Pontifícia Universidade Católica
do Rio de Janeiro

Juliana Christina Carvalho de Araújo

Aplicação de SRV e ESN à previsão de séries do Mercado de Seguros

Dissertação de Mestrado

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

Orientador: Prof. Reinaldo Castro Souza

Rio de Janeiro
Setembro de 2016

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Juliana Christina Carvalho de Araújo

Aplicação de SRV e ESN à previsão de séries do Mercado de Seguros

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.

Prof. Reinaldo Castro Souza
Orientador
Departamento de Engenharia Elétrica - PUC-Rio

Prof. Ruy Luiz Milidiú
Departamento de Informática – PUC-Rio

Prof. Eugênio Kahn Epprecht
Departamento de Engenharia Industrial – PUC-Rio

Prof. Eduardo Gonçalves
UFJF

Prof. Márcio da Silveira Carvalho
Coordenador do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 14 de Setembro de 2016

Juliana Christina Carvalho de Araújo

Graduou-se em Ciências Econômicas pela Universidade Federal de Juiz de Fora (UFJF). É membro do grupo de pesquisa "Modelagem Estatística e de Séries Temporais: Aplicações na Área de Energia". Participa de projetos de pesquisa e desenvolvimento (P&D) relacionados à modelagem e previsão de séries temporais. Suas áreas de interesse e pesquisa incluem: Finanças, Energia e Seguros.

Ficha Catalográfica

Araújo**,** Juliana Christina Carvalho de

Aplicação de SRV e ESN à previsão de séries do mercado de seguros / Juliana Christina Carvalho de Araújo; orientador: Reinaldo Castro Souza. – 2016.
85 f.: il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2016.
Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Prêmio de Seguros. 3. Previsão. 4. Aprendizado de Máquinas. I. Souza, Reinaldo Castro. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

Agradecimentos

Chegando ao final desta etapa, não posso deixar de agradecer a algumas pessoas especiais que estiveram de alguma maneira presentes durante a realização desse trabalho. Saibam que nada disso teria sido possível sem vocês.

Inicio os meus agradecimentos fazendo menção ao grande amor da minha vida, Bruno. Obrigado por ter aceitado embarcar comigo nessa jornada.

Agradeço imensamente a minha família. Iniciando por meus pais, Kleber e Rosimére, por sempre terem me dado todo o amor, incentivo e estrutura necessária para que eu pudesse superar todos os desafios impostos. E finalmente aos meus irmãos, Ricardo e Rosiany, que sempre estiveram ao meu lado. Amo vocês!

Ao meu tio Fábio ("*in memoriam"*), pelos sábios conselhos, sem os quais não teria tomado às decisões que me trouxeram até aqui. Muitas saudades!

Dedico um agradecimento especial ao meu orientador, Reinaldo Castro Souza, pela grande influência na minha formação acadêmica e por toda a atenção dada durante o mestrado. Muito obrigado!

Agradeço aos meus amigos e também companheiros do IEPUC pelas valorosas discussões teóricas e pela amizade. Sem vocês muito desse trabalho não teria sido feito.

A todo o corpo docente do Departamento de Engenharia Elétrica da PUC-Rio que tiveram participação na minha formação acadêmica. E também à PUC-Rio e a CAPES pelos auxílios concedidos.

Resumo

Araújo, Juliana Christina Carvalho de; Souza, Reinaldo Castro (Orientador). **Aplicação de SRV e ESN à previsão de séries do Mercado de Seguros**. Rio de Janeiro, 2016. 85 p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A previsão de seguros é essencial para a indústria de seguros e resseguros. Ela fornece subsídios para estratégias de negócios de longo-prazo, e pode servir como um primeiro passo para o planejamento de linhas específicas de produtos. No contexto brasileiro, a previsão de seguros é de especial relevância. O Brasil possui o maior mercado segurador da América-Latina e tem potencial para se tornar um dos mais importantes centros seguradores do mundo no médio-longo-prazo. A SUSEP e a CNseg realizam previsões de carteiras do mercado de seguros brasileiro com base em modelos estatísticos. Entretanto, as séries temporais de prêmios utilizadas para essas previsões exibem comportamento não estacionário e não linear. Assim, a utilização de algoritmos de *machine learning*, na modelagem de séries de seguros, se justifica em função da habilidade desses algoritmos em capturar componentes de natureza não linear e dinâmica que possam estar presentes nessas séries, sem a necessidade de realizar suposições sobre o processo gerador dos dados. Com base no exposto, este trabalho investiga o uso de redes neurais *Echo State* (ESN) e GA-SVR na previsão de prêmios de seguros do mercado brasileiro. A base de dados utilizada neste trabalho foi disponibilizada pela SUSEP e compreende as carteiras de Automóveis, Vida e Previdência. Foram realizadas previsões univariadas e multivariadas com ESN e GA-SVR para as três carteiras mencionadas. Os resultados demonstram superioridade preditiva da ESN.

Palavras-Chave

Prêmio de seguros; previsão; aprendizado de máquinas.

Abstract

Araújo, Juliana Christina Carvalho de; Souza, Reinaldo Castro (Advisor). **Applying SVR and ESN to forecast Insurance Market Series**. Rio de Janeiro, 2016. 85 p. MSc. Dissertation – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Insurance forecasting is essential for the insurance industry. It provides support for long-term business strategies and can serve as a first-step for planning specific lines of products. In the Brazilian context, insurance forecasting is of special relevance. In the Latin American insurance market, Brazil is the leader in premium, and could become one of the most important insurance centers of the world in the medium- or long-term. SUSEP and CNseg forecast insurance products of the Brazilian market with statistical models. Nevertheless, premium time series exhibit nonstationary and nonlinear behavior. Therefore, the use of machine learning algorithms in the modeling of insurance series is justified, due to the ability of these algorithms in capturing nonlinear and dynamic components, which may be present in those series, without making assumptions about the data generating process. Based on this, this work investigates the use of Echo State neural networks (ESN) and GA-SVR in the forecast of insurance premium of the Brazilian market. The database used in this work was provided by SUSEP and consists of the products *Automobiles*, *Life* and *Providence*. Univariate and multivariate forecasts were made with ESN and GA-SVR for the three aforementioned products. The results show predictive superiority of ESN.

Keywords

Insurance premium; forecasting; machine learning.

Sumário

1
Introdução

1.1
Séries temporais e algoritmos de machine learning

A previsão de séries temporais é de grande relevância em diversas áreas. Previsões acuradas podem servir de base para a tomada de decisão e planejamento gerencial e/ou estratégico em Economia, Finanças, Indústria e Comércio [1,2].

A maior parte da literatura de séries temporais clássica supõe que as séries são estacionárias [3] ou que podem ser transformadas em séries estacionárias através de alguma transformação simples, como a diferenciação. Modelos estatísticos clássicos assumem que o processo gerador da série temporal é um processo linear, e fazem suposições sobre as características da série temporal [4], como os modelos de Box-Jenkins [1].

Séries temporais reais, no entanto, frequentemente não apresentam características de um processo linear. De fato, a alta complexidade dos fenômenos do mundo real reflete-se por comportamentos caóticos não lineares [4]. Nas últimas décadas, algoritmos de aprendizado de máquinas (*machine learning)* têm sido amplamente utilizados como alternativa aos métodos estatísticos clássicos de previsão [5,6].

Para o problema de previsão, redes neurais [7,8,9] e regressão por vetores suporte [10,11] são os algoritmos de *machine learning* mais utilizados. Geralmente, esta última aparece na literatura combinada com metaheurísticas evolucionárias que otimizam seus parâmetros, como a otimização por enxame de partículas (*Particle Swarm Optimization*, PSO) [12,13,14] e os algoritmos genéticos (*Genetic Algorithms*, GA) [15,16,17].

Os algoritmos de *machine learning* assumem que o mecanismo de geração dos dados é complexo e desconhecido. Eles utilizam os dados de entrada (*input*) para prever a saída (*output*), realizando poucas suposições sobre o processo gerador dos dados [18]. Dado um conjunto de exemplos de treinamento $\boldsymbol{T}$, o modelo geral de aprendizado de *machine learning* é composto por, basicamente,

três elementos: um gerador (G) de vetores aleatórios, um elemento supervisor (S), e uma máquina de aprendizado (MA).

Em termos gerais, o gerador produz, de maneira independente, vetores aleatórios x pertencentes a um espaço n-dimensional, i.e., $x \in R^n$, a partir de uma função de distribuição de probabilidade desconhecida, $\boldsymbol{\mathcal{F}}$. O elemento supervisor, por sua vez, produz o valor de saída y para cada vetor de entrada x, de acordo com uma função de distribuição condicional $\boldsymbol{\mathcal{D}}$, também desconhecida.

A máquina de aprendizado, que fundamenta a teoria por trás das redes neurais e regressão por vetores suporte, implementa um conjunto de funções $f(x)$, com o intuito de, dada uma entrada x^*, produzir uma saída $\hat{y}^*$ que se aproxime da resposta do supervisor. A Figura 1.1 ilustra o modelo geral de aprendizado a partir de exemplos.

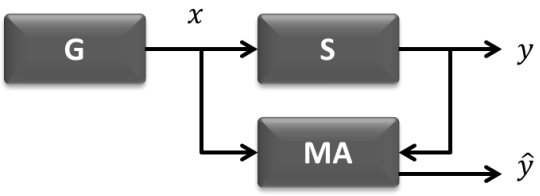

Figura 1.1 – Modelo básico de aprendizado de máquinas

1.2 Previsão de seguros e revisão da literatura

O setor de seguros é um setor de serviço relevante em qualquer economia. O desenvolvimento de todos os setores da economia nacional depende do desenvolvimento saudável do mercado de seguros e do apoio financeiro às companhias de seguros [19]. O seguro permite um planejamento futuro com mais segurança, evitando ou atenuando riscos específicos que são considerados uma ameaça para o processo comercial como um todo. Seguro é todo contrato pelo qual uma das partes, segurador, se obriga a indenizar a outra, segurado, em caso da ocorrência de determinado sinistro[1], em troca do recebimento de um montante (prêmio de seguro).

A previsão de seguros é de especial interesse para a indústria de seguros e resseguros. Ela pode fornecer subsídios para estratégias de negócios de longo-

[1] Sinistro é a ocorrência de acontecimentos previstos no contrato de seguro, de natureza súbita, involuntária e imprevista. Os prejuízos causados pelos riscos cobertos são indenizados ao segurado pela seguradora.

prazo, e pode servir como um primeiro passo para o planejamento de linhas específicas de produtos [20]. No contexto brasileiro, o estudo deste tipo de demanda é de especial relevância. Dados mostram que o setor de seguros é responsável por mais de 3,4% do produto interno bruto (PIB), o dobro do registrado no início da década de 90 [20,21]. No mercado da América Latina, o Brasil é líder em prêmios e oferece potencial para tornar-se um dos mais importantes centros seguradores do mundo no médio-longo prazo, em todos os segmentos, incluindo Seguros Gerais, Seguro de Vida e Produtos de Previdência Complementar [21].

Apesar da relevância para o setor, pesquisas bibliográficas na literatura acadêmica indicam a escassez de estudos envolvendo a previsão de demanda de Seguros. Quando se empreende pesquisas envolvendo palavras-chave como "seguro", "previsão", "modelos", "prêmio", "risco", "demanda de seguro" em bases de dados como *Scopus* ou *Web of Knowledge*, são encontrados muitos artigos fora do escopo desejado. Poucos tratam da especificamente da previsão de seguros [20,22,23]. A maioria utiliza métodos de estatística clássicos como Box e Jenkins [22,24], econometria [20,25,26] ou distribuições estatística para previsão de sinistros [27].

No Brasil, a Superintendência de Seguros Privados[2] (SUSEP) e a Confederação Nacional de Seguros Gerais, Previdência Privada e Vida, Saúde Suplementar e Capitalização (CNseg)[3] realizam previsões de carteiras do mercado de seguros com base em modelos estatísticos. Entretanto, as séries temporais de prêmios utilizadas para essas previsões exibem comportamento não estacionário e uma série de fatores que dificultam a sua previsão por modelos estatísticos clássicos:

1) Alterações no nível das carteiras de seguro: ao longo do tempo, algumas carteiras de produtos deixam de atuar nos mercados devido a fatores regulamentares, enquanto outros iniciam a sua atuação. Isso pode provocar alterações nos valores dos prêmios. No entanto, mesmo

[2] A SUSEP é responsável pela autorização, controle e fiscalização dos mercados de seguros, previdência complementar aberta, capitalização e resseguros no Brasil.

[3] A CNseg é responsável por congregar as principais lideranças, coordenar ações políticas, elaborar o planejamento estratégico do setor e representar o segmento perante o Governo, a sociedade e as entidades nacionais e internacionais.

que as carteiras permaneçam, modificação nos planos de cobertura do seguro podem provocar mudanças estruturais e/ou de nível.

2) Alterações econômicas ou climáticas: alterações econômicas ou climáticas significativas podem aumentar o nível de sinistralidade e/ou reduzir o valor das receitas das seguradoras por determinados períodos.
3) Alterações ao nível das empresas presentes nos mercados: No caso dos mercados de seguro, importantes alterações podem resultar, por exemplo, da fusão entre duas empresas concorrentes, ou da falência de uma empresa significativa para o mercado. Segundo dados da SUSEP, a participação no mercado das 10 maiores seguradoras representava 60,0% em 2001, atingindo o percentual máximo de 67,3% em 2015.
4) Efeito calendário em função da flutuação na quantidade de dias úteis em cada mês.

As séries temporais de seguros são, portanto, afetadas por regulamentações no setor. Isto faz com que dados históricos antigos possam não ser representativos do processo atual da série. Adicionalmente, como os mercados e a Economia estão em contínua mutação, mesmo os dados mais recentes representam um processo que pode sofrer alterações de funcionamento significativas em relação ao futuro imediato. Assim, a utilização de algoritmos de *machine learning*, na modelagem de séries de seguros, se justifica em função de suas habilidades em capturar a natureza não linear, dinâmica e caótica que possam estar presentes nessas séries, sem a necessidade de realizar suposições sobre a distribuição dos dados [28].

1.3 Objetivos e contribuições

Com base no que foi exposto, os principais objetivos e contribuições desta dissertação compreendem:

- Investigar o uso de algoritmos de *machine learning* na previsão de prêmios[4] de seguros;
- Criar um modelo automático híbrido (GA-SVR) que combine algoritmos genéticos (GAs) para otimização de parâmetros e

[4] Consideram-se os prêmios de seguro como o melhor indicador que mostra o tamanho da indústria de seguros e reflete a demanda do setor.

regressão por vetores suporte (SVR) para previsão no mercado de seguros;

- Investigar o uso de Redes Neurais com Estado de Eco (*echo-state*) no contexto do mercado segurador;
- Realizar previsões univariadas e multivariadas para as carteiras de Automóveis, Vida e Previdência do mercado brasileiro de seguros;
- Verificar a hipótese de não linearidade nas carteiras de Automóveis, Vida e Previdência do mercado brasileiro de seguros;
- Contribuir para a literatura de previsão de seguros.

1.4 Organização da dissertação

O presente capítulo motiva a dissertação, mostrando a sua relevância; caracteriza a inovação através de busca na literatura científica por trabalhos correlatos; apresenta o problema a ser resolvido e os objetivos da dissertação.

No segundo capítulo, realiza-se uma breve revisão do método de redes com estado de eco para previsão de séries temporais. O terceiro capítulo apresenta uma síntese sobre a teoria do aprendizado estatístico e sobre regressão por vetores suporte. No quarto Capítulo, é realizada uma introdução aos algoritmos genéticos e o seu uso na otimização de parâmetros da regressão por vetores suporte. Este capítulo sintetiza o GA-SVR para previsão.

No quinto capítulo, apresenta-se os modelos propostos para o mercado de seguro, as métricas de avaliação da acurácia das previsões realizadas e os modelos estatísticos para a comparação. No sexto capítulo, é apresentado um estudo de caso onde os modelos apresentados são aplicados na previsão de prêmio de três carteiras de seguro brasileira, a saber: Automóveis, Vida e Previdência. O sétimo capítulo apresenta as conclusões obtidas com o uso dos modelos propostos.

2
Redes neurais com estado de eco

2.1
Introdução

Uma Rede Neural (*Neural Network*, NN) é um sistema massivamente paralelo e distribuído, composto por unidades de processamento simples que possuem uma capacidade natural de armazenar e utilizar conhecimento [29][5]. As redes neurais são inspiradas na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

As NNs foram desenvolvidas, originalmente, na década de 40, a partir do neurônio artificial projetado por McCulloch, e Pitts [29]. Desde então, mais expressivamente a partir da década 80, diversos modelos de redes neurais têm surgido. Mais recentemente as Redes Neural Estado de Eco (*Echo State Networks*, ESN) [30,31,32] propostas por Jeager tornaram-se amplamente utilizado pela simplicidade no treinamento e pelos ótimos resultados obtidos em benchmarks como a *Japanese Vowel Competition* [37] e a *International Forecasting Competition* NN3 [9].

2.2
O neurônio artificial

O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico real. As principais partes do neurônio artificial genérico são:

i. As sinapses, caracterizadas pelos seus pesos associados,
ii. O a junção somadora e
iii. A função de ativação.

A Figura 2.1 ilustra o esquema de um neurônio artificial genérico.

[5] Ao leitor não familiarizado com os conceitos de Redes Neurais sugere-se a leitura de Haykin [29].

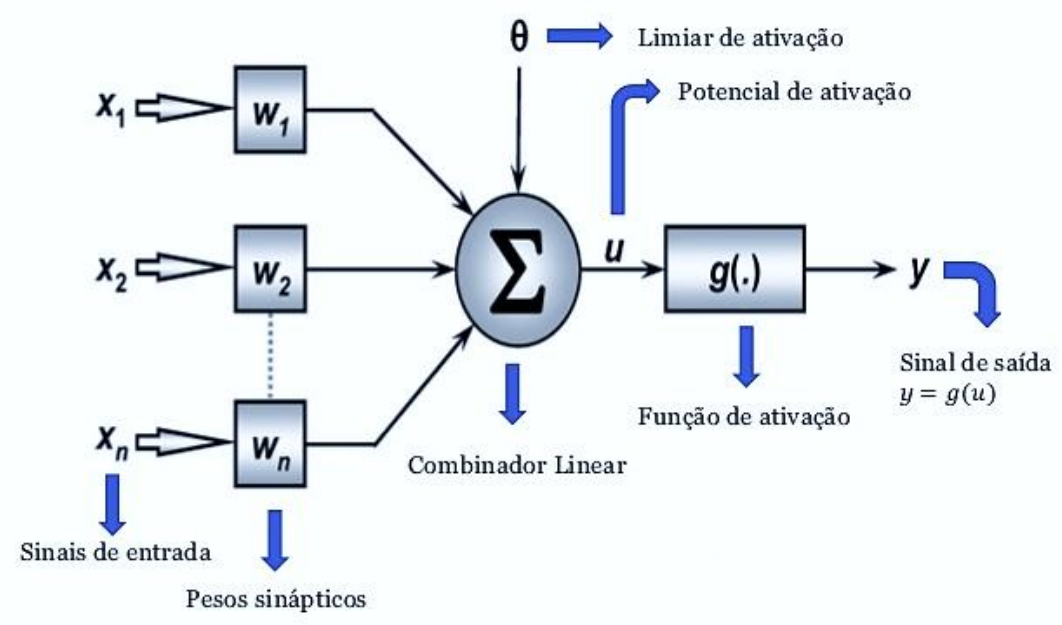

Figura 2.1 – Esquema de um neurônio artificial genérico.

Na representação ilustrada na Figura 2.1, a junção somadora soma todos os sinais de entrada ponderados pelos pesos das conexões. Assumindo os vetores de entrada e de pesos como sendo vetores coluna, esta operação corresponde ao produto interno do vetor de entradas x pelo vetor de pesos w, mais o limiar θ. A função de ativação é geralmente utilizada para limitar a saída do neurônio e introduzir não linearidade no modelo. O limiar θ tem a função de aumentar ou diminuir a influência do valor da entrada para a ativação do neurônio. Matematicamente, a saída do neurônio pode ser descrita como uma combinação linear das entradas pelos pesos associados, mais o limiar θ: $y = g(u) = f(\sum_{i=1}^{n} w_i x_i + \theta)$ ou $y = g(u) = f(\sum_{i=0}^{n} w_i x_i)$, onde x_0 é um sinal de entrada de valor um e peso associado $w_0 = \theta$.

2.3
Facetas elementares de redes neurais

Uma Rede Neural possui duas facetas elementares, a arquitetura e o algoritmo de aprendizagem:

i. A arquitetura da rede que descreve a conectividade entre os neurônios, e.
ii. O algoritmo de treinamento ou aprendizado, que consiste na estratégia de ajuste dos pesos sinápticos.

Existem basicamente três paradigmas de aprendizado [29]:

i. O aprendizado supervisionado é baseado em um conjunto de exemplos de estímulo resposta, ou em algum outro tipo de informação, que represente o comportamento que deve ser apresentado pela rede neural,
ii. O aprendizado não supervisionado que é baseado apenas nos estímulos recebidos pela rede neural. Basicamente, a rede deve aprender os "padrões" dos estímulos e
iii. O aprendizado por reforço onde o comportamento da rede é avaliado apenas com base em algum critério numérico, fornecido em instantes espaçados de tempo.

Quanto à arquitetura, as NNs podem ser classificadas em dois tipos:

i. As Redes Neurais Progressivas (*Feedforward Neural Networks*, FNN) [29], onde os sinais são propagados em um único sentido, e
ii. As Redes Neurais Recorrentes (*Recurrent Neural Networks*, RNN) [33]. Nesta rede existem laços de realimentação que se deslocam desde os neurônios de uma camada até de camadas anteriores.

2.4 Redes neurais recorrentes

RNNs são estruturas de processamento capazes de representar uma grande variedade de comportamentos dinâmicos não lineares. A presença de laços de realimentação (*feedback*) permite a RNN processar e armazenar informações temporais e sinais sequenciais, dando a ela a capacidade de reproduzir comportamentos complexos mesmo com um número reduzido de parâmetros. Por conseguinte, a rede pode gerar respostas distintas a um mesmo estímulo de entrada, dependendo de seu estado atual. Da perspectiva de processamento de sinais, RNNs se assemelham a filtros com resposta a um impulso infinito [34].

Foi demonstrado por Funahashi e Nakamura [35] e Schafer e Zimmermann [36] que a arquitetura de rede recorrente apresenta a capacidade de aproximação universal. Contudo devido à realimentação, pequenas alterações nos parâmetros podem levar a dinâmica da rede de pontos fixos estáveis para instáveis, o que

causa um súbito salto na medida de erro[6]. Neste cenário, as Redes com Estado de Eco e as NNs com padrão *Long Short-Term Memory* (LSTM) [37] representam soluções robustas e promissoras.

2.5 Redes neurais estado de eco

Uma ESN é, basicamente, uma rede neural recorrente com aprendizado supervisionado onde a camada oculta é uma camada densamente interconectada (denominada "reservatório de dinâmicas") e a camada de saída corresponde a um combinador linear ajustável. Diferentemente das abordagens tradicionais de redes neurais recorrentes, nas ESNs não há necessidade de treinamento dos pesos da camada de entrada e nem dos pesos internos da rede (*Reservoir*). Apenas os pesos da camada de saída são treinados. Os estados do reservatório são chamados de Estado de Eco *(Echo State)* e são os responsáveis pela memória dos padrões de entrada.

A arquitetura de uma ESN consiste de uma camada de entrada, uma camada densamente interconectada de Elementos de Processamento Não Linear (EPNL) que compõem um Reservatório de Dinâmicas (RD), e uma camada de saída responsável por combinar esses EPNLs. A estrutura básica de uma ESN com K sinais de entrada $\mathbf{u}$, N unidades internas $\mathbf{x}$ e L saídas y é ilustrada na Figura 2..

[6] Esta e outras dificuldades das RNNs são listadas por Jaeger [44].

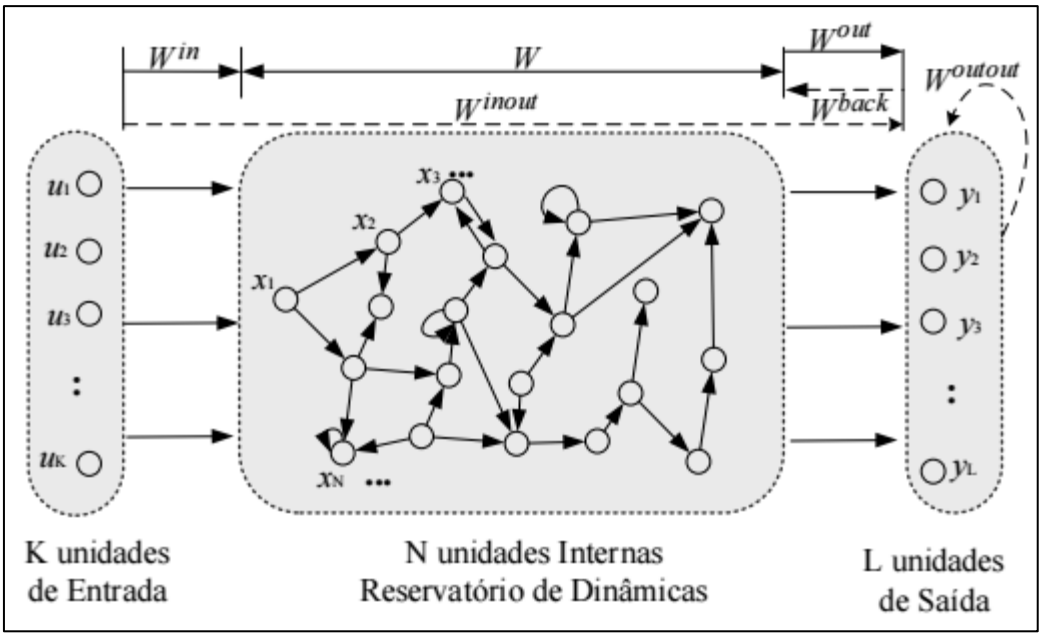

Figura 2.2 – Arquitetura genérica de uma ESN. Fonte: adaptado de [38].

Na Figura 2., temos que:

i. $\mathbf{W^{in}}$ representa a matriz dos pesos sinápticos entre a camada de entrada e o reservatório;
ii. $\boldsymbol{W}$ representa as conexões do reservatório;
iii. $\mathbf{W^{back}}$ são as conexões entre a camada de saída e o reservatório;
iv. $\mathbf{W^{inout}}$ são as conexões entre a camada de entrada e a camada de saída;
v. $\mathbf{W^{out}}$ são as conexões entre o reservatório e a camada de saída e
vi. $\mathbf{W^{outout}}$ representa as conexões recorrentes da camada de saída.

Apenas as conexões no sentido à camada de saída são treinadas ($\mathbf{W^{inout}}$, $\mathbf{W^{out}}$ e $\mathbf{W^{outout}}$).

2.6
Breve descrição matemática do modelo básico da ESN

Seja um vetor de entrada com K elementos para um tempo $t \geq 1$, i.e., $\mathbf{u}(n) = \left(\mathrm{u}_1(n), \ldots, u_k(n)\right)^t$; seja um vetor de estados dos neurônios do reservatório com N elementos, i.e., $\mathbf{x}(n) = \left(x_1(n), \ldots, x_N(n)\right)^t$; e seja um vetor de saída $\mathbf{y}(n) = \left(y_1(n), \ldots, y_L(n)\right)^t$ com L elementos. A atualização dos estados é definida pela Equação (2-1).

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W^{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W^{back}}\mathbf{y}(n)) \quad (2\text{-}1)$$

Onde **f** é a função de ativação das unidades internas do reservatório, tipicamente a função sigmoide ou tangente hiperbólica[7]. É possível aplicar um parâmetro α (*leak rate*) aos neurônios sigmoides da rede [39]. Este parâmetro pode ser aplicado antes ou depois da função de ativação do neurônio. Se escolhido corretamente, o parâmetro α pode efetivamente ajustar a dinâmica da ESN para coincidir com a escala de tempo de entrada, melhorando o desempenho da rede [40]. A Equação (2-2) apresenta as mudanças da Equação (2-1) de uma ESN depois de se adicionar o parâmetro α antes da função de ativação:

$$\mathbf{x}(n+1) = \mathbf{f}((1-\alpha)\mathbf{x}(\mathrm{n}) + \alpha(\mathbf{W^{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W^{back}}\mathbf{y}(n)+) \quad (2\text{-}2)$$

A saída da ESN é computada de acordo com a Equação (2-):

$$\mathbf{y}(n+1) = \mathbf{f^{out}}(\mathbf{W^{inout}}\mathbf{u}(n+1) + \mathbf{W^{out}}\mathbf{x}(n+1) + \mathbf{W^{outout}}\mathbf{y}(n)) \quad (2\text{-}3)$$

A Equação (2-4) do erro de predição e a Equação (2-5) da atualização do $\mathbf{W^{out}}$ através de um algoritmo estocástico de gradiente descendente são descritos a seguir:

$$\mathbf{e}(n) = \boldsymbol{y_d}(n) - \boldsymbol{y}(n) \quad (2\text{-}5)$$

$$\Delta\mathbf{W^{out}} = \eta \boldsymbol{e}(n)[\boldsymbol{u}(n); \boldsymbol{x}(n)]^T \quad (2\text{-}6)$$

Em que: $\mathbf{e}(n)$ é o erro, $\boldsymbol{y_d}(n)$ os valores de saída desejados, $\boldsymbol{y}(n)$ as saídas da rede, $\mathbf{W^{out}}$ representa os pesos sinápticos entre o reservatório e a camada de saída, η é a taxa de aprendizado, $\boldsymbol{u}(n)$ são as entradas da rede, e $\boldsymbol{x}(n)$ os estados do reservatório no instante n.

[7] Quando se utiliza função tangente hiperbólica 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. No caso da sigmoide o escalonamento deve ser realizado no intervalo entre 0 e 1. O objetivo é adaptar os dados de entrada ao intervalo das funções de ativação, aumentando a exatidão da previsão realizada pela ESN.

2.7
Propriedade de eco e geração do reservatório

A propriedade de eco[8] é crucial para que o modelo ESN funcione. Intuitivamente, uma rede recorrente exposta a sinais externos **u**(n) tem a propriedade de *Echo State* se as ativações dos neurônios **x**(n) forem variações sistemáticas dos sinais de entrada **u**(n). De uma maneira mais formal, isso significa que para cada neurônio interno x_i existe uma função de eco (*echo function*) e_i, tal que, se a rede foi executada por um tempo indefinidamente longo no passado, o estado corrente poderá ser escrito como na Equação (2-).

$$x_i(n) = e_i(\boldsymbol{u}(n), \boldsymbol{u}(n-1), \boldsymbol{u}(n-2), \dots) \tag{2-6}$$

Para a ESN de tempo discreto existem inúmeras definições alternativas (não-triviais) e caracterizações algébricas que estabelecem que as matrizes de pesos **W** da rede conduzem às redes de propriedade *Echo State* [41]. Para finalidades práticas, entretanto, basta ajustar o maior valor absoluto $|\boldsymbol{\lambda}_{max}|$ de um autovetor de **W**, denominado "raio espectral". O raio espectral é usado para escalar ou normalizar os pesos para que a rede não apresente um comportamento caótico e para que as ativações dos neurônios do Reservatório Dinâmico não sejam saturadas. Se $|\boldsymbol{\lambda}_{max}| > 1$, a rede não tem a propriedade de *Echo State*. Deste modo, para aplicações práticas, $|\boldsymbol{\lambda}_{max}| < 1$. É também importante que a dinâmica dos neurônios do reservatório seja bastante variada.

Recentemente, Yildiz, Jaeger e Kiebel demonstraram uma nova condição suficiente para a existência de estados de eco. Esta condição explora a ideia de estabilidade matricial de Schur, apresentada em [42].

2.8
Treinamento de uma ESN

O processo de aprendizado da ESN consiste em determinar os pesos utilizando um conjunto de dados de treinamento T, que consiste de pares de entradas e saídas desejadas $[\langle u(1), y_d(1)\rangle, \langle u(2), y_d(2)\rangle, \dots, \langle u(T), y_d(T)\rangle]$. Quando a ESN é alimentada com a entrada de treinamento u(n), a saída y(n) será próxima

[8] Os estados da rede tornam-se assintoticamente independentes da condição inicial, i.e., o efeito dos estados iniciais desaparece e as dinâmicas presentes no reservatório passam a ser governadas pelo histórico recente dos sinais de entrada - por isso o termo [30].

da saída desejada $y_d(n)$. A Equação (2-) mostra uma função *Normalized Root Mean Square Error* (NRMSE), que deve ser minimizada durante o processo de treinamento:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{2-7}$$

Onde o RMSE é calculado conforme a Equação (2-) e y_{max} e y_{min} são os valores máximo e mínimo de $y(n)$, respectivamente.

$$RMSE = \sqrt{\frac{1}{T}\sum_{n=1}^{T}(e(n))^2} \tag{2-8}$$

Onde $e(n)$ é o erro de predição da Equação (2-1).

O treinamento começa com a geração de uma RNN não treinada e aleatória que satisfaça a propriedade do estado de eco. Para a geração aleatória de pesos de uma ESN serão necessários dois parâmetros: o tamanho da rede N, e o raio espectral $\alpha \in [0,1]$.:

- **Passo 1:** Gerar uma matriz de pesos aleatórios W_0;
- **Passo 2:** Normalizar a matriz W_0, gerando a matriz W_1 com raio espectral λ de W_0, $W_1 = W_0/\lambda$ de forma que W_1 terá raio espectral unitário;
- **Passo 3:** Escalar a matriz W_1 para a matriz W com $\alpha < 1, W = \alpha W_1$. Assim W terá raio espectral α;
- **Passo 4:** Gerar as matrizes de pesos aleatórios $\mathbf{W^{in}}$ e $\mathbf{W^{back}}$.

Utilizando as matrizes de pesos inicializados, executa-se a ESN com o conjunto de dados de treinamento T, e aplicando-se a Equação (2-1). Em cada amostra, os estados das unidades de entrada e das unidades internas são armazenados na matriz de coleção de estados X de dimensão $T \times (K + N)$, sendo K, N e L os números de unidades de entrada, interna e saída, respectivamente.

Chamemos $\mathbf{W^{out}}$ todos os pesos do *readout* ($\mathbf{W^{inout}}$, $\mathbf{W^{out}}$, $\mathbf{W^{outout}}$ e $\mathbf{W^{biasout}}$), a formação do *readout* a partir de um reservatório é uma tarefa supervisionada não-temporal de mapeamento dos estados do reservatório x(n) para as saídas desejadas d(n). Sendo este um domínio bastante investigado em *Machine Learning*, existem vários métodos disponíveis. No método *batch*, o

treinamento dos pesos do $\mathbf{W^{out}}$ pode ser definido como uma solução de um sistema de equações lineares da Equação (2-.

$$\boldsymbol{W^{out}}\mathbf{X} = \boldsymbol{Y_d} \tag{2-9}$$

Em que: $\mathbf{X} \in \Re^{NXT}$ são todos os estados do reservatório e $\boldsymbol{Y_d} \in \Re^{LXT}$ são todos os valores desejados $d(n)$, ambos armazenados durante o período de treinamento $n = 1, \dots, T$.

O processo de treinamento consiste em minimizar o erro quadrático $E(\boldsymbol{Y_d}, \boldsymbol{W^{out}})$. Para solucionar a Equação (2-, são utilizados, com frequência, métodos de regressão linear. Os métodos comumente utilizados para o treinamento dos pedos da camada de saída (*readout*) são a pseudo-inversa de Moore-Penrose [39] e a regressão Ridge (ridge-regress) [43]. Entretanto, existem outros métodos, como a decomposição ponderada ou a busca evolucionária [39]. Um método direto consiste em calcular a pseudo-inversa de Moore-Penrose $\boldsymbol{X}^{\dagger}$ de $\mathbf{X}$ e $\mathbf{W}^{\boldsymbol{out}}$ através da Equação (2-:

$$\mathbf{W}^{\boldsymbol{out}} = \boldsymbol{Y_d}\boldsymbol{X}^{\dagger} \tag{2-10}$$

Com $\mathbf{W}^{\boldsymbol{out}}$ calculado, pode-se utilizar a Equação (2-) para se obter as saídas desejadas.

O cálculo direto através da pseudo-inversa mostra alta estabilidade numérica, mas demanda um alto custo computacional para grandes matrizes de coleta de estados, limitando assim o tamanho do reservatório N e/ou o número de amostras do conjunto de dados de treinamento T. Este problema é resolvido formulando equações a partir da Equação (2-), obtendo a Equação (2-1) e finalmente a sua solução é expressada segundo a Equação (2-). Desse modo, a solução passa a não depender do tamanho do conjunto de treinamento T[9].

$$\mathbf{W}^{\boldsymbol{out}}\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} = \boldsymbol{Y_d}\boldsymbol{X}^{\boldsymbol{T}} \tag{2-11}$$

$$\mathbf{W}^{\boldsymbol{out}} = \boldsymbol{Y_d}\boldsymbol{X}^{\boldsymbol{T}}\left(\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\right)^{-\mathbf{1}} \tag{2-12}$$

[9]Note que, neste caso, $\boldsymbol{Y_d} \in \boldsymbol{\Re}^{\boldsymbol{L}\times\boldsymbol{N}}$ e $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} \in \boldsymbol{\Re}^{\boldsymbol{N}\times\boldsymbol{N}}$.

2.9
Parâmetros da ESN

O ajuste de alguns parâmetros que fazem parte da topologia da ESN é fundamental para o bom desempenho da rede. Alguns desses parâmetros são o percentual de conexões (da matriz $\mathbf{W}$)[10], o tamanho N do reservatório e o raio espectral. Algumas recomendações de Jaeger [44] para ajuste de parâmetros da ESN são resumidas e apresentadas aqui:

- A matriz W deve ser esparsa, de tal forma a permitir a criação de uma rica variedade de dinâmicas no reservatório. A intuição subjacente a esta proposta é que um padrão de conexões esparso (cerca de 1-20%) tende a favorecer o desacoplamento de grupos de neurônios do reservatório, o que, por sua vez, pode contribuir para o desenvolvimento de dinâmicas individuais, isto é, pouco correlacionadas. Os pesos devem ser inicializados aleatoriamente de maneira que sejam homogêneos e distribuídos com média zero. Um reservatório esparso pode melhorar o tempo de processamento [39,45];
- O tamanho N do reservatório deve refletir a largura T do conjunto de dados de treinamento e a complexidade da tarefa. Como regra N não deve exceder o tamanho de $T/10$ a $T/2$. Trata-se de uma precaução contra *overffiting* ou sobreparametrização.
- A escolha do raio espectral λ, como já mencionado, é crítica para o desempenho do modelo. Ele deve ser pequeno para saídas desejadas com dinâmicas rápida e grande para dinâmicas lentas. A estabilidade do reservatório ESN é garantida quando o raio espectral é menor do que 1. Esta condição de estabilidade foi estabelecida na propriedade de estado de eco (ESP) [30].
- O tamanho absoluto da matriz de pesos de entrada $\mathbf{W}^{in}$ é importante. Valores grandes indicam que a rede é fortemente influenciada pelas entradas e pequenos indicam que a rede é excitada fracamente. Neste caso os neurônios irão operar próximo da região linear da função de ativação que estiver sendo utilizada, sigmoide ou tangente hiperbólica,

[10] O percentual de conexões indica a porcentagem de neurônios que vão ser conectados aleatoriamente dentro do RD normalmente de 60%, para garantir uma dinâmica variada.

por exemplo. Grandes valores de pesos de entrada farão as unidades da rede trabalharem próximo à região de saturação.

- As mesmas observações no item anterior são aplicáveis à matriz $\mathbf{W}^{back}$.

2.10 Desenvolvimentos da ESN

A maior parte das pesquisas em ESN atualmente é dedicada ao modelo do reservatório ideal e ou à natureza da camada de saída (ou *readout*). Há abordagens que tratam o projeto do reservatório da perspectiva de aprendizado supervisionado, recorrendo a ferramentas de otimização evolutiva para ajustar as conexões recorrentes do reservatório de dinâmicas, como realizado por [46] e detalhadas em [39]. A escolha por um combinador linear cujos coeficientes sejam ajustados visando minimizar o erro quadrático médio entre a saída da rede e um sinal de referência se mostra atraente pela disponibilidade de um vasto arcabouço teórico para seu treinamento. Contudo, há várias outras estruturas de processamento abordadas na literatura.

O uso de uma rede *Multilayer Perceptron* como camada de saída foi proposta por Babinec e Pospíchal [47]. Butcher *et al.* [48,49] propõem um modelo híbrido que constrói o *readout* com base em *extreme learning machines* [50]. Outra abordagem é apresentada por Shi e Han [51], com o uso de uma Máquina de Vetor Suporte (*Support Vector Machine*, SVM) [52,53] para treinamento de uma ESN, de modo a utilizar os princípios de uma SVM no treinamento do reservatório. Uma combinação de diferentes camadas de saída é sugerida por Bush e Anderson [54] e uma aplicação de saídas hierárquicas foi proposta por Jaeger [55]. Boccato *et al*. [56,57] faz uso da estrutura de um filtro de Volterra em lugar do combinador linear na camada de leitura. Neste caso, as saídas da rede são obtidas através de combinações lineares de termos polinomiais [58]. Em certo sentido, porém, todas estas abordagens mencionadas parecem caminhar na direção contrária ao espírito de simplicidade inerente às ESNs.

3
Regressão por vetores suporte

3.1
Introdução

O primeiro algoritmo a introduzir noções de Vetores Suporte foi proposto por Vapnik e Lerner [59] na década de 60 [60][11]. A forma atual das Máquinas de Vetores Suporte (SVMs) é uma generalização deste primeiro algoritmo.

Originalmente desenvolvidas para resolver problemas de reconhecimento de padrões (classificação) [52,53], as Máquinas de Vetores Suporte foram posteriormente estendidas para solucionar o problema de aproximação de funções [61,62]. A versão do SVM para regressão foi proposta por Drucker *et al.* [63] e recebeu o nome de Regressão por Vetores Suporte (*Support Vector Regression*, SVR, ou *Support Regression Vector, SRV*).

Baseada na Teoria de Aprendizado Estatístico [64,65,66], a regressão-SV possui a capacidade de aproximar qualquer função, realizando poucas suposições sobre o processo gerador dos dados [67,28]. Algumas das propriedades atrativas da SVR incluem: aproximação de funções tanto lineares como não-lineares; bom desempenho na generalização de dados e ausência de mínimos locais na resolução de problemas.

3.2
Teoria de aprendizagem estatística

A Teoria de Aprendizado Estatístico (TAE) [64,65,66] estabelece as condições matemáticas que auxiliam na escolha de um classificador ou regressor particular f a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador ou regressor no conjunto de treino e a sua complexidade, com o objetivo de obter um bom desempenho para novos

[11] Algoritmo *Generalized Portraits*.

dados do mesmo domínio. O problema do aprendizado de máquinas pode ser sintetizado conforme descrito a seguir.

Dado um espaço de entrada $\boldsymbol{\mathcal{X}}$ e um espaço de saída $\boldsymbol{\mathcal{Y}}$ oriundos de uma distribuição de probabilidade conjunta $\boldsymbol{\mathcal{D}}$ sobre $\boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{Y}}$, e de posse de um conjunto de exemplos (denominado dados de treinamento ou conjunto de entrada) $\boldsymbol{T} = \{(x_1, y_1), \ldots, (x_t, y_t)\}$ amostrados de maneira independente a partir de $\boldsymbol{\mathcal{D}}$[12], o objetivo do algoritmo de aprendizagem é encontrar uma função (de alguma classe de funções $\boldsymbol{\mathcal{F}}$) que melhor aproxime, a partir do conjunto de exemplos $\boldsymbol{T}$, o conjunto de dados amostrados de $\boldsymbol{\mathcal{D}}$ [68]. Ou seja, o objetivo é o de encontrar uma função que tenha boa capacidade de generalização. Ademais, a TAE estabelece que:

i. Nenhuma suposição é feita sobre a distribuição de $\boldsymbol{\mathcal{D}}$;
ii. $\boldsymbol{\mathcal{D}}$ é fixa, não se altera ao longo do tempo;
iii. No momento da aprendizagem, $\boldsymbol{\mathcal{D}}$ é desconhecida pela máquina;
iv. Devido a ruídos e sobreposição de classes, $\boldsymbol{T}$ não é determinístico.

A função de perda $\ell(f(x), y)$ mede a discrepância entra a resposta observada y do conjunto de exemplos $\boldsymbol{T}$, e a resposta da função f escolhida (para a entrada x). A perda esperada da função f de todos os pontos $x \in \boldsymbol{\mathcal{X}}$ gerados por $\boldsymbol{\mathcal{D}}$, denominado risco esperado de f [68] é definida pela Equação (3-1).

$$R(f) = E(\ell(f(x), y)) \tag{3-1}$$

O objetivo do aprendizado estatístico é encontrar a função $f \in \boldsymbol{\mathcal{F}}$ que minimize o risco esperado $R(f)$ da função de perda $\ell(f(x), y)$. No entanto, como $\boldsymbol{\mathcal{D}}$ é desconhecido pela máquina, não é possível calcular o risco esperado. Então, busca-se inferir uma função f que minimize o risco empírico $R_{emp}(f)$ na amostra de treinamento. Assim, aproxima-se o risco esperado por meio do risco empírico.

[12] Para o uso do aprendizado de máquina na previsão de séries temporais a hipótese de independência é relaxada.

3.2.1
Princípio da minimização do risco empírico

O objetivo do princípio da minimização do risco empírico (*Empirical Risk Minimization*, ERM) é encontrar um classificador (ou um regressor) f_n tal que [68]:

$$f_n = argmin_{f \in F} R_{emp}(f) \tag{3-2}$$

Dado o conjunto de treinamento ***T***, e utilizando os princípios indutivos da minimização do risco empírico, a minimização da função-risco desconhecida é substituída pela minimização da função conhecida do risco empírico, $R_{emp}(f)$.

$$R_{emp}(f) = \frac{1}{t}\sum_{i=1}^{t} \ell(f(x_i), y_i) \tag{3-3}$$

A motivação do Princípio da Minimização Empírica do Risco [69] está na Lei dos Grandes Números [70]. Pela Lei dos Grandes Números pode-se concluir que, para uma função fixa f, o risco empírico converge para o risco esperado à medida que o tamanho do conjunto de treinamento tende a infinito:

$$R_{emp}(f) = \frac{1}{t}\sum_{i=1}^{t} \ell(f(x_i), y_i) \rightarrow \ E(`\ell(f(x), y))\ para\ n \rightarrow \infty \tag{3-4}$$

Todavia, dado o tamanho limitado dos conjuntos de treinamento em problemas reais, a minimização do risco empírico não implica, necessariamente, na minimização do risco funcional[13]. Métodos clássicos para a solução de problemas de aprendizagem, como o da máxima-verossimilhança, o de mínimos e ao das redes neurais são instanciações do princípio de ERM para funções de perda específicas. A funcional de risco empírico $R_{emp}(f)$ difere da funcional de risco $R(f)$ em dois aspectos principais:

(i). Não depende, de forma explícita, da função de distribuição desconhecida;

[13] Nesse caso o conjunto de treinamento **T** pode se tornar pouco informativos para a tarefa de aprendizado, pois o classificador ou regressor induzido pode se super-ajustar a **T**.

(ii). Pode ser minimizada com relação aos parâmetros de aprendizagem (ou vetor peso no caso específico da máquina de aprendizagem ser uma Rede Neural).

3.2.2 Limites no risco esperado

Para garantir a consistência da Minimização Empírica do Risco, a Teoria do Aprendizado Estatístico mostra que é imperativo restringir o espaço de funções admissíveis onde f é escolhida para que a capacidade seja adequada à quantidade de dados de treinamento disponíveis [53]. Em aprendizado de máquina, essa questão é levada em conta por meio da complexidade (ou capacidade) da classe de funções que o algoritmo de aprendizado é capaz de obter [68][14]. Vapnik [71] propõe um limite ao desvio do risco empírico em relação ao risco esperado, que é dado pela soma do risco empírico e um termo de capacidade (h) que pode ser garantido com probabilidade $(1-\eta)$, onde $\eta \in [0,1]$:

$$R(f) \leq R_{emp}(f) + \underbrace{\sqrt{\frac{h\left(\ln(\frac{2T}{h}+1\right) - ln\left(\frac{\eta}{4}\right)}{T}}}_{Intervalo\ de\ confiança\ h} \qquad (3\text{-}5)$$

Onde h é a dimensão Vapnik Chernovenkis (VC) da classe de funções $\boldsymbol{\mathcal{F}}$, e onde T é o número de exemplos de treinamento. À medida que a razão t/h cresce, o termo de capacidade diminui e o risco esperado (erro de teste) se aproxima do risco empírico (erro de treinamento).

Em outras palavras, um algoritmo de aprendizagem será consistente (será capaz de generalizar) se, e somente se a função f é oriunda de uma classe de funções $\boldsymbol{\mathcal{F}}$ com dimensão VC finita [68].

3.2.3 Dimensão Vapnik Chernovenkis

A dimensão de Vapnik Chernovenkis (h) é uma das mais importantes medidas de complexidade de funções. Quanto maior o valor de h, mais complexas são as funções de classificação ou regressão que podem ser induzidas a partir de

[14] De maneira mais simples, Erro de teste = Erro de treino + Complexidade do conjunto de funções.

$\boldsymbol{\mathcal{F}}$. Para exemplificar esse conceito, a dimensão VC do conjunto de funções lineares no $\Re^2$, é três, uma vez que existe (pelo menos) uma configuração de três pontos nesse espaço que pode ser particionada por retas em todas as $2^3 = 8$ combinações binárias de rótulos.

A dimensão VC de uma classe de funções $\mathcal{F}$ é definida pelo número máximo de pontos que podem ser classificados de todas as maneiras possíveis por $\boldsymbol{\mathcal{F}}$. De forma genérica, para funções lineares no $\Re^n$, e para n ≥ 2, a dimensão VC (h), é dada pela expressão: $h = n + 1$.

De maneira mais formal, diz-se que uma amostra Z_n de tamanho n é quebrada por uma classe de funções $\boldsymbol{\mathcal{F}}$, se tal classe pode realizar qualquer classificação numa dada amostra, ou seja, a cardinalidade de $\mathcal{F}_{Z_{2n}} = 2n$ [68]. Então, a dimensão VC de $\mathcal{F}$ é definida como o maior número n tal que há uma amostra de tamanho n que pode ser quebrada por $\mathcal{F}$ [68].

3.2.4
Princípio da minimização estrutural do risco

Considerando a associação entre risco esperado e risco empírico, proposta por Vapnik (Equação (3-4), e a dimensão VC, a minimização do risco esperado recebe novo viés. Um que é baseado na minimização simultânea de um termo que depende do valor do risco empírico, e de outro que depende da dimensão VC do conjunto de funções [65]. Deste modo, o princípio da Minimização Estrutural do Risco (*Structural Risk Minimization*, SRM) [60,69] tem por objetivo encontrar uma função que minimize, simultaneamente, o risco empírico e a dimensão VC [68].

Sewell [72] descreve o SRM da seguinte forma:

i. Com base no conhecimento prévio do problema escolha alguma classe de funções $\boldsymbol{\mathcal{F}}$;
ii. Divida $\boldsymbol{\mathcal{F}}$ numa hierarquia de subconjuntos combinados em aumento crescente de complexidade $\boldsymbol{\mathcal{F}}_1 \in \boldsymbol{\mathcal{F}}_2 \in \cdots \boldsymbol{\mathcal{F}}_k$ com dimensões VC não-descrescentes ($h_1 \leq h_2 \leq \cdots h_k$);
iii. Para cada subconjunto $\boldsymbol{\mathcal{F}}_i$ encontre a função f_i que minimize o risco empírico; e

iv. Selecione a função (ou modelo) em que a soma do risco empírico e o termo que mede a complexidade da classe de funções seja mínima.

O SRM consiste em encontrar o subconjunto de funções que minimiza o limite sobre o risco esperado. Dessa forma, esse princípio garante, mensura, e controla a capacidade de generalização do algoritmo de aprendizagem.

Na Figura 3.1 fica evidente que quando a máquina tem uma capacidade grande (dimensão VC grande), ela apresenta um risco empírico baixo, mas não generaliza bem, pois o intervalo de confiança VC é grande. Com o uso do limite sobre o risco esperado, é possível escolher a função que tenha o menor erro de generalização [68].

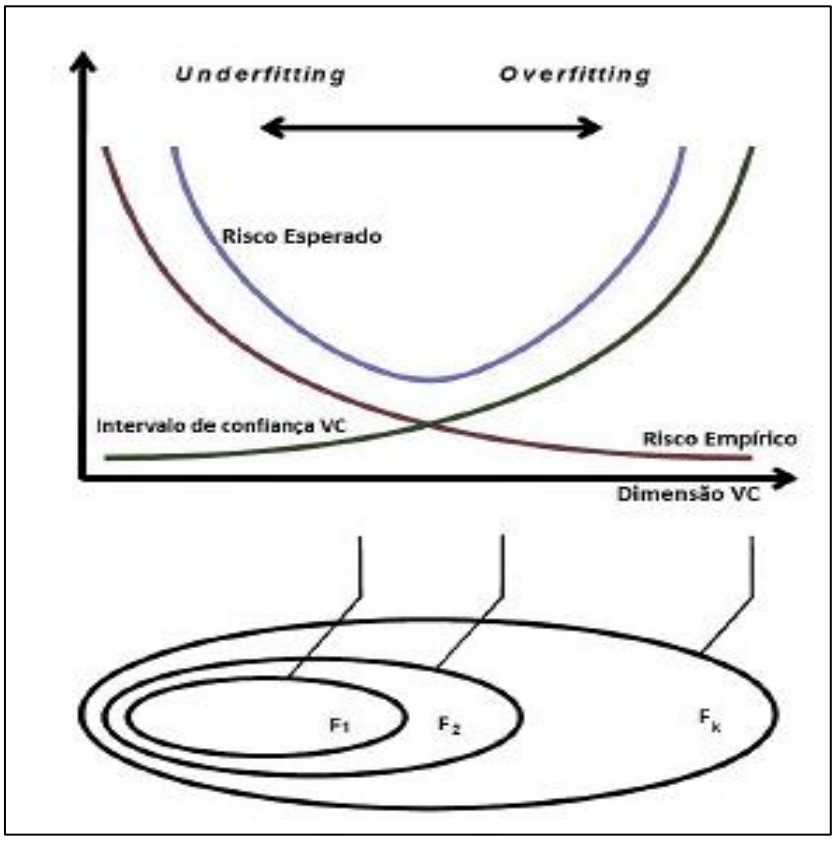

Figura 3.1 – Limite do risco esperado de uma máquina de aprendizado. Fonte: adaptado de [73].

3.3 Formulação matemática da SVR linear

Dado um conjunto de t amostras de dados temporais de treinamento $\mathbf{T} = \{(x_1, y_1), \ldots, (x_t, y_t)\} \subset \Re^n \times \Re$ em que $x \in \mathcal{X}$ é o vetor de entrada (*input*) e $y \in \Re$, o escalar de saída (*target*), o objetivo do SVR é encontrar um preditor

$f(x)$ para todo x ∈ $\mathcal{X}$, i.e., que aproxime o escalar y a menos de um erro de previsão ε previamente especificado [64]:

$$f(x) = w^T \Phi(\mathrm{x}) + \mathrm{b}, \text{com } \phi: \Re^n \to \boldsymbol{\mathcal{F}}, w \in \boldsymbol{\mathcal{F}} \quad (3\text{-}6)$$

Onde w é o vetor de pesos, b o termo de viés ou *bias* e $\phi(\mathrm{x})$ é a função de mapeamento, que projeta os vetores de entrada x no espaço característico de dimensão elevada $\boldsymbol{\mathcal{F}}$, onde a regressão linear está definida. Quanto maior a dimensão, maior é acurácia do SVR na aproximação da função.

Para estimar a regressão, é necessário mensurar a diferença entre os valores reais e as respectivas previsões por meio da função de perda ε-insensível linear, L_ε, proposta por [64]. O ε-SVR busca estimar $f(x)$ de modo que ela seja o mais suave possível e com erros menores que ε no espaço característico. Assim, a norma Euclideana do vetor de pesos $\|\mathrm{w}\|^2$ deve ser minimizada ao mesmo tempo em que se controla o erro referente às restrições de L_ε. Então, tem-se o seguinte problema de otimização convexa [64,60]:

$$Minimize \left\{ \frac{1}{2}\|\mathrm{w}\|^2 + \frac{C}{t}\sum_{i=1}^{t} \big(L_\varepsilon(f(x), y)\big); \right. \quad (3\text{-}7)$$

Em que $L_\varepsilon(f(x), y)$ é a função de perda ε-insensível:

$$L_\varepsilon(f(x), y) = \begin{cases} |\mathrm{y_i} - \mathrm{h(x)}| - \varepsilon, \text{se } |\mathrm{y_i} - \mathrm{h(x)}| > \varepsilon \\ 0, \quad \text{caso contrário} \end{cases} \quad (3\text{-}8)$$

É importante destacar que a variação de ε influencia o número de suportes vetoriais e, por conseguinte, afeta a complexidade do modelo [74]. Ao se considerar a tolerância $\varepsilon \geq 0$, pretende-se que a imagem de todos os pontos da amostra estejam a uma distância de no máximo ε da função a ser aproximada pelo preditor. Permite-se, no entanto, que o preditor esteja fora dessa região através das folgas ξ_i e ξ_i^*. As folgas aparecem também na função objetivo do problema (Equação (3-9), para que, ao resolvê-lo, sejam as menores possíveis. Obtém-se, então, o seguinte problema de otimização quadrática restrita para o SVR em sua forma primal, estabelecido por Vapnik [64]:

$$Minimize \left\{ \frac{1}{2}\|\mathrm{w}\|^2 + \mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{t}} (\xi_\mathrm{i} + \xi_\mathrm{i}^*) \right. \quad (3\text{-}9)$$

$$Sujeito\ a \begin{cases} y_i - w^T \Phi(x) - b \leq \varepsilon + \xi_i \\ w^T \Phi(x) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i\ , \xi_i^* \geq 0 \end{cases}$$

A constante C > 0 pondera o compromisso entre a complexidade do modelo e a quantidade de desvios maiores que ε admitidos [60]. Além disso, a programação quadrática convexa e as restrições lineares do problema primal da Equação (3-9) garantem que o SVR sempre terá solução única global ótima. Contudo, os parâmetros C e ε do ε-SVR necessitam ser adequadamente selecionados, a fim de garantir a capacidade de generalização do modelo SVR

3.3.1 Formulação dual

A formulação dual fornece o meio de estender a SVR para funções não lineares, através do uso de funções núcleo (*kernel*). Nesta formulação, introduz-se um conjunto de variáveis dual e multiplicadores de Lagrange $(\alpha, \alpha^*, \eta, \eta^*)$ na função de perda do SVR [64,60]:

$$\mathcal{L}(w, b, \xi_i^*, \xi_i\ ) \coloneqq \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{t}(\xi_i\ + \xi_i^*) - \sum_{i=1}^{t}(\eta_i\ \xi_i\ + \eta_i^* \xi_i^*) - \sum_{i=1}^{t}\alpha_i\ (\varepsilon + \xi_i\ - y_i - \langle w, \Phi(x)\rangle + b) - \sum_{i}^{t}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, \Phi(x)\rangle - b) \quad (3\text{-}10)$$

Onde que $\mathcal{L}$ é a função lagrangeana e $\alpha, \alpha^*, \eta\ e\ \eta^*$ os multiplicadores de Lagrange. Segundo Mangasarian [75], a função acima tem um ponto de sela em relação às variáveis primal e dual. Assim, derivando $\mathcal{L}$ em relação às variáveis de decisão w, b, $\xi_i\ e\ \xi_i^*$, é possível satisfazer a condição do ponto de sela:

$$\partial_b L = \sum_{i=1}^{t}\left(\alpha_i^* - \alpha_i\ \right) = 0 \quad (3\text{-}11)$$

$$\partial_w L = w - \sum_{i=1}^{t}(\alpha_i\ - \alpha_i^*)\,\Phi x_i = 0 \quad (3\text{-}12)$$

$$\partial_{\xi_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \quad (3\text{-}13)$$

Substituindo w e as variáveis duais η_i e η_i^* na Equação (3-10, obtém-se o problema de programação matemática na forma dual:

$$Maximize \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{t}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\Phi x_i . \Phi x_j) \\ -\varepsilon\sum_{i=1}^{t}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{t} y_i(\alpha_i - \alpha_i^*) \end{cases} \quad (3\text{-}14)$$

$$Sujeito\ a \begin{cases} \sum_{i=1}^{t}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

Como o problema de otimização não linear da Equação (3-14 tem restrições de desigualdade, as condições de Karush-Kunh-Tucker (KKT) [76,77] devem ser satisfeitas. Essas condições implicam que apenas as observações x_i, tais que $\alpha_i, \alpha_i^* \neq 0$, são os chamados vetores suporte, e que apenas elas e são usadas para estimar a função de decisão [60]. Deste modo, os vetores suporte vão corresponder às amostras para as quais um dos multiplicadores de Lagrange é diferente de zero, ou seja, a amostra fora do tubo ε-insensitiva.

Desta forma, a Equação (3-12) pode ser reescrita como $w = \sum_{i=1}^{t}(\alpha_i + \alpha_i^*)\Phi(x_i)$, e então:

$$f(x) = \sum_{i,j=1}^{t}(\alpha_i - \alpha_i^*)\langle\Phi(x_i), \Phi(x_j)\rangle + b \quad (3\text{-}15)$$

$$b = y_i - \sum_{i,j=1}^{t}(\alpha_i - \alpha_i^*)\langle\Phi(x_i), \Phi(x_j)\rangle + \varepsilon \quad (3\text{-}16)$$

Onde α_i, α_i^* são os multiplicadores de Lagrange otimizados e $\langle\Phi(x_i), \Phi(x_j)\rangle$ é o produto interno dos vetores no espaço característico. Quando $\varepsilon = 0$, obtém-se uma função de perda de Laplace e, e o problema de otimização é simplificado [61] [60].

3.4 Mapeamento não linear e funções *kernel*

O modelo para SVR demonstrado até então não leva em consideração possível não linearidade da amostra de treinamento, fenômeno que ocorre com frequência. Deste modo, os métodos SVR são estendidos para o problema de regressão não linear, com a introdução de funções núcleo ou *kernel* (K). Como o procedimento de derivação da função de decisão com K é muito similar ao caso linear, mostra-se apenas a forma final:

$$f(x) = \sum_{i=1}^{t} (\alpha_i - \alpha_i^*) K\langle x_i, x_j \rangle + b \quad (3\text{-}17)$$

$$b = \sum_{i=1}^{t} (\alpha_i - \alpha_i^*) K\langle x_i, x_j \rangle + \varepsilon \quad (3\text{-}18)$$

A função *kernel*, $K\langle x_i, x_j \rangle$, recebe dois dados de entrada x_i e x_j e calcula o produto interno destes dados no espaço de características. O *kernel trick* consiste na transformação (mapeamento) de dados não separáveis linearmente no espaço de entrada em linearmente separáveis no espaço característico:

$$K\langle x_i, x_j \rangle = \langle \Phi(x_i), \Phi(x_j) \rangle \quad (3\text{-}19)$$

As funções Φ devem pertencer a um domínio em que seja possível o cálculo de produtos internos. Para que um *kernel* seja válido, ele deve obedecer às condição de Mercer [60,78,79]. De forma simplificada, um *kernel* que satisfaz as condições de Mercer é caracterizado por originar matrizes positivas semi-definidas K, em que cada elemento $K_{i,j}$ é definido por $K_{i,j} = K\langle x_i, x_j \rangle$, para todo $i, j = 1, \ldots, n$ [80].

O uso de diferentes funções *kernel* possibilita a construção de máquinas de aprendizado com diferentes tipos de superfícies de decisão não-linear no espaço de entrada. Alguns dos *kernels* mais utilizados são os Polinomiais, os Gaussianos ou Funções de Base Radial (*Radial Basis Function*, *RBF*) e os Sigmoidais. Cada um deles apresenta parâmetros que devem ser determinados pelo usuário. O *kernel* Sigmoidal, em particular, satisfaz as condições de Mercer apenas para alguns valores de β_0 *e* β_1. Os *kernel* Polinomiais com $d = 1$ também são denominados lineares. A Tabela 3.1 mostra essas funções e os parâmetros determinados a priori pelo usuário.

Tabela 3.1 – Funções *Kernel* mais utilizadas em SVR

Tipo de *Kernel*	Função $K\langle x_i, x_j \rangle$	Parâmetros
Polinomial	$(x_i^T x_j + 1)^d$	d
Gaussiano ou RBF	$exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$	σ
Sigmoid	$\tanh(\beta_0 x_i^T x_j + \beta_1)$	β_0, β_1

Para a análise de séries temporais, a escolha do *kernel* ótimo é crucial para a qualidade da modelagem. Cada *kernel* modela diferentes hipóteses no processo gerador da série temporal. Segundo Cristianini e Shawe-Taylor [81], o problema de escolher uma arquitetura para uma aplicação de rede neural é substituído pela escolha do *kernel* adequado para SVR. Não obstante, até o momento não há nenhum método para escolha da função núcleo mais apropriada para tarefa de previsão [82].

3.5 Determinação dos parâmetros da SVR

A determinação dos parâmetros é uma etapa crucial nos modelos SVR [83,14,84]. Dependendo da escolha de parâmetros, a acurácia das previsões pode ser afetada.

A definição dos parâmetros não é uma tarefa fácil [85]. Uma das abordagens mais básicas para ajuste de parâmetros é aquela baseada em tentativa e erro. Esta, no entanto, além de poder consumir muito tempo, não realiza uma busca eficiente no espaço paramétrico [83].

Em geral, os parâmetros do SVR são obtidos por meio de *grid search*, um método de busca exaustiva. Contudo, o *grid search* apresenta os seguintes problemas: esbarra em ótimos locais, é demorado, requer informação a priori e não é capaz de otimizar concomitantemente o *kernel* e os parâmetros do SVR.

Nos últimos anos, técnicas computacionais vêm sendo utilizadas para determinar os parâmetros do SVR, destacando-se o uso de metaheurísticas evolucionárias como a Otimização por Enxame de Partículas (*Particle Swarm Optimization*, PSO) [84,12,13,14] e os Algoritmos Genéticos (*Genetic Algorithms*, GA) [13,15,16,17].

3.6 Aproximação por regressão e previsão de séries temporais

Uma forma de expressar o problema de previsão de séries temporais em uma estrutura de estimação por regressão é considerar as séries temporais como sistemas dinâmicos. Seja uma série temporal y_t, se $N \in \aleph$ e $h > 0$ forem

escolhidos adequadamente, então podemos prever y_t a partir de $(y(t-h), \ldots, y(t-N_h) \in \Re^n$. Portanto, pode-se considerar um problema de regressão onde o conjunto de treinamento consiste de entradas $(y(t-h), \ldots, y(t-N_h)$e saídas y_t para diferentes valores de t. Porém, conforme destacado por Schölkopf e Smola [79], várias características de previsão de séries temporais tornam o problema difícil para esta aproximação do vetor suporte. Dentre elas:

i. Series temporais são frequentemente não estacionárias, i.e., a distribuição de séries temporais muda com o tempo e, como consequência, exemplos de treinamento que são gerados como descrito acima se tornam menos úteis, se eles são tomados de um passado distante.
ii. Os diferentes exemplos de treinamento não são independentes e identicamente distribuídos (iid), o que é uma das hipóteses assumidas por modelos de aprendizagem estatística, como o SVR.

O SVR foi construído sob a hipótese de que os dados do conjunto de treinamento T são independentes e identicamente distribuídos [86]. No caso de séries temporais, essa hipótese é violada. Não obstante, Fender [87] demonstra que a maioria dos teoremas centrais envolvidos na minimização do risco estrutural continua válida para dados que possuem uma estrutura de dependência fraca [86].

Em que pese estas restrições, excelentes resultados têm sido obtidos utilizando SVR em problemas de previsão de séries temporais em diversas áreas como: biologia, química, meteorologia, medicina, contabilidade [88], economia e finanças [67,89,28,90,91]. Alguns estudos, inclusive, demonstraram que o SVR apresenta resultados empíricos superiores aos resultados de modelos estatísticos e econométricos tradicionais [28,92].

O recurso do SVR para previsão pode ser justificado pela sua capacidade de prever com precisão os valores futuros de uma série temporal, mesmo quando esta é não estacionária [92]. Enquanto as redes neurais artificiais convencionais (*Artificial Neural Networks*, ANN) possuem limitações na aprendizagem de padrões, quando os dados apresentavam dimensionalidade complexa e ruído elevado [67], o SVR apresenta bons resultados mesmo com quantidade de dados disponíveis reduzida [89].

Embora o SVR possua vantagens, algumas limitações também podem ser listadas, tais como a dificuldade na escolha de parâmetros e a baixa velocidade do treinamento, que é menor em relação a outras técnicas, como as ANN.

4
Algoritmos genéticos

4.1
Introdução

Algoritmos genéticos (*Genetic Algorithms*, GAs) pertencem aos algoritmos evolutivos de *Machine Learning* [93]. Os GAs são técnicas metaheurísticas não determinísticas de busca amplamente utilizadas em problemas de otimização complexos. Eles são particularmente interessantes em situações nas quais o número de parâmetros é muito grande e as soluções são muito difíceis, ou impossíveis, de serem obtidas analiticamente.

Os GAs manipulam um espaço de soluções potenciais utilizando mecanismos inspirados nas teorias de seleção natural de Darwin e na genética de Mendel, tais como, herança genética, mutação, seleção e recombinação (*crossover*). O princípio do algoritmo foi introduzido por Holland [94] em 1975 e posteriormente popularizado por Goldberg [95] para problemas de otimização.

Os tradicionais GAs, denominados GAs canônicos, baseiam-se em noções do *schema theorem* [96] e *building blocks* [96] e têm seus indivíduos representados por vetores binários [94]. Esta representação, entretanto, não é universalmente aceita na literatura. Alguns pesquisadores indicam a representação real (i.e., de ponto flutuante) em aplicações que necessitam de tratamento de valores contínuos. Sendo a representação binária mais apropriada em aplicações que requeiram o tratamento de valores discretos [97,98].

A evolução do GA normalmente inicia-se a partir de uma população P de indivíduos (cromossomos) de tamanho n. Então, para o instante $k = 0$, $P = \left\{P_1^{(0)}, P_2^{(0)}, \dots, P_n^{(0)}\right\}$. Cada membro da população é avaliado através de uma função de adequação (*fitness*), $f(P_i^{(k)})$, que nos casos mais simples é a própria função que se quer maximizar ou minimizar. Os indivíduos mais aptos são selecionados com probabilidade $p_i^{(k)}$, recombinados e mutados através de operadores de *crossover* e mutação de modo a formar uma nova população

$P = \left\{P_1^{(k+1)}, P_2^{(k+1)}, \ldots, P_n^{(k+1)}\right\}$. Em seguida, defina $k = k + 1$ e o algoritmo retorna para o passo de avaliação de adequação. Quando os critérios de convergência forem cumpridos, a evolução termina, e o algoritmo fornece $P^* \equiv \arg\max_{P_i^{(k)}} f(P_i^{(k)})$ como solução ótima.

Os procedimentos de execução de um GA canônico podem ser sintetizados pelo pseudocódigo de seu ciclo evolutivo, conforme apresentado no Quadro 4.1 abaixo.

Quadro 4.1 – Procedimento de execução de uma algoritmo GA canônico

```
GA ( )
{Algoritmo genético para otimização}
início
    k = 0;
    Inicializar população (P_i^(k));
    Avaliar P_i^(k);
    enquanto não (condição de término) faça
        k = k + 1;
        selecione P_i^(k)_i de P; {operador de reprodução}
        recombine P_i^(k); {operadores de crossover e mutação}
        avalie P_i;;
    fim enquanto
fim
```

4.2 Implementação de um GA

Usualmente, a implementação de um algoritmo genético para solução de um problema deve possuir os seguintes componentes [99]:

- **Inicialização da população:** a população inicial de parâmetros do algoritmo compreende P soluções. Cada uma das soluções consiste de vetores $P_i \in \{0,1\}$ (representação canônica) ou $P_i \in \Re^n$ (representação real). Estes parâmetros são inicializados de acordo com uma distribuição aleatória e uniforme. A população deve ter um tamanho N que não seja tão pequeno que comprometa a efetiva exploração do espaço de busca nem tão grande que afete a eficiência

do método. Muitos pesquisadores sugerem tamanhos de população entre 20 e 100.

- **Função de Avaliação (*Fitness*):** cada solução P_i, $i = [1, P]$ é classificada com relação ao cálculo da função de adequação (*fitness*). Ou seja, avalia-se o grau de adaptação de cada cromossomo da população em relação ao problema, e seleciona-se os indivíduos mais aptos de acordo com a estratégia de seleção;
- **Seleção:** Existem vários métodos de seleção presentes na literatura. Não há, no entanto, consenso em relação a qual método é melhor para determinado tipo de problema; isto ainda é uma questão aberta em GA [100]. A mais conhecida e utilizada forma de se fazer a seleção é por torneio. No torneio, a seleção é feita em função do número de vitórias de cada indivíduo em N competições, contra q oponentes aleatórios da população. Vence uma competição aquele que apresentar o melhor *fitness* (comparado ao de seu(s) oponente(s)). Uma importante propriedade da seleção por torneio é que esta não depende de um conhecimento global da população. Além disso, essa seleção não leva em consideração o *rank* que o indivíduo ocupa na população, permitindo uma seleção com menos tendências [101]. Outros exemplos de formas de seleção são: ranqueamento, *breeder*, *roulette wheel*, *steady-state*, *Boltzmann*, etc.[15];
- **Condições de mutação e *crossover*:** segundo Davis [97], diferentes taxas de mutação (p_m) e *crossover* (p_c) são apropriadas para diferentes momentos do algoritmo: no início é interessante empregar uma alta taxa de crossover, aumentando a taxa de mutação de acordo com a convergência da população. Geralmente a taxa de *crossover* varia entre 0,5 e 0,95, mas muitas vezes este valor é limitado dependendo do tipo de algoritmo. De Jong [102] sugere que a taxa de mutação seja inversamente proporcional ao tamanho da população;

[15] Uma adição aos métodos de seleção é o elitismo. O elitismo força os AGs a reter certo número de "melhores" indivíduos em cada geração. Tais indivíduos podem ser perdidos se não forem selecionados para reprodução ou se forem destruídos por *crossover* ou mutação. A utilização do elitismo reduz o tempo de convergência dos AGs, mas pode fazê-los convergir para um ótimo local.

- **Condição de término:** como os GAs são métodos de busca estocástica, é preciso impor limites para que a execução não seja infinita. As abordagens mais comuns são: o limite por tempo, o limite pela quantidade de avaliações de soluções (número máximo de gerações) e o limite por alguma propriedade atingida pela população.

De Jong [102] sugere para um bom desempenho do GA, a seguinte configuração de parâmetros: N=50, p_c=0,6 e p_m=0.001.

4.3 Operadores genéticos

O objetivo dos operadores genéticos é transformar a população através de sucessivas gerações, buscando melhorar a aptidão dos indivíduos. Os operadores genéticos são necessários para que a população se diversifique e mantenha as características de adaptação adquiridas pelas gerações anteriores. Neste trabalho serão apresentados os operadores mais difundidos na literatura do AG, o *crossover* (ou recombinação) e a mutação.

O operador de mutação é responsável pela exploração global do espaço de busca introduzindo novo material genético em indivíduos já existentes, sem destruir o progresso já obtido com a busca. No caso de problemas com codificação real, os operadores de mutação mais populares são a mutação uniforme e a mutação Gaussiana [103,104]. Outro operador de mutação bastante utilizado em problemas de otimização com restrições é o operador de mutação não uniforme. Na mutação não uniforme, um indivíduo P_m é selecionado para a mutação, resultando num vetor $P' = (P_1,..., P_m',..., P_n)$, tal que:

$$P_m' = \begin{cases} P_m + \Delta(k, a - P_m), com\ 50\%\ de\ probabilidade \\ P_m - \Delta(k, P_m - b), com\ 50\%\ de\ probabilidade \end{cases} \quad (4\text{-}1)$$

Onde a e b são os valores mínimo e máximo dos limites de valores do indivíduo P_m', respectivamente. A função Δ(k,y) retorna um valor no intervalo [0,y] tal que a probabilidade de Δ(k,y) inicia em zero e é incrementada de acordo com o número de gerações k. Michalewicz [98] propõe a seguinte função:

$$\Delta(k, y) = y.\left[1 - r^{\left(1-\frac{k}{K}\right)^{I}}\right] \quad (4\text{-}2)$$

Onde r é um número aleatório no intervalo [0, 1], K é o número máximo de gerações e I é um parâmetro que determina o grau de dependência do número de iterações[16]. Esta propriedade leva o operador a efetuar uma busca uniforme no espaço inicial, quando t é pequeno e, mais localmente nas gerações posteriores.

O operador genético do *crossover* cria novos indivíduos para a população através da recombinação de partes diferentes de dois cromossomos-pai escolhidos através do método de seleção. No *crossover* a partir de dois cromossomos-pai dois novos indivíduos são gerados, os quais são chamados de descendentes. Caso não ocorra o *crossover*, os descendentes serão iguais aos pais, permitindo a preservação de algumas soluções. No *crossover* blend, também chamado de BLX-α, os descentes são uma média ponderada dos pais. Inicialmente é gerado um número β aleatório com distribuição uniforme no intervalo [−α, 1 + α]. Sejam P_1 e P_2 dois indivíduos selecionados para *crossove*r, então os dois descendentes resultantes serão: $P_1^{"} = P_1 + \beta(P_2 - P_1)$ e $P_2^{"} = P_2 + \beta(P_1 - P_2)$. Usualmente $\alpha = 0{,}5$, garantido a diversidade da população inicial.

Em Deb *et al*. [104] são descritos outros operadores de *crossover* e mutação para a codificação real, tais como: Crossove*r* Linear; *Crossover* Aritmético; *Crossover* Binário Simulado; Operador de *Crossover* Fuzzy; *Crossover* Simplex; *Crossover* Baseado em Conectivos Fuzzy; entre outros; Mutação Aleatória; Mutação Polinomial; entre outras.

Mais detalhes sobre GA, i.e., outras configurações possíveis do algoritmo e operadores genéticos, podem ser encontrados em Goldberg [95], Davis [97], Back *et al.* [103,105], Yu e Gen [106] e Eiben e Smith [101].

4.4
GA e métodos de otimização tradicionais

Os GAs são robustos e eficientes em espaços de procura irregulares, multidimensionais e complexos, e segundo Goldberg [95], se diferenciam de outros métodos numéricos de otimização em quatro aspectos fundamentais:

[16] Valor proposto por Michalewicz [98]: I = 5.

i. Os GAs trabalham com codificação de parâmetros, ao invés dos pârametros originais do problema. São robustos e aplicáveis a uma grande variedade de problemas;

ii. Os GAs pesquisam soluções ótimas a partir de um conjunto de soluções, não a partir de uma única solução. Logo, não ficam presos, necessariamente, a ótimos locais como outros métodos de busca;

iii. Os GAs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar. Os algoritmos genéticos não requerem nenhum conhecimento ou informações de gradiente de uma superficie de resposta; a presença de descontinuidades tem pouco efeito no desempenho geral da otimização;

iv. Os GAs utilizam regras probabilísticas e não determinísticas na pesquisa de novas soluções. O GA realiza uma "busca orientada" pelo espaço de soluções, sem a necessidade de conhecimento prévio acerca da função objetivo. Apresenta bom desempenho para uma grande escala de problemas de otimização.

As principais críticas feitas aos algoritmos genéticos referem-se à incerteza da obtenção da solução ótima e ao grande número de avaliações da função objetivo que se faz necessário para obter a solução[17]. Em que pese essas restrições, os GAs constituem métodos competitivos em espaços de busca complexos [95,107,108,109,110,111,112,113,114,115]. Ainda que o algoritmo demande grande esforço computacional (de cálculo e de memória), é aceitável o tempo necessário para que a população convirja e sejam obtidas boas aproximações da solução ideal.

4.5
GA para otimização global de parâmetros da SVR

Os parâmetros C e ε, a escolha do *kernel e* os parâmetros do *kernel* afetam o desempenho de previsão da SVR, pois determinam a complexidade do modelo. No entanto, a correta seleção destes parâmetros da SVR não é tarefa trivial. Os GAs surgem como alternativa para solucionar esse problema. Algoritmos

[17] O trabalho de Chellapilla e Hoorfar**Fonte bibliográfica inválida especificada.** pode ser citado como exemplo de crítica aos GAs.

genéticos tem forte capacidade de pesquisa global [116]. Assim, esses algoritmos podem ser utilizados para encontrar as melhores combinações de parâmetros em SVR. O algoritmo genético implementado por Huang [114], por exemplo, otimiza os parâmetros C do SVM e σ^2 do *kernel* RBF (*Radial Basis Function*). Já o algoritmo implementado por Wu [116] utiliza um algoritmo genético para otimizar simultaneamente os parâmetros do SVR, os parâmetros do *kernel* e a escolha do *kernel.*

Nesta dissertação, propõe-se, para um problema de previsão com SVR, um GA-SVR com otimização simultânea (i) de C e ε (ii) dos parâmetros do *kernel* (φ), (iii) da escolha do *kernel* (K) e (iv) adicionalmente de $x_t = [x_t, \dots, x_{t-d}]^T$, onde X_t é um vetor de entradas no tempo t com d variáveis desfasadas. Em (iv), o GA procura o melhor valor de d.

A Figura 4.1 ilustra os processos de otimização de parâmetros e previsão com o GA-SVR. Algumas considerações sobre o procedimento ilustrado na Figura 4.1:

- A base de dados é dividida em treino, validação e teste, na proporção especificada pelo usuário;

A população inicial, com N candidatos, é escolhida aleatoriamente, considerando os limites definidos pelos intervalos dos parâmetros a serem ajustados. A população é composta por indivíduos (cromossomos), onde cada um destes indivíduos representa um possível modelo para previsão de séries temporais: os parâmetros da SVR, tipo de *kernel*, parâmetros do *kernel* e número de *lags* de entrada. O melhor cromossomo fornece a previsão mais acurada de acordo com a função de avaliação escolhida;

- Em cada geração, dois cromossomos na população serão selecionados para passar por uma operação de crossover pelo método proporcional ao *fitness*. O cromossomo com menor valor de *fitness* deve, portanto, ter maior chance de ser selecionado, uma vez que o objetivo é minimizar o erro de previsão da SVR.
- Após o processo de seleção, dois cromossomos (pais) são combinados para gerar novos cromossomos (filhos) por operações genéticas com elitismo. São aplicados os operadores de *crossover* e mutação com uma probabilidade pré-definida, p_c e p_m, respectivamente. Devido ao

elitismo, o melhor cromossomo em cada geração substitui o pior cromossomo da próxima geração com probabilidade p_e.

- Se o critério de convergência for atingido, o cromossomo ótimo encontrado pelo GA é utilizado para obter a previsão. O resultado é avaliado no conjunto de teste.

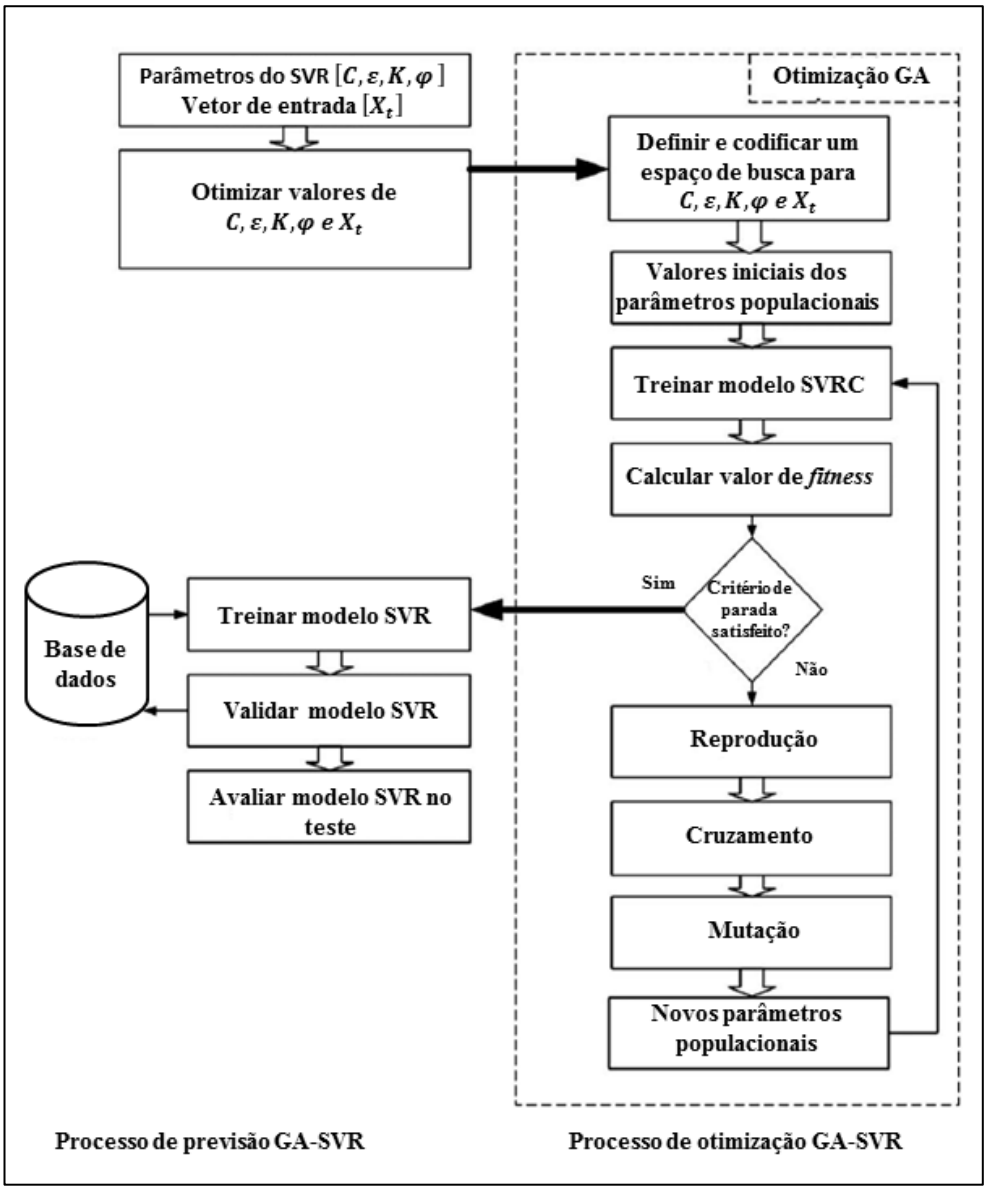

Figura 4.1 – Processo de otimização e previsão do GA-SVR

5
Modelagem empírica

Séries temporais são conjuntos de observações tomadas sequencialmente ao longo do tempo [117]. Séries temporais do mundo real geralmente apresentam características não-lineares e caóticas, de modo que não é o mais adequado o uso de modelos estatísticos clássicos para sua previsão. Sabendo disso, são desenvolvidos modelos baseados em técnicas de *machine learning* para a previsão de séries de prêmio de seguros, que apresentam características não-lineares.

Neste trabalho, são adotadas duas abordagens para a previsão de séries de prêmios de seguros. Uma abordagem é baseada em modelagem univariada, e a outra é baseada em modelagem multivariada. A primeira utiliza as informações da própria série de interesse para explicar seu comportamento futuro. A outra, diferentemente, utiliza informações não apenas da própria série, mas também de outras séries temporais, para explicar o comportamento futuro da série de interesse.

Para se prever as séries de prêmios de seguros, são desenvolvidos modelos univariados e multivariados a partir da rede *echo-state* (ESN) e a partir da hibridização de algoritmos genéticos e regressão por vetores suporte (GA-SVR). A seguir, são detalhados os modelos desenvolvidos.

5.1
Modelos propostos

5.1.1
Modelos univariado e multivariado ESN

Neste trabalho, são desenvolvidos um modelo univariado e um modelo multivariado de rede *echo-state* (ESN) para previsão de séries de prêmio de seguros. O modelo univariado ESN desenvolvido tem a estrutura exposta na Figura 5.1. As variáveis de entradas são variáveis defasadas em n da variável de interesse y_T (esta inclusa), que representam o valor da série no passado. As variáveis de saída são aquelas que representam o valor da série em até h passos à

frente, onde h é o horizonte da previsão. A seleção do número de defasagens é feita avaliando-se o modelo num conjunto de validação. São testadas três configurações de defasagens (3, 6 e 12).

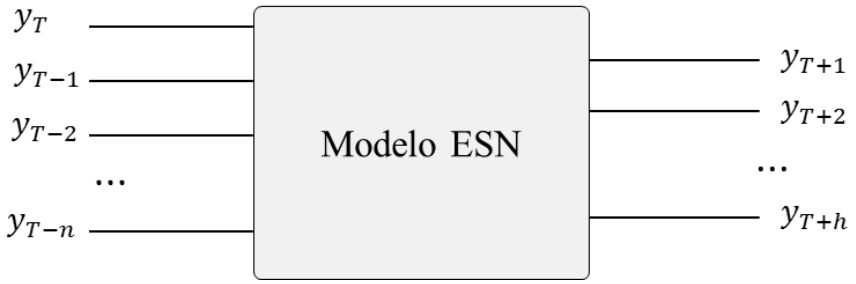

Figura 5.1 – Estrutura do modelo univariado ESN

O modelo multivariado ESN tem a estrutura ilustrada na Figura 5.2. As variáveis de entrada são variáveis defasadas das variáveis de interesse y_T e da variável explicativa $x_{1,T}$, que representa o Produto Interno Bruto (PIB) *per capita*, e da variável explicativa $x_{2,T}$, que representa, por sua vez, o índice de Inflação IPCA. As variáveis de saída são aquelas que representam o valor da série em até h passos à frente. A seleção das defasagens das variáveis de entrada é feita da mesma forma que para o modelo univariado.

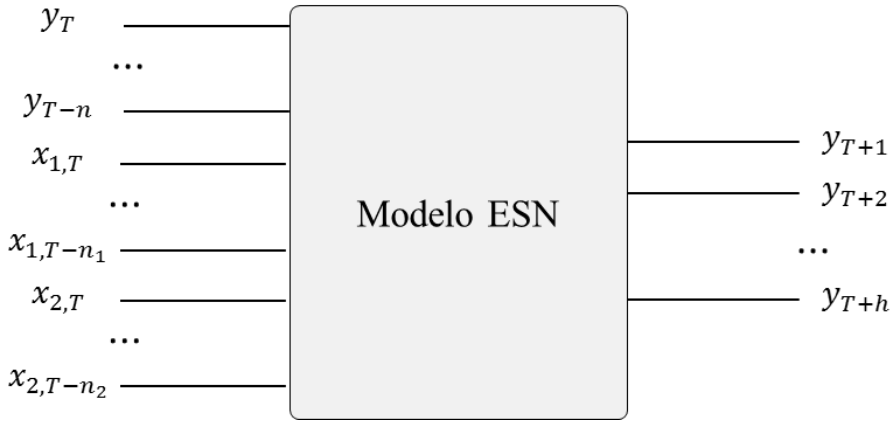

Figura 5.2 – Estrutura do modelo multivariado ESN

Os parâmetros da rede ESN são selecionados a partir de busca intensiva no espaço paramétrico apresentado na Tabela 5.1. O uso de busca intensiva foi

adotado para garantir o princípio de simplicidade da ESN, concebido por Jaeger [30].

Tabela 5.1 – Configuração dos Parâmetros da ESN

Parâmetros	Valores
Número de Unidades no Reservatório	20; 60; 90
Raio Espectral	0,2; 0,6; 0,8
Percentual de Conexões no Reservatório	0,2; 0,6; 0,8
Janela de Previsão (defasagens)	3; 6; 12
Função de Entrada	Tangente Hiperbólica
Função de Saída	Linear

Devido à utilização da função tangente hiperbólica na camada de entrada, os dados de entrada foram escalonados no intervalo ente -1 e 1.

5.1.2 Modelos univariado e multivariado GA-SVR

Nesta dissertação, o GA-SVR foi utilizado para previsão univariada e multivariada de séries de prêmio de seguros.

A estrutura do modelo univariado GA-SVR é ilustrada na Figura 5.3. As variáveis de entrada são defasagens da própria variável de interesse y_T, onde y_{T-i_j} pode ser qualquer defasagem de y_T, e i_1, i_2,..., i_n pode ser qualquer valor entre 1 e *n*. A saída do modelo é a previsão um passo à frente. A previsão para o horizonte *h* é obtida de maneira *multi-step*.

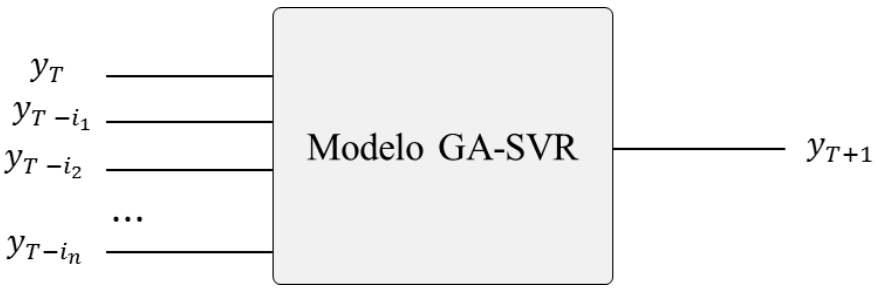

Figura 5.3 – Estrutura do modelo univariado GA-SVR

A estrutura do modelo multivariado GA-SVR é ilustrada na Figura 5.4. As variáveis de entrada são defasagens da própria variável de interesse, e das variáveis explicativas. Assim como no modelo univariado GA-SVR, a saída do modelo é a previsão um passo à frente. A previsão para o horizonte desejado também é obtida de maneira *multi-step*.

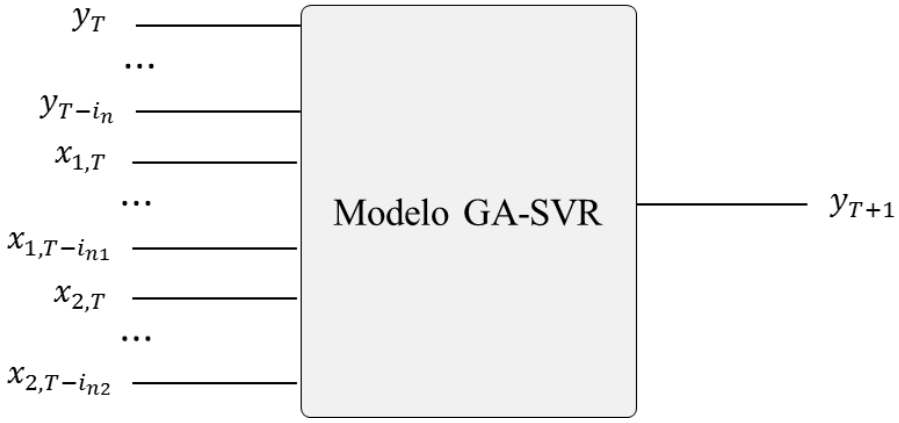

Figura 5.4 – Estrutura do modelo multivariado GA-SVR

A seleção das variáveis de entrada é feita com algoritmos genéticos. Estes são utilizados para otimização paramétrica simultânea do ε-SVR (C e ε), da determinação do tipo apropriado de função *Kernel*, dos valores ideais dos parâmetros do *Kernel* e do número ótimo de defasagens das séries temporais a serem previstas pelo SVR. Esta proposta constitui um diferencial deste trabalho, em relação a trabalhos correlatos na literatura.

A função de aptidão a ser otimizada pelo GA é o erro quadrático médico (*Mean Squared Error*, MSE)[18]. O objetivo do GA é buscar o menor valor de erro no conjunto de validação a partir das configurações dos parâmetros do SVR e da escolha do numero de defasagens. As Tabelas 5.2 e 5.3 a seguir detalham as configurações de GA utilizadas no modelo, e a gama de valores adotados para a busca da m/elhor solução. Foram utilizados os seguintes operadores genéticos de *crossover*, mutação e seleção: *crossover blend*, mutação não uniforme e seleção por torneio.

[18] $\text{MSE} = \text{média}(|e_{it}|)$, onde e_{it} é a diferença entre o vlor real e o valor previsto pelo GA − SVR.

Tabela 5.2 – Configuração dos Parâmetros do GA

Algoritmo Genético	
Tamanho da População	50
Número de Gerações	50
Função de Aptdão	MSE
Crossover (%)	0.85
Mutação (%)	0.02
Elitismo	Sim

Tabela 5.3 – Elementos a serem otimizados pelo GA

	Parâmetros	Valores da Codificação
SVR	C	0.1-10
	ε	0.001-5
Kernel	σ	0.1-10
	d	1-10
	β	1-10
Defasagem	-	1-13
Tipo de Kernel	Linear	1
	RBF	2
	Polinomial	3

5.2 Seleção de modelos computacionais via validação *holdout*

Em um problema de aprendizado, deseja-se encontrar um algoritmo que capture as principais características da amostra de treinamento, e que também seja capaz de prever de forma acurada os dados do processo que são desconhecidos pela máquina (generalização). Quando um modelo perde a capacidade de generalizar, ocorre um fenômeno denominado de sobreajuste (*overfitting*). No *overfitting* a complexidade da função $f(x)$ obtida é superior à necessidade do problema. Inversamente, quando a complexidade de $f(x)$ é inferior à necessidade do problema ocorre o chamado subajuste (*underfitting*) [118].

O desempenho na generalização fornece uma medida da qualidade da previsão do modelo escolhido. Hyndman [119] recomenda o uso de validação

cruzada em séries temporais, sempre que possível. Neste trabalho, utiliza-se a realização mais simples de validação cruzada para seleção do melhor modelo *Echo-State* e GA-SVR, denominada também método *holdout* [120].

No *holdout*, divide-se a base de dados em três conjuntos mutuamente exclusivos: treino, validação e teste [121]. Desse modo, é possível utilizar o conjunto de treinamento para atualização dos parâmetros e selecionar variáveis. A estimação da capacidade de generalização é realizada no conjunto de validação. O conjunto de teste é reservado para avaliar o desempenho de previsão (período fora da amostra). Neste trabalho, optou-se por separar 12 observações da base de dados para o conjunto de validação, 12 para o conjunto de teste e o restante das observações para o conjunto de treino.

5.3
Métricas de avaliação da previsão

Para avaliação dos diferentes métodos de previsão é necessário comparar o desempenho preditivo através de alguma função de perda estatística associada a um erro de previsão, ou seja, associada à diferença entre o valor real (y_i) e o valor previsto pelo modelo i ($\hat{y}_{it}$). Muitas medidas de erro são propostas para avaliar o desempenho dos métodos de previsão. Duas métricas consideradas robustas são baseadas no erro percentual (MAPE) e no erro escalado (MASE), discutidas a seguir:

- Erro Percentual: é dado por $p_i = \frac{e_{it}}{y_i}.100$. Por ser independente da escala é frequentemente utilizado para comparar o desempenho dos métodos de previsão entre diferentes conjuntos de dados. A medida MAPE (*Mean Absolute Percentage Error*) é sugerida por [122] e [123] devido a algumas características desejáveis, tais como ser independente de escala e menos afetado por "*outliers*" da previsão:

$$\text{MAPE} = \text{média}(|p_i|) \tag{5-1}$$

- Erro Escalado: proposto por [124], como uma alternativa ao uso dos erros percentuais quando comparado à precisão das previsões em toda série em diferentes escalas. A métrica MASE (*Mean Absolute Scaled Error*) pode ser facilmente interpretada como valores

inferiores a um indicam que o modelo de previsão adotado gera menores erros que o modelo ingênuo. Em contrapartida, valores maiores que um sinalizam que o modelo de previsão apresenta erros maiores que o modelo ingênuo:

$$\text{MASE} = \text{média}(|q_j|) \tag{5-2}$$

$$\text{Onde } q_j = \frac{e_{jt}}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}$$

É importante observar que quanto menor forem os valores de MAPE e MASE, melhor é a previsão.

5.4
Modelos estatísticos adotados como *benchmark*

A fim de verificar a eficiência dos métodos computacionais ESN e GA-SVR descritos nos capítulos 2, 3 e 4 desta dissertação, comparou-se a precisão destes com a precisão obtida por métodos estatísticos. Os *benchmarks* utilizados são descritos a seguir.

5.4.1
SARIMA

Box-Jenkins [125] compreende um classe específica de modelos estocásticos lineares [117]. A metodologia consiste em ajustar modelos autorregressivos integrados de média móveis a um conjunto de dados. É representado por ARIMA (p, d, q), onde p é o número de defasagens da série, d é a ordem de integração para tornar a série estacionária e q o número de defasagens dos erros aleatórios (a_t).

A modelagem Box-Jenkins pode ser estendida para captar efeitos sazonais (SARIMA). O modelo $\text{SARIMA}(p, d, q) \times (P, D, Q)_S$ pode ser descrito como:

$$\nabla_S^D \nabla^d \phi(B) \Phi(B^S) Z_t = \theta(B) \Theta(B^S) a_t \tag{5-3}$$

Onde, B é o operador de atraso; $\nabla^d = (1-B)^d$; $\nabla_S^D = (1-B^S)^D$; $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$; $\Phi(B^S) = 1 - \Phi_1 B^S - \cdots \Phi_P B^{PS}$; $\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q$; $\Theta(B^S) = 1 - \Theta_1 B^S - \cdots - \Theta_Q B^{QS}$.

Os dados que compõem a série devem ter distribuição normal e variância constante. Para construir o modelo seguimos um algoritmo composto por quatro etapas:

- Identificação – A estrutura do modelo é identificada analisando-se a função de autocorrelação (FAC[19]) e a função de autocorrelação parcial (FACP[20]); busca-se identificar os valores de p, q, e d;
- Estimação – Após a identificação da estrutura do modelo, os parâmetros φ_i's e θ_j's são estimados de forma a minimizar a soma dos quadrados dos resíduos;
- Verificação – As estruturas identificadas são validadas através de testes de sobrefixação. Verifica-se, ainda, se o modelo ajustado é adequado aos dados através de uma análise de resíduos (e.g. teste de Ljung-Box [126]). Se os resíduos são autocorrelacionados, então, a dinâmica da série não é completamente explicada pelos coeficientes do modelo ajustado, devendo-se voltar à fase de identificação;
- Previsão – Após a verificação, aplica-se a equação do modelo estimado para prever valores futuros da variável em estudo.

O algoritmo auto.arima() implementado no pacote forecast do software R [127] foi utilizado para ajustar o melhor modelo de acordo com o critério de AIC.

5.4.2 Amortecimento exponencial

O amortecimento exponencial (AE) deve ser considerado quando desejamos modelar o comportamento de uma série com nível (μ_t), ou nível (μ_t) acompanhado de tendência (b_t) com ou sem componente sazonal (S_t). O método compreende uma classe de modelos que descreve previsões como combinações ponderadas de observações passadas e presentes, sendo que o valor dos pesos das observações decresce exponencialmente à medida que as observações se tornam mais antigas.

Hyndman *et al.* [128] identificam 24 variações desse método. No entanto, existem três variações básicas de AE mais comumente usadas: amortecimento

[19] Medida padronizada da dependência linear de lag k: $\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{Cov[z_t, z_{t+k}]}{\sqrt{var(z_t).var(z_{t+k})}}$

[20] Medida de dependência linear ou correlação linear entre Zt e Zt+k eliminando a dependência dos termos intermediários Zt+1, Zt+2 . . . Zt+k-1

exponencial simples [129], amortecimento exponencial com correção de tendência [130], e amortecimento de Holt-Winter [131] com correção de tendência e sazonalidade.

Existem duas variações de Holt Winters. Estas diferem na natureza do componente sazonal. O método aditivo é preferido quando a amplitude das variações sazonais são mais ou menos constantes ao longo do tempo, enquanto o método multiplicativo é utilizado quando as variações sazonais se alteram proporcionalmente ao nível da série. Na Tabela 5.4 é possível visualizar as equações para os dois modelos.

Tabela 5.4 – Métodos aditivo e multiplicativo de Holt Winters

	Holt Winters Aditivo	**Holt Winters Multiplicativo**
Nível	$\mu_t = \alpha(y_t - S_{t-S}) + (1-\alpha)(\mu_{t-1} + b_{t-1})$	$\mu_t = \alpha \frac{y_t}{S_{t-S}} + (1-\alpha)(\mu_{t-1} + b_{t-1})$
Tendência	$b_t = \beta(\mu_t - \mu_{t-1}) + (1-\beta)b_{t-1}$	$b_t = \beta(\mu_t - \mu_{t-1}) + (1-\beta)b_{t-1}$
Sazonalidade	$S_t = \gamma(y_t - \mu_t) + (1-\gamma)S_{t-s}$	$S_t = \gamma \frac{y_t}{L_t} + (1-\gamma)S_{t-s}$
Previsão	$F_{t+h} = \mu_t + b_{t-h} + S_{t-s+h}$	$F_{t+m} = (\mu_t + b_{t-h})S_{t-s+h}$

Em que: S é comprimento da sazonalidade; F_{t+h} é a previsão para o período h adiante; y_t é o valor observado; $\alpha, \beta \, e \, \gamma$ são os parâmetros de amortecimento do nível, da tendência e da sazonalidade.

O algoritmo ets() (Error,Trend,Seasonal) implementado no pacote forecast do software R [127] foi utilizado para ajustar o melhor modelo de alisamento exponencial aos dados.

5.4.3
Regressão dinâmica

Os modelos de regressão dinâmica combinam a dinâmica de séries temporais e o efeito de variáveis explicativas ou preditoras. Atenta-se que o termo "regressão dinâmica" não indica que os parâmetros do modelo evoluem no tempo. Ao invés disso, a palavra "dinâmica" significa aqui um modelo de regressão no qual se inclui a estrutura de dependência de uma série temporal. Assim, a variável resposta y_t é explicada por seus valores defasados (*lags*) e pelos valores atuais e *lags* das variáveis explicativas x_t [132]. O modelo de regressão dinâmica pode ser descrito como:

$$\varphi(B)y_t = \beta x_t + \varepsilon_t \qquad (5\text{-}4)$$

Sendo β, vetor de coeficientes das variáveis causais; ε_t ruído aleatório associado ao modelo; $\varphi(B)$, polinômio auto-regressivo de ordem p, isto é: $\varphi(B) = 1 - \varphi_1 B - \varphi_2 B_2 - \cdots - \varphi_p B_p$, onde B o operador de atraso.

A estimação dos parâmetros da Equação (5-4 pode ser realizada utilizando-se o Método dos Mínimos Quadrados Ordinários (MQO), supondo que o termo estocástico ε_t tenha as propriedades adequadas como média zero ($E(\varepsilon_t) = 0$), homoscedasticidade ($Var(\varepsilon_t) = \sigma^2$) e erros não correlacionados ($E(\varepsilon_i, \varepsilon_j) =$ 0,εj) = 0, se i > j).

Neste estudo, a estimação do modelo de regressão dinâmica obedece à metodologia da Escola de Economia de Londres (*London School of Economics*) [133], conhecida como do geral para o específico "*the general to specific*". O método utilizado para as estimações econométricas realizadas, foi semelhante ao desenvolvido por Davidson *et al.* [134] para modelar uma função consumo para o Reino Unido.

O *software* Forecast Pro Windows (FPW) foi utilizado para ajustar o modelo.

6
Aplicação no mercado brasileiro de seguros – previsão de prêmio

6.1
Introdução

No Brasil, apenas em 2015, o mercado de seguros arrecadou R$ 363,1 bilhões em prêmios diretos de seguros, saúde suplementar, contribuições previdenciárias e em títulos de capitalização, o que significou 6,1% do PIB. Nos mercados de países desenvolvidos, a arrecadação anual de prêmios se situa próxima aos 8,3% do PIB. O Brasil tem, portanto, longo espaço de crescimento.

O mercado segurador brasileiro encontra-se concentrado em três sub-ramos: seguro saúde, seguro de pessoas (vida, acidentes e previdência) e automóveis. Pode ser observado na Figura 6.1 que juntos estes seguros detiveram 86,6% da receita em 2015, segundo dados da SUSEP (Superintendência de Seguros Privados) e da ANS (Associação Nacional de Saúde).

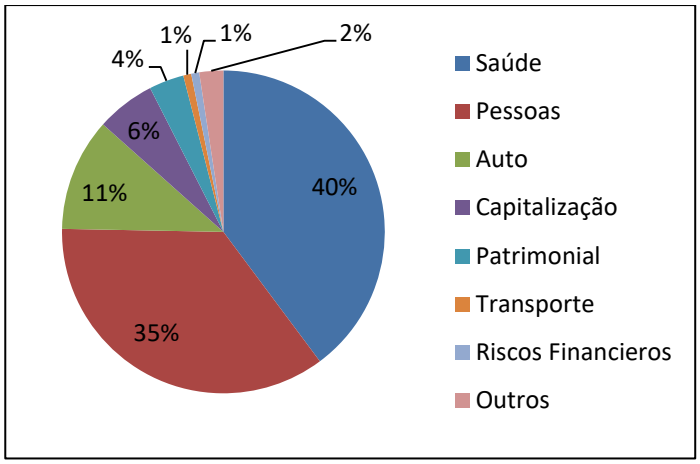

Figura 6.1 – Distribuição de Prêmios e Contribuições (2015)

Fonte: SUSEP e ANS

O seguro de automóveis é o item dessa indústria mais vendido no país. A maior parcela do faturamento do setor, no entanto, é gerada por um tipo de plano

de previdência complementar que tem características de investimento financeiro, o seguro Vida Gerador Benefício Livre (VGBL).

A indústria de seguros, crescentemente, suplementa o Estado no fornecimento de serviços cruciais nas áreas de saúde e de seguridade social. Seguros de vida e de previdência complementar aberta constituem importante elo nos mecanismos de proteção contra riscos da aposentadoria e da velhice.

No Brasil, a algumas décadas, esse risco era inexpressivo, vide a grande participação dos jovens na população total. Atualmente, isto não mais ocorre. De fato, a população brasileira com mais de 65 anos, que se mantivera em torno dos 3% do total até 1970, pode alcançar os 13% em 2020, e níveis de União Europeia em 2050 segundo dados do Instituto Brasileiro de Geografia e Estatística.

6.2
Base de dados e análise descritiva

Com base no exposto, este trabalho selecionou as séries de prêmio[21] dos seguros de automóveis, vida e previdência para serem modeladas. Os dados referentes à prêmios contemplam os prêmios diretos recebidos pelas seguradoras em cada mês. A unidade de medida é milhões de reais. A série de dados dos prêmios diretos foi retirada da base pública da Superintendência de Seguros Privados (SUSEP).

As séries de prêmio compreendem os períodos de Janeiro de 2001 a Dezembro de 2015, totalizando 180 observações. Como discutido no Capítulo 5, para a previsão com algoritmos de *machine learning*, as séries foram divididas em treinamento (2001 a 2013), validação (2014) e teste (previsão para o ano de 2015). A seguir são apresentadas e descritas as três séries temporais estudadas.

6.2.1
Automóveis

O seguro de automóveis tem o intuito de garantir ao segurado o pagamento de uma indenização em caso de perda total de seu veículo seja por colisão, incêndio ou roubo. Esse seguro também cobre o conserto do veículo mediante o

[21] O conceito de prêmio direto, definido pela SUSEP, engloba a arrecadação da empresa seguradora. Portanto, entende-se por prêmio de seguro o total de prêmios diretos de todas as empresas, que reflete a demanda de mercado em um determinado intervalo de tempo.

pagamento de uma franquia previamente informada no ato da contratação do seguro. Apesar de ser o seguro, mais vendido é o que apresenta o maior índice de sinistralidade.

A série de prêmio de seguros de automóveis utilizada neste trabalho é apresentada na Figura 6.2. É possível observar que a série apresenta tendência de crescimento ao longo do tempo, e que parece apresentar um padrão de variação mensal que se repete ao longo do tempo. A série aparenta, ainda, ser heterocedástica.

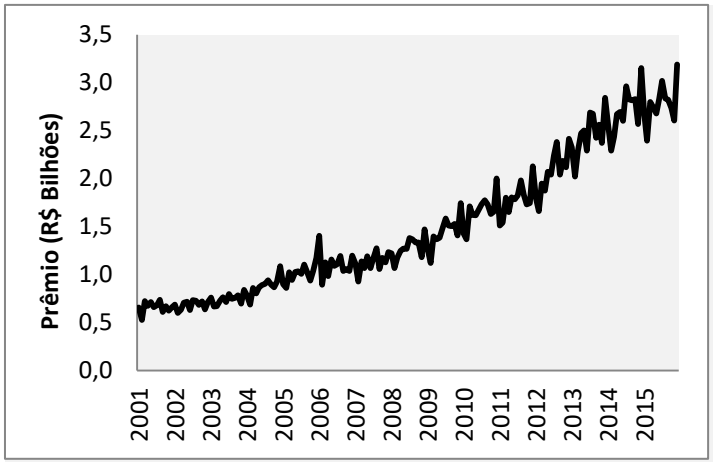

Figura 6.2 – Prêmio de seguro de Automóveis

As estatísticas descritivas da série Automóveis são apresentadas na Tabela 6.1. Elas sinalizam a não normalidade da série. Dados os valores dos coeficientes de assimetria e curtose, infere-se que a distribuição de prêmio em questão é mais achatada que a gaussiana[22] e apresenta cauda mais longa à direita. Isto indica que a modelagem desta série por modelos estatísticos (que assumem normalidade) não é a mais indicada.

Tabela 6.1 – Estatísticas descritivas da série Automóveis

Estatística Descritiva	
Média	1,49 (R$ Bilhões)
Desvio Padrão	0,71 (R$ Bilhões)
Mínimo	0,53 (R$ Bilhões)
Máximo	3,19 (R$ Bilhões)
Assimetria	0,65
Curtose	-0,78

[22] Para o leitor não familiarizado com conceitos de assimetria e curtose recomenda-se Casella e Berger [146].

6.2.2
Vida

A série de seguro de vida abrange as séries de seguro de vida individual, seguro de vida coletivo, prestamistas e acidentes pessoais. A série é apresentada na Figura 6.3. Nela é possível observar uma tendência de crescimento dos prêmios ao longo do tempo e a presença de um *outlier* no início da série. (Esse *outlier* se deve a um aumento atípico das receitas de prêmio de seguro de vida individual ocorrido em dezembro de 2002.) A série parece apresentar aumento de sua variância ao longo do tempo.

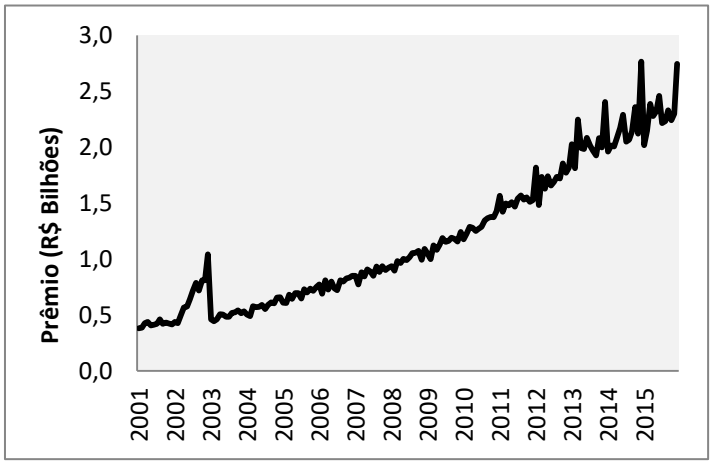

Figura 6.3 – Prêmio de seguro de Vida

As estatísticas descritivas da série Vida são apresentadas na Tabela 6.2. A partir dela, verifica-se que há evidencias de que a distribuição da série é não gaussiana, apresentando calda mais longa à direita. Isto indica que a modelagem desta série por modelos estatísticos pode ser ineficiente.

Tabela 6.2 – Estatísticas descritivas da série Vida

Estatística Descritiva	
Média	1,18 (R$ Bilhões)
Desvio Padrão	0,62 (R$ Bilhões)
Mínimo	0,38 (R$ Bilhões)
Máximo	2,77 (R$ Bilhões)
Assimetria	0,61
Curtose	-0,81

6.2.3
Previdência

A série de seguro de previdência agrega os dois principais ramos desse seguro: o PGBL (Plano Gerador de Beneficio Livre) e o VGBL (Vida Gerador de Benefício Livre). Sendo este último o seguro mais vendido, pois o imposto de renda incide apenas sobre os rendimentos do plano e não sobre o total acumulado.

A série é apresentada na Figura 6.4. Nela, é possível observar que a série apresenta tendência, comportamento ruidoso e variância crescente ao longo do tempo. Além disso, parece haver uma alteração estrutural entre 2013 e 2014.

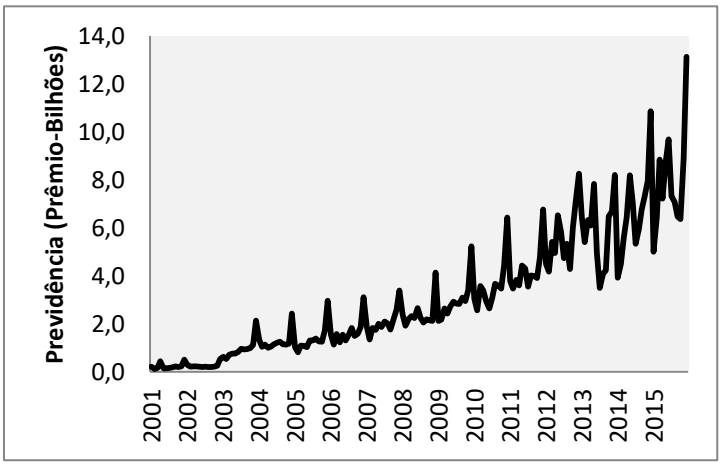

Figura 6.4 – Prêmio de seguro de Previdência

As estatísticas descritivas da série Previdência são apresentadas na Tabela 6.3. Elas sinalizam que a série de prêmio de previdência é altamente dispersa e apresta distribuição com calda bem mais longa à esquerda. Ademais a distribuição dessa série é mais afunilada (mais alta) que a gaussiana. Como consequência, é não recomendado o uso de modelos estatísticos para esta série.

Tabela 6.3 – Estatísticas descritivas da série Previdência

Estatística Descritiva	
Média	3,12 (R$ Bilhões)
Desvio Padrão	2,54 (R$ Bilhões)
Mínimo	0,12 (R$ Bilhões)
Máximo	13,13 (R$ Bilhões)
Assimetria	1,04
Curtose	0,78

6.2.4
Séries explicativas para modelagem multivariada

A relação entre instituições financeiras, variáveis macroeconômicas e crescimento econômico é objeto de estudo há longo tempo. Contudo, apesar da vasta literatura no assunto, proporcionalmente, pouca atenção foi dada ao mercado de seguros. Em geral alguns estudos optaram por estudar esse mercado em determinados países buscando entender suas relações com a economia. Li *et al.* [135] e Outreville [136] fazem uma revisão detalhada sobre o tema.

De forma genérica, a aquisição do seguro permite que o segurado dilua os riscos financeiros decorrentes da perda ou da imposição de danos a seus bens ou da redução de sua capacidade de geração de renda. Contador [137] mostra ser razoável pensar no seguro como um bem de luxo[23]. Uma possível explicação para este fato é que o custo de oportunidade associado ao evento negativo aumenta com a elevação da renda e torna os indivíduos mais avessos ao risco, sendo uma explicação para o aumento na demanda por seguros no Brasil nos últimos anos.

Babbel [138] afirma que a inflação deprecia o valor financeiro de ativos, reduzindo a atratividade do seguro de vida. Beck e Webb [139] corroboram parcialmente esses resultados, concluindo que a estabilidade econômica é importante.

Neste sentido, foram utilizadas como variáveis explicativas do prêmio de seguro, a inflação mediada pelo IPCA e o PIB *per capita.* Outras variáveis como o nível educação e crescimento populacional, foram exploradas por [23,26,135,136].

Os dados de inflação e de população foram extraídos do Instituto Brasileiro de Geografia e Estatística (IBGE). A série de PIB, por sua vez, foi obtida junto ao Banco Central do Brasil (Bacen). A Figura 6.5 e a Figura 6.6 apresentam os gráficos da série de PIB *per capita* e de inflação utilizadas neste trabalho, respectivamente.

[23] Denomina-se bem luxo, o bem cuja quantidade demandada aumenta em maior proporção ao se aumentar a renda.

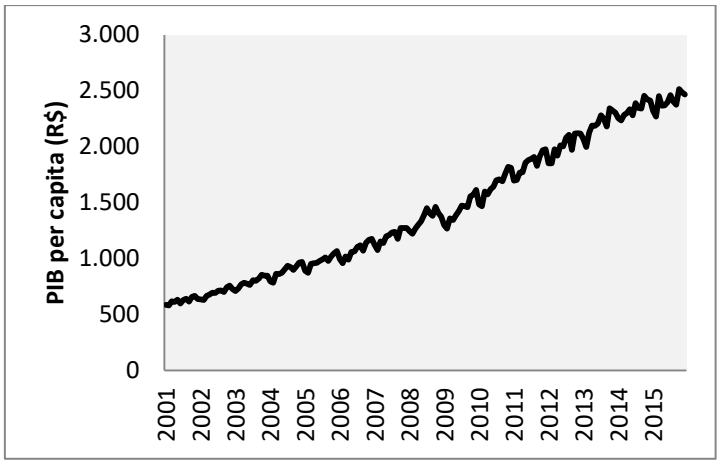

Figura 6.5 – Série de PIB *per capita* do Brasil

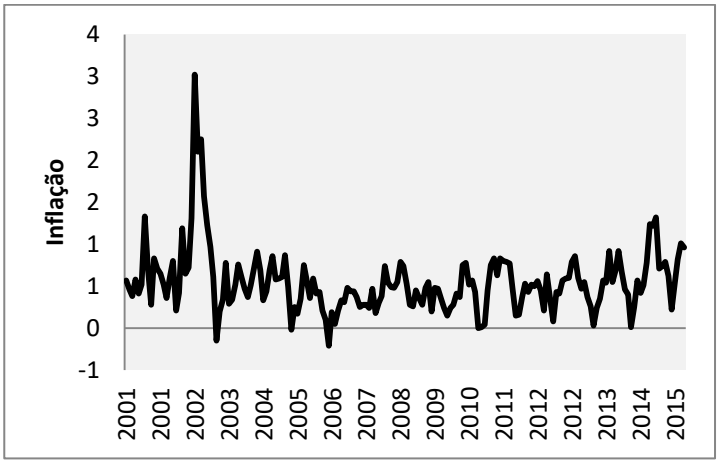

Figura 6.6 – Série de Inflação do Brasil

O objetivo da previsão multivariada é a construção de cenários. O aprendizado gerado da experiência de construir e de se capacitar a construir cenários permite enxergar oportunidades e ameaças que, de outra forma, permaneceriam ocultas aos gestores das empresas.

A qualidade da previsão multivariada depende da correta especificação das variáveis explicativas. Desta forma, o resultado da previsão multivariada não será diretamente comparado com as previsões univariadas. Os modelos de previsão multivariada construídos para esta dissertação são apenas uma simplificação para avaliar a viabilidade de utilização dos modelos propostos para previsão de prêmio de seguro.

6.3 Resultados

Os modelos univariados e multivariados ESN e GA-SVR propostos nesta dissertação, e os modelos utilizados para comparação, foram aplicados para a previsão das três séries de prêmio estudadas. A seguir são apresentados os resultados obtidos para os modelos unviariados (Seção 6.3.1) e para os modelos multivariados (Seção 6.3.2).

6.3.1 Previsão com Modelos Univariados

Os modelos univariados ESN e GA-SVR, selecionados para previsão das três séries de prêmio estudadas – i.e., automóveis, vida e previdência – têm seus parâmetros apresentados nas Tabelas 6.4 e 6.5. No apêndice, os gráficos das previsões são exibidos.

Tabela 6.4 – Parâmetros do modelo univariado ESN selecionado

Série	Unidades no reservatório	Percentual de conexão	Raio espectral	Janela de previsão
Automóveis	20	60%	0,80	12
Vida	60	60%	0,80	3
Previdência	20	20%	0,80	12

Tabela 6.5 – Parâmetros do modelo univariado GA-SVR selecionado

Série	Kernel	Defasagens	C	ε
Automóveis	linear	13	2,89	0,01
Vida	linear	13	6,81	0,01
Previdência	linear	13	9,93	0,11

Aplicando-se os modelos univariados ESN e GA-SVR selecionados, e os modelos de Amortecimento Exponencial (AES) e Box e Jenkins (B&J) às séries de prêmio de automóveis, vida e previdência, foram obtidas as previsões nos horizontes de até três, seis e doze passos à frente. Os resultados das previsões, em termos das métricas de acurácia MAPE e MASE, são apresentados nas Tabelas 6.6 e 6.7, respectivamente.

Tabela 6.6 – MAPE (%) das previsões univariadas

Série	Horizonte	AES	B&J	GA-SVR	ESN
Automóveis	1 a 3	5,51	5,81	5,14	1,07
	1 a 6	6,69	5,97	5,55	2,30
	1 a 12	7,76	6,97	5,97	3,18
Vida	1 a 3	18,11	9,76	12,07	0,88
	1 a 6	14,84	6,90	7,40	1,75
	1 a 12	15,98	10,20	6,09	3,02
Previdência	1 a 3	15,39	15,44	15,88	9,52
	1 a 6	13,95	15,11	18,49	9,55
	1 a 12	14,52	14,68	14,93	10,56

Tabela 6.7 – MASE das previsões univariadas

Série	Horizonte	AES	B&J	GA-SVR	ESN
Automóveis	1 a 3	0,38	0,38	0,34	0,07
	1 a 6	0,47	0,41	0,38	0,17
	1 a 12	0,56	0,50	0,43	0,24
Vida	1 a 3	0,84	0,44	0,56	0,04
	1 a 6	0,71	0,35	0,34	0,09
	1 a 12	0,78	0,50	0,30	0,16
Previdência	1 a 3	0,29	0,29	0,33	0,19
	1 a 6	0,36	0,35	0,38	0,22
	1 a 12	0,34	0,34	0,36	0,29

É possível observar nas Tabelas Tabela 6.6 e Tabela 6.7 que o modelo univariado de *Echo State* (ESN) apresenta os melhores resultados para todas as séries, e em todos os horizontes de previsão. O desempenho do ESN é bastante superior ao desempenho dos outros três modelos; por exemplo, para a série de prêmio de automóveis, o MAPE de até três passos à frente do ESN é de 1,07%, enquanto que os MAPEs dos outros modelos estão entre 5,00% e 6,00%. Este MAPE reduzido do ESN, em relação aos outros modelos, aparece em todas as séries e horizontes.

Os resultados de previsão do modelo GA-SVR se assemelham com os resultados dos modelos estatísticos clássicos. No caso da série de prêmio de automóveis, o GA-SVR tem desempenho melhor que os modelos AES e B&J. No

caso da série de prêmio de vida o modelo GA-SVR apresenta melhor desempenho que os dois modelos estatísticos (AES e B&J), em termos de MAPE, apenas para a previsão de doze passos à frente. Para três e seis passos à frente, o modelo B&J produz melhores resultados[24].

A justificativa para o melhor desempenho do ESN e a justificativa para o desempenho do GA-SVR similar ao dos modelos AES e B&J têm a mesma origem. O modelo GA-SVR selecionado tem função *kernel* linear (segundo Crone [140], o SVR com *kernel* linear é o mais adequado para séries com tendência), o que aproxima suas características àquelas de um modelo linear. Deste modo, o modelo ESN seria o único modelo inteiramente não-linear, dentre os quatro modelos utilizados. Por este motivo, seu desempenho na previsão é superior para as séries de seguros (que apresentam tendência, variância crescente, e sazonalidade ao longo do tempo).

É possível concluir que o modelo univariado ESN é o mais adequado para prever estas séries de prêmio de seguros, o que atesta seu potencial para o Mercado de Seguros Brasileiro.

6.3.2
Previsão com Modelos Multivariados

Os modelos multivariados ESN e GA-SVR selecionados para previsão das três séries de prêmio estudadas – i.e., automóveis, vida e previdência – têm seus parâmetros apresentados nas Tabelas 6.8 e 6.9. No apêndice, os gráficos das previsões são exibidos.

Tabela 6.8 – Parâmetros do modelo univariado ESN selecionado

Série	Unidades no reservatório	Percentual de conexão	Raio espectral	Janela de previsão
Automóveis	90	60%	0,80	12
Vida	60	80%	0,20	6
Previdência	90	60%	0,20	12

[24] De acordo com a métrica MASE, no entanto, o modelo GA-SVR apresenta melhores resultados tanto para as previsões de doze quanto para as previsões de seis passos à frente.

Tabela 6.9 – Parâmetros do modelo univariado GA-SVR selecionado

Série	Kernel	Defasagens	C	ε
Automóveis	linear	13	8,79	0,18
Vida	linear	9	9.86	0,27
Previdência	linear	9	7.43	0,28

Aplicando-se os modelos multivariados ESN e GA-SVR selecionados, e o modelo de Regressão Dinâmica às séries de prêmio de automóveis, vida e previdência, foram obtidas as previsões nos horizontes de até três, seis e doze passos à frente. Os resultados das previsões, em termos das métricas de acurácia MAPE e MASE, são apresentados nas Tabelas 6.10 e 6.11, respectivamente.

Tabela 6.10 – MAPE (%) das previsões multivariadas

Série	Horizonte	Regressão Dinâmica	GA-SVR	ESN
Automóveis	1 a 3	1,93	8,13	2,68
	1 a 6	2,79	7,05	2,51
	1 a 12	3,78	6,66	2,86
Vida	1 a 3	11,99	11,27	3,34
	1 a 6	7,78	6,65	2,75
	1 a 12	6,68	6,64	3,28
Previdência	1 a 3	28,73	18,15	10,2
	1 a 6	23,62	14,27	8,85
	1 a 12	19,67	15,71	10,42

Tabela 6.11 – MASE das previsões multivariadas

Série	Horizonte	Regressão Dinâmica	GA-SVR	ESN
Automóveis	1 a 3	0,13	0,55	0,18
	1 a 6	0,2	0,49	0,18
	1 a 12	0,28	0,48	0,21
Vida	1 a 3	0,55	0,51	0,16
	1 a 6	0,36	0,29	0,14
	1 a 12	0,33	0,33	0,18
Previdência	1 a 3	0,5	0,35	0,25
	1 a 6	0,46	0,34	0,23
	1 a 12	0,51	0,4	0,28

As previsões obtidas com o modelo multivariado ESN, levando-se em conta as variáveis explicativas de PIB *per capita* e Inflação, apresentam as melhores métricas de acurácia para todas as séries e todos os horizontes de previsão (com exceção da previsão de até três passos da série de prêmio de automóveis). As previsões com o modelo multivariado GA-SVR apresentam melhor acurácia que as previsões com a Regressão Dinâmica para duas séries (Vida e Previdência). A justificativa para a superioridade do ESN em relação ao GA-SVR e a Regressão Dinâmica passa pelos mesmos pontos daquela apresentada para os modelos univariados. No geral, os modelos ESN e GA-SVR desempenham melhor que o modelo estatístico clássico.

Um fato digno de nota diz respeito à acurácia das previsões para a série de previdência. Tanto para os modelos univariados quanto para os multivariados, os valores de MAPE são razoavelmente altos (próximos a 10%). Isto ocorre, muito provavelmente, devido às características da série (apresentadas na Seção 6.2).

7
Conclusões e trabalhos futuros

Modelos clássicos de previsão de séries temporais fazem suposições sobre as características da série temporal, geralmente trabalhando com hipóteses de normalidade. Séries temporais reais, no entanto, não apresentam características caóticas e não lineares. Como consequência, nas últimas décadas, modelos de aprendizado de máquinas (*machine learning)* têm sido cada vez mais utilizados como alternativa aos métodos estatísticos clássicos de previsão.

A previsão do prêmio de seguros é de grande importância para os agentes do Mercado de Seguros. As séries temporais deste mercado apresentam características que tornam inadequado o uso de modelos estatísticos clássicos de previsão. Apesar disso, a SUSEP e a CNseg utilizam-nos para previsão.

Neste contexto, este trabalho propôs o uso de modelos de *machine learning* para a previsão das séries de prêmio de seguros de automóveis, vida e previdência. Foram construídos e aplicados modelos univariados e multivariados de *echo-state* (ESN) e da híbridização entre algoritmos genéticos e regressão por vetores suporte (GA-SVR) para a previsão das séries. Os resultados dos modelos propostos foram comparados com resultados de modelos estatísticos clássicos.

Os modelos univariado e multivariado ESN demonstraram superioridade em relação aos modelos com os quais foram comparados. Em particular, quando séries temporais são geradas por um sistema dinâmico não linear e possuem dependências temporais de longo período (também conhecido como processos de memória longa), modelos de redes neurais como as ESN, geralmente tem melhor desempenho que as técnicas lineares tradicionais. Neste sentido, observou-se que os modelos ESN são promissores para as séries de prêmio de seguros. Os modelos GA-SVR apresentam resultados similares aos dos modelos estatísticos clássicos mesmo sem pré-processamento dos dados de entrada, e poderiam também ser utilizados. A partir destes resultados, é possível concluir que a adoção de modelos de aprendizado de máquinas pode trazer melhoras para as empresas que utilizam previsões de prêmio para auxiliar em seu planejamento.

Os trabalhos futuros decorrentes desta dissertação incluem:

- Utilização de outras técnicas para otimização de parâmetros do SVR como *Particle Swarm Optimization* [84].
- A investigação de técnicas de pré-processamento de dados para melhorar a acurácia da GA-SVR e da ESN como clusterização (e.g., Mapas Auto-Organizáveis [84], Fuzzy C-Means [15,141]), *Multivariable Adaptive Regression Splines* [142,143], *Correlation-based Feature Selection* [144], *Time Delay Coordinates* [16] e decomposição (e.g., [83,145]).
- Treinamento de uma ESN, modelando o reservatório como o *kernel* temporal de uma SVR, de modo a utilizar o principio da minimização estrutural do risco no treinamento da ESN.

8
Referências bibliográficas

[1] BOX, G. E. P.; JENKINS, G. M. **Time Series Analysis:** Forecasting and Control. San Francisco: Holden-Day Inc., 1976.

[2] CHATFIELD, C. **Time-series Forecasting**. Boca Raton: Chapman & Hall / CRC, 2000.

[3] YAN, W. Toward automatic time-series forecasting using neural networks. **IEEE Transactions on Neural Networks and Learning Systems**, 23, n. 7, June 2012. 1028-1039.

[4] SUDHEER, G.; SUSEELATHA, A. Short term load forecasting using wavelet transform combined with Holt-Winters and weighted nearest neighbor models. **International Journal of Electrical Power and Energy Systems**, 64, January 2015. 340-346.

[5] KHOSRAVI, A. et al. Interval type-2 fuzzy logic systems for load forecasting: A comparative study. **IEEE Transactions on Power Systems**, 27, n. 3, February 2012. 1274-1282.

[6] KANTZ, H.; SCHREIBER, T. **Nonlinear time series analysis**. Cambridge university press, v. 7, 2004.

[7] KAASTRA, I.; BOYD, M. Designing a neural network for forecasting financial and economic time series. **Neurocomputing**, 10, n. 3, April 1996. 215-236.

[8] HIPPERT, H. S.; PEDREIRA, C. E.; SOUZA, R. C. Neural networks for short-term load forecasting: A review and evaluation. **IEEE Transactions on Power Systems**, 16, n. 1, February 2001. 44-55.

[9] ILIES, I. et al. Stepping forward through echoes of the past: forecasting with echo state networks. **Artificial Neural Networks & Computational Intelligence Forecasting Competition**, 2007.

[10] SANTAMARÍA-BONFIL, G.; REYES-BALLESTEROS, A.;

GERSHENSON, C. Wind speed forecasting for wind farms: A method based on support vector regression. **Renewable Energy**, 85, January 2016. 790-809.

[11] PAI, P.-F. et al. Time series forecasting by a seasonal support vector regression model. **Expert Systems with Applications**, 37, n. 6, June 2010. 4261-4265.

[12] CEPERIC, E.; CEPERIC, V.; BARIC, A. A strategy for short-term load forecasting by support vector regression machines. **IEEE Transactions on Power Systems**, 28, n. 4, July 2013. 4356-4364.

[13] HONG, W.-C. Hybrid evolutionary algorithms in a SVR-based electric load forecasting model. **International Journal of Electrical Power and Energy Systems**, 31, n. 7-8, September 2009. 409-417.

[14] GENG, J. et al. Hybridization of seasonal chaotic cloud simulated annealing algorithm in a SVR-based load forecasting model. **Neurocomputing**, 151, n. P3, March 2015. 1362-1373.

[15] PAI, P.-F.; HUNG, K.-C.; LIN, K.-P. Tourism demand forecasting using novel hybrid system. **Expert Systems with Applications**, 41, n. 8, June 2014. 3691-3702.

[16] SANTAMARÍA-BONFIL, G.; REYES-BALLESTEROS, A.; GERSHENSON, C. Wind speed forecasting for wind farms: A method based on support vector regression. **Renewable Energy**, 85, January 2016. 790-809.

[17] CHEN, R. et al. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. **Applied Soft Computing Journal**, 26, January 2015. 435-443.

[18] BREIMAN, L. Statistical modeling: The two cultures. **Quality control and applied statistics**, v. 48, n. 1, p. 81-82, 2003.

[19] PRIKAZYUK, N.; MOTASHKO, T. Security of the insurance market of Ukraine and the determining factors. **Procedia Economics and Finance**, v. 27, p. 288-310, 2015.

[20] RANGER, N.; SURMINSKI, S. A preliminary assessment of the impact of climate change on non-life insurance demand in the BRICS economies.

International Journal of Disaster Risk Reduction, 3, March 2013. 14-30.

[21] STAIB, D.; PUTTAIAH, M. **World Insurance in 2014: Back to Life.** Swiss Re. Zurich, Switzerland. 2015.

[22] ANDREESKI, C.; MILOšEVIć, B.; NJEGOMIR, V. Analysis of the life insurance market in the Republic of Macedonia. **Economic Annals**, 57, July – September 2012. 107-122.

[23] EBRAHIM FOULADVAND, M.; DAROONEH, A. H. Premium forecasting of an insurance company: Automobile insurance. **International Journal of Modern Physics C**, 16, n. 3, March 2005. 377-387.

[24] CUMMINS, J. D.; GRIEPENTROG, G. L. Forecasting automobile insurance paid claim costs using econometric and ARIMA models. **International Journal of Forecasting**, v. 1, n. 3, p. 203–215, 1985.

[25] ENZ, R. The S-Curve Relation between Per-Capita Income and Insurance Penetration. **Geneva Papers on Risk and Insurance: Issues and Practice**, v. 25, n. 3, p. 396-406, 2000.

[26] CRISTEA, M. . M. N.; CÂRSTINA, S. The relationship between insurance and economic growth in Romania compared to the main results in Europe–a theoretical and empirical analysis. **Procedia Economics and Finance**, v. 8, p. 226-235, 2014.

[27] ANGUS, J. E. Poisson compounding of dependent random variables: A stochastic model for total claim costs. **Mathematical and Computer Modelling**, v. 18, n. 5, p. 97-105, September 1993.

[28] CAVALCANTE, R. C. et al. Computational Intelligence and Financial Markets: A Survey and Future Directions. **Expert Systems with Applications**, 55, 2016. 194-211.

[29] HAYKIN, S. **Neural networks:** a comprehensive foundation. 2. ed. New Jersey: Prentice Hall, 1999.

[30] JAEGER, H. The Echo state approach to analyzing and training recurrent neuralnetworks. **German National Research Center for Information Technology GMD Technical Report**, v. 148, n. 24, 2001.

[31] JAEGER, H. Short-term memory in echo state networks. **Technical report, GDM 152, German National Resource Center for Information**

Technology, 2002.

[32] JAEGER, H.; HAAS, H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. **Science**, v. 304, n. 5667, p. 78–80, 2004.

[33] MEDSKER, L. R.; JAIN, L. C. **Recurrent neural networks:** Design and Applications. London, New York: CRC Press, 2001.

[34] NERRAND, O. et al. Neural networks and nonlinear adaptive filtering: unifying concepts and new algorithms. **Neural computation**, v. 5, n. 2, p. 165-199, 1993.

[35] FUNAHASHI, K. I.; NAKAMURA, Y. Approximation of dynamical systems by continuous time recurrent neural networks. **Neural networks**, v. 6, n. 6, p. 801-806, 1993.

[36] SCHAFER, A. M.; ZIMMERMANN, H. G. Recurrent neural networks are universal approximators. **International Journal of Neural Systems**, v. 17, n. 5, p. 253–263, 2007.

[37] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, 9, n. 8, 1997. 1735-1780.

[38] PENG, Y. et al. A modified echo state network based remaining useful life estimation approach. **Prognostics and Health Management (PHM), 2012 IEEE Conference**, p. 1-7, 2012.

[39] LUKOšEVIčIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, v. 3, n. 3, p. 127-149, 2009.

[40] ANTONELO, E. A.; SCHRAUWEN, B.; STROOBANDT, D. Event detection and localization for small mobile robots using reservoir computing. **Neural Networks**, v. 21, n. 6, p. 862-871, 2008.

[41] JAEGER, H. **The " echo state " approach to analysing and training recurrent neural networks – with an Erratum note 1**. Bonn, Germany, p. 1-47. 2010.

[42] YILDIZ, I. B. . J. H. . &. K. S. J. Re-visiting the echo state property. **Neural networks**, v. 35, p. 1-9, 2002.

[43] BISHOP, C. M. **Pattern recognition and Machine Learning**. New York: Springer, 2006.

[44] JAEGER, H. **A tutorial on training recurrent neural networks , covering BPPT , RTRL , EKF and the " echo state network " approach**, p. 1-46. 2013.

[45] LUKOšEVIčIUS, M. A practical guide to applying echo state networks. In: _____ **Neural networks:** Tricks of the trade. Springer Berlin Heidelberg, 2012. p. 659-686.

[46] SCHMIDHUBER, J. et al. Training recurrent networks by evolino. **Neural computation**, v. 19, n. 3, p. 757-779, 2007.

[47] BABINEC, Š.; POSPÍCHAL, J. Merging echo state and feedforward neural networks for time series forecasting. In: _____ **International Conference on Artificial Neural Networks**. Springer Berlin Heidelberg, 2006. p. 367-375.

[48] BUTCHER, J. . V. D. et al. Extending reservoir computing with random static projections: a hybrid between extreme learning and RC. In: _____ **18th European Symposium on Artificial Neural Networks (ESANN 2010)**. D-Side, 2010. p. 303-308.

[49] BUTCHER, J. B. . V. D. et al. Reservoir computing and extreme learning machines for non-linear time-series data analysis. **Neural networks**, v. 38, p. 76-89, 2013.

[50] HUANG, G. B.; WANG, D. H.; LAN, Y. Extreme learning machines: a survey. **Journal of Machine Learning and Cybernetics**, 2, n. 2, 2011. 107-122.

[51] SHI, Z.; HAN, M. Support vector echo-state machine for chaotic time-series prediction. **IEEE Transactions on Neural Networks**, v. 18, n. 2, p. 359-372, 2007.

[52] BOSER, B.; GUYON, I.; VAPNIK, V. A training algorithm for optimal margin classifiers. **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**, 1992. 144-152.

[53] BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, v. 2, n. 2, p. 121-167,

1998.

[54] BUSH, K.; ANDERSON, C. **Modeling reward functions for incomplete state representations via echo state networks**. 2005 International Joint Conference on Neural Networks. Montreal: IEEE. 2005. p. 2995-3000.

[55] JAEGER, H. **Discovering multiscale dynamical features with hierarchical echo state networks**. 2007.

[56] BOCCATO, L. et al. An extended echo state network using Volterra filtering and principal component analysis. **Neural Networks**, v. 32, p. 292-302, 2012.

[57] BOCCATO, L. et al. An echo state network architecture based on Volterra filtering and PCA with application to the channel equalization problem. In: **_____ Neural Networks (IJCNN), The 2011 International Joint Conference**. [S.l.]: IEEE, 2011. p. 580-587.

[58] MATHEWS, V. J. Adaptive polynomial filters. **IEEE Signal Processing Magazine**, v. 8, n. 3, p. 10-26, 1991.

[59] VAPNIK, V. N.; LERNER, A. Pattern recognition using generalized portraits. **Automation and Remote Control**, 24, n. 6, June 1963. 774-7780.

[60] SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and Computing**, 14, 2004. 199-222.

[61] VAPNIK, V.; GOLOWICH, S.; SMOLA, A. Support vector method for function approximation, regression estimation, and sinal processing. In: MOZER, M. C.; JORDAN , M. I.; PETSCHE, T. **Advanced in Neural Information Processing Systems 9**. Cambridge: MIT Press, 1997. p. 281-287.

[62] CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, 20, n. 3, September 1995. 273-297.

[63] DRUCKER, H. et al. Support vector regression machines. In: MOZER, M. C.; JORDAN, M. I.; PETSCHE, T. **Advanced in Neural Information Processing Systems 9**. Cambridge: MIT Press, 1997. p. 155-161.

[64] VAPNIK, V. **The nature of statistical learning theory**. Springer Science+Business Media, 1995.

[65] VAPNIK, V. **Statistical learning theory**. 1ª. ed. Jonh Wiley & Sons, 1998.

[66] VAPNIK, V. An overview of statistical learning theory. **IEEE Transactions on neural networks**, 10, n. 5, September 1999. 988-999.

[67] CAO, L.; TAY, F. E. H. Financial forecasting using support vector machines. **Neural Computing & Applications**, 10, n. 2, 2001. 184-192.

[68] LUXBURG, U. V.; SCHÖLKOPF, B. **Statistical learning theory:** models, concepts and results. ArXiv e-prints, 2008. 1-40 p.

[69] VAPNIK, V. Principles of risk minimization for learning theory. **Advances in Neural Information Processing Systems**, p. 831-838, 1992.

[70] LOÈVE, M. **Probability theory 1**. 4ª. ed. Springer Verlag, 1977.

[71] VAPNIK, V. **Estimation of Dependences Based on Empirical Data**. New York: Springer-Verlag, v. 4, 1982.

[72] SEWELL, M. **Structural risk minimization**. 2008.

[73] CHERKASSKY, V.; MULIER, F. M. **Learning from data:** concepts, theory, and methods. John Wiley & Sons, 2007.

[74] CHERKASSKY, V.; MA, Y. Practical selection of SVM parameters and noise estimation for SVM regression. **Neural Networks**, 17, n. 1, January 2004. 113-126.

[75] MANGASARIAN, O. L. **Nonlinear Programming**. Society for Industrial and Applied Mathematics, 1994.

[76] KARUSH, W. Minima of functions of several variables with inequalities, 1939.

[77] KUNH, H. W.; TUCHER, A. W. **Nonlinear programming**. University California Press, 1951.

[78] MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. **Philosophical Transactions of the Royal Society of London**, 209, n. A, 1909. 415-446.

[79] SCHOLKOPF, B.; SMOLA, A. J. **Learning with kernels:** support vector machines, regularization, optimization, and beyond. MIT press, 2001.

[80] HERBRICH, R. **Learning kernel classifiers:** theory and algorithms. Mit Press, 2001.

[81] CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. Cambridge university press, 2000.

[82] SANGEETHA, R.; KALPANA, B. A comparative study and choice of an appropriate kernel for support vector machines. **Communications in Computer and Information Science**, 101, 2010. 549-553.

[83] DE OLIVEIRA, J. F. L.; LUDERMIR, T. B. A hybrid evolutionary decomposition system for time series forecasting. **Neurocomputing**, 180, March 2016. 27-34.

[84] DONG, Z. et al. A novel hybrid approach based on self-organizing maps, support vector regression and particle swarm optimization to forecast solar irradiance. **Energy**, 82, March 2015. 570-577.

[85] PENG, H.-W. et al. Time series forecasting with a neuro-fuzzy modeling scheme. **Applied Soft Computing Journal**, 32, July 2015. 481-493.

[86] RÜPING, S.; MORIK, K. **Support vector machines and learning about time**. IEEE International Conference on Acoustics, Speech, and Signal. 2003. p. 864-867.

[87] FENDER, T. **Empirische Risiko-Minimierung für dynamische Datenstrukturen**. Tese de Doutorado, Universität Dortmund. Dortmund: 2003.

[88] SONG, X. et al. Application of machine learning methods to risk assessment of financial statement fraud:evidence from China. **Jounal of Forecasting**, 33, n. 8, 2014. 611-626.

[89] KIM, K. Financial time series forecasting using support vector machines. **Neurocomputing**, February 2003. 307-319.

[90] ZIMMERMANN, T. **Inductive Learning and Theory Testing : Applications in Finance**. Havard University. 2015.

[91] VARIAN, H. R. Big Data : New Tricks for Econometrics. **Journal of Economic Perpectives**, 28, n. 3, 2014. 3-28.

[92] SANKAR, R.; SAPANKEVYCH, N. I. Time Series Prediction using Support Vector Machines: A Survey. **Computational Intelligence Magazine**, p. 24-38, May 2009.

[93] DE JONG, K. Learning with genetic algorithms: An overview. **Machine learning**, v. 3, n. 2-3, p. 121-138, 1988.

[94] HOLLAND, J. H. **Adaptation in natural and artificial systems:** an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, 1975.

[95] GOLDBERG, D. E. **Genetic algorithms in search, optimization and machine learning**. Boston: Adilson-Wesley, 1989.

[96] SALOMON, R. **Short notes on the schema theorem and the building block hypothesis in genetic algorithms**. International Conference on Evolutionary Programming. Springer Berlin Heidelberg. 1998. p. 113-122.

[97] DAVIS, L. **Handbook of genetic algorithms**. New York: Van Nostrand Reinhold, 1991.

[98] MICHALEWICZ, Z. **Genetic algorithms+ data structures= evolution programs**. Berlin Heidelberg: Springer, 1996.

[99] GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**. Boston, MA: Kluwer Academic Publishers, 2003.

[100] MITCHELL, M. **An introduction to genetic algorithms**. MIT press, 1998.

[101] EIBEN, A. E.; SMITH, J. E. **Introduction to Evolucionary Computing**. Berlin Heidelberg: Springer-Verlag, 2003.

[102] DE JONG, K. An analysis of the behavior of a class of genetic adaptive systems. In: _____ **Tese de Ph. D.** University of Michigan, 1975.

[103] BÄCK, T.; FOGEL, D. B.; & MICHALEWICZ, Z. **Evolutionary computation 1:** basic algorithms and operators. IOP Publishing, Bristol and Philadelphia, v. 1 , 2000.

[104] DEB, K.; ZOPE, P.; JAIN, A. Distributed computing of pareto-optimal solutions with evolutionary algorithms. In: _____ **International Conference on Evolutionary Multi-Criterion Optimization**. Berlin Heidelberg: Springer, 2003. p. 534-549.

[105] BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. **Evolutionary computation 2:** advanced algorithms and operators. IOP Publishing, Bristol and Philadelphia, v. 2, 2000.

[106] YU, X.; GEN, M. **Introduction to Evolutionary Algorithms**. Berlin: Springer-Verlag, 2010.

[107] CAO, Y. J.; WU, Q. H. Optimization of control parameters in genetic algorithms: a stochastic approach. **International Journal of Systems Science**, v. 30, n. 5, p. 551-559, 1999.

[108] FOGEL, D. B. An introduction to simulated evolutionary optimization. **IEEE transactions on neural networks**, v. 5, n. 1, p. 3-10, 1994.

[109] ALBA, E.; DORRONSORO, B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 2, p. 126-142, 2005.

[110] AURNHAMMER, M.; TONNIES, K. D. A genetic algorithm for automated horizon correlation across faults in seismic images. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 2, p. 201-210, 2005.

[111] VENKATRAMAN, S.; YEN, G. G. A generic framework for constrained optimization using genetic algorithms. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 4, p. 424-435, 2005.

[112] WALTERS, C. D.; SHEBLENG, G. B. Genetic algorithm solution of economic dispatch with valve point loading. **IEEE transactions on Power Systems**, v. 8, n. 3, p. 1325-1332, 1993.

[113] MCCALL, J. Genetic algorithms for modelling and optimisation. **Journal of Computational and Applied Mathematics**, v. 184, n. 1, p. 205-222, 2005.

[114] HUANG, C. L.; WANG, C. J. A GA-based feature selection and parameters optimizationfor support vector machines. **Expert Systems with applications**, v. 31, n. 2, p. 231-240, 2006.

[115] BÄCK, T.; SCHWEFEL, H. P. An overview of evolutionary algorithms for parameter optimization. **Evolutionary computation**, v. 1, n. 1, p. 1-23, 1993.

[116] WU, C. H.; TZENG, G. H.; LIN, R. H. A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. **Expert Systems with Applications**, v. 36, n. 3, p. 4725-4735, 2009.

[117] CHATFIELD, C. **The analysis of time series:** Theory and practice.

London: Chapman and Hall, 1975.

[118] GAMA, C. et al. **Extração de Conhecimento de Dados, Data Mining**. [S.l.]: Sílabo, 2012.

[119] HYNDMAN, R. J. Measuring forecast accuracy, 2014.

[120] KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. **Ijcai**, v. 14, n. 2, p. 1137-1145, August 1995.

[121] SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning:** from theory to algorithms. Cambridge University Press, 2014.

[122] MAKRIDAKIS, S.; HIBON, M.; MOSER, C. Accuracy of forecasting: An empirical investigation. **Journal of the Royal Statistical Society. Series A (General)**, 142, n. 2, 1979. 97-145.

[123] LAWRENCE, R. H.; EDMUNDSON, R. H.; O'CONNOR, M. J. An examination of the accuracy of judgmental extrapolation of time series. **International Journal of Forecasting**, 1, n. 1, 1985. 25-35.

[124] HYNDMAN, R. J.; KOEHLER, A. B. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. **International journal of forecasting**, 22, n. 4, October–December 2006. 679-688.

[125] BOX, G. E. P.; JENKINS, G. M. **Time Series Analysis:** Forecasting and Control. San Francisco: Holden-Day Inc., 1976.

[126] LJUNG, G. M.; BOX, G. E. On a measure of lack of fit in time series models. **Biometrika**, v. 65, n. 2, p. 297-303, 1978.

[127] HYNDMAN, R. J.; KHANDAKAR, Y. Automatic time series for forecasting: the forecast package for R (No. 6/07). **Journal of Statistical Software**, 27, n. 3, July 2008.

[128] HYNDMAN, R. J. et al. A Stade Space Framework for Automatic Forecasting Using Exponential Smoothing Methods. **International Journal of Forecasting**, 18, 2002. 439-454.

[129] BROWN, R. G. **Statistical Forecasting for Inventory Control**. New York: McGraw-Hill, 1959.

[130] HOLT, C. C. Forecasting Trend Seasonal by Exponentially Wheighted

Averages. **International Journal of Forecasting**, 20, 2004. 5-13.

[131] WINTERS, P. R. Forecasting Sales by Exponentially Weighted Moving Averages. **Management Science**, 6, n. 3, 1960. 324-342.

[132] PANKRATZ, A. **Forecasting with dynamic regression models**. John Wiley & Sons, v. 935, 2012.

[133] CAMPOS, J.; ERICSSON, N. R.; HENDRY, D. F. General-to-specific modeling: an overview and selected bibliography. **FRB International Finance Discussion Paper** , n. 838, 2005.

[134] DAVIDSON, J. E. et al. Econometric modelling of the aggregate time-series relationship between consumers' expenditure and income in the United Kingdom. **The Economic Journal**, p. 661-692, 1978.

[135] LI, D. . M. F. . N. P. . W. T. The demand for life insurance in OECD countries. **The Journal of Risk and Insurance**, v. 74, n. 3, p. 637-652, 2007.

[136] OUTREVILLE, J. F. The relationship between insurance growth and economic development – 80 empiricalpapers for a review of the literature. **ICER Working Papers 12, International Center for Economic Research**, Torino, Italy, v. 12, p. 263-278, 2011.

[137] CONTADOR, C. R. Mercado de Seguros, crescimento econômico e inflação: uma análise internacional. In: CONTADOR, C. R. **Desafios e oportunidades no mercado de seguros**. Rio de Janeiro: Ediouro: COPPEAD/UFRJ, 1999. p. 10-24.

[138] BABBEL, D. F. Inflation, Indexation, and Life Insurance Sales in Brazil. **The Journal of Risk and Insurance**, v. 49, p. 111-135, 1981.

[139] BECK, T. . W. I. Economic, Demographic, and Institutional Determinants of Life Insurance Consumption Across Countries. **World Bank Economic Review**, v. 17, p. 51-88, 2003.

[140] CRONE, S. F. . G. J. . &. W. R. A study on the ability of Support Vector Regression and Neural Networks to Forecast Basic Time Series Patterns. **IFIP International Conference on Artificial Intelligence in Theory and Practice**, p. 149-158, 2006.

[141] FEIJOO, F.; SILVA, W.; DAS, T. K. A computationally efficient electricity

price forecasting model for real time energy markets. **Energy Conversion and Management**, 113, April 2016. 27-35.

[142] GENG, J. et al. Port throughput forecasting by MARS-RSVR with chaotic simulated annealing particle swarm optimization algorithm. **Neurocomputing**, 147, n. 1, January 2015. 239-250.

[143] LU, C.-J. Sales forecasting of computer products based on variable selection scheme and support vector regression. **Neurocomputing**, 128, March 2014. 491-499.

[144] RANA, M.; KOPRINSKA, I.; AGELIDIS, V. G. Univariate and multivariate methods for very short-term solar photovoltaic power forecasting. **Energy Conversion and Management**, 121, August 2016. 380-390.

[145] FAN, G.-F. et al. Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. **Neurocomputing**, 173, January 2016. 958-970.

[146] CASELLA, G.; BERGER, R. L. **Statistical inference**. Pacific Grove, CA: Duxbury, v. 2, 2002.

Apêndice

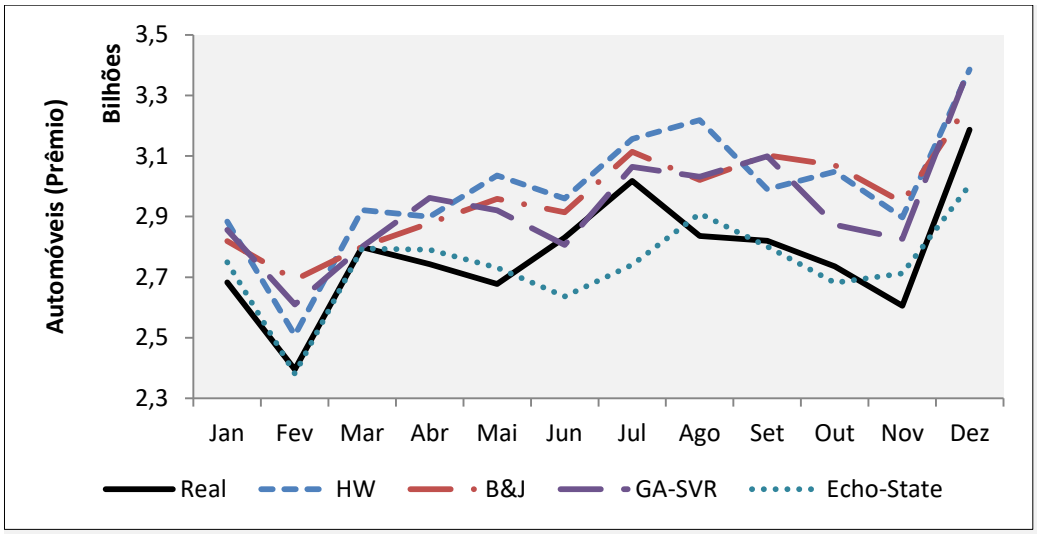

Figura 1: Previsão Univariada - Prêmio de Seguro Automóveis

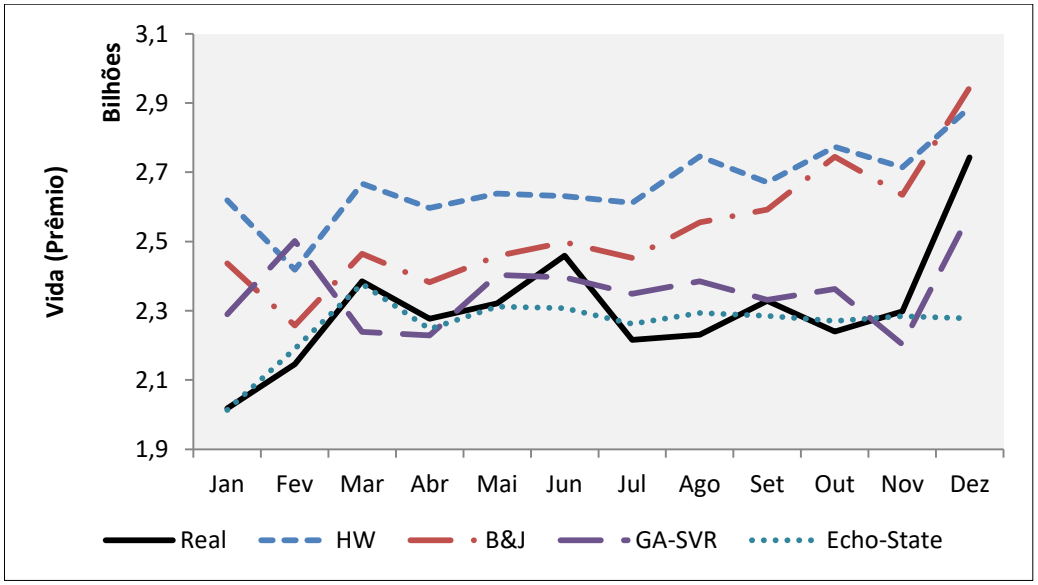

Figura 2: Previsão Univariada - Prêmio de Seguro Vida

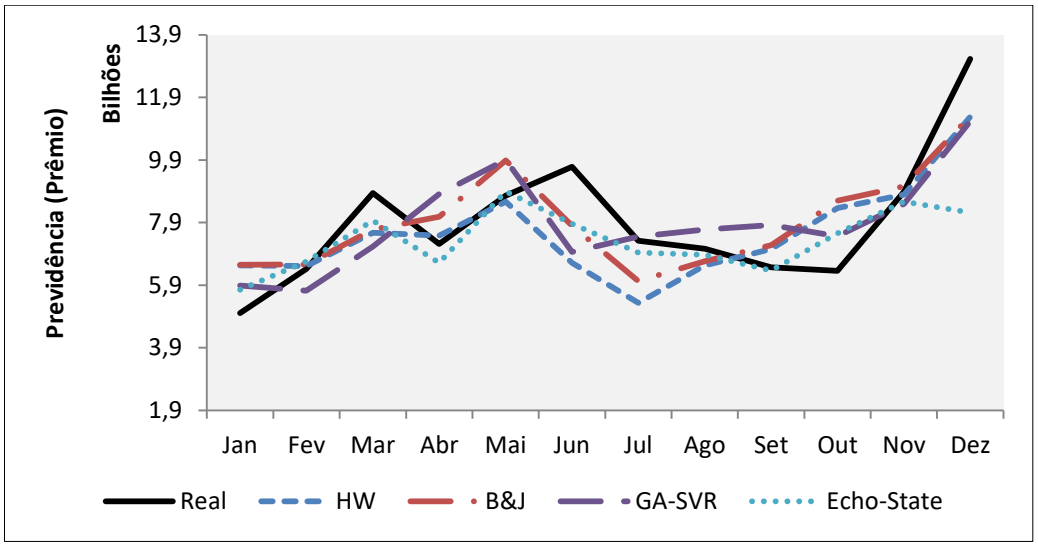

Figura 3: Previsão Univariada - Prêmio de Seguro Previdência

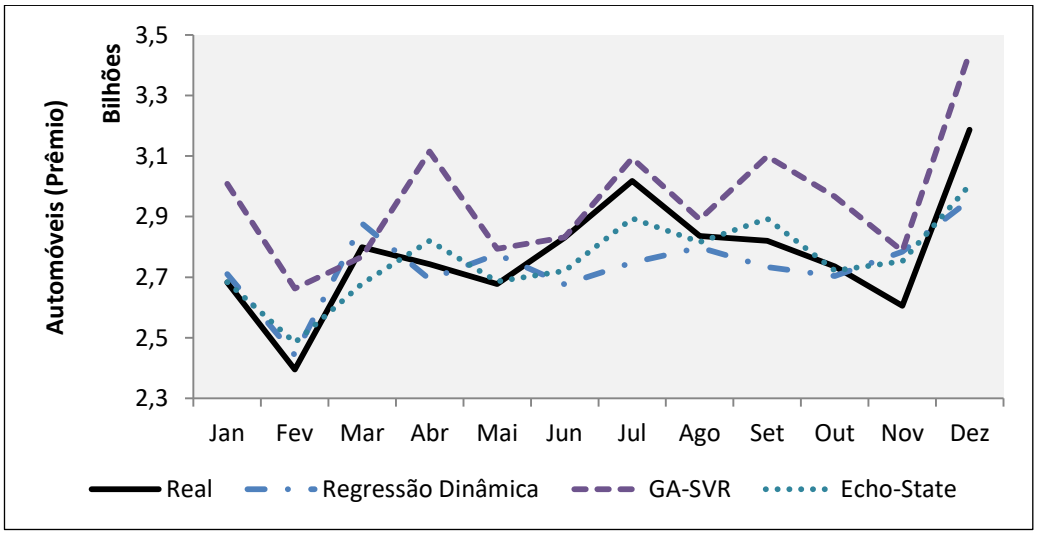

Figura 4: Previsão Multivariada - Prêmio de Seguro Automóveis

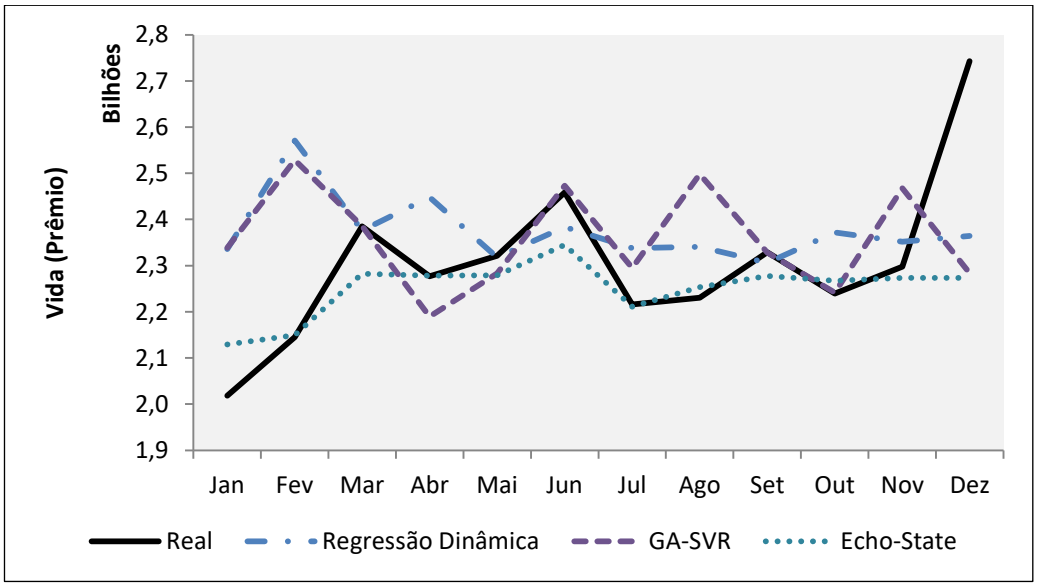

Figura 5: Previsão Multivariada - Prêmio de Seguro Vida

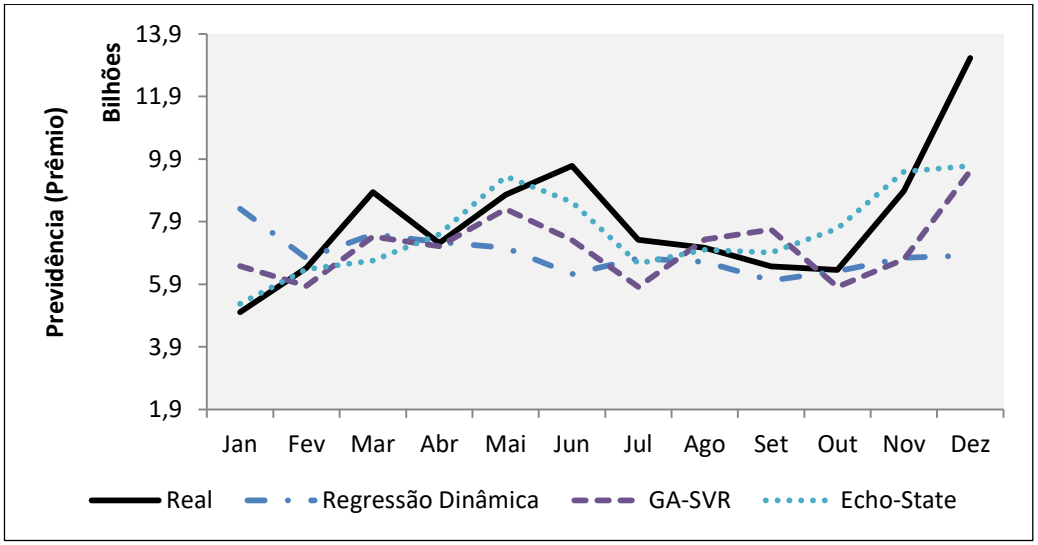

Figura 6: Previsão Multivariada - Prêmio de Seguro Previdência