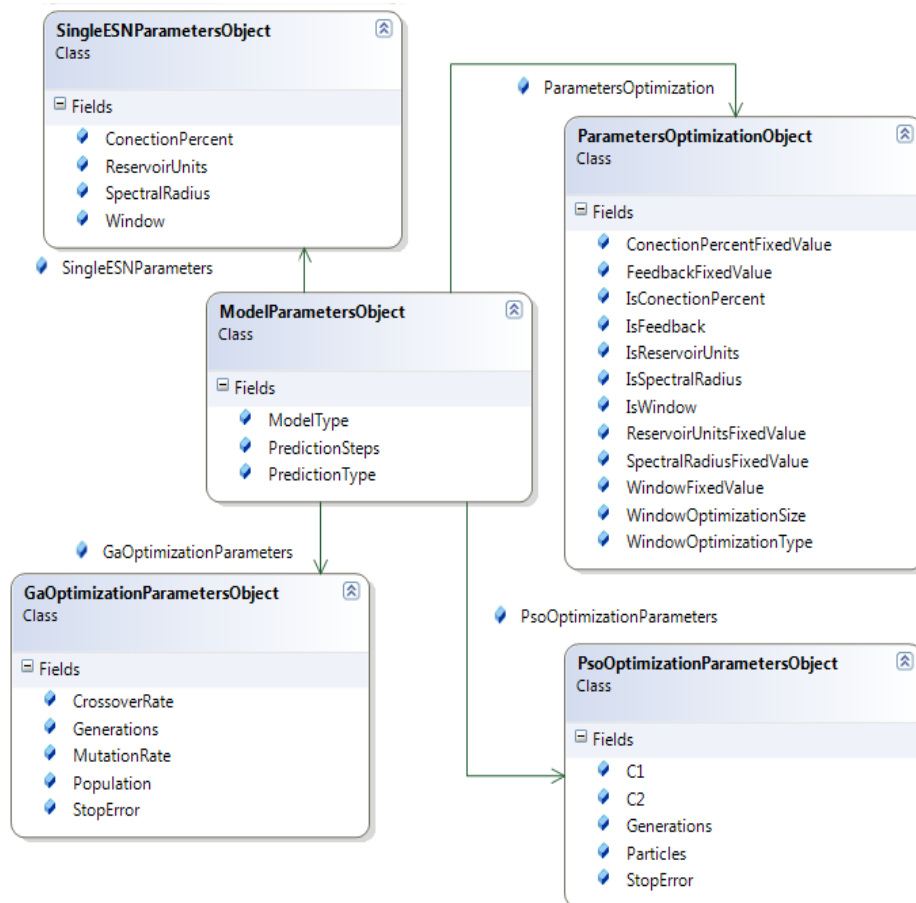


Figura 68. Diagrama de Parâmetros do Modelo

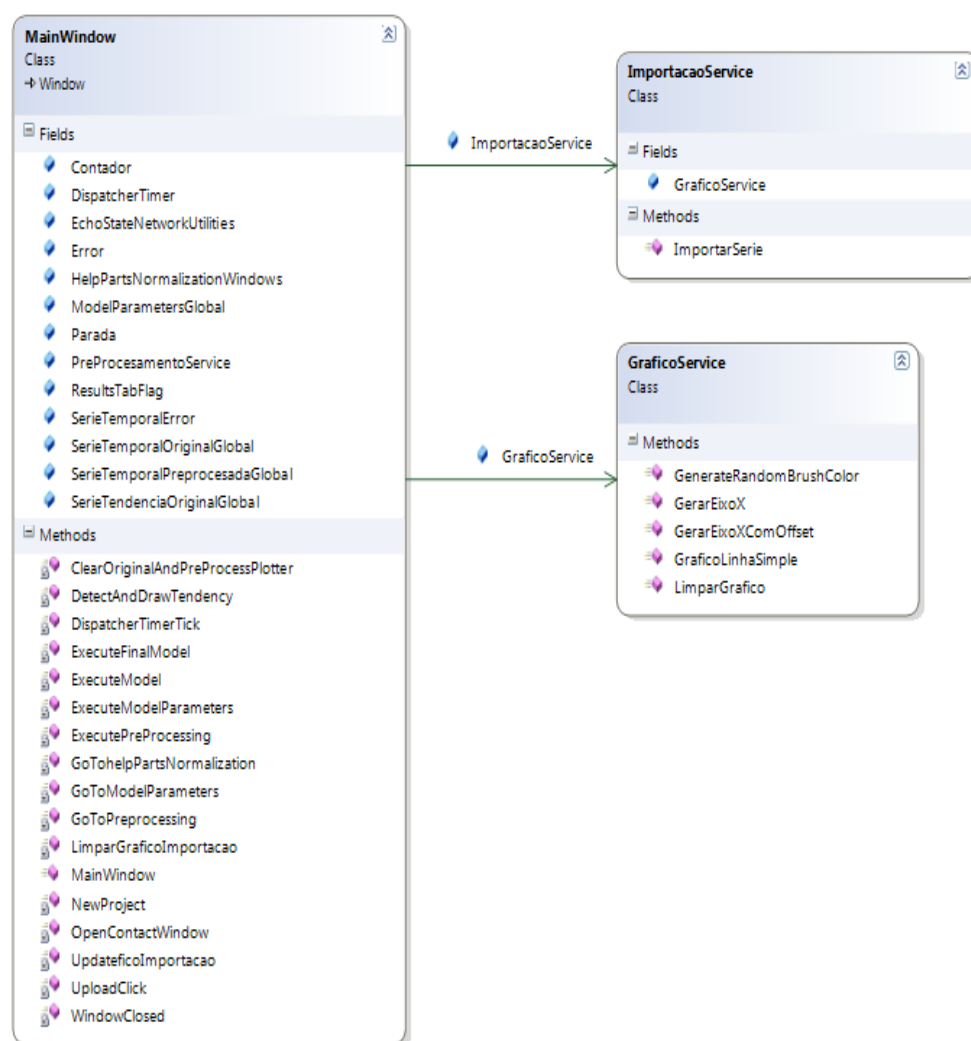


Figura 69. Diagrama Tela Principal