DISSERTAÇÃO DE MESTRADO

PREVISÃO DE PRECIPITAÇÃO USANDO MÁQUINAS DE VETORES DE SUPORTE VISANDO SUA IMPLEMENTAÇÃO EM SISTEMAS EMBARCADOS

Everaldo José Rabêlo dos Santos

Brasília, junho de 2019

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

PREVISÃO DE PRECIPITAÇÃO USANDO MÁQUINAS DE VETORES DE SUPORTE VISANDO SUA IMPLEMENTAÇÃO EM SISTEMAS EMBARCADOS

Everaldo José Rabêlo dos Santos

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica como requisito parcial para obtenção do grau de Mestre em Sistemas Mecatrônicos

Banca Examinadora

Prof. Dr. Carlos H. Llanos Quintero, ENM/FT/UnB _____
Orientador

Dr. Antônio Marcos Mendonça, Analista Científico do CNPq _____
Examinador externo

Prof. Dr. Daniel M. Muñoz - FGA/UnB _____
Examinador interno

FICHA CATALOGRÁFICA

SANTOS, EVERALDO RABÊLO
PREVISÃO DE PRECIPITAÇÃO USANDO MÁQUINAS DE VETORES DE SUPORTE VISANDO SUA IMPLEMENTAÇÃO EM SISTEMAS EMBARCADOS [Distrito Federal] 2019.
xvi, 122 p., 210 x 297 mm (ENM/FT/UnB, Mestre, Engenharia Mecânica, 2019).
Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Mecânica

1. Meteorologia	2. Máquinas de Vetores de Suporte
3. Aprendizagem de Máquinas	4. Previsão de Precipitação
I. ENM/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

SANTOS, E.J.R. (2019). *PREVISÃO DE PRECIPITAÇÃO USANDO MÁQUINAS DE VETORES DE SUPORTE VISANDO SUA IMPLEMENTAÇÃO EM SISTEMAS EMBARCADOS*. Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 122 p.

CESSÃO DE DIREITOS

Everaldo José Rabêlo dos Santos
Depto. de Engenharia Mecânica (ENM) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

Dedicatória

A minha família, em especial minha esposa e meus filhos, sendo uma infinita motivação diária, sempre confiante em romper os meus limites imaginários com o intuito de atingir meus objetivos sem receios.
Ao meu maravilhoso núcleo familiar: minha mãe Ignês, e meus irmãos Evandro, Aurora e Evaldo que sempre estiveram e estão ao meu lado, sem se importarem com as condições externas, contribuindo, mesmo distante, continuamente para o fortalecimento dos laços familiares.
À minha tia Nazaré Assumpção e sua família, pelo total apoio dado à mim durante todas as minhas estadas em Brasília.
Finalmente saúdo também à todos os membros de minha família que apoiaram direta ou indiretamente meus projetos de vida.

Everaldo José Rabêlo dos Santos

Agradecimentos

Primeiramente aos meus orientadores Prof. Dr. Carlos Humberto Llanos Quintero e Profa. Dra. Maria Aurora Mota, pela excelência no apoio e orientação na condução deste trabalho. Sem eles esta etapa da minha vida acadêmica seria quase impossível.
Ao PPMEC da UNB, e especialmente ao professor Edson, por me ter dado a oportunidade de concluir esta etapa acadêmica.
A Faculdade de Meteorologia da UFPA pelo apoio nas duvidas e orientações nas questões de meteorologia, por mim pouco dominadas.
Ao IFPA por me ter dado a oportunidade de fazer concretizar um ideal de vida.
Ao PPMEC da UFBa, onde comecei esta etapa acadêmica.
Ao INMET pelo fornecimento dos dados utilizados nesta pesquisa.
Ao colega e amigo Prof. Dr. Carlos Santos, que ao longo do trabalho me prestou total apoio para conclusão desta pesquisa.
A todas as pessoas que de alguma maneira, direta e indiretamente, contribuíram para que este sonho pudesse hoje estar concluído, através de uma palavra, um abraço, uma caminhada. Detalhes que marcam as vidas de pessoas que convivem na mesma realidade.

Everaldo José Rabêlo dos Santos

RESUMO

A previsão e a evolução dos parâmetros climáticos (como radiação solar, índices de precipitações, temperatura atmosférica e da umidade relativa) são informações importantes para a sociedade e suas áreas de aplicação. As novas tecnologias (como estações meteorológicas inteligentes, colheitadeiras, robôs e drones) necessitam de informações atuais e futuras sobre as condições de clima e tempo de sua localidade, para melhorar a eficiência no uso de recursos e permitir o funcionamento sustentável destes dispositivos. Adicionalmente os indicadores de clima e tempo para a região amazônica apresentam baixa acurácia nas previsões com os modelos disponíveis, devido a fatores como a dinâmica tropical da região, instrumentação escassa e dificuldade logística da região-acesso e energia elétrica. Este trabalho apresenta uma proposta de aplicação de algoritmos de Inteligência Artificial (IA), para dispositivos inteligentes com sistemas embarcados, como: drones, robôs e estações meteorológicas, para previsão de classes de chuva (precipitação). Como classificador foi utilizado o algoritmo SVM, combinados com o algoritmos bio-inspirados - MOPSO e MODE. Para validação desta proposta foram utilizados dados reais de uma localidade da região amazônica, a cidade de Belém-PA. Essas técnicas foram propostas recentemente na literatura e apresentaram boa capacidade de previsão de chuva na Europa, China e India. Os resultados qualitativos e quantitativos, deste trabalho, demostraram que o desempenho destes algoritmos foram bons, e mostrou que seu desempenho depende das reais necessidades do problema a ser aplicado. Os modelos apresentaram uma boa eficiência computacional, para aplicações em sistemas embarcados, já que não requerem grandes recursos de hardware e software. Os modelos apresentaram, também, uma boa acurácia, boa precisão e recall eficientes, para as classes de precipitação utilizadas, podendo ser implementado para previsão de curto prazo com baixo custo computacional.

Palavras-chave: Máquinas de vetores de suporte. Aprendizagem de máquinas. Meteorologia. Previsão de Tempo, Classe de Chuvas.

ABSTRACT

Prediction and evolution of climate parameters (such as solar radiation, precipitation rates, atmospheric temperature and relative humidity) are important information for society and its application areas. New technologies (such as smart weather stations, harvesters, robots, and drones) need current and future information on your local weather and weather conditions to improve resource efficiency and enable these devices to function sustainably. Additionally, climate and weather indicators for the Amazon region present low accuracy in the predictions with the available models, due to factors such as the region's tropical dynamics, scarce instrumentation and logistical difficulty of the access region and electricity. This work presents a proposal for the application of Artificial Intelligence (AI) algorithms for intelligent devices in embedded systems, such as: drones, robots and meteorological stations, to predict rainfall classes. For the classifier the SVM algorithm was used, combined with the bio-inspired algorithms - MOPSO and MODE. To validate this proposal, real data were used from a locality in the Amazon region, the city of Belém-PA. These techniques were recently proposed in the literature and showed good rainfall prediction in Europe, China and India. The qualitative and quantitative results of this work demonstrated that the performance of these algorithms were good and showed that their performance depends on the real needs of the problem to be applied. The models presented a good computational efficiency, for applications in embedded systems, since they do not require great hardware and software resources. The models also presented a good accuracy, good precision and efficient recall for the precipitation classes used, being able to be implemented for short term prediction with low computational cost

Keywords: Support Vectors Machines, Machine Learning, Meteorology, Rain Prediction, Rain Classes.

SUMÁRIO

1 INTRODUÇÃO **1**
1.1 Contextualização 1
1.2 Definição do Problema 2
1.3 Aplicações da Solução Proposta 3
1.4 Objetivo Geral 3
1.5 Objetivos Específicos 3
1.6 Possíveis Impactos da Proposta 4
1.7 Apresentação do Documento 4

2 REFERENCIAL TEÓRICO E TRABALHOS CORRELATOS **5**
2.1 Processos e Modelos de Previsão em Meteorologia 5
2.1.1 O Processo Operacional de Previsão 5
2.1.2 Os Modelos Meteorológicos 7
2.2 Técnicas de Inteligência Artificial 8
2.2.1 Aprendizado de Máquina - ML 8
2.2.2 Máquinas de Vetores de Suporte - SVM 9
2.3 Máquinas de Vetores de Suporte para Classificaçãos 12
2.3.1 SVMs - Linearmente Separáveis 12
2.4 Algoritmos Bioinspirados 15
2.4.1 Algoritmo PSO 15
2.4.2 Algoritmo PSO básico 16
2.4.3 Algoritmo DE 17
2.4.4 Algoritmos MOPSO e MODE 18
2.5 A Ferramenta NIOTS Desenvolvida no LEIA-UnB 30
2.5.1 NIOTS: Ambiente Comitê de Máquinas - *Ensembles* 32
2.5.2 NIOTS: Ambiente de Otimização 32
2.5.3 NIOTS: Ambiente de Predição 36
2.5.4 NIOTS: Ambiente de Estatística 38
2.6 Sistemas Embarcados e Dispositivos Inteligentes 40
2.7 Inteligência Artificial na Meteorologia 41
2.8 Dispositivos Inteligentes na Meteorologia 44
2.9 Conclusões do Capítulo 45

3 METODOLOGIA APLICADA AO DESENVOLVIMENTO DESTA PESQUISA 46
3.1 Diretrizes do Trabalho . . . 46
3.1.1 Uso de dados locais . . . 46
3.1.2 Facilidade de Implementação . . . 47
3.1.3 Baixo Custo Computacional da Técnica a ser Desenvolvida . . . 47
3.1.4 Previsão de Precipitação para Período de 24 hs . . . 47
3.2 Visão Geral da Metodologia . . . 48
3.3 Etapa de Definição de Requisitos . . . 49
3.3.1 Definição das Classes . . . 49
3.4 Etapa de Aquisição de Dados . . . 49
3.5 Etapa de Análise dos Dados . . . 50
3.5.1 Análise de Dados Faltantes . . . 51
3.5.2 Análise Relevância das Variáveis . . . 51
3.5.3 Análise da Distribuição dos Dados . . . 52
3.5.4 Análise da Sazonalidade dos Dados . . . 52
3.6 Etapa de Preparação dos Dados . . . 52
3.6.1 Ordenamento dos Dados Brutos . . . 53
3.6.2 Carga da Tabela de Relevância das Variáveis . . . 53
3.6.3 Tratamento de Dados Incompletos . . . 53
3.6.4 Definição das Classes de Classificação . . . 53
3.6.5 Seleção da Janela de Previsão . . . 54
3.6.6 Cálculo da Precipitação Acumulada para Janela . . . 54
3.6.7 Aplicação das Classes de Precipitação . . . 54
3.6.8 Geração dos Dados . . . 54
3.7 Etapa de Treinamento . . . 55
3.7.1 Considerações sobre os Experimentos . . . 55
3.7.2 Métrica Utilizada . . . 55
3.8 Etapa de Previsão . . . 56
3.9 Etapa de Avaliação dos Resultados . . . 56
3.9.1 Avaliação por Especialista . . . 57
3.9.2 Gráficos Histograma . . . 57
3.9.3 Matriz de Confusão . . . 57
3.9.4 Curva ROC . . . 59
3.10 Etapa de Implementação do Algoritmo . . . 59
3.11 Conclusões do Capítulo . . . 60

4 ESTUDO DE CASO-PREVISÃO DE PRECIPITAÇÃO EM BELÉM 61
4.1 Local de Estudo . . . 61
4.2 Dados Utilizados . . . 62
4.2.1 Aquisição de Dados . . . 63
4.3 Análise dos Dados . . . 63

4.3.1 Dados Faltantes 64
4.3.2 Relevância de Variáveis 64
4.3.3 Avaliação com Especialista 69
4.4 Parametrização das Classes de Precipitação 70
4.4.1 Distribuição do Dados 71
4.4.2 Sazonalidade do Dados 72
4.5 Preparação dos Dados 72
4.5.1 Tratamento de Dados 73
4.5.2 Geração de Dados 75
4.6 Treinamento 75
4.6.1 Relatório do Treinamento 75
4.7 Previsão 76
4.8 Resultados da Previsão 77
4.8.1 Previsão com Algorítimo MODE 77
4.8.2 Previsão com Algorítimo MOPSO 86
4.9 Avaliação dos Resultados 94
4.9.1 Ranking dos Resultados Quantitativos - Histogramas 94
4.9.2 Ranking dos Resultados Qualitativos - Matriz de Confusão 97
4.9.3 Ranking dos Resultados - Curvas ROC 104
4.10 Conclusões do Capítulo 107

5 CONCLUSÃO **109**

REFERÊNCIAS BIBLIOGRÁFICAS **111**

APÊNDICES **115**
A Máquinas de Vetores de Suporte para Regressão - SVR 115
B Programa Gerador de Dados 117
C Programa Relatório de Resultados 121

LISTA DE FIGURAS

2.1 O Processo Operacional de Previsão-fonte (NOVAK, 2015) 6
2.2 Estrutura dos Modelos Numéricos *fonte-(SAMPAIO; SILVA DIAS, 2014)* 7
2.3 Tipos de Aprendizado de Maquinas *(ML)*- Fonte (TECHSPARKS, 2019) 8
2.4 Gráfico SVM*(fonte:(WIKIPEDIA,))* 10
2.5 SVM - Margens rígidas. 13
2.6 Diagrama de fluxo do algoritmo MOPSO. 20
2.7 Comparação entre SLHD e distribuição uniforme. 22
2.8 Diagrama de fluxo do algoritmo APMT-MODE - Fonte (SANTOS, 2019) 25
2.9 Comportamento da variância σ. 26
2.10 Exemplo de modelo da Fronteira de Pareto. 31
2.11 Relatório contendo o Conjunto e a Fronteira de Pareto. 32
2.12 Diagrama de fluxo de dados e opções do NIOTS. Fonte (SANTOS, 2019) 33
2.13 Ambiente gráfico NIOTS-Otimização. Fonte (SANTOS, 2019) 34
2.14 Ambiente gráfico NIOTS-Ensembles. Fonte (SANTOS, 2019) 34
2.15 Ambiente gráfico NIOTS - Otimização. Fonte (SANTOS, 2019). 37
2.16 Gráficos gerados pela opção *Correlation Plot*. Fonte (SANTOS, 2019) 38
2.17 Coeficiente de Correlação. 38
2.18 Gráfico da métrica ROC. 39
2.19 Ambiente gráfico NIOTS - Estatística. Fonte (SANTOS, 2019) 40
2.20 Distribuição dos Algorítimos 43
2.21 Utilização de Sistemas Embarcados 43
2.22 Distribuição do Uso de Dados para Previsão 44
2.23 EstacaoAutomatica*(Fonte : Vaisala, 2019 Weather Station)* 44

3.1 Ação da Máquina de Previsão- Variáveis x Tempo-*fonte: o autor.* 47
3.2 Etapas da Metodologia - *Fonte : o autor* 48
3.3 Fluxo de Preparação dos Dados 53
3.4 Matriz de Confusão - Fonte :(TING, 2010) 57
3.5 Curva ROC 59
3.6 Fluxo de Dados Simplificado em uma Estação Inteligente-fonte:(MESTRE et al., 2015) 60

4.1 Localização Geográfica de Belém-Pa 62
4.2 Localização da Estação Meteorologia do INMET Belém 62

4.3 Gráfico Dispersão PCA-Scatter Plot 66
4.4 Gráfico de Cargas - PC1 67
4.5 Gráfico de Cargas - PC1 68
4.6 Distribuição dos Dados por Classe de Precipitação 71
4.7 Variação da PRP mensal Média Climatológica, período de 30 anos (barra azul) e Média Histórica de 118 anos (barra verde) em Belém-PA.Fonte: SANTOS, J.S. (SANTOS et al., 2014) 72
4.8 Distribuição de Dados Gerados por Classe de Precipitação. 75
4.9 Relatório NIOTS-Treinamento 76
4.10 Parâmetros para previsão 77
4.11 Histograma de Previsão 24 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 78
4.12 Histograma de Previsão 48 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 79
4.13 Histograma de Previsão 72 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 80
4.14 Matriz de Confusão Algorítimo MODE-24 horas. 81
4.15 Matriz de Confusão Algorítimo MODE-48 horas. 82
4.16 Matriz de Confusão Algorítimo MODE-72 horas. 83
4.17 Curvas ROC Algorítimo MODE-Previsão 24 hs. 83
4.18 Curvas ROC Algorítimo MODE-Previsão 48 hs. 85
4.19 Curvas ROC Algorítimo MODE-Previsão 72 hs. 86
4.20 Histograma de Previsão 24 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 87
4.21 Histograma de Previsão 48 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 88
4.22 Histograma de Previsão 72 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 89
4.23 Matriz de Confusão Algorítimo MOPSO-24 horas. 90
4.24 Matriz de Confusão Algorítimo MOPSO-48 horas. 90
4.25 Matriz de Confusão Algorítimo MOPSO-72 horas. 91
4.26 Curvas ROC Algorítimo MOPSO-Previsão 24 hs. 92
4.27 Curvas ROC Algorítimo MOPSO-Previsão 48 hs. 93
4.28 Curvas ROC Algorítimo MOPSO-Previsão 72 hs. 94

A.1 Representação da função ϵ-insensitive, função aproximadora $\hat{f}(x)$. Adaptado de (SMOLA; SCHÖLKOPF, 2004). 116

LISTA DE TABELAS

2.1 Comparação da métrica *diversidade* entre distribuição uniforme e SLHD. 22
2.2 Descrição das opções do ambiente Otimização. 34
2.3 Artigos Pesquisados 42

3.1 Classes de Precipitação 49
3.2 Variáveis Meteorológicas das Estações Automatizadas - Fonte INMET 50

4.1 Lay-out do arquivo das Estação Automatizada do INMET 63
4.2 Distribuição de Dados Tratados 64
4.3 Distribuição de Carga PCA 65
4.4 Classificação das Variaveis - PC1 67
4.5 Classificação das Variaveis - PC2 69
4.6 Tabela de Relevância 70
4.7 Parâmetros das Classes de Precipitação - Acumulado de 24 Horas 71
4.8 Distribuição por Classe de Precipitação 71
4.9 Lay-out do arquivo de Saida - Etapa Tratamento de Dados 74
4.10 Erro Relativo de Previsão 24 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 78
4.11 Erro Relativo - Previsão 48 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 79
4.12 Erro Relativo - Previsão 72 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial 80
4.13 AUC das Curvas ROC Algorítimo MODE-Previsão 24 hs. 84
4.14 AUC das Curvas ROC Algorítimo MODE-Previsão 48 hs. 85
4.15 AUC das Curvas ROC Algorítimo MODE-Previsão 72 hs. 86
4.16 Erro Relativo - Previsão 24 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 87
4.17 Erro Relativo - Previsão 48 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 88
4.18 Erro Relativo - Previsão 72 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial 89
4.19 AUC das Curvas ROC Algorítimo MOPSO-Previsão 24 hs. 92
4.20 AUC das Curvas ROC Algorítimo MOPSO-Previsão 48 hs. 93
4.21 AUC das Curvas ROC Algorítimo MOPSO-Previsão 72 hs. 94

4.22 Ranking de Histogramas Previsão 24 hs 95
4.23 Ranking de Histogramas Previsão 48 hs 96
4.24 Ranking de Histogramas Previsão 72 hs 97
4.25 Ranking de Acurácia - Previsão 24 horas 98
4.26 Ranking de Acurácia - Previsão 48 horas 98
4.27 Ranking de Acurácia - Previsão 72 horas 98
4.28 Ranking de Precisão - Previsão 24 horas 99
4.29 Ranking de Precisão - Previsão 48 horas 100
4.30 Ranking de Precisão - Previsão 72 horas 101
4.31 Ranking de Recall - Previsão 24 horas 102
4.32 Ranking de Recall - Previsão 48 horas 103
4.33 Ranking de Recall - Previsão 72 horas 104
4.34 Ranking de ROC (AUC) - Previsão 24 horas 105
4.35 Ranking de ROC (AUC) - Previsão 48 horas 106
4.36 Ranking de ROC (AUC) - Previsão 72 horas 107

LISTA DE QUADROS

1 Pseudocódigo do PSO básico. 17
2 Pseudocódigo do algoritmo DE. 18
3 Pseudocódigo do algoritmo SLHD. 21

LISTA DE SÍMBOLOS

Símbolos Latinos

K	Matriz ou função *kernel*
N	Cardinalidade do conjunto de treinamento
C	Parâmetro de regularização
$\mathbf{w}$	Vetor diretor do hiperplano classificador
$\mathbf{x}$	Vetor caracterí stica do dado de treinamento
b	*bias*

Símbolos Gregos

α_i	i-é sima variável Lagrangeana
ξ_i	i-é sima variável de folga do problema de otimização
Φ	Função de transformação
γ	Parâmetro do *kernel* gaussiano
β	Coeficiente do *kernel* polinomial
κ	Termo independente do *kernel* polinomial
π	Constante irracional pi
δ	Parâmetro do *kernel* sigmoide
ρ	Parâmetro do *kernel* sigmoide
ϵ	Largura do tubo ϵ-*insensitive*

Grupos Adimensionais

e	Número de Euler

Sobrescritos

α^*	Variável de decisão ótima
$\hat{f}$	Preditor, função aproximadora f
$\bar{y}$	Média da variável y

Siglas

ABNT	Associação Brasileira de Normas Técnicas
AG	Algoritmo Genético
AP-MODE	*Adaptive Parameter Multi-Objective Differential Evolution*
APMT-MODE	*Adaptive Parameter with Mutant Tournament* - MODE
AR	Atrativo Repulsivo
AUC	*Area Under Curve*
CLO	*Criticism of Lexicographic Ordering*
DE	*Differential Evolution*
EA	*Evolutionary Algorithm*
FP	Falso positivo
FPGA	*Field Programmable Gates Arrays*
FSM	*Finite State Machines*
GRACO	Grupo de Automação e Controle
GS	*Grid Search*
GUIDE	*Grafical User Interface Development Environment*
H-C	Hermite-Chebyshev
H-C	Hermite-Gaussian
HDL	*Hardware Description Language*
INMET	Instituto Nacional de Meteorologia
KKT	Karush-Kuhn-Tucker
LEIA	*Laboratory of Embedded Systems and Integrated Circuits Applications*
MAC	*Multiply-Accumulate*
MCGAs	Modelos de Circulação Geral da Atmosfera
MODE	*Multi-Objective Differential Evolution*
MOOP	*Multi-objective Optimization Problem*
MOPSO	*Multi-Objective Particle Swarm Optimization*
MOUD	*Muti-objective uniform design*
MSE	*Mean Squared Error*
NIOTS	*Nature Inspired Optimization Tools for SVM*
NP-Completo	Não Polinomial Completo
NSGA-II	*Non Sorting Genetic Algorithm II*
OAO	*One Against One*
OBL	*Oposition Based Learning*
PF	*Pareto Front*
PD	Problema de Decisão
PO	Pareto Ótimo
PSM	Problema de Seleção de Modelos
PSO	*Particle Swarm Optimization*
PSP	Problema de Seleção de Parâmetros
RBF	*Radial Basis Function*

Siglas

RNA	Redes Neurais Artificiais
ROC	*Receiver Operating Characteristic*
SA	*Simulated Annealing*
SLHD	*Symmetric Latin Hypercube Design*
SMO	*Sequential Minimal Optimization*
SRM	*Structural Risk Minimization*
SV	*Support Vectors*
SVM	*Support Vectors Machine*
SVR	*Support Vectors Regressor*
T	Turing
TIC	Tecnologia da Informação e Comunicação
UCI	*University of California, Irvine*
VHDL	*Very High Description Language*
VC	Vapnik Chervonenkis
ZCIT	Zona de Convergência Intertropical

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A medição de parâmetros climáticos tais como radiação solar, índices de precipitações, temperatura atmosférica e da umidade relativa, bem como a disponibilidade de previsões de sua evolução, ao longo do tempo, são informações importantes para a sociedade e suas áreas de aplicação, tais como agricultura, energia renovável e gerenciamento de energia, ou conforto térmico pessoal.

Cada vez mais as tecnologias (colheitadeiras, robôs, drones e outros tipos de dispositivos) necessitam de informações atuais e futuras sobre as condições de clima e tempo de sua localidade, a fim de melhorar a eficiência no uso de recursos e também permitir o funcionamento sustentável desses dispositivos. Então as medidas de clima e tempo terão, cada dia mais, um impacto significativo nas tecnologias nos próximos anos. Assim, as previsões de tempo e clima de curto prazo desempenham um papel estratégico, ao dar suporte às tecnologias emergentes.

Por outro lado, os indicadores de clima e tempo para a região amazônica apresentam baixa acurácia nas previsões com os modelos numéricos correntes, devido a diversos fatores tais como a dinâmica tropical da região, instrumentação escassa e dificuldade logística da região-acesso e energia elétrica, entre outras.

Este trabalho visa aplicar uma técnica de IA, especificamente de Aprendizado de Máquina (*Machine-Learning*) para previsão de *classes de precipitação*. Tais técnicas, recentemente propostas na literatura, vêm apresentando uma habilidade de previsão de tempo e clima para diversas localidades na China, Índia e Europa.

A utilização de algoritmos simples, como SVM (*Support Vector Machine*) em sistemas embarcados, poderá permitir maior autonomia dos dispositivos inteligentes, tais como drones e robôs; deixando que os mesmos, por exemplo, decidam se realizam ou não um vôo, tarefa ou missão. Criar soluções de baixo consumo de recursos computacionais, tal como é o caso dos sistemas embarcados, visando desenvolver dispositivos inteligentes, é uma necessidade para as atuais metas do Brasil e da sociedade.

A utilização de *classificadores* a fim de fazer uma previsão de chuvas mais eficiente, neste contexto, tem por objetivo classificar as chuvas por *tipos de precipitação*, o qual apresenta uma nova visão para previsões que tenham restrições nos dados, especificamente quando se utilizam somente dados locais. Este aspecto é importante para ser seriamente pesquisado, tendo em conta uma carência na pesquisa de técnicas de previsão de chuvas para pequenas estações meteorológicas, como aquelas requisitadas na região amazônica.

Adicionalmente, nas condições supracitadas (para a região amazônica) os classificadores a

serem desenvolvidos devem apresentar características de simplicidade computacional, tendo em conta que objetiva-se embarcar os mesmos em sistemas computacionais com características de tempo real, e com restrições em desempenho computacional, consumo de energia, custos, condições adversas (temperatura, chuva, etc) como aquelas associadas a plataformas computacionais denominadas de *Sistemas Embarcados.* (descritos em 2.6)

Nos últimos anos foram desenvolvidas técnicas de previsão do tempo utilizando modelos meteorológicos de clima e tempo, e Aprendizado de Máquina (*Machine-Learning*) com regressores, estes últimos, na sua maioria usando Redes Neurais Artificiais (RNAs) ((MESTRE et al., 2015), Máquinas de Vetores Suporte (SVMs) ((CHEN; CHANG; LIN, 2004) e (SIVAPRAGASAM; LIONG; PASHA, 2001)) e Regressores Lineares (LR) (ZHENG et al., 2015). Todavia, poucos são os modelos que usam classes de chuva, pois, normalmente são os modelos climáticos que se utilizam deste recurso, por se tratar de previsão de longo prazo (de meses a anos). Enquanto que técnicas de Aprendizado de Máquina (*Machine-Learning*) são usadas para desenvolver regressores para previsão de precipitação ((SIVAPRAGASAM; LIONG; PASHA, 2001)).

Apesar de existirem estudos sobre classificação de precipitação (chuva), como em (LLASAT, 2001) e (SOUZA; AZEVEDO; ARAÚJO, 2012), eles são desenvolvidos para análise e comparação entre as variáveis (por exemplo, temperatura, pressão e umidade) e suas consequências para sociedade e ambiente, porém nenhum emprega técnicas de *Aprendizado de Máquina*, como que está sendo proposto nesta pesquisa.

1.2 DEFINIÇÃO DO PROBLEMA

A utilização de dados de clima e tempo nas áreas da Amazônia é de vital importância a fim de evitar grandes perdas de recursos e possíveis catástrofes. Com os dados de previsões de clima e tempo pode-se, por exemplo, evitar a perda de minérios de ferro, ao longo do seu transporte, já que suas viagens levam dias de duração, e são feitas a céu aberto. Outro exemplo é a utilização em estações meteorológicas remotas, para previsão de possíveis tempestades (eventos extremos de precipitações) em locais de riscos, tais como encostas e áreas ribeirinhas.

A maioria das soluções disponíveis no mercado, para previsão de clima e tempo, utiliza não somente dados locais, mas também dados globais (por exemplo, índices *El Niño* e *La Niña*). Entretanto, tais dados estão poucos disponíveis em dispositivos inteligentes que estão trabalhando em ambiente adverso (com restrições de fontes de energia elétrica e sem acesso à internet).

Outro ponto a considerar, é o fato da maioria das informações de clima e tempo são baseadas nos dados para uma grande área de cobertura (120 km^2), o que ocasiona pouca precisão e grandes falhas nas previsões para pequenas localidades. Assim soluções com uso de poucos recursos computacionais e trabalhando somente com os dados locais disponíveis para o dispositivo, tendem a ganhar importância e utilidade.

Existem alguns modelos e técnicas de classificação estudadas na área de *Aprendizado de Má-*

quina ((MESTRE et al., 2015), (HONG, 2008) e (ZHENG et al., 2015)), as quais utilizam as Redes Neurais Artificiais (RNAs), Máquinas de Vetores de Suporte (SVMs - Support Vector Machines) e regressão linear (LR). Entretanto, em sua grande maioria, são dedicadas a trabalhar principalmente com dados globais.

1.3 APLICAÇÕES DA SOLUÇÃO PROPOSTA

Considerando o problema descrito, esta pesquisa coloca, como uma primeira hipótese, que é possível implementar técnicas de previsão de eventos de precipitação (chuvas) para serem utilizadas em estações meteorológicas de poucos recursos computacionais, trabalhando com dados locais, usando uma abordagem de classificação por tipos de precipitação, utilizando técnicas de *Machine-Learning*.

A segunda hipótese é que estas técnicas podem ser direcionadas a serem implementadas em *sistemas embarcados*, os quais exigem pouca complexidade computacional dos algoritmos, tendo em conta suas restrições em desempenho: (a) baixa frequência de *clock*, (b) baixo consumo de energia (a fim de garantir sua autonomia, caso precisem ser alimentados por bancos de baterias), e (c) baixo custo.

Como foi sugerido anteriormente, as técnicas desenvolvidas podem ser utilizadas para monitoração de transportes de minérios em linhas de trem, apoio a navegação de robôs e apoio a vôo de drones, para maior autonomia destes dispositivos.

1.4 OBJETIVO GERAL

O objetivo geral desta dissertação é desenvolver um protótipo de uma solução de *software* para utilização em estação meteorológica inteligente, capaz de realizar previsão de *classes de precipitação*, utilizando técnicas de *Aprendizado de Máquina*. Especificamente, serão utilizadas técnicas baseadas em SVMs, cujos parâmetros são sintonizados mediante a utilização de metaheurísticas bio-inspiradas. O modelo tem que ser simples e eficiente, a fim de poder ser embarcado em pequenas plataformas computacionais, tais como os sistemas embarcados, envolvendo sérias restrições de recursos computacionais tais como baixas taxas de *clock*, baixo consumo de energia e baixos recursos de memória

1.5 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral no presente trabalho, foram estabelecidos os objetivos específicos a seguir:

- Analisar dados de precipitação, com base em informações locais das estações meteorológicas.
- Avaliar a eficiência dos algoritmos de IA para aplicação em problemas de previsão de classes de precipitação.
- Comparar e validar os resultados das previsões geradas em relação a dados reais.
- Gerar um modelo de classificação usando SVMs para previsão de classes de precipitação.

1.6 POSSÍVEIS IMPACTOS DA PROPOSTA

As dificuldades econômicas, cientificas e sociais na Amazônia são comprovadamente enormes, pois, os obstáculos são maiores e mais numerosos do que nas demais regiões brasileiras. A Amazônia é uma região vasta, pouco conhecida e onde o clima e o ambiente são nada amigáveis.

A implementação de uma solução computacional deste tipo, poderá causar grande impacto, para comunidade cientifica, pois, irá gerar maior volume de dados e um significativo aumento dos pontos de coleta de informações de clima e tempo da região. Permitindo assim uma melhor presivibilidade para as ações dos trabalhos científicos e de pesquisa nestas áreas. Além disto, irá provocar uma redução de custos operacionais para empresas e instituições, pelo fato de permitir um melhor aproveitamento dos fatores climáticos e de tempo da região, permitindo um melhor planejamento para as atividades das empresas, como, por exemplo, viagens e transportes, normalmente feitas em longos percursos e condições adversas. Finalmente, os resultados sociais para a população, com a implementação de uma solução desta, permitira um melhor planejamento para os impactos dos eventos extremos, causadores das enchentes e deslizamentos que afetam diretamente as populações ribeirinhas, possibilitando assim, uma utilização destas estações pelas autoridades locais e possivelmente pela própria população.

1.7 APRESENTAÇÃO DO DOCUMENTO

O presente trabalho está composto por cinco capítulos, incluindo esta introdução (1, e três apêndices. Inicialmente é apresentada uma revisão de literatura com trabalhos correlatos 2. Em seguida é apresentada a metodologia proposta para o desenvolvimento da pesquisa (3). No capitulo quatro (4) é feito um estudo de caso, considerando as precipitações ocorridas na cidade de Belém-Pa. A conclusão do trabalho é descrita no capitulo cinco (5) com sugestão para continuidade da pesquisa, bem como a aplicabilidade da solução proposta.

Nos apêndices são detalhados os aplicativos utilizados e algoritmos utilizados, apresentando seus códigos e descrição.

2 REFERENCIAL TEÓRICO E TRABALHOS CORRELATOS

Neste capítulo são apresentados a fundamentação teórica e o estado da arte associados à implementação da técnicas de *Aprendizado de Máquina* em dispositivos inteligentes em meteorologia. Assim, é apresentada toda base teórica utilizada neste trabalho, bem com as publicações de IA, associado ao *Aprendizado de Máquina* envolvendo aspectos de meteorologia, para sistemas embarcados.

2.1 PROCESSOS E MODELOS DE PREVISÃO EM METEOROLOGIA

Conforme a Enciclopédia de Ciências Atmosféricas - Segunda Edição (NOVAK, 2015), no início dos anos 1950, o grupo de meteorologia de Princeton já tinha completado as análises matemáticas necessárias e desenhado um algoritmo numérico para resolver o sistema de equações um pouco mais complexo que o utilizado por Charney (1955) e Von Neumann (1950) (o chamado sistema quase-geostrófico). Arranjos foram feitos para fazer uma integração do conjunto de equações no Eniac em Aberdeen, Maryland, EUA. Foram realizadas quatro previsões de 24 horas, e os resultados claramente indicaram que poderiam ser previstos os padrões de larga escala (da ordem de poucos milhares de quilômetros) do fluxo da média troposfera (cerca de 5.000 m). Entretanto, cada integração de 24 horas levava cerca de 24 horas para ser realizada no Eniac, ou seja, sem valor prático, porém muito úteis do ponto de vista teórico. As investigações seguiram também na linha de estudos de inicialização (determinação do estado inicial da atmosfera), tais como os estudos de Charney (1955) e Phillips (1960), dentre outros.

2.1.1 O Processo Operacional de Previsão

A previsão operacional moderna envolve múltiplos aspectos, que são mostrados esquematicamente na Figura 2.1. Tudo começa com observações da atmosfera, oceanos e superfície terrestre (NOVAK, 2015).

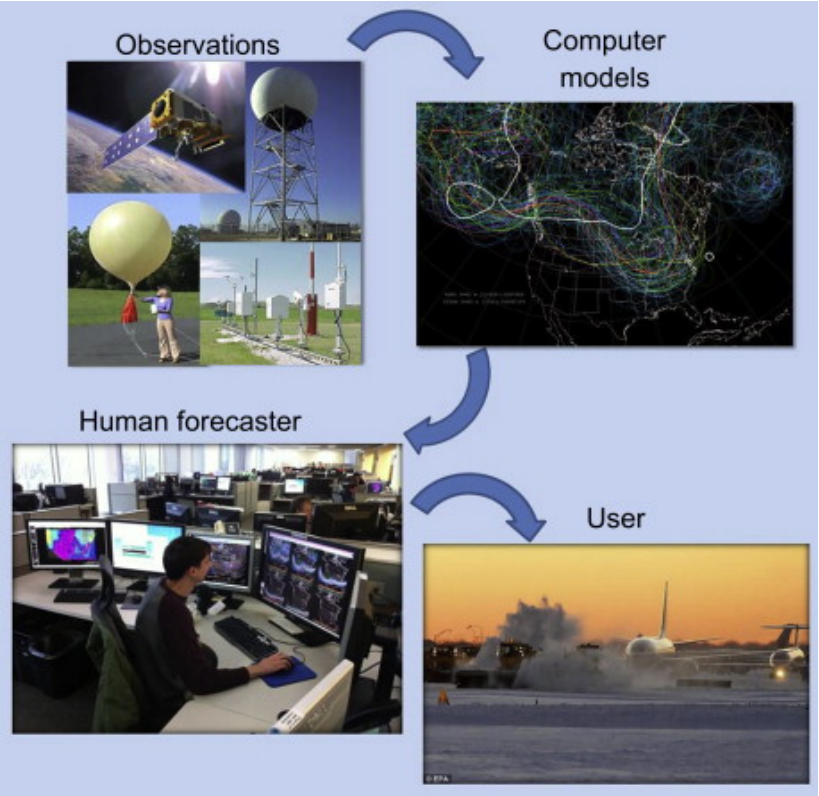


Figura 2.1 – O Processo Operacional de Previsão-fonte (NOVAK, 2015)

As observações são coletadas de plataformas *in situ*, como estações meteorológicas de superfície e balões meteorológicos, ou plataformas de sensoriamento remoto, como radares Doppler e de dupla polarização e sofisticados sistemas de satélite. Essas observações são então convertidas em uma forma que os modelos computacionais numéricos possam usar. Esses modelos de computador são baseados nas leis físicas da ciência atmosférica. Os modelos atmosféricos são extremamente complexos e prevêem variáveis como: (a) temperatura, (b) umidade, e (c) vento em todo o mundo, em incrementos de tempo na ordem de dezenas de segundos. Esses modelos devem considerar fatores como cobertura de neve, umidade do solo, vegetação, temperatura da superfície do mar, radiação solar e muitos outros. Portanto, esses modelos são computacionalmente intensivos e testam continuamente os limites da ciência da computação, pela alta complexidade computacional dos algoritmos envolvidos (NOVAK, 2015).

A saída desses modelos de computador é visualizada e examinada por meteorologistas previsores, que usam seu conhecimento especializado para fazer modificações. Por exemplo, o modelo pode mostrar a precipitação esperada para a tarde ou para amanhã. Assim, o previsor considera fatores como o viés de modelo, realismo físico, verificação passada e consenso entre vários sistemas de modelagem para desenvolver uma previsão mais provável. Os procedimentos e a forma em que as previsões são geradas variam em todo o mundo, mas está se tornando uma prática comum criar bancos de dados de grande porte, contendo elementos meteorológicos de superfí-

cie, que podem ser exibidos graficamente e consultados para criar descrições de texto narrativo (NOVAK, 2015).

2.1.2 Os Modelos Meteorológicos

Os modelos de circulação geral da atmosfera (MCGAs) possibilitam prever as condições do tempo para vários dias, dependendo da região e do estado da atmosfera, com alto grau de confiança, de até 7 a 12 dias (SAMPAIO; SILVA DIAS, 2014).

A meteorologia se utiliza de diversas técnicas para realizar previsões de clima e tempo. Tais técnicas podem ser divididas em 2 grupos: (a) previsão de tempo e (b) previsão climática. O modelo de previsão de tempo utiliza definições físicas que são distribuídas em pontos de grade para cada região ou para o globo terrestre. Então, o sistema de equações de um modelo meteorológico pode ser discretizado nas quatro dimensões (latitude, longitude, altitude – os chamados "pontos de grade" – e tempo). Os resultados de um modelo de circulação geral da atmosfera, ou seja, as previsões, são apresentados em pontos de grade, conforme apresentado na Figura 2.2, (SAMPAIO; SILVA DIAS, 2014).

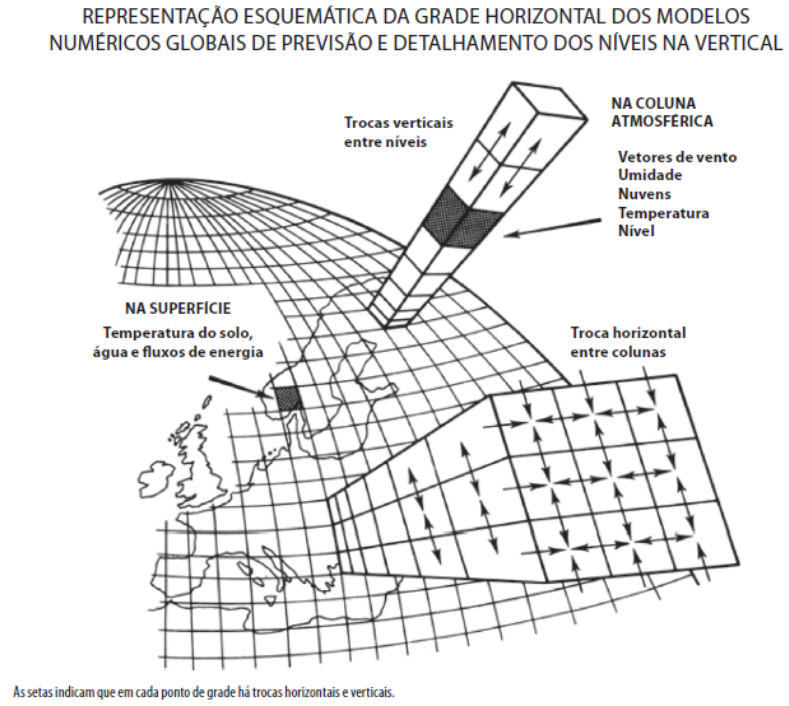


Figura 2.2 – Estrutura dos Modelos Numéricos *fonte-(SAMPAIO; SILVA DIAS, 2014)*

A base científica para as previsões numéricas de clima deriva, principalmente, da previsibilidade das condições de contorno, sobretudo da temperatura da superfície do mar (TSM) e da grande influência desta na determinação das condições atmosféricas futuras. Ou seja, a variabili-

dade climática sazonal é controlada, principalmente, pelas lentas variações das temperaturas dos oceanos (SAMPAIO; SILVA DIAS, 2014).

Buscando uma melhor acurácia nos modelos meteorológicos, técnicas computacionais, como as de Inteligências artificial, tem sido aplicadas.

2.2 TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) está mudando fundamentalmente a maneira como as empresas operam em todos os setores, incluindo áreas como manufatura, saúde, TI e transporte. Os avanços na IA na última década estão apresentando oportunidades para as empresas automatizarem os processos de negócios, e transformarem as experiências dos clientes, diferenciando as ofertas de produtos. Pioneiros da IA como o Google e a Amazon, que adotaram essas novas tecnologias para criar uma vantagem competitiva crescente, já testemunharam os benefícios finais de suas estratégias de inteligência artificial (IEEE, 2018).

O crescimento de aplicação das técnicas computacionais de IA (Inteligência Artificial) em diversas áreas como meteorologia, robótica, medicina e engenharia aeronáutica, tem motivado o aumento constante das pesquisas e técnicas relacionadas a este assunto. Neste caso particular, a computação evolucionária com seus Algoritmos Evolucionários (EA - *Evolutionary algorithm*) e a aplicação das técnicas de otimização avançam de forma crescente para atender estas necessidades (EIBEN; SMITH et al., 2003).

2.2.1 Aprendizado de Máquina - ML

O Aprendizado de Máquina (*Machine Learning* - ML), é um dos segmentos de pesquisa da IA. O mesmo é dividido em dois métodos de aprendizado, tal como apresentado na Figura 2.3: (a) aprendizado supervisionado e (b) aprendizado não-supervisionado. As definições para os dois tipos são dadas a seguir:

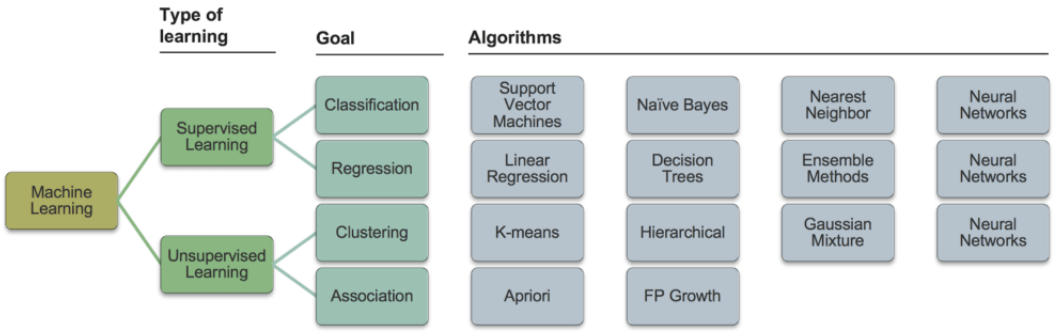


Figura 2.3 – Tipos de Aprendizado de Maquinas *(ML)*- Fonte (TECHSPARKS, 2019)

Aprendizado supervisionado: usa um conjunto de dados conhecido para fazer inferências com base em dados de entrada e saída rotulados.

Aprendizado não supervisionado: extrai inferências de conjuntos de dados contendo dados sem saídas rotuladas.

O método mais comum no trabalho de hoje em dia é o aprendizado supervisionado, enquanto que o aprendizado não supervisionado mostra grande promessa para aplicações mais amplas, que envolvem a geração de *clusters* dos dados e/ou caracterização de redundâncias nos mesmos, entre outros. Dentro de cada método de aprendizado, existem várias categorias de algoritmos que podem ser selecionados dependendo da aplicação. As decisões aqui variam dependendo do tipo de problema ou do resultado desejado.

No presente trabalho foi escolhido a técnica de Máquinas de Vetores de Suporte (SVM - *Support Vector Machine*), em vez de outros métodos de aprendizado supervisionados, devido à sua boa capacidade de generalização, boa precisão, e capacidade de lidar com a não linearidade nos dados. Adicionalmente, SVMs fornecem um modelo matemático explícito do problema (modelo de caixa branca), ao contrário das Redes Neurais Artificiais (ANNs - *Artificial Neural Network*) que são consideradas modelos de caixa preta. Adicionalmente, no LEIA (Laboratório de Sistemas Embarcados e Aplicações de Circuitos Integrados) uma ferramenta denominada de NIOTS, descrita em 2.5, foi desenvolvida, a qual consegue sintonizar os parâmetros de configuração de uma SVM dependendo dos dados ligados a uma aplicação específica.

2.2.2 Máquinas de Vetores de Suporte - SVM

Uma Máquina de Vetores de Suporte (SVM) é um conceito na ciência da computação para um conjunto de métodos do aprendizado supervisionado que analisam os dados e reconhecem padrões, visando gerar um processo eficiente para classificação e/ou análise de regressão.

As SVMs foram desenvolvidas para terem uma baixa utilização de recursos e um bom desempenho, razão pelos quais vem ganhando popularidade. As SVMs foram primeiramente desenvolvidas para resolver o problema de classificação e, em seguida, estenderam-se a problemas de regressão (SCHÖLKOPF; BURGES; VAPNIK, 1996).

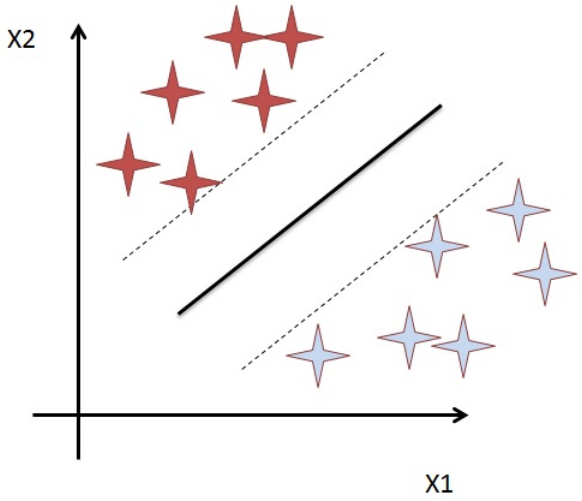


Figura 2.4 – Gráfico SVM*(fonte:(WIKIPEDIA,))*

Tendo como entrada um conjunto de dados, o SVM padrão, prediz para qual, das duas possíveis classes, cada ocorrência pertence. Isto torna as SVMs um classificador linear binário não probabilístico. Como um SVM é um classificador, ele recebe um conjunto de exemplos de treinamento, cada um marcado como pertencendo a uma das duas classes. Assim, um algoritmo de treinamento para um SVM constrói um modelo que atribui novos exemplos em uma categoria ou o de outros. Intuitivamente, um modelo SVM é uma representação das amostras das classes como pontos no espaço (Figura 2.4), mapeadas para que as amostras sejam separadas por uma lacuna clara, que deve ser a mais ampla possível. Novos exemplos são então mapeados no mesmo espaço e previstos para pertencer a uma categoria, com base em qual lado da lacuna eles caem.

As SVMs são obtidas a partir de um subconjunto finito de dados conhecidos e de seus respectivos rótulos (conjunto de treinamento), que, por meio de uma técnica de indução, buscam classificar dados que não fazem parte do conjunto de treinamento. Portanto, a SVM é uma função induzida a partir de um conjunto de treinamento, capaz de separar os dados em classes, baseada na informação fornecida pelo conjunto de treinamento (LIMA, 2004),(BURGES, 1998).

O processo de aprendizagem ocorre dependendo da forma como o algoritmo de indução atua sobre o conjunto de treinamento. Este processo é dividido em três paradigmas de aprendizagem: (a) *supervisionada*, (b) *não supervisionada* e (c) *por reforço*. Na aprendizagem supervisionada, o processo ocorre com auxílio de um professor externo, que possui conhecimento sobre os dados de treinamento e seus respectivos rótulos e atua avaliando o erro cometido e corrigindo-os, aprimorando a função classificadora em cada etapa. No processo não supervisionado, os dados não são rotulados e, portando, não existe um conhecimento inicial das classes, nesse caso, os dados são classificados por afinidade. Neste contexto, a função das SVMs é induzida pelo paradigma supervisionado.

A eficiência de uma função classificadora é medida tanto pela complexidade quanto pela capacidade de generalização. A *capacidade de generalização* é a habilidade da função em classificar

corretamente dados que não pertencem ao conjunto de treinamento, enquanto que a complexidade se refere à quantidade de elementos que são necessários para compor a função, como, por exemplo, os pontos do conjunto de treinamento das SVMs. Neste trabalho, entende-se por *complexidade* a quantidade de pontos do conjunto de treinamento necessários para compor a função classificadora ou aproximadora.

Entretanto, minimizar a complexidade e maximizar a capacidade de generalização são objetivos divergentes ou contraditórios e, portanto, o que se busca nestas máquinas é o compromisso entre esses dois quesitos, tendo em vista a aplicação prática que elas terão.

Conforme (SMOLA et al., 1999), entre as principais características de uma SVM cita-se as seguintes:

- Grande capacidade de generalização.
- Robustez para dados de grande dimensões, como, por exemplo, imagens.
- O problema de otimização que modela o treinamento é quadrático e convexo, o que garante uma solução ótima única para o problema de treinamento.

O conjunto de treinamento é composto por $X = \{(\mathbf{x}_i, y_i)\}$, $i = 1, 2, ...N$ tal que $\mathbf{x}_i \in \mathbb{R}^n$ e $y_i \in \{-1, 1\}$, em que $\mathbf{x}_i$ é o vetor das características, y_i os rótulos e N a cardinalidade de X.

A função $f(\mathbf{x}_i, \alpha)$ é uma máquina treinada a partir do conjunto X, sendo α o parâmetro de ajuste da função obtido no processo de treinamento e $f(\mathbf{x}_i, \alpha)$ o rótulo atribuído ao vetor $\mathbf{x}_i$ pelo modelo. Para vetores que não pertencem ao conjunto X, o erro cometido pela máquina é calculado por 2.1

$$R(\alpha) = \frac{1}{2} \int c(f(\mathbf{x}, \alpha), y) dP(\mathbf{x}, y), \tag{2.1}$$

em que c é uma métrica para a diferença entre $f(\mathbf{x}, \alpha)$ e y, $R(\alpha)$ é denominado *risco esperado* ou simplesmente *risco*. Entretanto, não é possível calcular o risco esperado, pois a distribuição de probabilidade $P(\mathbf{x}, y)$ é desconhecida, assim o que se pode calcular é o *risco empírico* definido pela Equação 2.2

$$R_{emp}(\alpha) = \frac{1}{2N} \sum_{i=1}^{N} c(f(\mathbf{x}_i, \alpha), y_i). \tag{2.2}$$

O risco empírico $R_{emp}(\alpha)$ é determinado sobre o conjunto de treinamento por meio da técnica de *cross-validation* ou por meio de um conjunto de vetores característicos com rótulos conhecidos, mas que não pertencem ao conjunto de treinamento.

No treinamento de uma máquina é importante conhecer o risco esperado, mas, como a distribuição de probabilidade é desconhecida, define-se um limite superior para o risco com probabilidade $\eta - 1$ tal que $0 < \eta < 1$. Este limite superior é estimado pela Equação 2.3

$$R(\alpha) \leq R_{emp} + \sqrt{\left(\frac{h(log(2N/h) + 1) - log(\eta/4)}{N} \right)}. \tag{2.3}$$

Implícito à Equação 2.3 está o conceito de dimensão de Vapnik Chervonenkis - VC definido como: dado um conjunto X de vetores de características com cardinalidade N e $y_i \in \{-1, 1\}$, então existe 2^N formas de atribuir rótulos ao conjunto X. A classe de funções $\{f(\mathbf{x}, \alpha_j)\}$ é dita ter dimensão VC igual a N se esta classe é capaz de classificar todas as 2^N possibilidades de rotular X.

A variável h na Equação 2.3 é a dimensão de VC, sendo relacionada com a complexidade da SVM e, consequentemente, com sua capacidade de classificação. A parcela que soma ao risco empírico na Equação 2.3 é denominada de *termo de capacidade*.

A minimização do limite superior do risco esperado é conhecido como *minimização do risco estrutural*. Entretanto, minimizar o risco estrutural envolve objetivos contraditórios (risco empírico e termo de capacidade), quanto menor o risco empírico, maior o termo de capacidade e vice-versa.

Um modelo com risco empírico mínimo para um dado conjunto de treinamento não garante que este tenha boa capacidade de generalização, esse fenômeno é conhecido como *sobreajuste* ou, em inglês, *overfitting*. Um modelo sobreajustado perde a capacidade de generalização, pois este assimila durante o processo de treinamento ruídos e informações indesejadas oriundas de X. O contrário é chamado de *subajuste* ou, em inglês, *underfitting*, ou seja, o risco empírico não atende às restrições do problema no conjunto de treinamento.

2.3 MÁQUINAS DE VETORES DE SUPORTE PARA CLASSIFICAÇÃOS

As SVMs têm como objetivo definir um hiperplano que separe os dados de treinamento, garantindo uma boa generalização. Assim é introduzido o conceito de *margens*, que são hiperplanos paralelos ao hiperplano classificador, sendo estes definidos por pontos, com propriedades inerentes ao conjunto de treinamento denominados de *vetores de suporte*. Para que a generalização seja a melhor possível, as margens devem ser definidas de tal forma que a distância entre elas seja máxima com o mínimo de risco empírico (LIMA, 2004).

Por questões didáticas, o modelo matemático das SVMs foi dividido em duas categorias: (a) linearmente separáveis e (b) linearmente não separáveis. As SVMs linearmente separáveis se dividem em: (a) SVMs de margens rígidas e (b) SVMs de margens suaves. Entretanto, em aplicações reais, as SVMs linearmente não separáveis são transformadas em margens suaves com o uso de *funções kernel*, descritas em (SANTOS, 2019).

2.3.1 SVMs - Linearmente Separáveis

Um conjunto de treinamento X é denominado linearmente separável se existe um hiperplano que separe suas classes sem qualquer erro. A SVM, nesse caso, define o hiperplano classificador

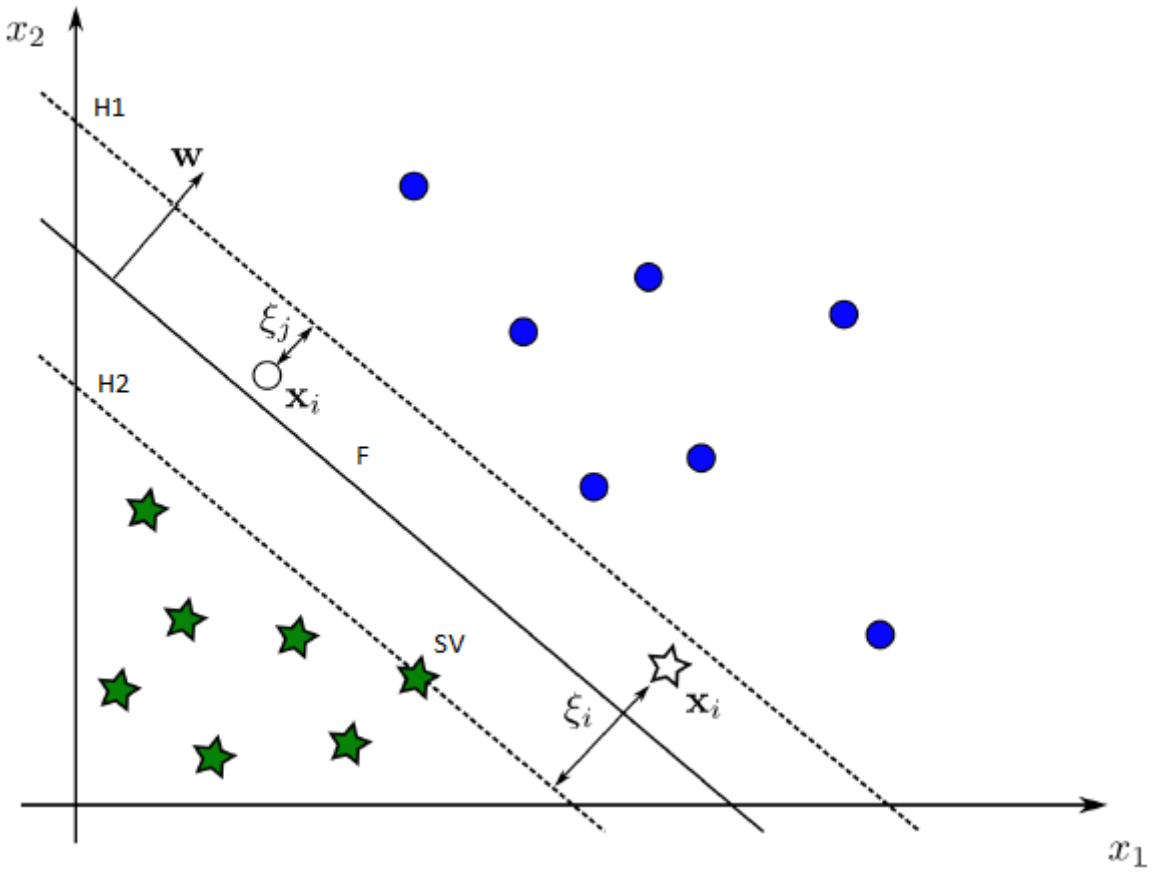


Figura 2.5 – SVM - Margens rígidas.

com margens mais distantes possível, minimizando o erro de generalização, para este conjunto de treinamento.

Na Figura 2.5, as retas tracejadas H_1 e H_2 são as margens da reta classificadora F, e os pontos que satisfazem as equações das margens são denominados SV. O objetivo é identificar quais são os vetores de suporte que definam margens de largura máxima que separem as classes, sem qualquer tipo de erro, nessas condições, as margens são denominadas de *margens rígidas* (CORTES; VAPNIK, 1995).

Os pontos que estão sobre as margens satisfazem a Equação 2.4. Esta equação representa a restrição do problema de treinamento e garante que não haja erro empírico na classificação.

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant 0. \tag{2.4}$$

No processo de treinamento busca-se maximizar a distância entre as margens H_1 e H_2 dada pela Equação 2.5.

$$d(H_1, H_2) = \frac{2}{\|\mathbf{w}\|}. \tag{2.5}$$

As margens máximas são calculadas minimizando $\|\mathbf{w}\|$ (é a norma de $\mathbf{w}$, sendo $\mathbf{w}$ o vetor

normal do hiperplano) na Equação 2.5, sujeito à restrição ($\mathbf{s.a.}$) da Equação 2.4, gerando, assim, o problema de otimização primal 2.6 nas variáveis $\mathbf{w}$ e b.

$$\begin{aligned} &\text{Min} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 \\ &s.a. \\ &y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geqslant 0 \qquad i = 1, ..., N, \end{aligned} \tag{2.6}$$

em que N é a cardinalidade do conjunto de treinamento.

O problema de otimização, descrito em 2.6, possui dois tipos de variáveis distintas e N restrições, que torna o problema complexo de ser resolvido. Para simplificar o modelo de treinamento, empregam-se os multiplicadores de Lagrange e o teorema de Karush-Kuhn-Tucker - KKT para transformar o problema primal no seu correspondente dual (GRIVA; NASH; SOFER, 2008). Segundo o teorema da dualidade fraca, os dois problemas têm a mesma solução ótima e, nesse caso, o problema dual possui conceitos que permite a sua generalização para conjuntos com dados não linearmente separáveis. O Problema de Otimização 2.7 é conhecido como dual de Wolf, nas variáveis α_i.

$$\begin{aligned} &\text{Max} \quad L_d(\alpha_i) = \textstyle\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{s.a} \\ &\textstyle\sum_{i=1}^{N} \alpha_i y_i = 0 \\ &\alpha_i \geqslant 0 \qquad\qquad i = 1, ..., N, \end{aligned} \tag{2.7}$$

em que α_i são os multiplicadores de Lagrange, ou seja, as variáveis do problema de otimização, $\mathbf{x}$ e y são os vetores característica e os rótulos de X respectivamente.

No problema definido por 2.7, a função L_d é uma função quadrática e a única restrição linear, ambas convexas, sendo que esta característica é condição necessária e suficiente (segundo o teorema de KKT) para encontrar uma solução ótima única. Essa propriedade garante que o classificador obtido por este método seja o ótimo global, dado um conjunto de treinamento e os parâmetros do modelo (GRIVA; NASH; SOFER, 2009).

Ao resolver o problema 2.7, têm-se os valores ótimos de α_i^*. Uma vez conhecidos os α^*, calcula-se o $\mathbf{w}^*$ a partir da Equação 2.8.

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i. \tag{2.8}$$

O b^* é calculado implicitamente isolando-o na equação $\alpha_i^*[y_i(\mathbf{w}^*\cdot\mathbf{x}_i + b^*) - 1] = 0$; sendo que as margens são equidistantes do hiperplano classificador e o termo b^* determina uma translação em relação à origem, toma-se sua média, assim tem-se:

$$b^* = \frac{1}{\#SV}\sum_{i=1}^{\#SV}\left(\frac{1}{y_i} - \mathbf{w}^*\cdot\mathbf{x}_i\right). \tag{2.9}$$

Usando os resultado das equações 2.9 e 2.8, obtém-se a função classificadora dependente dos α^*, cujos índices pertencem ao conjunto $SV = \{\alpha^* : \alpha^* \neq 0\}$, dada pela Equação 2.10

$$f(\mathbf{x}) = sgn\left(\sum_{i=1}^{\#SV} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b\right). \tag{2.10}$$

A função 2.10 é capaz de classificar corretamente todos os dados do conjunto de treinamento com a melhor generalização possível, mas, no caso dos problemas não serem linearmente separáveis, pode-se relaxar as restrições do Problema de Otimização 2.6. Essas SVMs são denominadas de margens suaves.

2.4 ALGORITMOS BIOINSPIRADOS

Nas técnicas de IA, os algoritmos de otimização de custos e recursos são um dos principais motivadores das pesquisas. Neste particular destacam-se os algoritmos baseados em comportamentos de enxames, conhecidos como algoritmos bio-inspirados. Dentre os algoritmos bio-inspirados temos o PSO e o DE como os dois mais utilizados pela comunidade acadêmica. Os algoritmos PSO (*Particle Swarm Optimization*) e DE(*Differential Evolution*) são algoritmos de otimização estocástica e baseado em populações. Que se utiliza dos princípios de inteligência coletiva de um enxame, baseada em princípios psicosociológicos. Estes algoritmos podem ser usados para demonstrar comportamentos sociais ou aplicações de engenharia ((GONG et al., 2017)).

2.4.1 Algoritmo PSO

O algoritmo *Particle Swarm Optimization* (PSO) busca otimizar um problema de forma iterativa, procurando melhorar a solução candidata com respeito a uma dada medida de qualidade. Este método foi proposto por Kennedy e Eberhart em 1995 (KENNEDY; EBERHART,).

PSO é baseado no comportamento social e cooperativo de várias espécies . Além disso, PSO é uma meta-heurística, pois realiza poucas ou nenhumas premissas sobre o problema que está sendo otimizado e pode procurar soluções candidatas em espaços de grandes dimensões.

A grande vantagem de utilizar o PSO é a sua fácil implementação, usando somente estruturas primitivas e operadores matemáticos sem grande custo computacional. Obviamente, como toda heurística, o PSO não garante a solução ótima e é comum o método cair em mínimos locais. Por conta disso, existem diversas modificações no algoritmo canônico do PSO, mas nenhuma garante o ótimo. O algoritmo PSO é formado por duas simples equações, que tratam da atualização de velocidade e posição das partículas no espaço de busca até que uma solução suficientemente boa seja encontrada.

No algoritmo PSO cada partícula tem uma velocidade aleatória associada, permitindo que as soluções potenciais (posição das partículas) se movimentem pelo espaço de busca do problema de otimização. Entretanto, cada partícula mensura a sua aptidão mediante a avaliação da função custo e conserva seu conhecimento do melhor valor de aptidão, utilizando uma memória individual e uma memória coletiva. A memória individual permite que a partícula lembre a posição em que encontrou um melhor valor de aptidão, enquanto a memória coletiva permite que as partículas lembrem a posição em que o enxame encontrou o melhor valor global de aptidão.

Os termos que serão utilizados neste trabalho são:

1. *Partícula ou agente*: individuo do enxame.
2. *Enxame*: coleção de indivíduos.
3. *Posição (x)*: coordenadas de uma partícula no espaço N-dimensional que representa uma possível solução ao problema.
4. *Aptidão*: o valor que representa quão boa é uma solução. Geralmente é a avaliação da função objetivo $f(x)$
5. *pbest (y_i)*: posição da melhor aptidão para uma determinada partícula.
6. *gbest (y_s)*: posição da melhor aptidão para o enxame inteiro.
7. v_{max}: velocidade máxima permitida em uma direção determinada.

2.4.2 Algoritmo PSO básico

Considerando um espaço de busca N-dimensional e um enxame com S partículas, a posição da i-ésima partícula do enxame na j-ésima dimensão pode ser atualizada mediante as equações (2.11) e (2.12).

$$v_{ij}^{(t+1)} = v_{ij}^{(t)} + c_1 U_{1j}(y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2 U_{2j}(y_{sj}^{(t)} - x_{ij}^{(t)}) \tag{2.11}$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \tag{2.12}$$

onde U_{1j} e U_{2j} são números aleatórios uniformemente distribuídos entre 0 e 1, y_{ij} é a melhor posição individual da i-ésima partícula na j-ésima dimensão e y_{sj} é a melhor posição global entre todas as partículas na j-ésima dimensão. As velocidades v_{ij} estão limitadas na faixa $[-v_{max}, v_{max}]$ evitando assim que as partículas abandonem o espaço de busca.

Os parâmetros c_1 e c_2 representam o coeficiente cognitivo e o coeficiente social, respectivamente. O comportamento do algoritmo PSO muda radicalmente dependendo dos valores desses

parâmetros. Um valor grande do coeficiente cognitivo c_1 indica partículas com alta autoconfiança na sua experiência, enquanto um valor grande do coeficiente social c_2 proporciona às partículas maior confiança no enxame (BERGH; ENGELBRECHT, 2004). Para funções objetivo unimodais é aconselhável utilizar pequenos valores do coeficiente cognitivo e valores grandes para o coeficiente social, enquanto para funções multimodais é necessário encontrar um balanço entre os dois coeficientes visando melhorar o desempenho do algoritmo (KENNEDY; EBERHART,), (BERGH; ENGELBRECHT, 2004).

O Algoritmo 1 apresenta o pseudocódigo do PSO. O mesmo pode ser visto como um algoritmo iterativo, em que a cada iteração é calculada uma nova posição para cada partícula no enxame. No intuito de conferir a conveniência das posições geradas, as mesmas são avaliadas utilizando a função objetivo f. Para cada partícula, se o valor da função objetivo na posição atual $f(x_i)$ é menor que o valor da função objetivo da melhor posição individual $fmin_k$ (vide a linha 12 do Algoritmo 1), então a melhor posição individual é substituída pela posição atual da partícula ($y_i = x$). Se o valor da função objetivo da posição atual $f(x_i)$ é menor que o valor da função objetivo da melhor posição global $f(y_s)$ então a melhor posição global é substituída pela posição atual da partícula ($y_s = x$).

Algorithm 1: Pseudocódigo do PSO básico.

Input: $S, N, c_1, c_2, x_{max}, Max_{ite}, threshold$
Output: posição da melhor partícula e sua aptidão $f(x)$

1: **Início:**
2: inicializa o enxame
3: **repeat**
4: **for** $k = 1 : S$ **do**
5: **if** $f(x_k) \leq f(y_{ik})$ **then**
6: $y_{ik} = x_k$
7: **end if**
8: **end for**
9: Calcule y_s usando os S valores de aptidão $f(y_{ik})$
10: **for** $k = 1 : S$ **do**
11: **for** $j = 1 : N$ **do**
12: $v_{kj} = v_{kj} + c_1 U_1(y_{kj} - x_{kj}) + c_2 U_2(y_{sj} - x_{kj})$
13: $v_{kj} = x_{kj} + v_{kj}$
14: **end for**
15: **end for**
16: **until** $f(y_s) < threshold$
17: **fim**

2.4.3 Algoritmo DE

Na computação evolutiva, a evolução diferencial (*Differential Evolution-DE*) é um método que otimiza um problema, tentando iterativamente melhorar uma solução candidata em relação a uma determinada medida de qualidade. Tais métodos são comumente conhecidos como metaheurísticas, pois fazem poucas ou nenhuma suposição sobre o problema que está sendo otimizado, e

podem procurar espaços muito amplos de soluções candidatas. No entanto, as metaheurísticas, como DE, não garantem uma solução ótima.(STORN; PRICE, 1997a)

O DE é usado para funções multidimensionais de valor real, mas não usa o gradiente do problema que está sendo otimizado, o que significa que o DE não exige que o problema de otimização seja diferenciável como é exigido pelos métodos clássicos de otimização, como o método de gradiente descendente. O DE pode, portanto, também ser usado em problemas de otimização cuja função custo não são contínuas, e que mudam ao longo do tempo, etc.(ROCCA; OLIVERI; MASSA, 2011).

O DE otimiza um problema, mantendo uma população de soluções candidatas e criando novas soluções candidatas, combinando as existentes de acordo com suas fórmulas simples e, em seguida, mantendo a solução candidata com a melhor pontuação ou aptidão sobre o problema de otimização em questão. O Algoritmo 2 apresenta o pseudocódigo do DE básico.

Algorithm 2: Pseudocódigo do algoritmo DE.

Input: NP, Cr, F, Max_{ite}

Início:
Inicializa a população
Avalia cada indivíduo
$G = 1$
repeat
  for $i = 1 : NP$ **do**
    Escolha aleatoriamente r_1, r_2 e r_3.
    $\mathbf{v}_{i,G} = \mathbf{x}_{r_1^i,G} + F \cdot (\mathbf{x}_{r_2^i,G} - \mathbf{x}_{r_3^i,G})$
    for $j = 1 : n$ **do**
      if $(rand_{i,j} \leq Cr) \vee (j = j_{rand})$ **then**
        $u_{j,i,G} = \mathbf{v}_{i,j,G}$
      else
        $u_{j,i,G} = \mathbf{x}_{i,j,G}$
      end if
    end for
  end for
  Avalia cada indivíduo u_i
  $C = U \cup X$
  $X =$ seleção(C, NP)
  $G = G + 1$
until $G \geq Max_{ite}$
fim

2.4.4 Algoritmos MOPSO e MODE

Os algoritmos *Multi-Objective Particle Swarm Optimization-MOPSO* e *Multi-Objective Differential Evolution-MODE* são adaptações feitas nos algorítimos PSO e DE para tratamento de múltiplos objetivos. Diferentes dos originais (PSO e DE), estes algorítimos apresentam múltiplos

resultados, em vez de somente um, como nos algorítimos mono-objetivos. Tais resultados são normalmente ajustados pelas técnicas de Pareto.

Aqui são apresentados os algoritmos *Adaptive Parameter with Mutant Tournament Multi-Objective Differential Evolution - APMT-MODE*, *Adaptive Parameter Multi-Objective Differential Evolution - AP-MODE* e *Multi-Objective Particle Swarm Optimization - MOPSO*, desenvolvidos para encontrar o *Conjunto e a Fronteira de Pareto - PF* para o *Problema de Seleção de Modelos - PSP* das SVM/SVRs, bem como, as modificações realizadas nas estratégias utilizadas para aumentar a eficiência dos algoritmos em encontrar hiperâmetros que gerem modelos de SVM/SVRs que atendam as restrições das aplicações.

O algoritmo PSO foi modificado para tratar *Multi-objective Optimization Problem - MOOP*, agregando a função *truncate*, a técnica *Criticism of Lexicographic Ordering - CLO* e o *Symmetric Latin Hypercube Design - SLHD* para inicialização da população inicial, agregando técnicas eficientes para problemas distintos, mas que possui características semelhantes ao problema de seleção de modelo das SVMs. O desenvolvimento do APMT-MODE seguiu metodologia semelhante de agregação de técnicas distintas que ora foram utilizadas para algoritmos mono-objetivo e, neste trabalho, foi adaptado para tratar o problema como MOOP. O APMT-MODE e o MOPSO foram validados utilizando diferentes *benchmarks* classificadores e regressores, como também aplicações práticas a problemas de engenharia.

O detalhamento destes algoritmo e suas especificidades, utilizados nesta pesquisa, serão descritas a seguir.

2.4.4.1 Otimização Multiobjetivo por Enxame de Partículas - MOPSO

O PSO básico é um algoritmo inspirado na natureza que simula o comportamento social de pássaros e cardumes de peixes. Devido à sua simplicidade na implementação, e na rápida convergência, tem atraído o interesse da comunidade científica, sendo utilizado para resolver diversos problemas aplicados à engenharia (LIN et al., 2015). Portanto, devido às características mencionadas, o PSO foi adaptado para resolver problemas multiobjetivos. As principais modificações aplicada ao PSO básico foram: (a) estratégia de inicialização das partículas; (b) aplicação da técnica de CLO para determinação do melhor individual e global.

Essas adaptações consistem em contribuições distintas para o PSO desenvolvido denominado como *Multi-Objective Particle Swarm Optimization* - MOPSO, sendo cada uma delas descritas com detalhes no decorrer desta seção.

A Figura 2.6 é o diagrama de fluxo do MOPSO, no qual o passo *Inicialização*, os parâmetros NP, w_0, w_f, c_1, c_2 e v_{max}, tamanho da população, fator de inércia inicial e final, coeficientes cognitivos e social e a velocidade máxima, respectivamente são inicializados.

Muitas implementações do MOPSO usam a distribuição uniforme para inicializar as partículas, o que não garante um espalhamento adequado delas sobre o espaço de busca. Assim, para

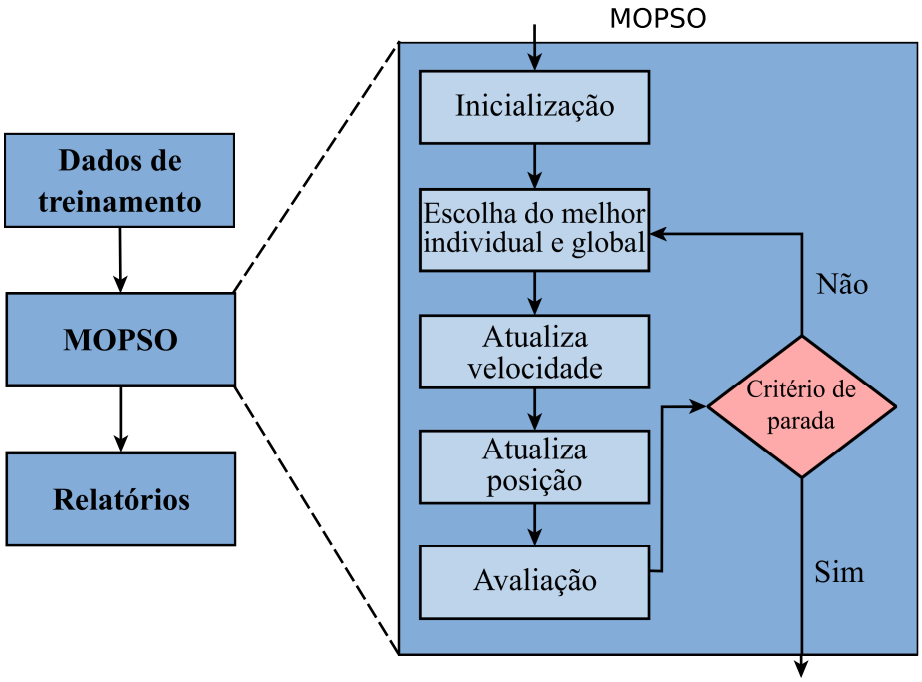


Figura 2.6 – Diagrama de fluxo do algoritmo MOPSO.

gerar o enxame inicial, foi usado o método denominado *Symmetric Latin Hypercube Design* - SLHD (ZHAO et al., 2016). O pseudocódigo do SLHD é apresentado na Algoritmo 3.

O SLHD gera uma matriz de números aleatórios simetricamente distribuídos entre $[0, 1]$ em todas as dimensões do espaço de busca. Assim, a distribuição evita eventuais concentrações em determinadas regiões do espaço de busca, que pode ocorrer com a distribuição uniforme.

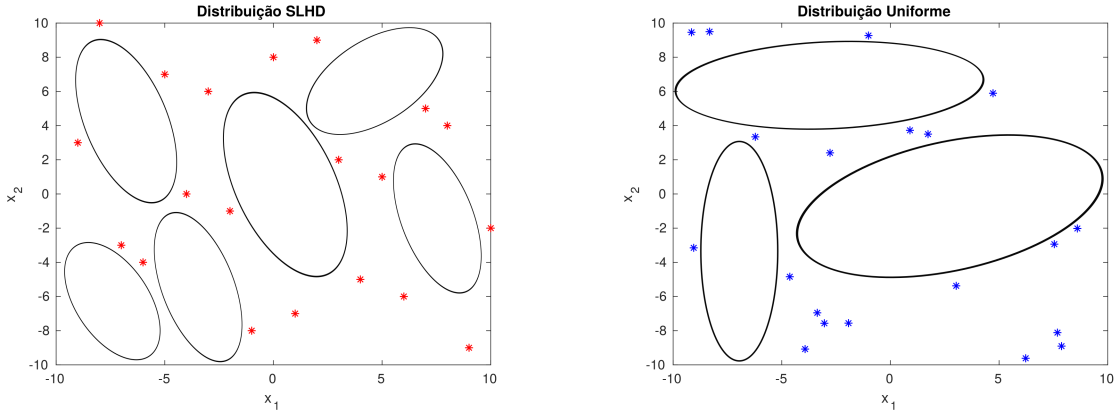
Para evidenciar a diferença entre a população gerada pela distribuição uniforme e o SLHD, foi utilizada a métrica de diversidade do método de adição de diversidade AR para avaliar as populações iniciais. Na Figura 2.7, os asteriscos vermelhos representam as partículas geradas pelo SLHD, enquanto os asteriscos azuis representam as partículas geradas pela distribuição uniforme.

As elipses evidenciam as regiões contíguas que não possuem partículas. Enquanto no SLHD, são necessárias mais elipses menores, a população gerada pela distribuição uniforme possui elipses maiores, assim, conclui-se que a distribuição das partículas pelo SLHD é melhor.

Algorithm 3: Pseudocódigo do algoritmo SLHD.

Input: NP, n

Início:
Inicializa matriz $NP \times n$
if NP for ímpar **then**
    $M((NP+1)/2, j) = (NP+1)/2$, para $j = 1, ..., n$.
end if
for $j = 1 : n$ **do**
    Escolha aleatoriamente a permutação de $1, ..., k$ e denote-a por ϕ_j
end for
for cada par (i, j), em que $i = 1, ...k$ e $j = 1, ..., n$ **do**
    Gere um número aleatório com distribuição uniforme $w_{ij} \in [0, 1]$.
    if $w_{ij} \leq 0,5$ **then**
        $M(i, j) = \phi_j(i)$
        $M(NP + 1 - i, j) = NP + 1 - \phi_j(i)$
    else
        $M(i, j) = NP + 1 - \phi_j(i)$
        $M(NP + 1 - i, j) = \phi_j(i)$
    end if
end for
for $j = 1 : n$ **do**
    $\pi_j = M(:, j)$
    Particione o intervalo $[a_j, b_j]$ em NP subintervalos de tamanhos iguais, sendo $c_j^{(i)}$ o ponto médio de $[a_j, b_j]$.
end for
for $I = 1 : NP$ **do**
    o i-ésimo ponto SLHD é dado por $(c_1^{(\pi_1(i))}, c_2^{(\pi_2(i))}, ..., c_n^{(\pi_n(i))})$
end for



(a) População obtida pelo SLHD. (b) População obtida pela distribuição uniforme.

Figura 2.7 – Comparação entre SLHD e distribuição uniforme.

A diversidade é medida por um escalar d calculado pela Equação 2.13, sendo a métrica $d(S)$ independente do tamanho do enxame, da dimensionalidade do problema e dos limites do espaço de busca. Na iteração em que $d(S)$ é menor que o limite inferior d_{min}, a técnica entra na fase repulsiva e, se $d(S)$ for maior que d_{max}, o algorítimo entra na fase atrativa

$$d(S) = \frac{1}{\#S \cdot |L|} \cdot \sum_{i=1}^{\#S} \sqrt{\sum_{i=1}^{n} (p_{ij} - \bar{p}_j)^2}, \tag{2.13}$$

em que S é o conjunto que representa o exame, $\#S$ a cardinalidade do exame, $|L|$ a maior diagonal do espaço do busca, n é a dimensionalidade do problema, p_{ij} é o valor da i-ésima partícula na j-ésima dimensão e $\bar{p}_j$ é o valor da j-ésima média do ponto $\bar{p}$ (RIGET; VESTERSTRØM, 2002).

A métrica diversidade, dada pela Equação 2.13, foi utilizada para avaliar a distribuição da população inicial gerada pelo SLHD e pela distribuição uniforme. Foram geradas trinta e duas amostras independentes de populações por cada estratégia e calculada suas respectivas média, mediana e desvio-padrão (DP) da métrica diversidade para os dois casos. Estes dados são apresentados na Tabela 2.1.

Tabela 2.1 – Comparação da métrica *diversidade* entre distribuição uniforme e SLHD.

Distribuição	**Média**	**Mediana**	**DP**
SLHD	0,276	0,276	1,13E-16
Uniforme	0,208	0,208	5,64E-17

Os dados da Tabela 2.1 mostram que o SLHD gera populações com diversidade média e mediana maior que a distribuição uniforme, enquanto o desvio-padrão do SLHD é menor. Isto indica que o SLHD gera populações iniciais melhores distribuídas na maioria das vezes em relação à

distribuição uniforme. O SLHD não acrescenta grande esforço computacional ao MOPSO, pois é utilizado apenas para gerar a população inicial.

O próximo passo consiste em definir as partículas, *melhor individual* e *melhor global*, ou seja, a melhor posição visitada por cada indivíduo e a melhor partícula já encontrada pelo enxame respectivamente pois, a velocidade de cada partícula depende da identificação destas partículas. Entretanto, devido ao conceito de dominância, a melhor partícula pode ser qualquer uma do conjunto de soluções não dominadas.

A melhor partícula individual é obtida ordenando por *rank* de não dominância (função *Truncate*), todas as posições já visitadas pela partícula, e entre as partículas que possuem *rank* igual é escolhida aquela que possui menor função objetivo definida pela técnica CLO. A melhor partícula global é obtida a partir de um conjunto criado com todas as soluções não dominantes visitadas pelo algoritmo até a iteração atual, deste conjunto de soluções não dominantes uma é escolhida de acordo com o critério (função objetivo) definido pelo CLO.

O passo *Atualiza velocidade* define a direção, o sentido e o módulo do vetor velocidade de cada partícula do enxame. O vetor velocidade é calculado pela Equação (2.11).

Em seguida, o passo *Atualiza posição* movimenta a partícula a partir da posição atual no sentido, na direção e no módulo do vetor velocidade v_{ij}^{t+1}, conforme Equação (2.12).

No passo *Avaliação*, cada partícula representa os hiperparâmetros de uma SVM/SVR. Os modelos são gerados pela ferramenta LibSVM (CHANG; LIN, 2011) que utiliza os hiperâmetros juntamente com os dados de treinamento para gerar o problema de otimização e resolvê-lo, entregando como solução os modelos de SVM/SVR. O modelo gerado pela LibSVM é avaliado com os dados de validação e que produz os valores das funções objetivo, que mede a capacidade de generalização do modelo e sua complexidade. O critério de parada é o número máximo de iterações definido pelo usuário.

No MOPSO, a posição do enxame na iteração $t + 1$ não é aceita diretamente, como ocorre no PSO básico, ao invés disso, a matriz que representa o enxame na nova posição é concatenada ao enxame da posição atual, formando um uma matriz de ordem $2NP \times n$. Então, essa matriz é processada pela função *Truncate* que faz ranqueamento das soluções não dominadas e as NP melhores soluções consistem no posicionamento do enxame na iteração $t + 1$. Assim, de uma iteração para outra, somente as partículas com posições melhores e não dominadas são aceitas. Neste processo, as partículas tendem sempre a melhorar seus valores funcionais, tendo uma grande capacidade de refinamento das soluções.

Na Seção 2.4.4.2, uma versão do DE foi adaptada para tratar o problema de seleção de parâmetros como um MOOP. Este algoritmo possui dois parâmetros ajustados por uma técnica adaptativa, assim como são empregadas diferentes estratégias de mutação.

2.4.4.2 Parâmetros Adaptativos com Torneio MultiObjetivo Evolução Diferencial - APMT-MODE

O algoritmo de *Parâmetros Adaptativos com Torneio Multiobjetivo Evolução Diferencial - APMT-MODE* foi baseado no algoritmo DE básico proposto por Storn and Price (STORN; PRICE, 1997b) e no processo adaptativo desenvolvido originalmente por Fan e Zhang (FAN; ZHANG, 2016).

O DE básico possui algumas características que são desejadas em aplicações práticas, tais como habilidade de tratar funções objetivos não diferenciáveis, não lineares e multimodais, capacidade de paralelização para trabalhar com funções complexas, poucos parâmetros para serem ajustados, alta capacidade de convergência para soluções de boa qualidade em várias execuções independentes (STORN; PRICE, 1997b).

As modificações realizadas no DE básico objetivam contornar algumas dificuldades que este possui na solução do problema de seleção de parâmetros, tais como: (a) diferentes espaço de busca obtidos pela combinação dos conjuntos de treinamento e kernels disponíveis e (b) problema com dois objetivos contraditórios.

Na dificuldade (a), cada conjunto de treinamento das SVM/SVR possui características diferentes e, portanto, exige ajustes distintos do parâmetros Cr, F e, também, diferentes estratégias de mutação e recombinação, então, foram utilizadas técnicas de adaptação propostas por (FAN; ZHANG, 2016). No item (b), a função *Truncate* é utilizada para definir os conjuntos de soluções não dominadas e a técnica CLO para definir os melhores indivíduos.

O fluxograma de solução do problema de seleção de parâmetros é apresentado na Figura 2.8. O sistema representado pelo fluxograma é iniciado fornecendo os dados de treinamento para o bloco *Otimizador* que gera o Conjunto e a Fronteira de Pareto correspondente, assim como seus modelos. As linhas pontilhadas no bloco Otimizador evidenciam, no fluxograma, o algoritmo APMT-MODE, descrito em detalhes nos parágrafos subsequentes.

1. *Inicialização*: Os indivíduos da população inicial possuem importante papel na qualidade final das soluções, pois uma distribuição pobre dos indivíduos sobre o espaço de busca pode induzir o algoritmo para mínimo local. Várias implementações do DE usam a distribuição uniforme para gerar a população inicial, o que não garante uma espalhamento razoável sobre o espaço de busca. Assim, para gerar a população inicial, foi usado o SLHD (ZHAO et al., 2016), como descrito na Seção 2.4.4.1.

2. *Adaptação dos parâmetros*: Os parâmetros F e Cr são atualizados pela Equação 2.14 e Equação 2.15 respectivamente.

$$F_i^{G+1} = N\left(F_w^{G+1}, \sigma\right) \tag{2.14}$$

$$Cr_i^{G+1} = N\left(Cr_w^{G+1}, \sigma\right) \tag{2.15}$$

em que $\sigma = 0,1 + 0,40 \times (1 - (G/G_{max})^2)$ é a variância, F_w^{G+1} e Cr_w^{G+1} são as médias da

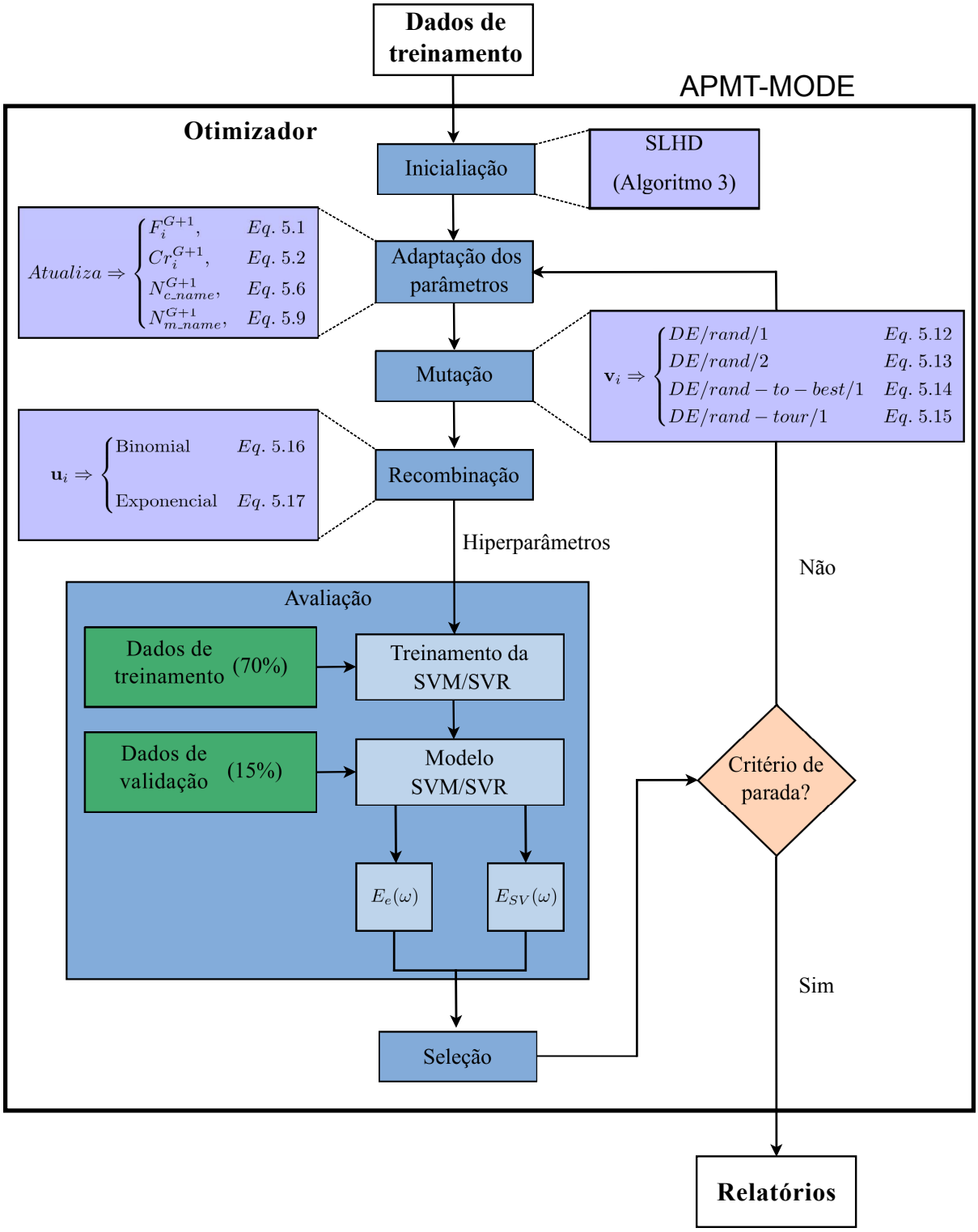


Figura 2.8 – Diagrama de fluxo do algoritmo APMT-MODE - Fonte (SANTOS, 2019)

distribuição Normal definidas pela Equação 2.16 e Equação 2.17 respectivamente,

$$F_w^{G+1} = \sum_{i=1}^{NP} w_i^G \times F_i^G \tag{2.16}$$

$$Cr_w^{G+1} = \sum_{i=1}^{NP} w_i^G \times Cr_i^G \tag{2.17}$$

em que w_i^{G+1} é calculado por Equação 2.18

$$w_i^{G+1} = \frac{\left| f_j\left(\mathbf{x}_i^G\right) - \max\left(f_j\left(\mathbf{x}^G\right)\right)\right|}{\sum_{i=1}^{NP}\left|f_j\left(\mathbf{x}_i^G\right) - \max\left(f_j\left(\mathbf{x}^G\right)\right)\right|}, \tag{2.18}$$

em que w_i^{G+1} é a média ponderada do parâmetros de controle e o peso do parâmetro de recombinação (Cr_i^{G+1}) é calculado por Equação 2.17.

O parâmetro σ controla a variância da distribuição Normal e decresce quadraticamente de $0,5$ a $0,1$ diminuindo variedade dos vetores mutantes nas iterações finais do processo de otimização, este comportamento aumenta a capacidade de refinamento do APMT-MODE, conforme apresentado na Figura 2.9.

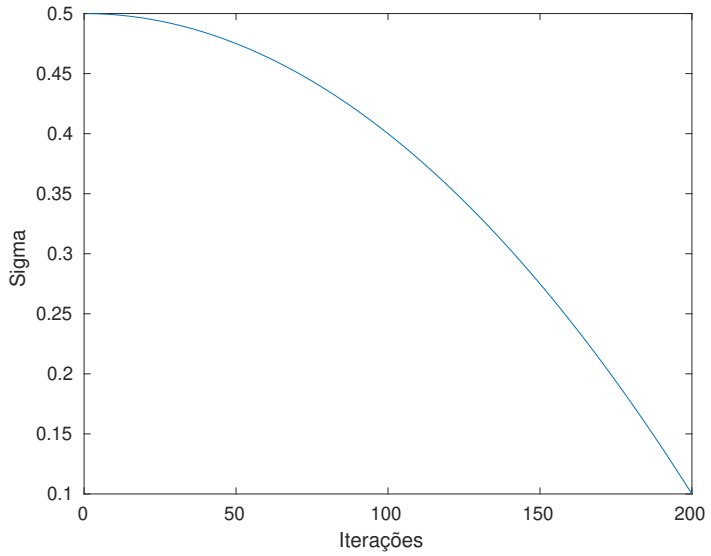


Figura 2.9 – Comportamento da variância σ.

Nesta etapa, também é definida cardinalidade dos subgrupos de indivíduos que são afetados por cada estratégia de recombinação, calculada pela Equação 2.19.

$$N_{c\_name}^{G+1} = \begin{cases} N_{c\_name}^G + 1, \text{ se } N_{c\_name}^{G+1} > N_{c\_name}^G \\ N_{c\_name}^G - 1, \text{ se } N_{c\_name}^{G+1} < N_{c\_name}^G \\ N_{c\_name}^G, \quad \text{caso contrário} \end{cases} \tag{2.19}$$

sendo G a geração atual, c_name o nome da estratégia de recombinação e $N_{c\_name}^{G+1}$ calculado pela Equação 2.20

$$N_{c\_name}^{G+1} = round\left(PS \times \frac{S_{c\_name}^{G+1}}{S^{G+1}}\right), \tag{2.20}$$

sendo PS o número de indivíduos na população, $round$ é uma função de arredondamento,

$S^{G+1} = S^{G+1}_{cross\_bin} + S^{G+1}_{cross\_exp}$ e $S^{G+1}_{c\_name}$ é a soma das diferenças calculadas pela Equação 2.21

$$S^{G+1}_{c\_name} = \sum_{k=1}^{N^G_{c\_name}} \left| f_j\left(\mathbf{x}^G_{c\_name,k}\right) - \max\left(f_j\left(\mathbf{x}^G\right)\right)\right| \tag{2.21}$$

onde $f_j\left(\mathbf{x}^G_{c\_name,k}\right)$ é o valor funcional da partícula da k-ésima partícula do conjunto referente a estratégia de recombinação c_{name}, para a j-ésima função objetivo.

Finalmente, o *Adaptação dos parâmetros* também designa o melhor número de indivíduos para cada estratégia de mutação, a cada cinco iterações, esta quantidade é definida pela Equação 2.22

$$N^{G+1}_{m\_name} = \begin{cases} N^G_{m\_name} + 1, \text{ se } N^{G+1}_{m\_name} > N^G_{m\_name} \\ N^G_{m\_name} - 1, \text{ se } N^{G+1}_{m\_name} < N^G_{m\_name} \\ N^G_{m\_name}, \quad \text{caso contrário} \end{cases} \tag{2.22}$$

sendo $N^{G+1}_{m\_name}$ calculado pela Equação 2.23,

$$N^{G+1}_{m\_name} = round\left(PS \times \frac{S^{G+1}_{m\_name}}{S^{G+1}}\right) \tag{2.23}$$

em que, S^{G+1} é a diferença entre o valor objetivo de cada indivíduo e o máximo valor objetivo da população, como mostrado na Equação 2.24

$$S^{G+1}_{m\_name} = \sum_{k=1}^{N^G_{m\_name}} \left| f_j\left(\mathbf{x}^G_{m\_name,k}\right) - \max\left(f_j\left(\mathbf{x}^G\right)\right)\right|, \tag{2.24}$$

em que m_name são os índices que representam a estratégia de mutação, o f_i é a mesma função empregada na estratégia de recombinação.

3. *Mutação*: produz NP mutantes distribuídas entre as estratégias de mutação definidas pela Equação 2.22. As estratégias de mutação utilizadas no APM-MODE são apresentadas nas equações 2.25 - 2.28,

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i\left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) \tag{2.25}$$

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i\left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + F_i\left(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\right) \tag{2.26}$$

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i\left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + F_i\left(\mathbf{x}_{best} - \mathbf{x}_i\right) \tag{2.27}$$

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F_i\left(\mathbf{x}_{t\_best} - \mathbf{x}_i\right) \tag{2.28}$$

em que r_1, r_2, r_3, r_4 e r_5 são números aleatórios mutuamente diferentes escolhidos da população de indivíduos da geração atual, $\mathbf{x}_{best}$ é o melhor indivíduo da população atual e $\mathbf{x}_{t\_best}$ é o indivíduo com a melhor avaliação da função objetivo de um subconjunto esco-

lhido aleatoriamente da população. Note que o parâmetro F_i é ajustado pelo passo (b) do algoritmo.

4. *Crossover*: A recombinação binomial verifica cada posição do vetor mutante e o altera, como descrito pela Equação 2.29.

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{se } rand_{i,j} \leq Cr \quad \text{ou} \quad j = j_{rand} \\ x_{ij}^{G}, & \text{caso contrário,} \end{cases} \tag{2.29}$$

onde $\mathbf{x}$ são os vetores pais, $\mathbf{u}$ são os vetores filhos, j é o *j*-ésimo posição do *i*-ésimo indivíduo, $rand_{i,j}$ é um número aleatório de distribuição uniforme que pertence aos intervalo $[0, 1]$, e j_{rand} é um inteiro escolhido aleatoriamente na faixa $[1, D_m]$ onde D_m é a dimensionalidade do indivíduo. A condição $j = j_{rand}$ assegura que ao menos uma posição dos filhos tenha uma informação do vetor mutante. A recombinação exponencial divide o vetor mutante o pai em uma posição aleatória e troca suas partes para criar um novo vetor como definido pela Equação 2.30:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{se } j = \langle n \rangle_{D_m}, \langle n+1 \rangle_{D_m}, ..., \langle n+L-1 \rangle_{D_m} \\ x_{ij}^{G}, & \text{caso contrário,} \end{cases} \tag{2.30}$$

onde $\langle \rangle_D$ denota o operador módulo com módulo D_m. O número L é um inteiro obtido do intervalo $[1, D_m]$ e n é um número inteiro.

Vale observar que as equações 2.29 e 2.30 são aplicadas sobre conjuntos disjuntos escolhidos aleatoriamente da população com NP indivíduos. Inicialmente, os subconjuntos possuem a mesma cardinalidade, e depois, a cada cinco iterações, a cardinalidade dos subconjuntos são atualizadas, e neste caso, dependendo da Equação 2.19.

5. *Avaliação*: neste passo, os indivíduos representam os hiperparâmetros necessários para treinar uma SVM/SVR. O treinamento de uma Máquina de Vetores de Suporte consiste em resolver um problema de otimização definido pela Equação 2.31 para um dado conjunto de treinamento.

$$\begin{array}{ll} \text{Max} & L_d(\alpha_i) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.a} & \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \xi_i \geqslant 0 \qquad\qquad i = 1, ..., N \\ & 0 \leqslant \alpha_i \leqslant C \qquad\quad i = 1, ..., N. \end{array} \tag{2.31}$$

O modelo é gerado pela biblioteca LibSVM (CHANG; LIN, 2011), é validado com os conjunto de dados de validação, este processo permite calcular o $E_e(\omega)$, ou seja, o MSE para as SVRs e a precisão para as SVMs, assim como a cardinalidade do conjunto de vetores de suporte ($E_{SV}(\omega)$). O passo *Avaliação* é descrito pelo diagrama de fluxo da Figura 2.8.

6. *Seleção*: os indivíduos filhos são concatenados ao conjunto população e, então, a função *Trucante* é aplicada à matriz com dimensão duas vezes o tamanho da população. Assim, somente os indivíduos não dominantes sobrevivem à próxima geração.

O processo iterativo repete após o passo (f) voltando ao passo (b) até que o número de iterações pré estabelecido seja alcançado.

A notação das estratégias de mutação seguem o padrão $DE/x/y$, em que DE designa o algoritmo Evolução Diferencial, x nomeia o vetor a ser alterado pelo operador mutação e y é quantidade de vetores diferenças (STORN; PRICE, 1997b). As estratégias de mutação utilizadas são apresentadas nas equações 2.25, 2.26, 2.27 e 2.28 denominadas *DE/rand/1*, *DE/rand/2*, *DE/rand-to-best/1*, *DE/rand-tournament/1* respectivamente.

A estratégia de mutação definida na Equação 2.28 é denominada *torneio*. Esta estratégia de mutação busca o balanço entre a exploração e o refinamento. Primeiro, um subconjunto da população é criado com vinte e cinco porcento dos indivíduos escolhidos aleatoriamente. Então, o indivíduo com a melhor avaliação da função objetivo é escolhido deste subconjunto para ser o $\mathbf{x}_{t\_best}$. Esse processo garante que um indivíduo, que possui baixa qualidade, tenha chance de ser o melhor e direcionar o movimento de um subconjunto população, evitando possíveis mínimos locais.

Uma variação do APMT-MODE, denominada de *Adaptive Parameters Multi-Objective Differential Evolution* - AP-MODE, foi desenvolvida substituindo a mutação torneio, do APMT-MODE, pela estratégia definida na Equação 2.32

$$\mathbf{v}_i = \mathbf{x}_i + F_i \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + F_i \left(\mathbf{x}_{c\_best} - \mathbf{x}_i\right). \tag{2.32}$$

A estratégia de mutação definida na Equação 2.32, gera um novo indivíduo nas proximidades da melhor solução atual explorando a sua vizinhança (GONG et al., 2017).

Nos problemas multiobjetivos, todas as funções objetivos têm o mesmo grau de importância. Porém, para escolher o melhor indivíduo no APMT-MODE, uma das funções objetivos é escolhida aleatoriamente para ser o critério de decisão. Este método é chamado *Criticism of Lexicographic Ordering* - CLO (COELLO; VELDHUIZEN, 2007).

O CLO consiste em eleger uma das funções objetivo, por sorteio, como critério para definir o melhor indivíduo dentre as soluções não dominantes. O CLO é empregado a cada cinco gerações, esta frequência, na escolha, oferece a oportunidade dos indivíduos melhorarem em relação a um determinado critério antes de realizar outro sorteio. Todas as estratégias de mutação, que de alguma forma empregam a definição de melhor individuo utilizam CLO.

A principal desvantagem do CLO é a aleatoriedade da escolha da função objetivo, que, no caso de várias funções, tende a favorecer funções específicas. Entretanto, no nosso caso, o problema aqui abordado possui apenas duas funções objetivos e, nesse caso, este problema é minimizado (COELLO; VELDHUIZEN, 2007).

A principal vantagem do CLO é que ele é capaz de descrever uma Fronteira de Pareto côncava, embora isto dependa da distribuição da população e do problema em si (COELLO; VELDHUIZEN, 2007). Assim sendo, o método SLHD foi utilizado para inicializar a população do APMT-MODE, assim como o CLO, como estratégia de definição do melhor indivíduo nas estratégias de mutação e ajuste de parâmetros.

2.5 A FERRAMENTA NIOTS DESENVOLVIDA NO LEIA-UNB

A ferramenta denominada *Nature Inspired Optimization Tools for SVM - NIOTS*, busca tornar mais prática a avaliação e o desenvolvimento de técnicas e a utilização dos algoritmos APMT-MODE e MOPSO, possibilitando que projetistas possam produzir bons resultados aos problema de seleção de parâmetros das SVMs. Algumas técnicas promissoras envolvem a análise do conjunto não dominante para definir: (a) o movimento das partículas/indivíduos; (b) as métricas para ordenar o conjunto de soluções não dominantes; (c) a adoção de técnicas adaptativas para controlar os parâmetros do APMT-MODE e suas metodologia de mutação e recombinação; (d) as metodologias de escolha dos *gbest* e *lbest* no MOPSO; (e) as distribuições alternativas de variáveis aleatórias, como Normal, Cauchy e Levy; e (f) a geração da população inicial com uso da técnica *Symmetric Latin Hypercube Design*.

Os algoritmos meta-heurísticos APMT-MODE e MOPSO, desenvolvidos por (SANTOS, 2019), foram codificados em linguagem Matlab com auxílio da ferramenta GUIDE - *Grafical User Interface Development Environment*, originando o sistema denominado NIOTS. Este sistema é composto dos ambientes de (a) Otimização, (b) *Ensembles*, (c) Predição e (d) Estatística. Cada um desses ambientes são descritos nas subseções 2.5.2, 2.5.1, 2.5.3 e 2.5.4.

O sistema possibilita a utilização dos conceitos associados a SVM/SVR apresentados no Capítulo 2 (SANTOS, 2019), os algoritmos de otimização multiobjetivo APMT-MODE, AP-MODE e MOPSO e as técnicas de Adição de Diversidade descritas no Capítulo 3 (SANTOS, 2019). O NIOTS possibilita que usuários com interesse em obter modelos de SVM/SVRs para áreas específicas do conhecimento, sem a realização de estudos profundos dos conceitos envolvidos.

Adicionalmente, o NIOTS gera relatórios e modelos que podem ser testados e analisados estatisticamente nos ambientes Predição e Estatística.

O ambiente *Ensembles* é uma versão inicial de trabalhos futuros, em que será desenvolvido um algoritmo meta-heurístico para escolha do melhor conjunto de modelos para gerar o *ensemble* a partir de Conjuntos e Fronteiras de Pareto obtido no ambiente de Otimização. Na versão atual do NIOTS, os *ensembles* são obtidos testando todas as combinações possíveis de modelos de dois, três e quatro modelos de SVR provenientes de uma fronteira de Pareto (PF) escolhida.

A ferramenta LiBSVM realiza o treinamento utilizando os hiperparâmetros fornecidos pelos algoritmos APMT-MODE e MOPSO. Esses modelos são avaliados com os dados do conjunto de validação, retornando as métricas da capacidade de generalização e complexidade do modelo.

No fluxograma da Figura 2.12, a variável ω é o vetor dos hiperparâmetros e os valores funcionais E_e e C_{SV} são as métricas da capacidade de generalização e da complexidade, respectivamente. As setas preenchidas, em preto, representam o fluxo de dados entre os blocos, enquanto as setas brancas indicam as possíveis opções que o usuário pode escolher no processo de otimização, também apresentadas na Tabela 2.2.

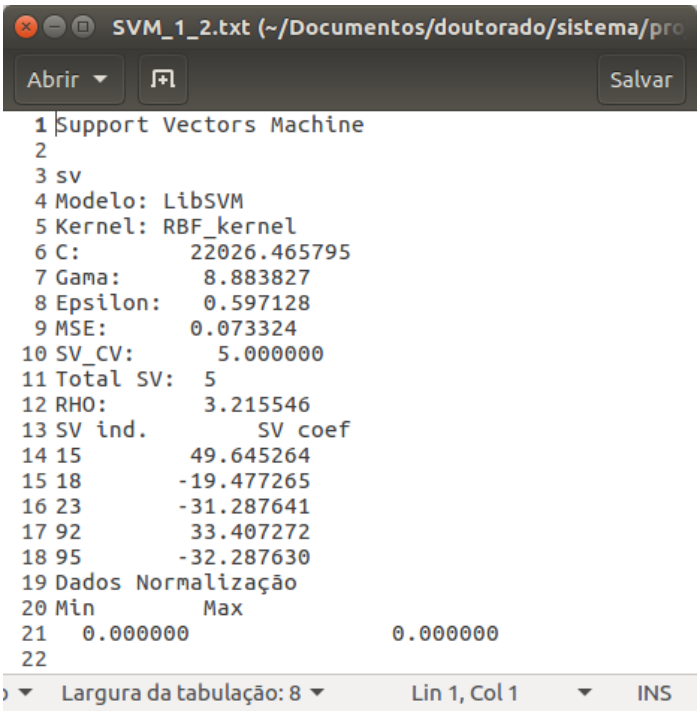


Figura 2.10 – Exemplo de modelo da Fronteira de Pareto.

O relatório *Pareto Front* (Fronteira de Pareto), Figura 2.11, contém os parâmetros do APMT-MODE/MOPSO definidos pelo usuário e os elementos do conjunto de Pareto, assim como seus respectivos objetivos da Fronteira de Pareto. As métricas de qualidade da PF são avaliadas a cada iteração e gravadas no arquivo relatório *Métricas de Qualidade*. As métricas utilizadas para avaliar a Fronteira de Pareto são a cardinalidade do conjunto de soluções não dominadas e o *Spacing*.

Essas métricas são calculadas a cada iteração, permitindo a análise de: (a) convergência do algoritmo e (b) da qualidade das soluções obtidas para cada processo de otimização.

Além dos painéis descritos na Tabela 2.2, o ambiente possui três botões. O botão *carregar dados de treinamento* faz a leitura do arquivo que contém os dados de treinamento. Para carregar os dados de validação, utiliza-se o botão *carregar dados de validação*; esta opção estará disponível se a opção *cross-validation* não estiver ativa. No botão *Salvar relatório*, deve ser escolhido o nome e o local onde os relatórios serão gravados e o botão *rodar* inicia o processo de otimização.

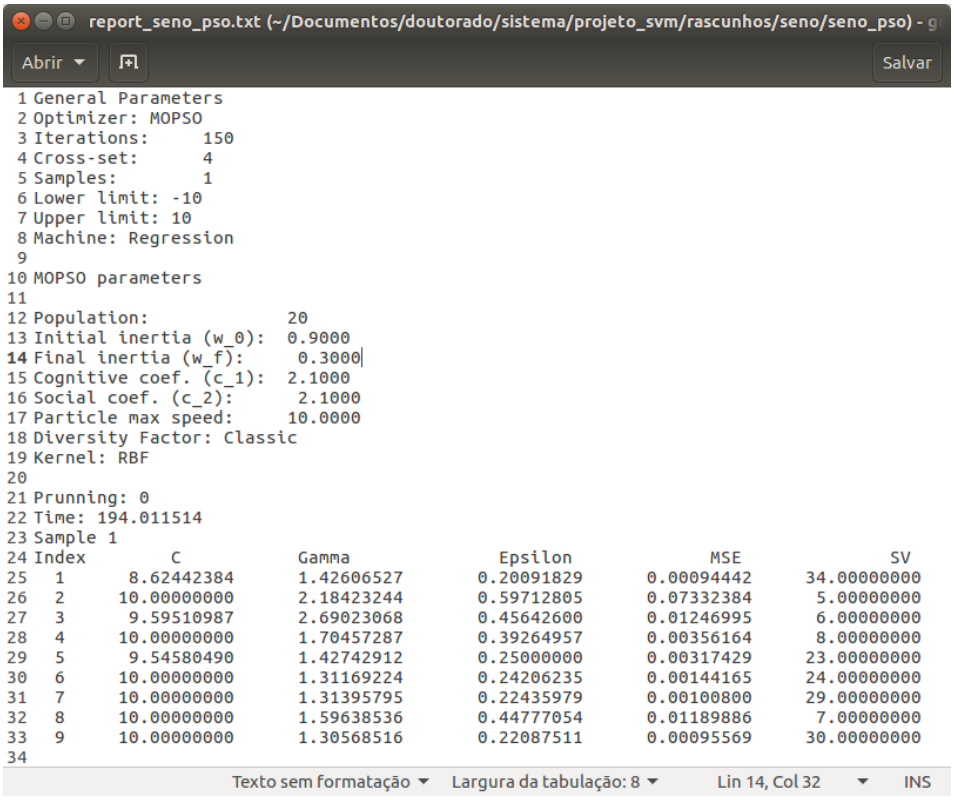


Figura 2.11 – Relatório contendo o Conjunto e a Fronteira de Pareto.

2.5.1 NIOTS: Ambiente Comitê de Máquinas - *Ensembles*

Os métodos de *ensembles* utilizam múltiplos modelos com características distintas para fornecer uma melhor capacidade de generalização, a qual os modelos individualmente possam produzir. Os modelos que compõe os *ensembles* diferem-se pela diversidade: (a) dos dados, (b) de parâmetros, (c) estrutural, (d) divisão e conquista, (e) otimização multiobjetivo e (f) *fuzzy ensemble* (REN; ZHANG; SUGANTHAN, 2016).

O ambiente *Ensemble* do NIOTS foi desenvolvido para criar *ensembles* baseados na diversidade dos parâmetros obtidos a partir do conjunto de Pareto e na estrutural, em que se pode escolher mais de um tipo de *kernel*. A Figura 2.14 apresenta o ambiente *Ensembles* no qual foram carregadas duas PF com *kernels* RBF e Polinomial.

No final do processo de otimização, são gerados três tipos de relatórios representados na Figura 2.12 pelo bloco *Relatórios*. Para cada hiperparâmetro do conjunto de Pareto, é gerado um arquivo relatório, denominado *Dados do Modelo*, que possui todos os dados necessários para implementação do modelo em questão. A Figura 2.10 é um exemplo desse relatório.

2.5.2 NIOTS: Ambiente de Otimização

No ambiente de otimização do NIOTS, é possível realizar a otimização dos hiperparâmetros de classificadores e regressores, usando os algoritmos meta-heurísticos MOPSO e APMT-MODE

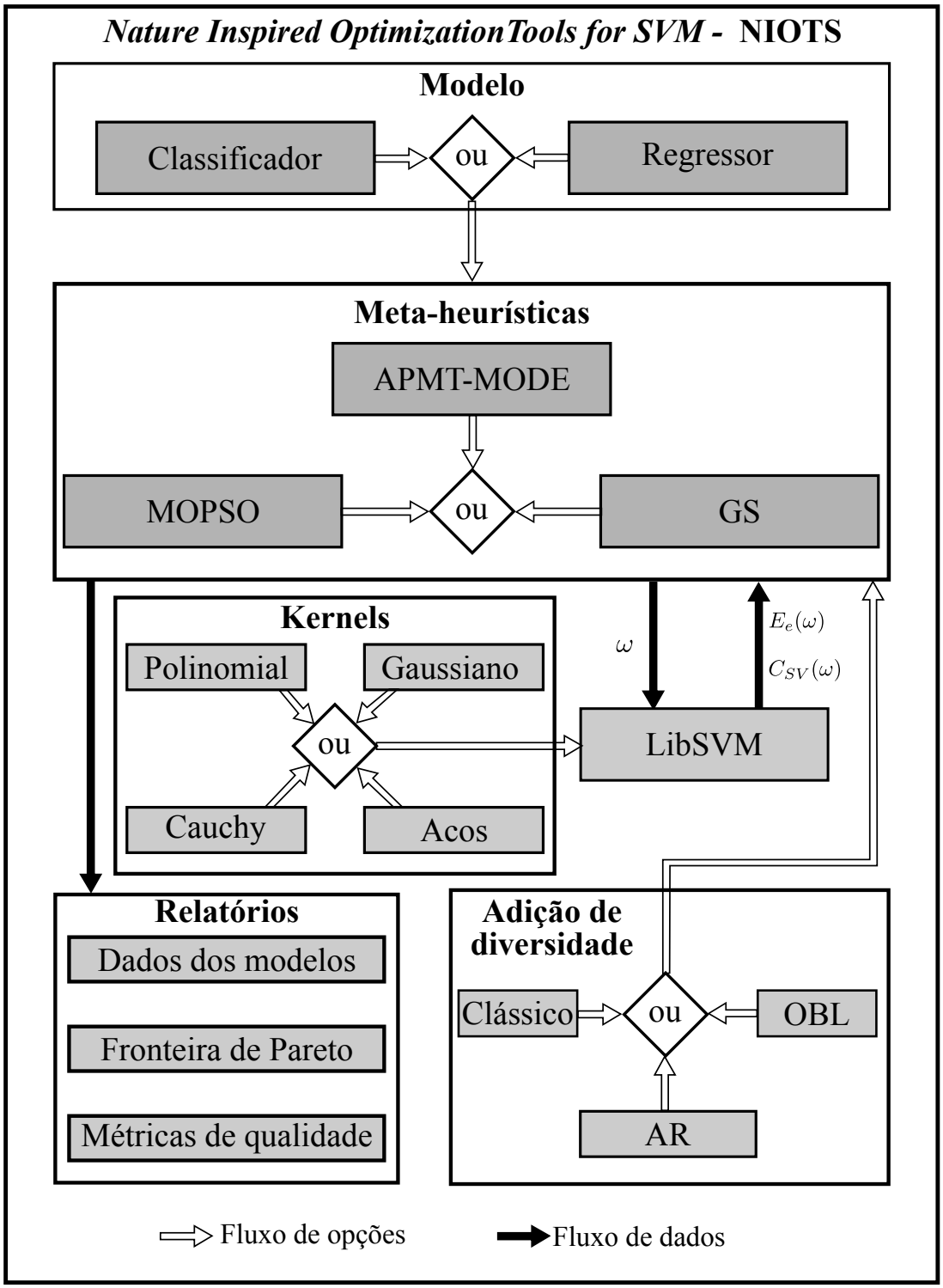


Figura 2.12 – Diagrama de fluxo de dados e opções do NIOTS. Fonte (SANTOS, 2019)

.

para resolver o problema de seleção de parâmetros formulado como um MOOP. Os algoritmos MOPSO e APMT-MODE definem as soluções candidatas que são os parâmetros da ferramenta de treinamento LiBSVM (CHANG; LIN, 2011).

A interface gráfica do NIOTS, Figura 2.13, foi desenvolvida para facilitar o emprego das

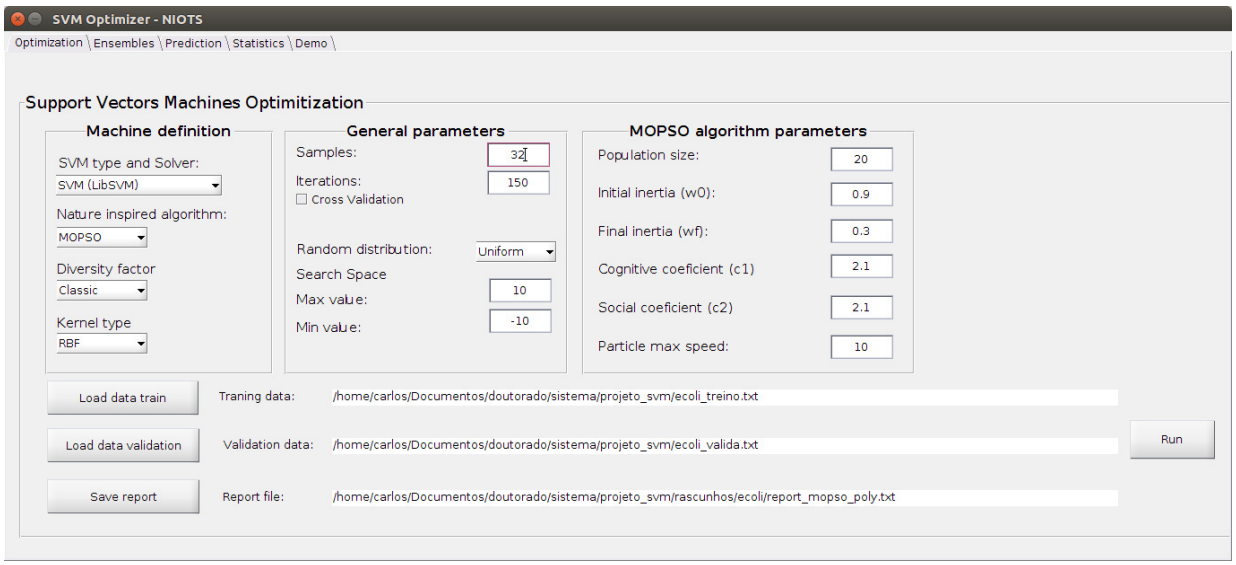


Figura 2.13 – Ambiente gráfico NIOTS-Otimização. Fonte (SANTOS, 2019)

.

amplas possibilidades do sistema pelos usuários. O ambiente de otimização está dividido em três painéis: (a) *Definição da Máquina*, (b) *Parâmetros Gerais* e (c) *Parâmetros dos Algoritmos*, sendo cada opção desses painéis apresentado na Tabela 2.2.

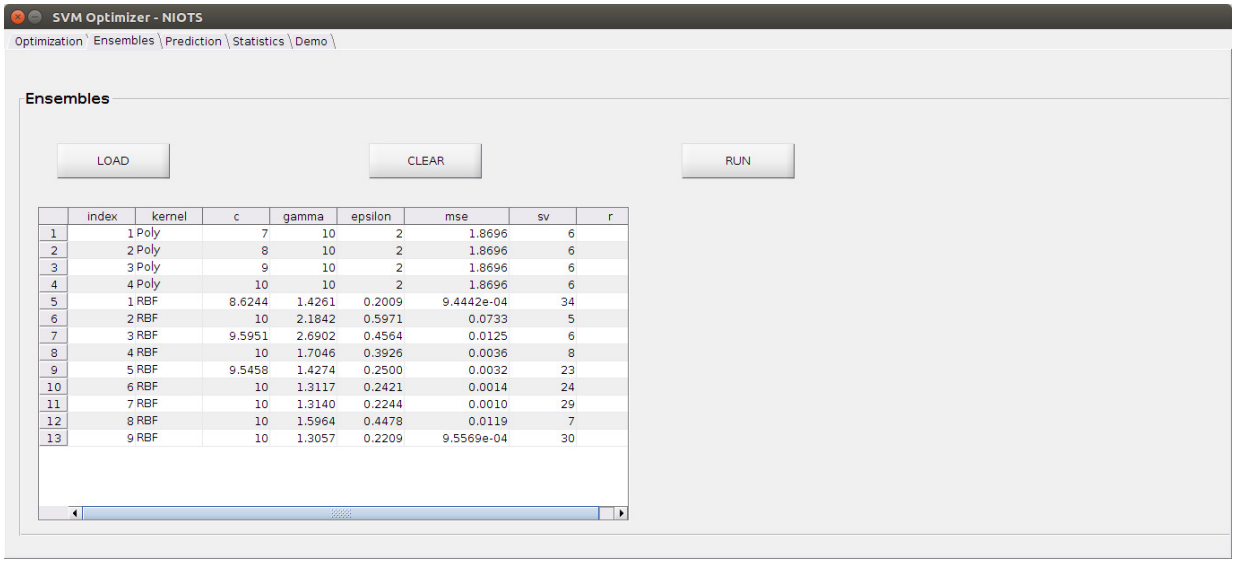


Figura 2.14 – Ambiente gráfico NIOTS-Ensembles. Fonte (SANTOS, 2019)

.

Tabela 2.2 – Descrição das opções do ambiente Otimização.

Painel	**Menu/Campo**	**Opções/descrição**
	Tipo de Modelo	SVM/LiBSVM
		SVR/LiBSVM
		Grid Search SVR

Continua na próxima página.

Definição da Máquina

Tabela 2.2 – *Continuando da página anterior.*

Painel	**Menu/Campo**	**Opções/descrição**
		Grid Search SVM
	Algoritmo de Otimização	MOPSO
		APMT-MODE
	Fator de diversidade	Clássico
		AR
		OBL
	Tipo de Kernel	RBF
		Polinomial
		Acos
		Cauchy
Parâmetros Gerais	Amostras	Quantidade de Amostras independentes
	Iterações	Número de iterações do algoritmo de otimização
	Cross Validation	Esta opção, ao ser desativada, é necessário carregar um arquivo com o conjunto de de validação
		Caso contrário utiliza a Validação Cruzada
	Distribuição	Uniforme
		Normal
		Cauchy
	Espaço de busca	Limite superior de C e Gama
		Limite inferior de C e Gama
Parâmetros do MOPSO	Tamanho da população	Quantidade de partículas do enxame
	Inércia inicial	Fator de inércia inicial
	Inércia final	Fator de inércia final
	Coeficiente Cognitivo	Grau de confiança na partícula
	Coeficiente Social	Grau de confiança no enxame

Continua na próxima página.

Tabela 2.2 – *Continuando da página anterior.*

Painel	**Menu/Campo**	**Opções/descrição**
	Velocidade máxima da partícula	Define a velocidade máxima permitida pela partícula durante o processo de otimização.
Parâmetros do APMT-MODE	Tamanho da população	
	Fator de escala	Fator inicial
	Taxa de recombinação	Taxa inicial

A leitura das PF é realizada ao acionar o botão *LOAD* no qual deve ser indicado o arquivo da PF gerada pelo ambiente Otimização. Quando o botão *RUN* for acionado, serão solicitados dois arquivos, um para o treinamento dos modelos e outro para validação dos *Ensembles*.

A ferramenta, de fato, treina novamente cada modelo com os hiperparâmetros das Fronteiras de Pareto carregadas e realiza todas as combinações possíveis, como os modelos com *Ensembles* de dois, três e quatro modelos. Ao final, a função *Truncate* é utilizada para determinar o conjunto de *Ensembles* não dominados, considerando o MSE e o número total de vetores de suporte.

O resultado é gravado em um arquivo relatório, na mesma pasta do conjunto de treinamento, indicando os elementos dos *Ensembles* e o quanto cada um foi melhor que o melhor modelo individual pertencente ao *Ensemble*.

Para gerar os *Ensembles*, o NIOTS testa todas as possibilidades de dois, três e quatro elementos que, na prática, para grupos com mais elementos, se torna impraticável, sendo deixado para trabalhos futuros o desenvolvimento de um algoritmo Meta-heurístico capaz de definir quais e quantos modelos são necessários para que a complexidade dos *Ensembles* seja minimizada, enquanto a capacidade de generalização seja maximizada. Na Subseção 2.5.3, é descrito o ambiente Predição, no qual os modelos podem ser testados individualmente e gráficos comparativos são gerados.

2.5.3 NIOTS: Ambiente de Predição

No ambiente *Predição*, apresentado na Figura 2.15, podem ser realizados testes com os modelos de funções aproximadoras e dos classificadores gerados pelas soluções dos algoritmos de otimização dos hiperparâmetros. Inicialmente, é necessário escolher entre os modelos *Single-target-SVR* e *Classify-SVM*, em seguida, é definido se os dados que serão lidos possuem ou não rótulos/imagens.

Acionando o botão *Load SV*, uma caixa de diálogo é aberta para que seja indicado o arquivo que contém o modelo a ser avaliado pelo ambiente *Predict*. A Figura 2.10 é um exemplo desse

tipo de arquivo que contém todas as informações necessárias para gerar o modelo. Os dados de teste são carregados ao acionar o botão *Features*, que possui o mesmo formato do arquivo de treinamento.

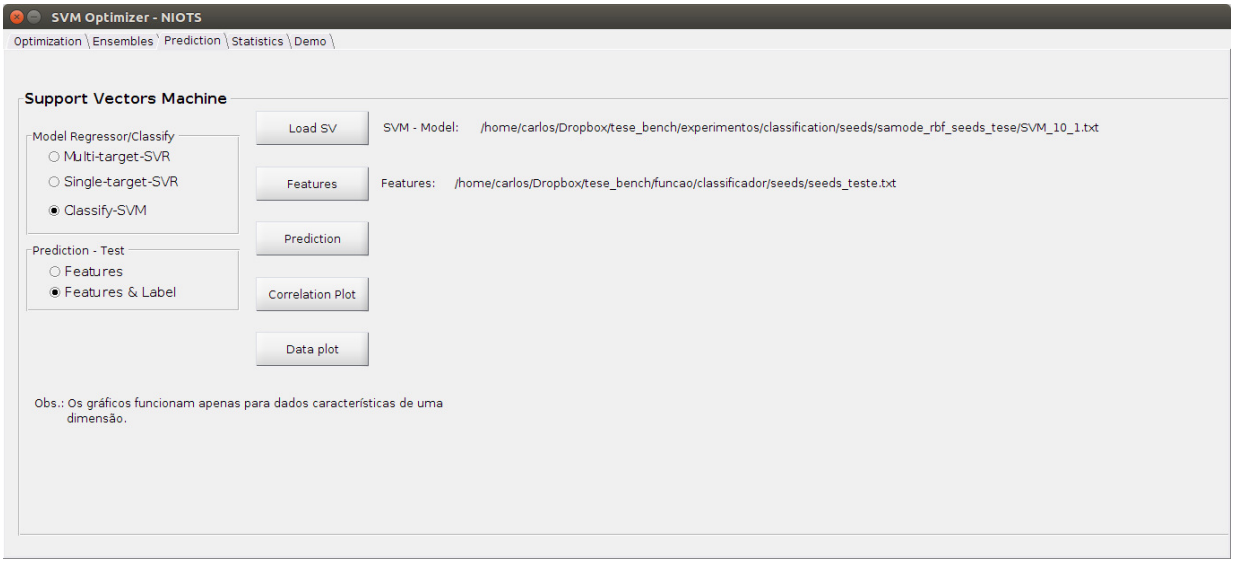


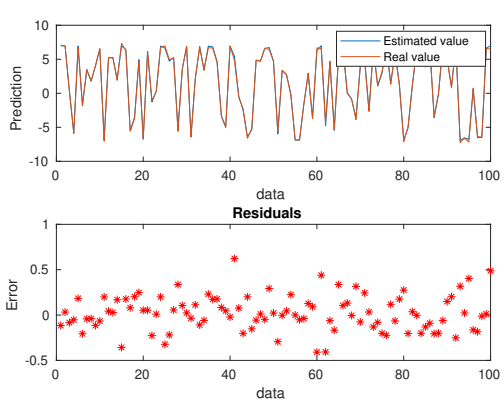
Figura 2.15 – Ambiente gráfico NIOTS - Otimização. Fonte (SANTOS, 2019).

O botão *Prediction* pode ser acionado depois de carregado o arquivo do modelo de SVM/SVR, o arquivo de testes, e escolhido o tipo de modelo que se deseja testar. O acionamento deste retorna o MSE ou AUC para regressores ou classificadores dependendo do modelo em questão. O botão *Prediction* produz, ainda, dois gráficos distintos para as SVRs. A Figura 2.16a confronta as saídas produzidas pelo modelo e as do conjunto de testes e gráficos de resíduos.

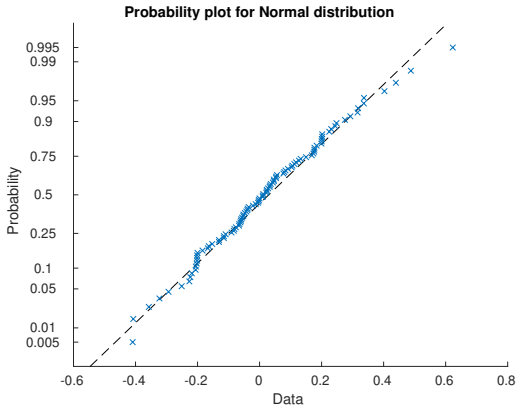
Os resíduos possuem informações importantes para análise do modelo. No gráfico dos resíduos, não deve ser observado qualquer padrão, o que significa que o modelo absorveu toda a informação disponível e suas diferenças são, em sua maioria, devido a ruídos dos dados. A Figura 2.16b representa o teste de normalidade. Neste gráfico, quanto mais próximos os pontos estiverem da linha pontilhada, maior é a normalidade dos resíduos. A normalidade dos resíduos é outro critério utilizado para avaliar a qualidade do regressor em questão.

O gráfico da Figura 2.17 é obtido com o acionamento do botão *Correlation Plot*, no caso de configurada a opção *Single-target-SVR*. No rodapé do gráfico, é impresso o coeficiente de correlação que varia no intervalo $[0, 1]$, quanto mais próximo de um, mehor é o ajuste do modelo. No gráfico, pode-se realizar esta análise visualmente, tendo em conta que melhor é a qualidade do modelo em questão, quando os pontos estão mais próximos à reta pontilhada.

Para classificadores, o acionamento do botão *Correlation Plot* gera o gráfico da curva *Receiver Operating Characteristic* - ROC, sendo *Area Under Curve* - AUC a métrica que mede o quanto um classificador é capaz de distinguir entre duas classes. Os valores do AUC variam entre zero e um, sendo que, quanto maior o valor da métrica, melhor será o classificador. O modelo é considerado ruim para valores próximos de $0,5$.



(a) Gráfico de predição e resíduos.



(b) Teste de normalidade dos resíduos.

Figura 2.16 – Gráficos gerados pela opção *Correlation Plot*. Fonte (SANTOS, 2019)

.

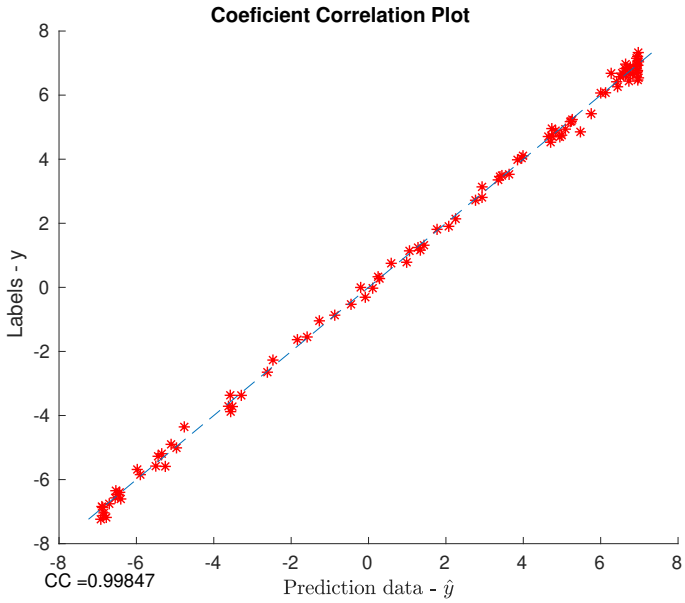


Figura 2.17 – Coeficiente de Correlação.

Na Subseção 2.5.4, o ambiente *Estatística* possibilita a análise estatística dos algoritmos implementados, considerando cada conjunto de treinamento.

2.5.4 NIOTS: Ambiente de Estatística

Quando se trabalha com vários algoritmos de otimização e estes possuem alguns parâmetros a serem configurados, torna-se necessário um estudo estatístico, para que seja possível discernir quais são as opções mais viáveis. O estudo estatístico valida ou não as propostas de algoritmos e suas configurações para determinados problemas. O ambiente *Estatistics* foi implementado para tornar prático e rápido os testes estatísticos realizados.

O ambiente *Estatistics* possui três painéis, o primeiro gera gráficos das métricas *hipervolume*,

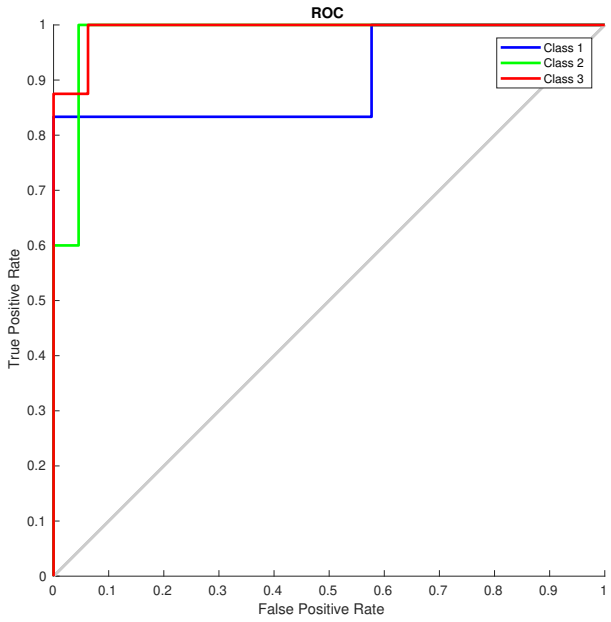


Figura 2.18 – Gráfico da métrica ROC.

cardinalidade do conjunto de não dominados e *spacing* de cada iteração do conjunto de soluções não dominadas. No painel, têm-se as opções *sample means*, *standard deviation* e *median*, conforme Figura 2.19. O cálculo de cada uma dessas estatísticas é realizado com base em cada iteração das amostras. A opção *Pareto front* gera o gráfico da fronteira de Pareto de todas as amostras.

Neste painel, o botão *Data load* possibilita a leitura do arquivo "metricas.txt", que contém os dados necessários para gerar os gráficos (o botão *Run* gera os gráficos).

O painel *Plot Multi-target* tem os mesmos objetivos e funcionalidades do painel *plot single-target*.

No painel *Hypothesis test*, é possível fazer a leitura de duas séries de dados que serão comparadas. As séries são ajustadas e, então, é aplicado o teste Kolmogorov-Smirnov para verificar se a distribuição das amostas provém ou não de uma distribuição Normal. Caso as duas distribuições advenham de uma distribuição Normal, é aplicado o teste ANOVA e são gerados os gráficos de *Boxplot*. Caso as séries não advenham de uma distribuição Normal, é aplicado o teste de Wilconxon e são gerados os gráfico de *Boxplot*.

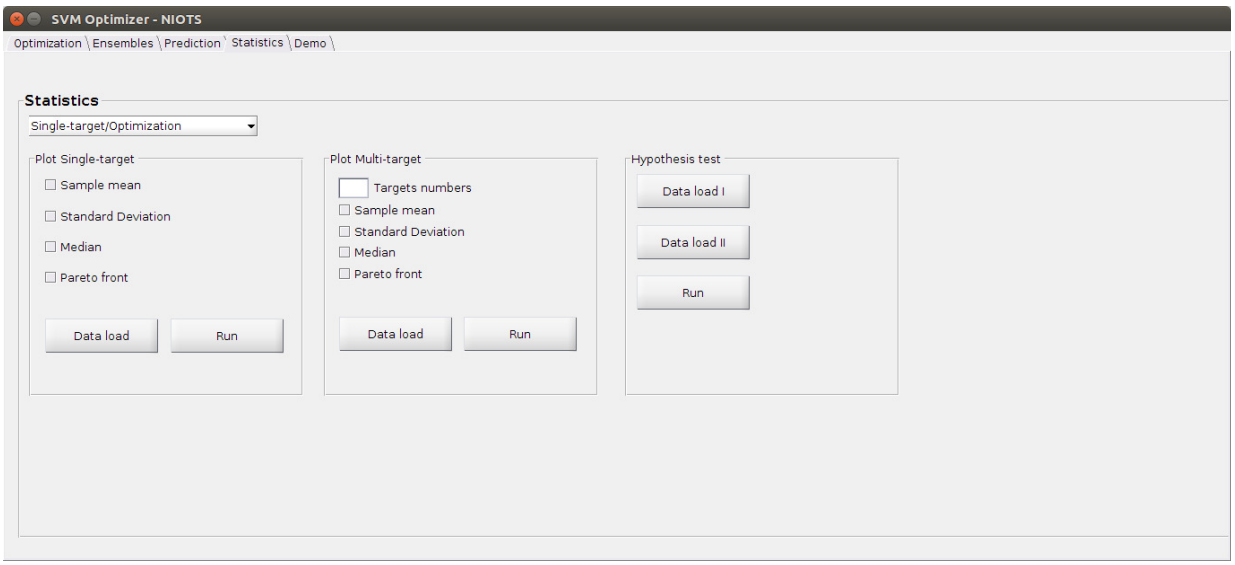


Figura 2.19 – Ambiente gráfico NIOTS - Estatística. Fonte (SANTOS, 2019)

.

2.6 SISTEMAS EMBARCADOS E DISPOSITIVOS INTELIGENTES

Um dispositivo inteligente é um dispositivo eletrônico, geralmente conectado a outros dispositivos ou redes através de diferentes protocolos sem fio, como Bluetooth, NFC, Wi-Fi, LiFi, 3G, etc., que podem operar de forma interativa e autônoma.

Existem vários tipos de dispositivos inteligentes, como : smartphones, carros inteligentes, termostatos inteligentes, campainhas inteligentes, fechaduras inteligentes, refrigeradores inteligentes, tablets, smartwatches, smart bands, chaveiros inteligentes, alto - falantes inteligentes e outros. O termo também pode se referir a um dispositivo que exibe algumas propriedades de computação onipresente, incluindo - embora não necessariamente-inteligência artificial.(VALHOULI, 2010)

Conforme NOERGAARD, um sistema embarcado é um sistema de computação aplicada, diferente de outros tipos de sistemas de computadores, como computadores pessoais (PCs) ou supercomputadores. A definição de "sistema embarcado"é dinâmica e difícil de definir, pois evolui constantemente com avanços tecnológicos e reduções drásticas no custo de implementação de vários componentes de hardware e software. Dentre as características mais comuns de um sistema embarcado, podemos citar:

- Nos sistemas embarcados às suas funcionalidades de hardware e/ou software são mais limitados do que um PC.
- Um sistema embarcado é projetado para executar uma função dedicada.
- Em um sistema embarcado os requisitos de qualidade e confiabilidade são, normalmente, mais altos do que outros tipos de sistemas de computação.

2.7 INTELIGÊNCIA ARTIFICIAL NA METEOROLOGIA

A meteorologia se utiliza de recursos computacionais há bastante tempo. Isto deve-se ao fato de suas necessidades envolverem, normalmente, grande volume de dados e altos requisitos de processamento, para a previsão de eventos de tempo e clima.

A utilização da computação na meteorologia envolve não somente aspectos operacionais, como também pesquisas cientificas. Muitas destas pesquisas buscam a otimização dos modelos matemáticos atuais para previsão de eventos climáticos e de tempo. Entretanto, a utilização de IA em meteorologia aconteceu a partir da década de 1990. As pesquisas mais recentes demonstram um crescimento substancial nesta área.

Para este trabalho foram tabuladas algumas destas publicações, conforme a Tabela 2.3. Sendo que foram considerados principalmente fatores que dizem respeito a este trabalho, como: (a) tipo de uso da previsão (regressor ou classificador), (b) quais dados foram utilizados (local ou global), (c) o algoritmo utilizado para previsão, e (d) se a solução foi ou não utilizada em um sistema embarcado.

Tabela 2.3 – Artigos Pesquisados

Ref	Autor	Tipo	Dados	Algoritimo	Embar.	Ano
1	Zheng et al.	Regressor	Local	LR	Não	2015
			Global	NN		
2	Mestre et al	Regressor	Local	NEN	Sim	2015
			Global	NN		
				MOGA		
3	Chen et al.	Regressor	Global	SVM	Não	2004
			Local	SVR		
4	Das and Ghosh	Regressor	Global	FBN	Não	2014
			Local			
5	Sharma et al	Regressor	Global	SVM	Não	2011
			Local			
6	Wu et al.	Regressor	Global	NN	Não	2003
			Local			
7	Krasnopolsky and Rabinovitz	Regressor		NN	Não	2005
8	Rasouli et al.	Regressor		BNN	Não	2012
				SVM		
				MLR		
				GP		
9	Lu et al.	Regressor		SVM	Não	2002
10	Trafalis et al	Regressor	Global	SVR	Não	2003
11	Wang et al	Regressor		NN	Não	2003
12	Nasseri et al	Regressor		NN/GA	Não	2008
13	Nurcahyo et al	Regressor		NN/GA	Não	2014
14	Wel	Regressor		SVM	Não	2012
15	Seo et al	Regressor		SVM	Não	2014
				k-NN		
16	Ferreira et al	Regressor	Local	SVM	Não	2019
			Global	NN		
17	sfetsos and Coonick	Regressor	Global	NN	Não	2000
18	Fuentes et al	Regressor	Local		Sim	2014
19	Lu et al.	Regressor	Local	SVM	Não	2002
			Global			
20	Hong	Regressor	Local	NN	Não	2008
			Local	SVM		
21	Kisi and Cimen	Regressor	Local	NN	Não	2012
		Regressor	Local	SVM		
22	Sivapragasam et al	Regressor	Global	SVM	Não	2001
23	Hema and Kant	Regressor	Local	SVM	Não	2016

Sigla	Algoritimo	Sigla	Algoritimo
BNN	*Bayesian Neural Network*	MLR	*Multiple Linear Regression*
FBN	*Fuzzy-Bayesian networks*	MOGA	*Multi-objective Genetic algorithms*
GA	*Genetic Algorithms*	NEN	*Nearest-neighbors*
GP	*Gaussian process*	NN	*Neural Network*
k-NN	*k-Nearest Neighbors*	SVM	*Support Vector Machine*
LR	*Linear Regression*	SVR	*Support Vector Regression*

A Figura 2.20, apresenta a distribuição do uso de cada um dos algorítimos para o universo das publicações pesquisadas. Como pode ser observado, existe uma grande tendência na utilização dos algorítimos SVM e NN, por fatores como: (a) facilidades de implementação e (b) eficiência.

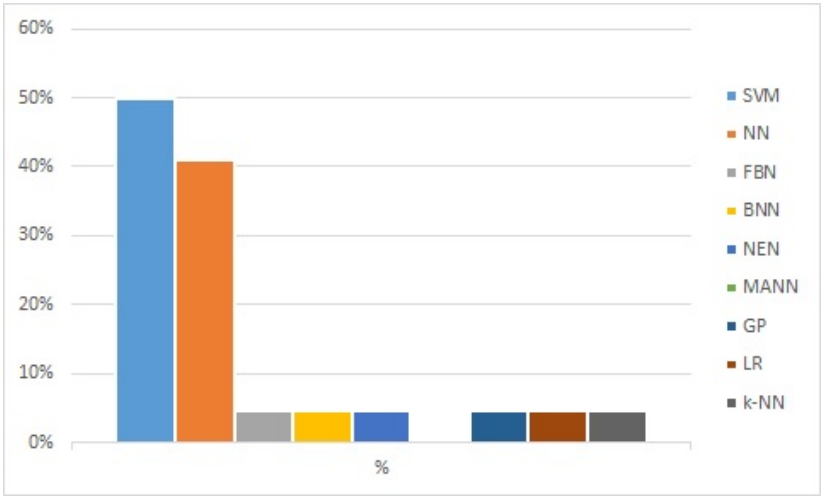


Figura 2.20 – Distribuição dos Algorítimos

Na análise para quantificação de sistemas embarcados, para o universo dos artigos pesquisados, foi observada uma pequena utilização destes sistemas (aproximadamente 9%), conforme demonstra a Figura 2.21.

Figura 2.21 – Utilização de Sistemas Embarcados

A Figura 2.22, apresenta a distribuição do uso de dados para realização da previsão, no universo das publicações pesquisadas. Como se pode observar, não existe grandes diferenças entre o uso de dados locais e dados globais.

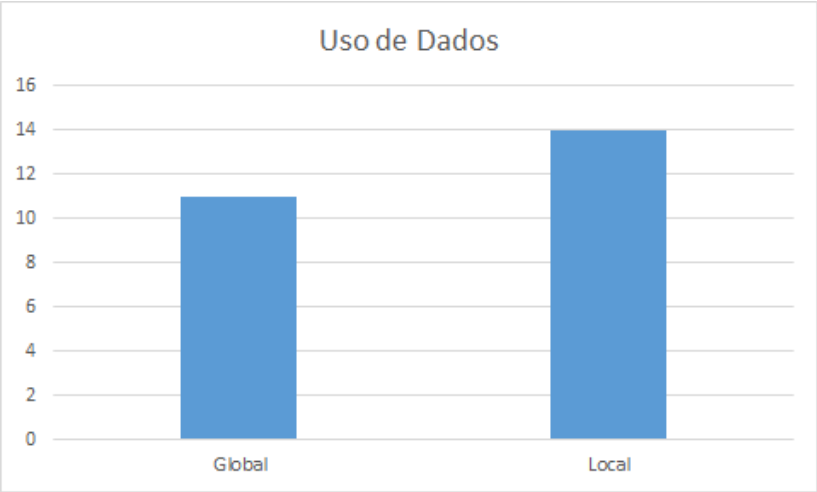


Figura 2.22 – Distribuição do Uso de Dados para Previsão

2.8 DISPOSITIVOS INTELIGENTES NA METEOROLOGIA

Atualmente, empresas disponibilizam estações automáticas, para obtenção de dados meteorológicos em tempo real (Figura 2.23). Essas estações visam substituir e automatizar as observações convencionais, que eram realizadas em intervalos de 6 horas, por observadores, o qual se tornou padrão de intervalo para obtenção de dados. Porém, estas estações não utilizam sistemas embarcados para desenvolver os sistemas de previsão de tempo.

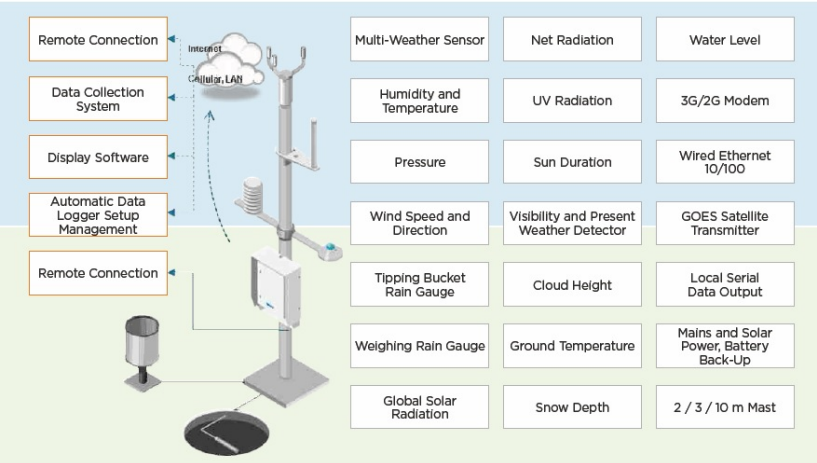


Figura 2.23 – EstacaoAutomatica*(Fonte : Vaisala, 2019 Weather Station)*

Uma rede de dispositivos (estação meteorológica, sensor de umidade e rede de sensores sem fio) capaz de realizar apoio para a irrigação inteligente, feita pela extração de dados climáticos, obitidos de históricos externos, dados baseados na WEB e dos dados dos próprios dispositivos, foi desenvolvido por HEMA; KANT, em 2016. Nesta proposta o programa de irrigação inteligente

com sensoriamento e a tomada de decisão da irrigação dependem dos dados climáticos atuais. No entanto, o sistema pode ser melhorado usando a precipitação em tempo real da vizinhança, para estimar a precipitação local aproximada. Este método pode resultar em benefícios como apoio para economia de água e melhoria do plantio.

2.9 CONCLUSÕES DO CAPÍTULO

O referencial teórico e os trabalhos correlatos apresentados neste capítulo foram utilizados para o desenvolvimento desta pesquisa. Pôde ser verificado que a escolha das SVMs representa uma tendência da modelagem de previsão de chuva que envolvem soluções usando IA. Porém, a utilização de classificadores, para estes tipos de previsões, não foi encontrada na literatura, o que reforça a proposta deste trabalho.

3 METODOLOGIA APLICADA AO DESENVOLVIMENTO DESTA PESQUISA

Este capítulo descreve a metodologia utilizada para o desenvolvimento deste trabalho. Inicialmente, são apresentadas as diretrizes que nortearam a metodologia, em seguida é apresentada uma visão geral da metodologia, e finalmente são detalhadas cada uma das etapas da metodologia com suas respectivas tarefas.

A metodologia proposta apresentada neste capítulo, poderá ser uma contribuição relevante para a literatura, devido a falta de uma metodologia específica para os processos de uso de algoritmos de Aprendizado de Máquina.

3.1 DIRETRIZES DO TRABALHO

Visando uma adequação aos objetivos desta pesquisa, foram estabelecidas diretrizes para a formulação e a condução da metodologia a ser utilizada no trabalho. Assim ficaram estabelecidos os seguintes pilares:

- Uso de dados locais.
- Fácil implementação do classificador.
- Baixo custo computacional.
- Baixo volume de dados.
- Previsão de precipitação para um período de 24 horas.

3.1.1 Uso de dados locais

O uso de dados locais foi estabelecido como diretriz, visto que para obtenção de dados globais, tais como índices *Elniño/Laniña* e dados de radio sondagens, se tornaria de grande dificuldade, para aquisição em estações remotas nas diversas localidades da Amazônia. Os dados obtidos localmente, pela própria estação, são, entre outros: (a) precipitação e (b) temperatura. Assim, estes dois conjuntos de dados são de fácil obtenção localmente e poderão ser utilizados nas previsões de classes de precipitações, à qual este trabalho se propõe.

3.1.2 Facilidade de Implementação

Para uma melhor eficiência e eficácia da aplicação em um dispositivo inteligente, a utilização de baixo volume de dados assim como a utilização de algorítimos de fácil implementação foi uma das diretrizes para o desenvolvimento dos trabalhos, dadas as limitações dos recursos disponíveis nos dispositivos à serem utilizados. Isto deve-se ao fato das condições adversas com as quais esses dispositivos operam.

A utilização de algorítimos de fácil implementação foi uma das diretrizes para o desenvolvimento dos trabalhos, dadas as limitações dos recursos disponíveis nos dispositivos a serem usados.

3.1.3 Baixo Custo Computacional da Técnica a ser Desenvolvida

Uma das aplicações da solução proposta, por este trabalho, será para o desenvolvimento de estações meteorológicas inteligentes. Neste caso, se faz necessário o controle dos recursos computacionais a serem utilizados, tendo em conta que estações meteorológicas, normalmente, possuem baixos recursos computacionais, pois operam em ambientes adversos, sujeitos a fatores como: (a) chuvas, (b) ventos, (c) variações de temperatura, e (d) ausência de fontes de energia.

3.1.4 Previsão de Precipitação para Período de 24 hs

O processo de previsão, desenvolvido neste trabalho, envolve a aplicação do algorítimo em um determinado instante do tempo, conhecido como *ação da maquina de previsão*. Assim, neste momento, serão previstos os próximos acontecimentos com base nos dados históricos de diversas variáveis, e que são obtidos por um determinado período (janela histórica). Com isto pretende-se realizar a previsão para o período desejado, conforme ilustra a Figura 3.1.

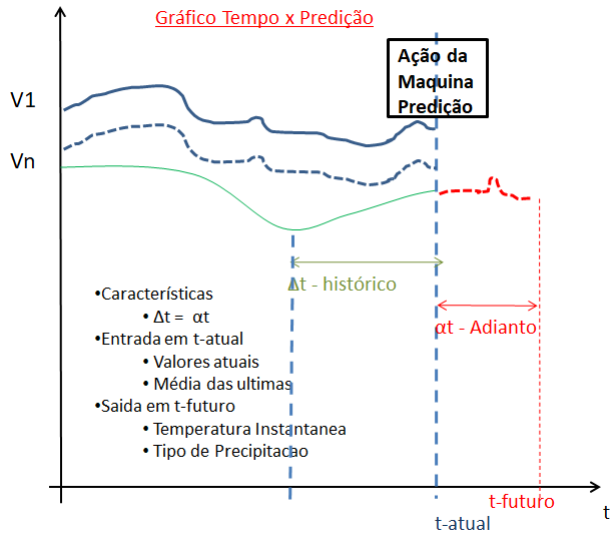


Figura 3.1 – Ação da Máquina de Previsão- Variáveis x Tempo-*fonte: o autor.*

Para a previsão de curto prazo, foi estabelecido um período mínimo de 24 horas; podendo ser períodos de: (a) 24 (vinte quatro) horas, (b) 48 horas e (c) 72 horas. Assim para previsão de 24 horas, são utilizados os dados históricos das últimas 24 horas, para previsão de 48 horas os dados históricos das ultimas 48 horas e para 72 horas os dados históricos das últimas 72 horas. Assim estabeleceu-se os conceitos de *Janela de Históricos* e *Janela Previsão*, conforme discutido a seguir:

3.1.4.1 Janela de Previsão

A janela de previsão, olha o futuro e trabalha com a periodicidade mínima de 24 horas, ou seja, com base no momento atual será previsto o que ocorrerá nas próximas 24 horas no local do dispositivo; podendo ser de até 72 horas em múltiplos de 24 horas. O anterior está representado por $\alpha\ t$ na Figura 3.1, a frente da máquina de previsão (SVM).

3.1.4.2 Janela de Históricos

Uma janela de dados históricos será utilizada para a obtenção dos dados passados, onde os mesmos darão suporte ao processo de previsão. Desta forma, os dados históricos a serem utilizados para previsão das classes de precipitação seguirão os mesmos períodos da janela da previsão (24 hs, 48 hs ou 72 hs). Isto é representado por Δt na Figura 3.1, que ocorre anteriormente a máquina de previsão (SVM).

3.2 VISÃO GERAL DA METODOLOGIA

Visando uma melhor condução das atividades deste trabalho, foi desenvolvida uma metodologia para aplicação no problema proposto. Esta metodologia foi construída com base na experiência do autor e em propostas metodológicas utilizadas para aprendizado de máquina.

As atividades desta metodologia foram agrupadas em etapas, e sequenciadas conforme apresentada na figura 3.2.

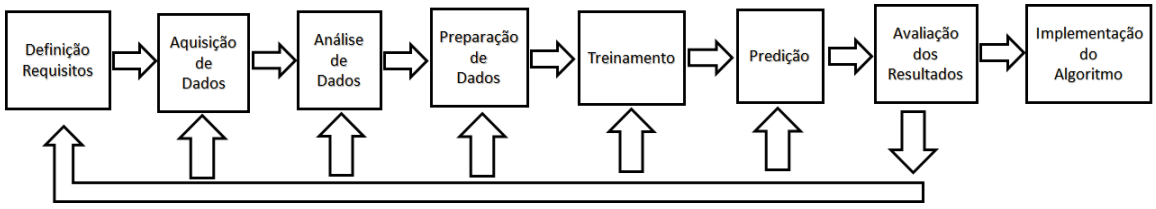


Figura 3.2 – Etapas da Metodologia - *Fonte : o autor*

Apesar da proposta de etapas ser de forma sequencial, se faz necessário eventuais retornos para etapas anteriores em função dos resultados obtidos por cada uma das etapas, para realização

de ajustes visando assim atender os objetivos do trabalho. Cada etapa possui um objetivo e escopo de atividades próprias, conforme serão descritas a seguir.

3.3 ETAPA DE DEFINIÇÃO DE REQUISITOS

Esta etapa visa nortear todo o processo do estudo de caso. Portanto, envolve todas as atividades de definição dos parâmetros e procedimentos iniciais necessários para o desenvolvimento das etapas seguintes.

Trata-se de uma das etapas mais importantes, pois estas definições e parâmetros irão conduzir e influenciar todas as demais etapas.

3.3.1 Definição das Classes

A tarefa de definição das classes consiste em identificar quais as classes serão utilizadas para os algoritmos de classificação de Aprendizado de Máquina.

Usando como base a proposta de SOUZA; AZEVEDO; ARAÚJO 2012 para classes de precipitação, foi estabelecida uma Tabela de Classes conforme descrito em Tabela 3.1.

Tabela 3.1 – Classes de Precipitação

Classe	**Descrição**
1	Sem Chuva
2	Chuva fraca
3	Chuva moderada
4	Chuva forte
5	Chuva muito forte (Tempestade)

3.4 ETAPA DE AQUISIÇÃO DE DADOS

A tarefa de aquisição de dados consiste na escolha e obtenção dos dados das fontes de dados disponíveis. Tais fontes de dados devem ser as mais reais possíveis, já que a qualidade dos dados são fatores decisivos para obtenção dos melhores e mais confiáveis resultados.

3.5 ETAPA DE ANÁLISE DOS DADOS

As tarefas desta etapa consistiram em avaliar os dados recebidos e que foram utilizados na SVM. Portanto, esta etapa é de fundamental importância, pois os resultados desta análise implicaram diretamente na qualidade dos resultados.

Para esta análise foram necessários os conhecimentos e experiências de meteorologistas, além do uso de técnicas como: (a) diagramação de dados estatísticos e (b) seleção e classificação das variáveis. Assim, neste etapa, foram utilizadas diversas técnicas como: (a) análise de dados faltantes, (b) diagramação dos dados, (c) análise multivariada-PCA, (d) distribuição de dados, e (e) avaliação de especialistas.

A base para análise dos dados desta etapa foram os dados das variáveis normalmente utilizados nas estações meteorológicas. Estas variáveis, e suas unidades, se encontram descritas na Tabela 3.2, e foram utilizadas para todo o processo proposto por está metodologia.

Tabela 3.2 – Variáveis Meteorológicas das Estações Automatizadas - Fonte INMET

Variável	**Unidade**
Nome da estação observada	
Data do evento	
Hora GMT do evento	
Valor da temperatura instantânea	^{o}C
Valor da temperatura máxima	^{o}C
Valor da temperatura miníma	^{o}C
Valor da umidade instantânea	%
Valor da umidade máxima	%
Valor da umidade miníma	%
Valor da temperatura instantânea do ponto de orvalho	^{o}C
Valor da temperatura máxima do ponto de orvalho	^{o}C
Valor da temperatura miníma do ponto de orvalho	^{o}C
Valor da pressão instantânea	hPa
Valor da pressão máxima	hPa
Valor da pressão miníma	hPa
Direção do vento	m/s
Velocidade do vento	m/s
Rajada	m/s
Radiação do evento	W
Precipitação do evento	mm

3.5.1 Análise de Dados Faltantes

Apesar da busca por uma completa utilização dos dados disponíveis, para realização dos experimentos, a realidade mostra que sempre irão ocorrer falta de dados. Os motivos são diversos, sejam por falha nos dispositivos de aquisição ou por falta de observação humanas. Estes fatos são reais e não podem ser desprezados, pois a ausência de dados pode implicar, diretamente, na qualidade dos resultados; tendo em conta que eles deverão produzir, pelos algorítimos utilizados, as saídas desejadas.

A análise de dados faltantes, será feita segundo a proposta HOENS; CHAWLA 2013, que consiste em verificar como será tratada a ausência dos dados provocados por estas falhas; podendo ser resolvidos por interpolação ou por descarte.

3.5.2 Análise Relevância das Variáveis

Como normalmente existe uma gama muito grande de variáveis envolvidas nos processos, recomenda-se utilizar técnicas estatísticas para análise da importância destas variáveis, visando a otimização dos recursos e melhoria dos resultados. A aplicação de análise multivariada (tal como o PCA (WOLD; ESBENSEN; GELADI, 1987)) para estes casos, além de uma avaliação por especialista, são práticas normalmente utilizadas.

3.5.2.1 Análise Multivariada

As análises multivariadas são utilizadas principalmente para encontrar a variável menos representativa e eliminá-la; simplificando assim os modelos estatísticos, em que o número de variáveis torna-se um problema para compreender a relação entre os vários grupos de dados.

A Análise de Componentes Principais (ACP) ou *Principal Component Analysis* (PCA)((WOLD; ESBENSEN; GELADI, 1987)), é um procedimento matemático que utiliza uma transformação ortogonal (ortogonalização de vetores) para converter um conjunto de observações de variáveis possivelmente correlacionadas em um conjunto de valores de variáveis linearmente não correlacionadas, chamadas de *componentes principais*. O número de componentes principais é menor ou igual ao número de variáveis originais. Esta transformação é definida de forma que o primeiro componente principal tem a maior variância possível (ou seja, é responsável pelo máximo de variabilidade nos dados), e cada componente seguinte, por sua vez, tem a máxima variância sob a restrição de ser ortogonal, por exemplo, não correlacionado com os componentes anteriores.

Os componentes principais são garantidamente independentes apenas se os dados forem normalmente distribuídos (conjuntamente). O PCA é sensível à escala relativa das variáveis originais, dependendo da área de aplicação. O PCA é também conhecido como transformada de Karhunen-Loève (KLT) discreta, transformada de Hotelling ou decomposição ortogonal própria (POD).

3.5.2.2 Avaliação de Especialista

Mesmo com utilização das diversas técnicas propostas o conhecimento e prática de especialistas é de extrema importância para o processo. Visando a melhor qualificação dos dados recebidos a consulta a especialistas em meteorologia é a finalidade desta tarefa.

3.5.3 Análise da Distribuição dos Dados

Consiste em plotar gráficos estatísticos visando identificar as distribuições dos dados à serem utilizados. A distribuição de percentuais e quantitativas dos dados dentro de cada uma das classes, a serem classificadas, é de extrema importância. Uma boa e realística distribuição implica diretamente na qualidade dos resultados a serem obtidos.

3.5.4 Análise da Sazonalidade dos Dados

A visão histórica dos dados, dentro de um período, poderá identificar uma sazonalidade das ocorrências. Está identificação poderá gerar inclusões, alterações dos atributos dos dados para uma melhoria dos resultados.

3.6 ETAPA DE PREPARAÇÃO DOS DADOS

Finda a etapa de análise de dados, se fez necessária a preparação dos dados para realização dos experimentos computacionais. Nesta etapa, foram realizadas todas as tarefas de preparação dos dados, para serem aplicados nas etapas seguintes. Assim nesta etapa, foram realizadas as seguintes atividades:

- Ordenamento dos Dados Brutos
- Carga da Tabela de Relevância das Variáveis
- Tratamento de Dados incompletos
- Definição das Classes de Precipitação
- Seleção da Janela de Previsão
- Cálculo da Precipitação Acumulada para Janela
- Cálculo das Média/Janela/Janelinha
- Aplicação Classes de Precipitação
- Separação dos Dados - Treinamento/Validação/Testes

Esta etapa apresenta o seguinte diagrama, conforme Figura 3.3 das entradas recebidas com suas respectivas saídas geradas :

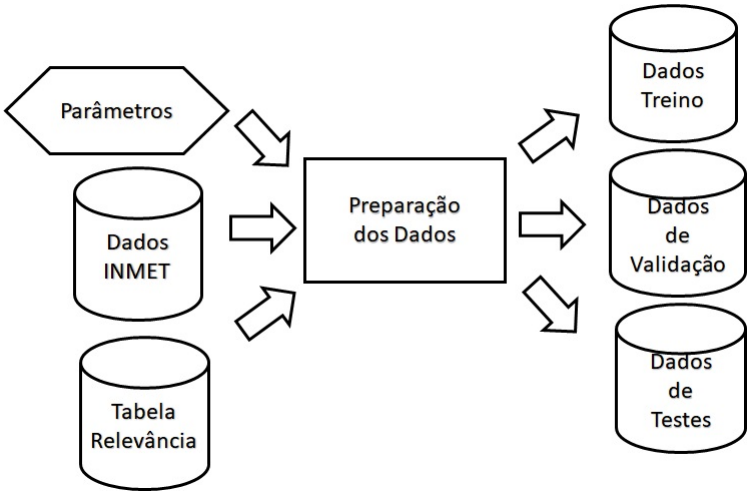


Figura 3.3 – Fluxo de Preparação dos Dados

3.6.1 Ordenamento dos Dados Brutos

Esta tarefa consiste em classificar os dados recebidos por ordem crescente de tempo (ano, mês, dia e hora) da ocorrência do evento. Esta tarefa se faz necessária porque, normalmente, os dados recebidos não vem ordenados de forma desejada.

3.6.2 Carga da Tabela de Relevância das Variáveis

Em função do número de variáveis envolvidas nos processo, nem todas elas são de total importância para a previsão desejada. Com base nos resultados obtidos na etapa anterior (análise de dados), é criada uma *Tabela de Relevância* de variáveis a qual irá conduzir o uso destas no processo de classificação.

3.6.3 Tratamento de Dados Incompletos

Realizar o processo de *Tratamento de Dados Incompletos* é o objetivo desta tarefa. Ela deve ser realizada com base nos resultados da etapa anterior.

3.6.4 Definição das Classes de Classificação

Com base nos dados obtidos, deve-se estabelecer quais valores das variáveis serão utilizadas por cada uma das classes que serão usadas para classificação. Neste caso, técnicas como média e quartis podem ser utilizadas.

3.6.5 Seleção da Janela de Previsão

Estabelecer os valores para as janelas de tempo que serão utilizadas na previsão é o objetivo desta tarefa. Assim, dentro do escopo do trabalho, poderão ser utilizadas janelas de períodos mínimos de 24 (vinte e quatro) horas e seus múltiplos, limitando-se ao máximo de 72(setenta e duas) horas.

3.6.6 Cálculo da Precipitação Acumulada para Janela

Consiste em calcular, para cada ocorrência, a precipitação acumulada nas últimas horas da janela a ser utilizada para previsão. Isto visa encontrar valores para uma série histórica da precipitação.

3.6.7 Aplicação das Classes de Precipitação

Consiste em identificar, em cada amostra de dado horário, qual classe da precipitação que ela corresponde dentro das classes de precipitações estabelecidas. Assim, para cada amostra, se verifica o valor de sua precipitação e a enquadra dentre uma das classes de precipitação correspondente da *Tabela de Classes* Tabelatabclasses, e que foi previamente definida.

3.6.8 Geração dos Dados

Esta tarefa visa, a partir dos dados gerados pelas atividades anteriores, gerar os dados necessários para o treinamento, validação e teste dos algoritmos de SVM. Para isto os dados recebidos, após tratamento, serão separados em três grupos: (a) Treinamento, (b) Validação e (c) Teste. Para a separação dos dados, foi desenvolvido um *script* para realizar as seguintes tarefas:

- Separação dos Dados - Treinamento
- Separação dos Dados - Validação
- Separação dos Dados - Testes

3.6.8.1 Dados de Treinamento

São os dados utilizados para realização do treinamento dos algoritmos de treinamento e configuração da SVM (seguindo a estrutura do NIOTs). Normalmente, representam o maior volume da massa de dados (em torno de 70% do volume total).

3.6.8.2 Dados de Validação

São os dados utilizados para realizar a validação dos resultados dos algoritmos de treinamento da SVM. A utilização destes dados, ocorre durante a etapa de treinamento a fim de identificar a qualidade dos resultados gerados. Estes dados representam 15 % do volume total dos dados.

3.6.8.3 Dados de Teste

São os dados utilizados para teste da SVM obtida a partir do NIOTs. A utilização destes dados ocorre durante a etapa de treinamento e previsão, no NIOTS. Representam 15 % do volume total dos dados.

3.7 ETAPA DE TREINAMENTO

Trata-se de uma das etapas mais importante deste trabalho. Nela foram realizadas as atividades de treinamento dos algorítimos para posterior aplicação no processo de previsão. Nesta etapa foram aplicadas as técnicas de IA para realizar o treinamento e validação, através de ciclos de experimentos, utilizando o aplicativo NIOTS, descrito em 2.5.

3.7.1 Considerações sobre os Experimentos

Os experimentos foram realizados durante esta fase, e obedeceram os princípios de treinamentos mínimos para produção de resultados significativos, e que podem ser avaliados dentro das métricas e objetivos do trabalho.

Visando comparar os resultados das previsões, foi utilizada uma função de Base Radial (RBF) e uma função polinomial(Poly), como funções *kernel* para o algorítimo SVM. Os experimentos de treinamento e previsão foram divididos em 2 grupos, cada um usando um dos 2 algorítimos de otimização (MODE e MOSPO), para cada Kernel do SVM. resultando assim em 4 conjuntos para treinamento e previsão, para cada período (24, 48 e 72 horas). Cada experimento foi realizado utilizando o aplicativo NIOTS, descrito em 2.5.

3.7.2 Métrica Utilizada

Para modelos classificadores, a precisão é largamente utilizada como métrica para avaliar o desempenho destes modelos. Entretanto, a precisão não é capaz de capturar diferentes fatores que caracterizam a eficiência de um classificador. Por exemplo, a precisão é especialmente enganosa na classificação de dados desbalanceados, ou seja, um conjunto de treinamento que possui 80% dos dados de uma classe, e se o modelo independente da entrada sempre decidir por essa classe, ter-se-á uma precisão de 80%. Desse modo, para problemas desbalanceados e multiclasses, a mé-

trica proposta por Carbonero-Ruz *et al.* em (CARBONERO-RUZ et al., 2017) adapta-se melhor para direcionar os algoritmos meta-heurísticos no processo de otimização. Em um problema com Q classes e N instâncias, em que N_q são as instâncias pertencentes a q-ésima classe, a precisão do classificador é dada pelas Equações 3.1 e 3.2

$$C = \sum_{q=1}^{Q} \frac{C_q}{N_q} \frac{N_q}{N}, \tag{3.1}$$

em que C_q é a quantidade de dados classificados corretamente pertencente à classe q, $\frac{C_q}{N_q}$ é a precisão na classe q e $\frac{N_q}{N}$ é a probabilidade (baseado na disponibilidade do conjunto de treinamento) que uma instância tem de pertencer à mesma classe,

$$D = \sqrt{\sum_{q=1}^{Q} \left(\frac{C_q}{N_q} - C \right)^2 \frac{N_q}{N}}. \tag{3.2}$$

Valores de D próximos de zero corresponde à grande representatividade de C. Um modelo tem valor ótimo D quando este é exatamente igual a zero. Isto somente acontece no caso de a taxa de sucesso ser igual em todas as classes. Por outro lado, D aumenta com diferentes taxas de sucesso em cada classe. Isto se torna muito útil quando o equilíbrio de precisão entre as classes é importante (CARBONERO-RUZ et al., 2017).

3.8 ETAPA DE PREVISÃO

Esta etapa trata de realizar as tarefas relativas ao objetivo principal deste trabalho. Nela foram realizadas as atividades de previsão, usando como base os resultados obtidos durante a etapa de treinamento (máquinas de vetores), aplicando os dados reais, para validação dos resultados. Assim, foram aplicados, em cada previsão um conjunto de dados selecionados para testes, utilizando uma das SVMs resultante do treinamento.

3.9 ETAPA DE AVALIAÇÃO DOS RESULTADOS

Nesta etapa são aplicadas métricas para avaliação dos resultados. Tais métricas devem estar em consonância aos objetivos do trabalho, devendo conciliar as necessidades dos assuntos envolvidos. Para avaliação dos resultados foram utilizados as seguintes técnicas e métricas:

3.9.1 Avaliação por Especialista

Esta tarefa consiste em validar com especialistas da área de meteorologia a qualidade dos resultados obtidos nos experimentos.

3.9.2 Gráficos Histograma

Utilizar gráficos estatísticos, como histogramas, contribuem para a avaliação quantitativa dos resultados.

3.9.3 Matriz de Confusão

É um *layout* de tabela específica que permite a visualização do desempenho de um algoritmo, tipicamente em um aprendizado supervisionado (em aprendizagem não supervisionado é geralmente chamado de matriz de correspondência) (WIKIPEDIA, 2019).

Cada linha da matriz representa as instâncias em uma classe prevista, enquanto cada coluna representa as instâncias de uma classe real (ou vice-versa). O nome deriva do fato de que fica mais fácil ver se o sistema está confundindo duas classes (ou seja, comumente errando um nome como outro).

A mesma é um tipo especial de *Tabela de Contingência*, com duas dimensões ("real"e "previsto") e conjuntos idênticos de "classes"em ambas as dimensões (cada combinação de dimensão e classe é uma variável na tabela de contingência), vide Figura 3.4.

		True condition			
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	F_1 score = $\frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$

Figura 3.4 – Matriz de Confusão - Fonte :(TING, 2010)

Onde :
P = numero de casos positivos
N = numero de casos negativos
TP = verdadeiro positivo
TN = verdaeiro negativo
FP = falso positivo
FN = falso negativo

TPR = Taxa de Verdadeiro Positivo ou Sensibilidade

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR \tag{3.3}$$

TNR = Taxa de Verdadeiro Negativo, especificidade ou seletividade

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR \tag{3.4}$$

PPV = precisão ou valor preditivo positivo

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDRPPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = 1 - \mathrm{FDR} \tag{3.5}$$

NPV = valor preditivo negativo

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{FORNPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} = 1 - \mathrm{FOR} \tag{3.6}$$

FNR = taxa de perda (falta) ou taxa de falso negativo

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPRFNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR} \tag{3.7}$$

FPR = taxa falso positivo ou falha

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNRFPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR} \tag{3.8}$$

FDR = taxa de falsa descoberta

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPVFDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV} \tag{3.9}$$

FOR = taxa de omissão falsa

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPVFOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV} \tag{3.10}$$

ACC = precisão (acurácia)

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{3.11}$$

escore F1 é a média harmônica de precisão e sensibilidade

$$F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{3.12}$$

3.9.4 Curva ROC

A técnica curvas de *Receiver Operating Characteristic* (ROC) é uma representação gráfica que ilustra o desempenho (ou performance) de um sistema classificador binário e como o seu limiar de discriminação é variado.

A curva ROC foi desenvolvida pela primeira vez por engenheiros elétricos e engenheiros de radar durante a Segunda Guerra Mundial para detectar objetos inimigos em campos de batalha e logo foi apresentada à psicologia para explicar a detecção perceptual de estímulos. A análise ROC desde então tem sido usada em medicina, radiologia, biometria, previsão de perigos naturais, meteorologia, avaliação de desempenho de modelos e outras áreas, há muitas décadas, sendo cada vez mais usada em aprendizado de máquina e mineração de dados pesquisa.

A Figura 3.5 apresenta uma visão simplificada de uma curva ROC.

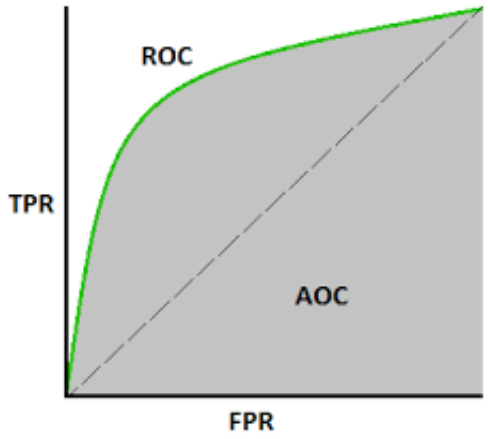


Figura 3.5 – Curva ROC

O traçado da curva é obtido pela representação das coordenadas da fração de *Positivos Verdadeiros* sobre os *Positivos Totais* (RPV=PV/P) versus a fracção de *Positivos Falsos* dos *Negativos Totais* (RPF=PF/N), em várias configurações do limite. O RPV é também conhecido como *sensibilidade*, e o RPF é um menos a especificidade ou o *Razão de Negativos Verdadeiros* RNV.

3.10 ETAPA DE IMPLEMENTAÇÃO DO ALGORITMO

Nesta etapa são desenvolvidas as tarefas necessárias para conversão dos códigos obtidos como resultado na etapa anterior para a plataforma de hardware e software do dispositivo inteligente.

Estes algorítimos devem ser implantados e embarcados em um dispositivo inteligente com uma arquitetura semelhante a proposta por MESTRE et al., conforme apresentado na Figura 3.6.

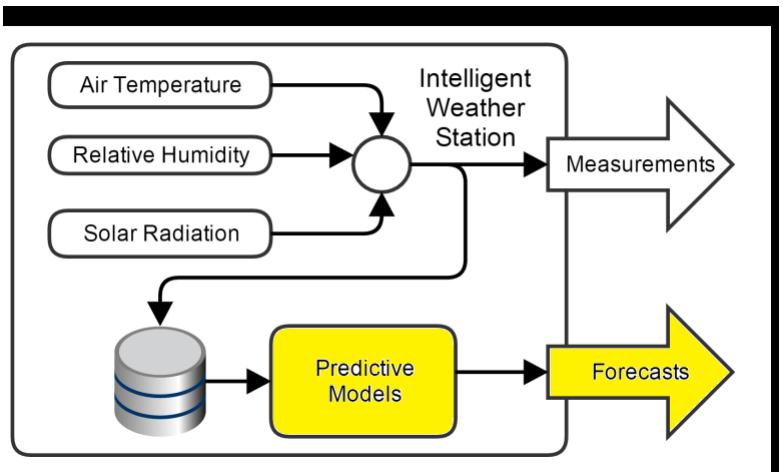


Figura 3.6 – Fluxo de Dados Simplificado em uma Estação Inteligente-fonte:(MESTRE et al., 2015)

As marcações em destaques, coloração em amarelo da Figura 3.6, representam as áreas do escopo deste trabalho. Refere-se, portanto, onde devem ser implementados os algorítimos propostos nesta pesquisa, para obter-se as previsões das precipitações (saída), a partir dos dados (entradas), capturados e disponibilizados localmente pela estação.

3.11 CONCLUSÕES DO CAPÍTULO

A metodologia apresentada é adequada para o desenvolvimento da pesquisa proposta. Pois, ela pode direcionar de forma mais planejada a execução de etapas, para melhor condução de cada umas das tarefas relacionadas ao trabalho. Embora a metologia siga as linhas propostas pela academia para trabalhos de *Machine Learning*, foram feitas modificações, inclusões e adequações para utilização em sistemas embarcados, de maneira que ficasses eficiente e apropriada para os objetivos a serem alcançados.

4 ESTUDO DE CASO-PREVISÃO DE PRECIPITAÇÃO EM BELÉM

Cada seção deste capítulo descreve a aplicação da metodologia proposta para o desenvolvimento da pesquisa, considerando um estudo de caso real. Assim, os resultados apresentados, bem como os dados utilizados refletem valores reais.

4.1 LOCAL DE ESTUDO

A região Amazônica possui uma área de extensão de 5.500.000 km^2, o que corresponde a cerca de 30 % da área da América do Sul. Isto faz que os impactos de clima e tempo desta região afetem também os variações de clima e tempo do planeta Terra. Apesar desta importância, a região amazônica possui uma baixa e má distribuição de estações meteorológicas, o que tem como consequência uma ausência de dados sistemáticos das estações, dada sua localização e das mesmas. Portanto, foi escolhida a cidade de Belém-Pa, tendo em conta a mesma possuir um grande volume de dados históricos, e de ser uma cidade de grande importância para região amazônica.

A cidade de Belém-PA (Figura 4.1) possui uma população de 1.393.399 habitantes (IBGE 2010), com uma área urbana da unidade territorial de 1.059,406 quilômetros quadrados. A mema está situada na região equatorial (latitude 1º 27' S, Longitude 48º 28' W e altitude de 16 m acima do nível médio do mar), ficando a 100 km do Oceâno Atlântico, sendo banhada pelo Rio Guamá ao sul, Baía do Guajará e rio Pará ao oeste, e ao leste limita com o município de Ananindeua-PA. De acordo com a classificação de Köppen (1948), o clima é do tipo tropical Am (quente e úmido), com temperatura média de 26,0 ºC, umidade relativa do ar variando entre 80 a 90 % e precipitação média anual de 2.783,8 mm variando entre 2.054,7 a 3.753,9 mm (BASTOS et al., 2002).

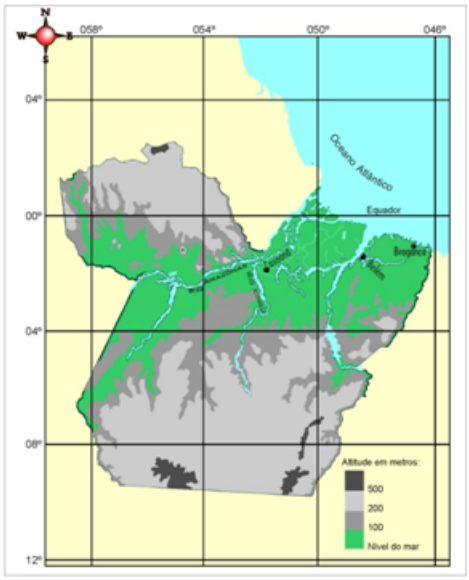


Figura 4.1 – Localização Geográfica de Belém-Pa

4.2 DADOS UTILIZADOS

Para este trabalho foram utilizados os dados da estação meteorológica automática, da cidade de Belém, do Instituto Nacional de Meteorologia-INMET, do período de 06 de novembro de 2017 a 01 de maio de 2019, disponibilizados, de forma on-line, através de seu sitio www.inmet.gov.br (INMET, 2019). Tal estação encontra-se na região da periferia de Belém, conforme apresentado na Figura 4.2. Também foram utilizados dados históricos de 118 anos (01 de janeiro de 1900 a 31 de dezembro de 2017), obtidos junto a Faculdade de Meteorologia da UFPa.

Figura 4.2 – Localização da Estação Meteorologia do INMET Belém

4.2.1 Aquisição de Dados

O INMET disponibiliza dados diários de suas estações automáticas e manuais distribuídas pelo Brasil. Para cada dia são disponibilizados os valores de cada variável que possuem uma identificação como apresentado na Tabela 4.1

Tabela 4.1 – Lay-out do arquivo das Estação Automatizada do INMET

Variavel	Descrição	Unidade
codigo_estacao	nome da estação obervada	
data	data do evento	
hora	hora GMT do evento	
temp_inst	valor da temperatura instantanea	^{o}C
temp_max	valor da temperatura maxima	^{o}C
temp_min	valor da temperatura minima	^{o}C
umid_inst	valor da umidade instantanea	%
umid_max	valor da umidade maxima	%
umid_min	valor da umidade minima	%
pto_orvalho_inst	valor da temperatura instantanea do ponto de orvalho	^{o}C
pto_orvalho_max	valor da temperatura maxima do ponto de orvalho	^{o}C
pto_orvalho_min	valor da temperatura minima do ponto de orvalho	^{o}C
pressao	valor da pressão instantanea	hPa
pressao_max	valor da pressão maxima	hPa
pressao_min	valor da pressão minima	hPa
vento_direcao	direção do vento	m/s
vento_vel	velocidade do vento	m/s
vento_rajada	rajada	m/s
radiacao	radiacao do evento	W
precipitacao	precipitacao do evento	mm

A partir dos dados disponibilizados pelo INMET, foram realizadas duas tarefas para seleção e classificação das variáveis, que serão utilizadas nas próximas etapas do processo de previsão (vide Capítulo 3).

4.3 ANÁLISE DOS DADOS

Para esta análise foram utilizadas as propostas da metologia: conhecimentos e experiências de meteorologistas, técnicas estatísticas, análise multivariada. Nas subsecções seguintes serão apresentados os processos realizados por cada tarefa (propostas pela metodologia), bem como seus resultados quando aplicáveis.

4.3.1 Dados Faltantes

Com os dados disponíveis, observou-se uma baixa taxa de ausência de dados (perda de 8,2%), conforme demonstra a Tabela 4.2. Após esta análise, e usando a proposta de HOENS; CHAWLA 2013, optou-se pelo descarte das ocorrências com dados faltantes, com a simples remoção das mesmas. As demais propostas, para o tratamento deste dados, foram descartadas, já que a utilização por exemplo da inclusão de dados utilizando técnicas de interpolação, usando a média de precipitação do período faltante, causariam uma distorção na caracterização dos fenômenos naturais.

Tabela 4.2 – Distribuição de Dados Tratados

Dados	**Qtd**	%
Tratado	11.930	91,8%
Bruto	12.994	100,0%
Perda	1.064	8,2%

4.3.2 Relevância de Variáveis

Para esta análise foram consideradas as técnicas propostas para SVM e a experiência dos especialistas. Assim foram realizadas as tarefas de: (a) análise multivariada e (b) analise de especilistas; conforme descritas a seguir.

4.3.2.1 Análise Multivariada

Para melhor seleção e classificação das variáveis foi utilizada a técnica de Analise de Componente Principal (PCA), para a análise multivariada. Tal técnica permite uma identificação da importância das variáveis baseada na relevância dos dados a serem utilizados.

Para isto, foi utilizado o aplicativo PAST((HAMMER et al., 2001)), disponível na internet sem custos. Assim, foram realizados ensaios com os dados, para todo o período disponível, com todas as variáveis disponibilizadas pelo INMET, e considerando para análise as metas de nossa previsão: *Classe de Precipitação.* Como resultado obteve-se a Tabela 4.3, para os componentes principais da análise PCA:

Tabela 4.3 – Distribuição de Carga PCA

PC	Eigenvalue	% variance
1	7,498	44,348
2	3,05325	18,059
3	2,60841	15,428
4	1,25254	7,4083
5	0,969192	5,7324
6	0,550484	3,2559
7	0,384806	2,276
8	0,19811	1,1717
9	0,160079	0,9468
10	0,100447	0,59411
11	0,0485048	0,28689
12	0,0417194	0,24675
13	0,013701	0,081036
14	0,0112709	0,066663
15	0,00833442	0,049295
16	0,00736027	0,043533
17	0,00110709	0,006548

Complementando esta análise, foi feito o gráfico de dispersão, para cada uma das variáveis apresentado na Figura 4.3.

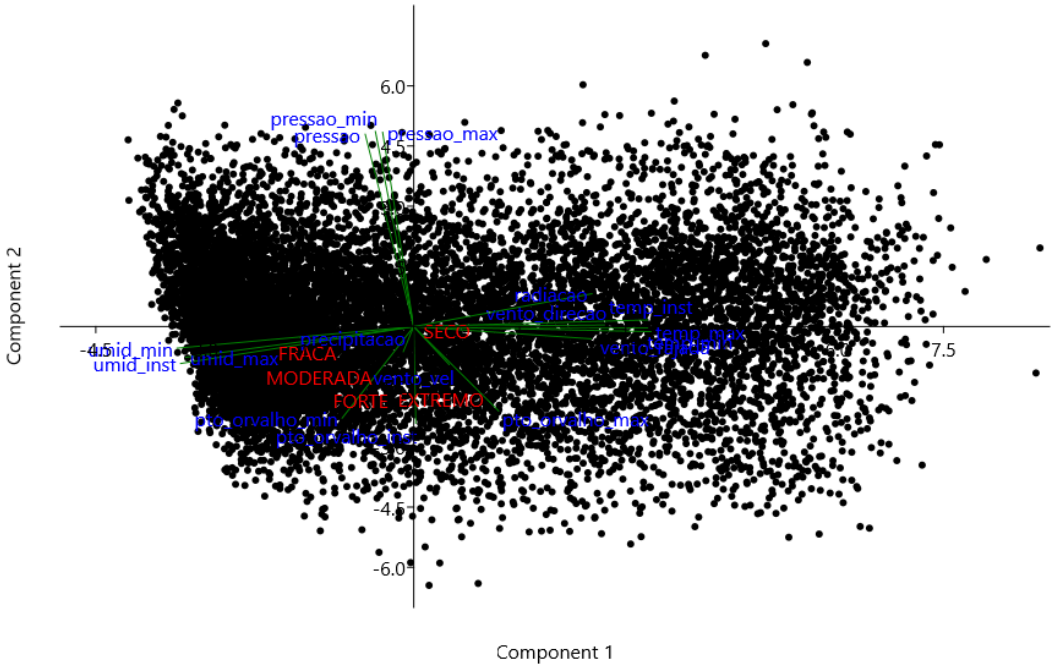


Figura 4.3 – Gráfico Dispersão PCA-Scatter Plot

Como pode se observado, as maiores cargas ficaram concentradas nos componentes PC1, PC2 e PC3 (77,84%). Entretanto, foram considerados somente os 2 maiores: neste caso PC1 e PC2, que representam um total de 62,41% do total.

Obteve-se assim a classificação de cada uma das variáveis mais relevantes. Para o *escore 1* (PC1) foram obtidos os resultados apresentados na Figura 4.4 e na Tabela 4.4, que apresenta o grau de importância de cada uma das variáveis, sob a ótica do *escore 1* (PC1). As variáveis de v*alores positivos* são consideradas como as mais relevantes.

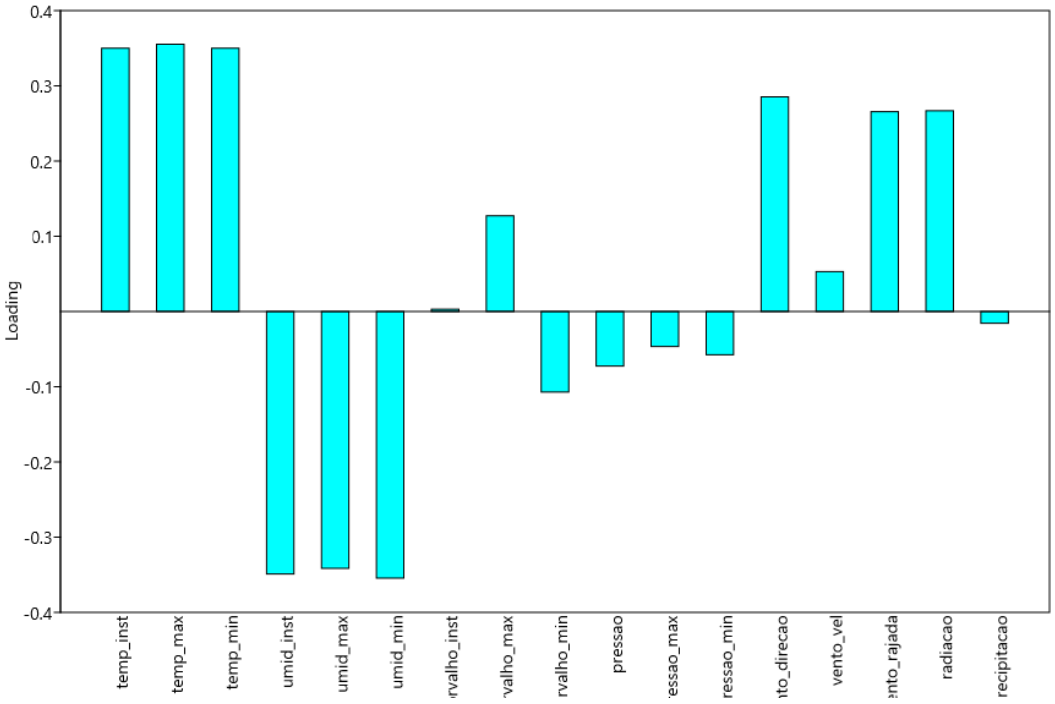


Figura 4.4 – Gráfico de Cargas - PC1

Tabela 4.4 – Classificação das Variaveis - PC1

Variavel	**PC 1**
temp_max	0,35528
temp_inst	0,35007
temp_min	0,35007
vento_direcao	0,28546
radiacao	0,26682
vento_rajada	0,26564
pto_orvalho_max	0,12717
vento_vel	0,05306
pto_orvalho_inst	0,00306
precipitacao	-0,01563
pressao_max	-0,04658
pressao_min	-0,05744
pressao	-0,07265
pto_orvalho_min	-0,10711
umid_max	-0,34140
umid_inst	-0,34900
umid_min	-0,35443

Para o *escore 2* (PC2) , a classificação das variáveis são apresentadas na Figura 4.5 e na Tabela 4.5. Semelhante ao escore 1 (PC1), a Figura 4.5 apresenta o grau de importância de cada uma das variáveis, sob a ótica do escore 2 (PC2). As variáveis de valores positivos, são consideradas as mais relevantes.

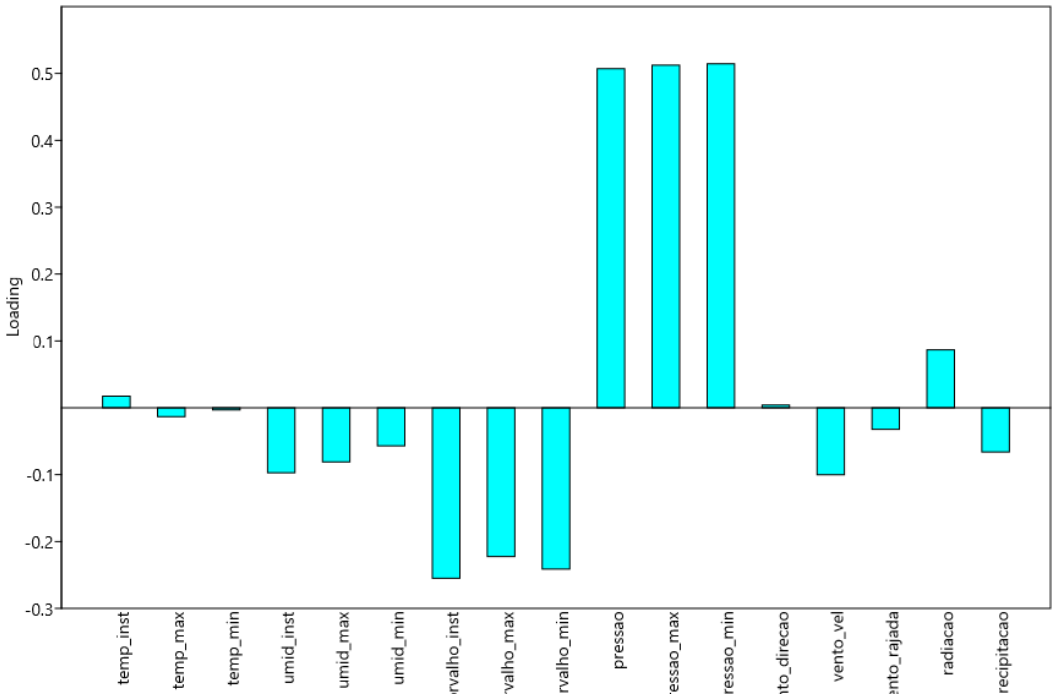


Figura 4.5 – Gráfico de Cargas - PC1

Tabela 4.5 – Classificação das Variaveis - PC2

Variavel	**PC 2**
pressao_min	0,51458
pressao_max	0,51237
pressao	0,50724
radiacao	0,08656
temp_inst	0,01745
vento_direcao	0,00418
temp_min	-0,00329
temp_max	-0,01327
vento_rajada	-0,03215
umid_min	-0,05696
precipitacao	-0,06602
umid_max	-0,08103
umid_inst	-0,09727
vento_vel	-0,10012
pto_orvalho_max	-0,22246
pto_orvalho_min	-0,24119
pto_orvalho_inst	-0,25502

4.3.3 Avaliação com Especialista

Como base nos princípios da meteorologia para formação de nuvens e precipitação, além da consulta com especialistas da área, foi gerada a Tabela 4.6, que classifica a influência de cada umas das variáveis, disponibilizadas pelo INMET, em relação a precipitação. Considerando na escala o valor de 1, para a variável de maior influência, e 5, para a de menor influência:

Tabela 4.6 – Tabela de Relevância

Variável	Relevância
precipitacao	1
umid_inst	2
umid_max	2
umid_min	2
pressao	2
pressao_max	2
pressao_min	2
temp_inst	3
temp_max	3
temp_min	3
pto_orvalho_inst	3
pto_orvalho_max	3
pto_orvalho_min	3
vento_direcao	3
vento_vel	3
vento_rajada	3
radiacao	3
data	4
hora	4
codigo_estacao	5

4.4 PARAMETRIZAÇÃO DAS CLASSES DE PRECIPITAÇÃO

Com base nas classes de precipitação, previamente definidas pela metodologia (Tabela 3.1), e com os resultados das análises de dados e dos experimentos foram definidos os parâmetros para cada uma das classes. Para períodos de 24 horas ficaram estabelecidas as seguintes regras de classificação com seus respectivos limiares, conforme descrito na Tabela 4.7

Tabela 4.7 – Parâmetros das Classes de Precipitação - Acumulado de 24 Horas

Classe	Descrição	Regra de classificação	Limiar
1	Sem Chuva	Sem Precipitação	0 mm/24 h
2	Chuva fraca	Precipitação cuja intensidade é menor do que	15,6 mm/24 h
3	Chuva moderada	Precipitação cuja intensidade está compreendida entre	15,6 mm/24 h e 32,5 mm/24 h
4	Chuva forte	Precipitação cuja intensidade está compreendida entre	32,5 mm/24 h e 50,4 mm/24 h
5	Chuva muito forte (Tempestade)	Precipitação cuja intensidade é maior do que	50,4 mm/24 h

4.4.1 Distribuição do Dados

Para cada um dos experimentos realizados foi feita uma análise da distribuição quantitativa dos dados em relação a cada classe de precipitação, conforme apresentado na Tabela 4.8 e Figura 4.6.

Tabela 4.8 – Distribuição por Classe de Precipitação

Classe	Count	Percent
1	2631	26.48%
2	4735	47.66%
3	1479	14.89%
4	686	6.91%
5	403	4.06%

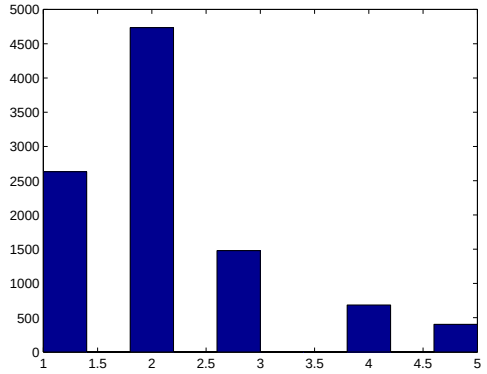


Figura 4.6 – Distribuição dos Dados por Classe de Precipitação

Isto se fez necessário para se obter uma distribuição de classes o mais realista e homogênea possível, e com isto se ter um melhoria dos dados trabalhados.

4.4.2 Sazonalidade do Dados

A sazonalidade da precipitação, foi obtida através da: (a) média mensal dos dados históricos de 118 anos de registros de precipitação em Belém, no período de 1900 a 2017, e (b) a média climatológica para os últimos 30 anos.(vide Figura 4.7)

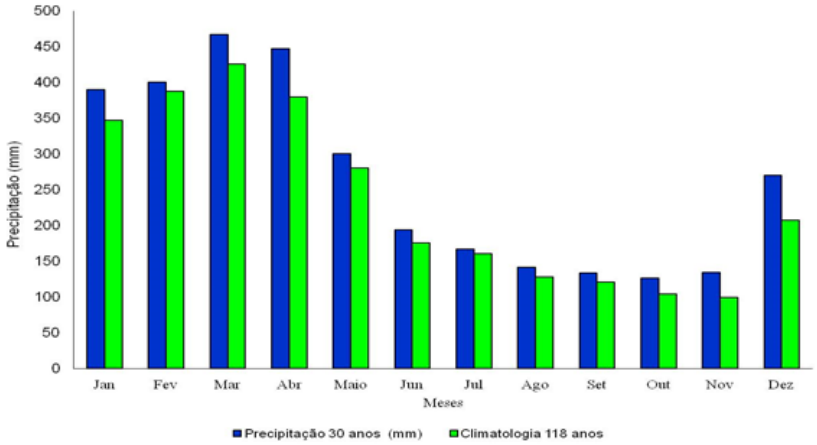


Figura 4.7 – Variação da PRP mensal Média Climatológica, período de 30 anos (barra azul) e Média Histórica de 118 anos (barra verde) em Belém-PA.Fonte: SANTOS, J.S. (SANTOS et al., 2014)

Como pode ser observado, existe um comportamento sazonal das precipitações em Belém, possuindo dois períodos bem definidos: (a) um chuvoso que vai de dezembro a maio, e (b) menos chuvoso de junho a novembro (Figueroa e Nobre, 1990; Marengo et al., 2001; e De Souza e Ambrizzi, 2003).

Isso ocorre, principalmente, devido à migração latitudinal da Zona de Convergência Intertropical (ZCIT), que durante o verão austral, está posicionada mais abaixo da linha do Equador, podendo alcançar até 5º S de latitude, provocando assim intensas chuvas nessa região (período chuvoso); enquanto no inverno austral, está mais ao norte, podendo alcançar até 10º N, e como consequência ocorre redução das chuvas na Amazônia (Cavalcanti et al., 2009). Isto sugeriu a inclusão dos dados do mês de ocorrência, como variável importante para previsão da classe de precipitação. Este comportamento influenciou diretamente para a escolha das funções *kernel*, utilizadas pelos algorimtos de SVM.

4.5 PREPARAÇÃO DOS DADOS

As tarefas desta etapa foram realizadas, quase que em sua totalidade, de forma automática. Foram desenvolvidos algorítimos em MATLAB para a maior automação e segurança das tarefas. Os *scripits* em MATLAb foram utilizados para a realização das tarefas de Tratamento e Separação de Dados, criando-se assim dois *scripts*, e que estão listados nos apêndices A e B . As tarefas de cada um destes *scripts* serão descritas a seguir:

4.5.1 Tratamento de Dados

Para esta ação, foi desenvolvido um script para realizar as seguintes tarefas :

- Ordenamento dos Dados Bruto
- Carga da Tabela de Relevância dos Dados
- Exclusão de Dados Incompletos
- Definição das Classes de Precipitação
- Aplicação da Janela de previsão
- Cálculo da Precipitação Acumulada para Janela
- Cálculo das Médias para cada Janela
- Aplicação Classes de Precipitação

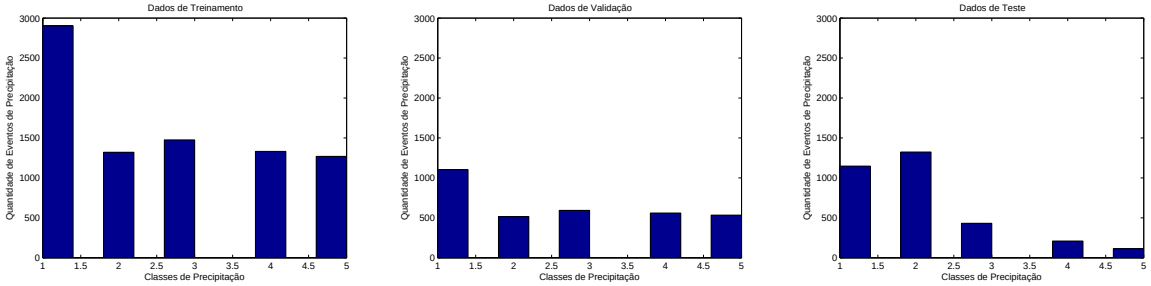
Assim com base nos dados disponibilizados pelo INMET no *layout* apresentado na Tabela 4.1, foram gerados novos dados, com o seguinte *layout*, conforme apresentado na Tabela 4.9:

Tabela 4.9 – Lay-out do arquivo de Saida - Etapa Tratamento de Dados

Variável	Descrição	Unidade
temp_inst	valor da temperatura instantânea	^{o}C
temp_max	valor da temperatura máxima	^{o}C
temp_min	valor da temperatura mínima	^{o}C
umid_inst	valor da umidade instantânea	%
umid_max	valor da umidade máxima	%
umid_min	valor da umidade mínima	%
pto_orvalho_inst	valor da temperatura instantânea do ponto de orvalho	^{o}C
pto_orvalho_max	valor da temperatura máxima do ponto de orvalho	^{o}C
pto_orvalho_min	valor da temperatura mínima do ponto de orvalho	^{o}C
pressao	valor da pressão instantânea	hPa
pressao_max	valor da pressão máxima	hPa
pressao_min	valor da pressão mínima	hPa
vento_direcao	direção do vento	^{o}
vento_vel	velocidade do vento	nós
vento_rajada	vento de rajada	nós
radiacao	radiação do evento	w
precipitacao	precipitação do evento	mm
precipitacao	precipitação acumulada das ultimas horas da janela de previsão	mm
mês	mês da ocorrêcia	-
temp_inst	média valor da temperatura instantânea	^{o}C
temp_max	média valor da temperatura máxima	^{o}C
temp_min	média valor da temperatura mínima	^{o}C
umid_inst	média valor da umidade instantânea	%
umid_max	média valor da umidade máxima	%
umid_min	média valor da umidade mínima	%
pto_orvalho_inst	média valor da temperatura instantânea do ponto de orvalho	^{o}C
pto_orvalho_max	média valor da temperatura máxima do ponto de orvalho	^{o}C
pto_orvalho_min	média valor da temperatura mínima do ponto de orvalho	^{o}C
pressao	média valor da pressão instantânea	hPa
pressao_max	média valor da pressão máxima	hPa
pressao_min	média valor da pressão mínima	hPa
vento_direcao	direção do vento	^{o}
vento_vel	velocidade do vento	nós
vento_rajada	vento de rajada	nós
radiacao	radiação do evento	w
precipitacao	precipitação do evento	mm

4.5.2 Geração de Dados

Utilizando os dados obtidos junto ao INMET (período 06/11/2017 a 01/05/2019) foram realizadas uma distribuição percentual dos dados, segundo as recomendações propostas pela literatura de IA, ou seja : 70 % dos dados para treinamento, os restantes (30 %) distribuídos entre dados de validação e testes. A escolha dos dados foram feitas de forma aleatória, para evitar comprometer o comportamentos dos algoritmos para as tarefas de classificação e previsão. As distribuições, por classes de precipitação de cada grupo de dados, são apresentadas na Figura 4.8.



(a) Dados de Treinamento (b) Dados de Validação (c) Dados de Testel

Figura 4.8 – Distribuição de Dados Gerados por Classe de Precipitação.

4.6 TREINAMENTO

Com a utilização do NIOTS (2.5), nesta etapa foram aplicados os dados para a geração das SVMs, o que envolveu a realização das tarefas de treinamento e validação. Visando uma busca por melhores resultados, foram utilizados diversas combinações de técnicas de SVMs.

4.6.1 Relatório do Treinamento

Os parâmetros e os resultados da etapa de treinamento realizados são apresentados através de um relatório (Figura 4.9) gerado pelo aplicativo NIOTS. Como pode se verificar, a primeira parte deste relatório (*General Parametrs*) apresenta os parâmetros fornecidos como entrada para o treinamento. Já a segunda parte (Sample 1) descreve os resultados deste treinamento, aferidos pelas métricas aplicadas durante este processo. Para cada experimento foram gerados 12 relatórios, totalizando 600 amostras .

```
General Parameters
Optimizer: MODE
Iterations:     150
Cross-set:      4
Samples:        1
Machine: Classification
MODE parameters

Population:             20
Scale factor:  0.7000
Crossover rate:          0.4000
Lower limit:    -10.0000
Upper limit:    10.0000
Diversity factor: classic
Kernel: RBF

Sample 1
Index          C           Gamma          Error             SV
  1      5.79784654     -1.51286354     0.11580727   3611.00000000
  2      9.78860678     -4.23816950     0.14497041   3096.00000000
  3      5.53899083     -1.87134169     0.12426036   3441.00000000
  4      8.77111883     -2.81984690     0.12468301   3273.00000000
  5      7.09029644     -2.97793447     0.12890955   3218.00000000
  6      9.50955429     -3.90975345     0.13947591   3136.00000000
  7      7.77599820     -3.33812671     0.13567202   3182.00000000
  8      8.38531633     -3.33812671     0.13398140   3195.00000000
  9      4.32486447     -1.74772530     0.12130178   3498.00000000
 10      8.32775726     -1.52802971     0.11792054   3581.00000000
 11      9.85165979     -3.91080993     0.14074387   3117.00000000
 12      9.48833957     -2.85803101     0.12764159   3243.00000000
 13      9.66419158     -1.59779597     0.11918850   3577.00000000
 14      8.44422672     -1.81749156     0.12256974   3462.00000000
 15      6.29621766     -1.69144537     0.11961116   3531.00000000
 16      9.41056451     -4.23816950     0.14285714   3111.00000000
 17      8.87873922     -2.83420865     0.12595097   3249.00000000
 18      9.73806814     -4.23816950     0.14370245   3107.00000000
 19      5.41123522     -1.81749156     0.12341505   3458.00000000
 20      5.35730697     -1.72390295     0.12045647   3522.00000000
 21      7.74653041     -3.33812671     0.13482671   3189.00000000
 22      9.29927869     -2.80926193     0.12552832   3267.00000000
 23      6.88332413     -1.70954120     0.12003381   3524.00000000
 24      9.01414198     -2.81451881     0.12510566   3270.00000000
```

Figura 4.9 – Relatório NIOTS-Treinamento

4.7 PREVISÃO

Para a tarefa de previsão, foram utilizados os melhores resultados apresentados em cada experimento da etapa de treinamento, considerando o menor erro (Coluna Error). Também foi utilizado o aplicativo NIOTS (2.5), conforme apresentados na Figura 4.10 :

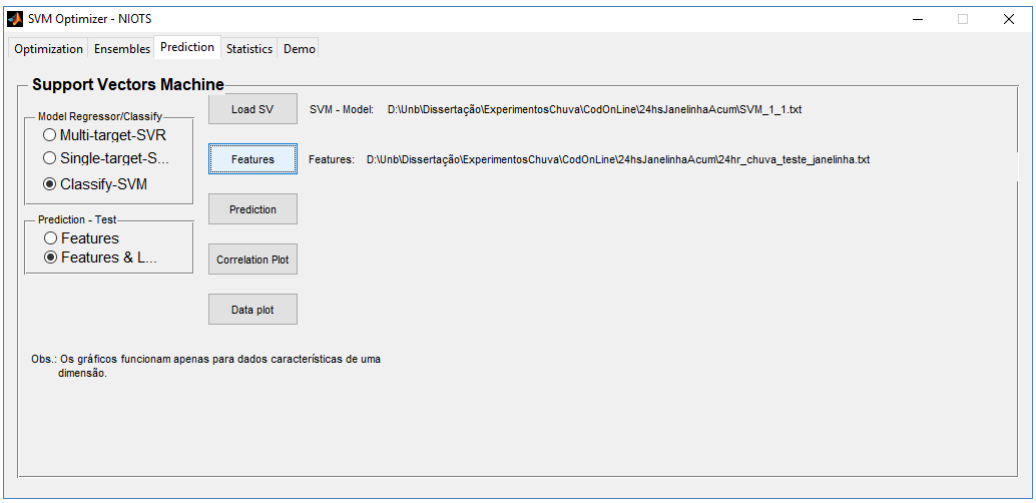


Figura 4.10 – Parâmetros para previsão

4.8 RESULTADOS DA PREVISÃO

Os resultados, da etapa de previsão realizadas pelo algorítimo SVM, com uso do NIOTS, são apresentados, a seguir, através dos gráficos de histograma, matriz de confusão e curva ROC.

4.8.1 Previsão com Algorítimo MODE

4.8.1.1 Histograma-24 horas

A Figura 4.11 e a Tabela 4.10 apresentam o histograma os respectivos valores de erro relativo, para previsão de 24 horas utilizando o algoritmo MODE, como otimizador, e as funções *Kernel* RBF e Polinomial, para o algorítimo SVM. Os gráficos de histograma apresentam os resultados da previsão, comparando os dados reais, das diversas classes de precipitações, com os previstos pelos algorítimos de IA. Este gráfico apresenta os valores quantitativos para cada uma das classes de precipitação, o eixo X contém as classes de precipitação, enquanto o eixo Y, corresponde a quantidade de eventos para cada classe de precipitação. As barras em azul, representam as classes reais, enquanto que as barras em verde são para a função *Kernel* RBF e as vermelhas para a função *Kernel* Polinomial, representando os valores previstos. As tabelas apresentam os valores de erro relativo, para cada classe de precipitação, calculados com base nos valores reais das precipitações.

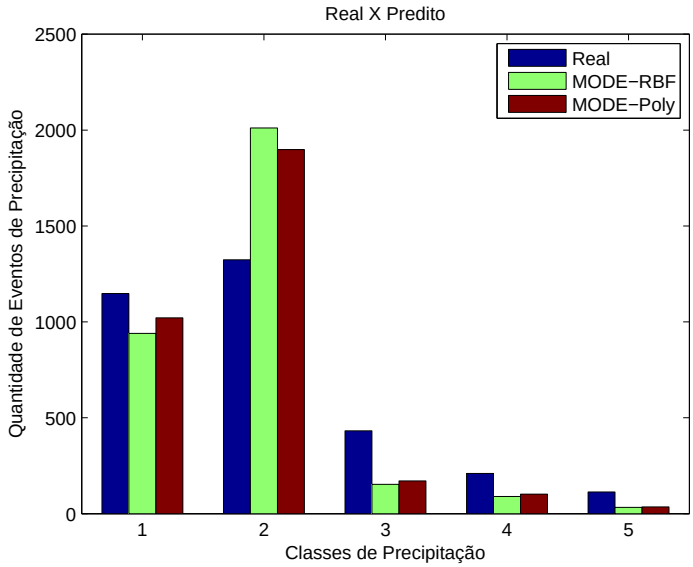


Figura 4.11 – Histograma de Previsão 24 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Tabela 4.10 – Erro Relativo de Previsão 24 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Classe	**Kernel**	**Erro**
1	Rbf	18,12%
1	Poly	11,06%
2	Rbf	51,89%
2	Poly	43,35%
3	Rbf	64,58%
3	Poly	60,42%
4	Rbf	57,14%
4	Poly	51,43%
5	Rbf	70,80%
5	Poly	69,03%

Comparando as previsões feitas com os dois *kernels*, percebe-se que não existem grandes diferenças entre eles. Porém, a classe 2 apresentou um quantitativo previsto maior que o real, diferente das demais classes em os quantitativos previstos são menores que os reais, apresentando um erro médio de aproximadamente 45 %.

4.8.1.2 Histograma-48 horas

Para previsão de 48 horas, os dados foram redistribuídos, gerando um novo quantitativo de ocorrências para cada classe de previsão. A Figura 4.12 e a Tabela 4.11 apresentam o histograma e os respectivos valores de erro relativo para previsão de 48 horas utilizando o algoritmo MODE, como otimizador, e as funções *Kernel* RBF e Polinomial, para o algorítimo SVM. Pode

ser observado que a previsão com a função *kernel Polinomial*, apresenta erros significativos, com diferenças expressivas, entre os quantitativos previsto em comparação aos valores reais, superestimando os valores na classe 1 e subestimando nas demais classes. Entretanto, a função *kernel* RBF, apresentou um quantitativo de acertos maiores, onde apenas a classe 3 teve seu valor subestimado, com diferença significativa,destaque para classe 2, cujo acerto foi aproximadamente de 85%. A outras classes (4,5 e 1) apresentaram uma diferença média de aproximadamente 32 %, entre o previsto e o real.

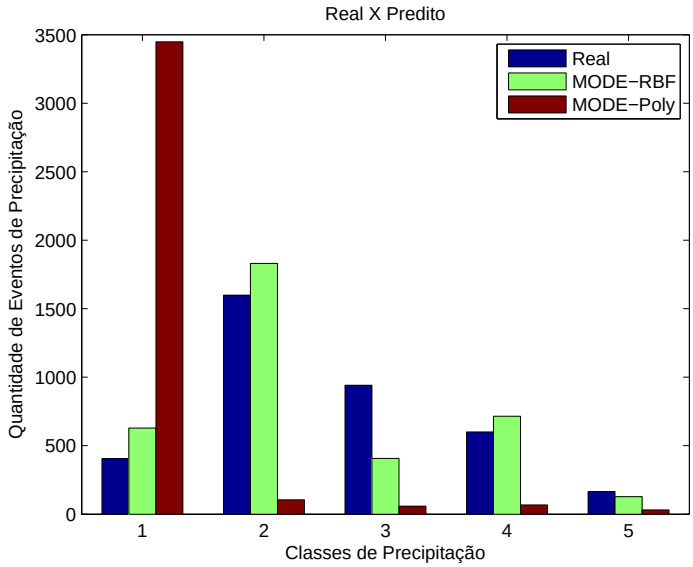


Figura 4.12 – Histograma de Previsão 48 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Tabela 4.11 – Erro Relativo - Previsão 48 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Classe	**Kernel**	**Erro**
1	Rbf	55,31%
1	Poly	751,36%
2	Rbf	14,51%
2	Poly	93,43%
3	Rbf	56,75%
3	Poly	93,84%
4	Rbf	19,17%
4	Poly	88,67%
5	Rbf	22,42%
5	Poly	81,21%

4.8.1.3 Histograma-72 horas

Novamente, os dados foram redistribuídos, gerando um novo quantitativo de ocorrências para cada classe de previsão para 72 horas. A Figura 4.13 e a Tabela 4.12 apresentam os histogramas

e os respectivos valores de erro relativo para previsão de 72 horas utilizando o algoritmo MODE, como otimizador, e as funções *kernel* RBF e Polinomial, para o algorítimo SVM. Pode ser observado que a previsão com a função *kernel* Polinomial, volta a apresentar erros significativos, como ocorreu com a previsão de 48 horas, sendo que neste caso, superestimou as classes 1 e 5, e subestimou as classes 2, 3 e 4. A função *kernel* RBF, apresenta os melhores resultados com poucos erros maiores que 70%.

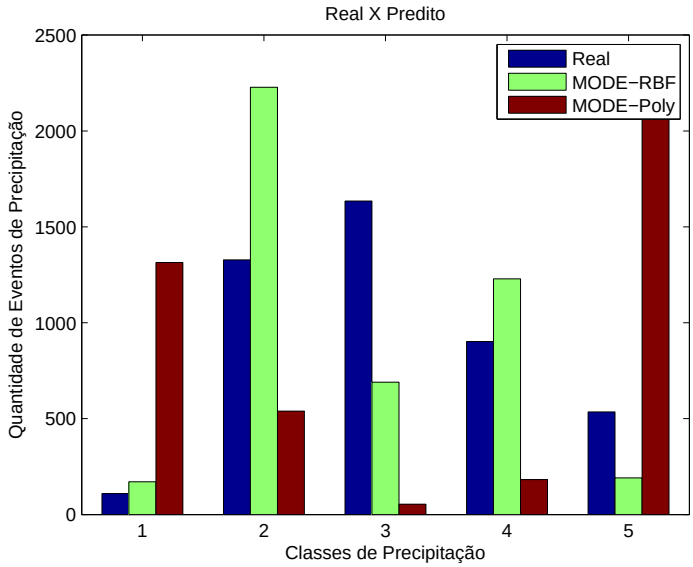


Figura 4.13 – Histograma de Previsão 72 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Tabela 4.12 – Erro Relativo - Previsão 72 hs utilizando o algorítimo MODE-*Kernel* RBF e Polinomial

Classe	**Kernel**	**Erro**
1	Rbf	56,88%
1	Poly	1105,50%
2	Rbf	67,77%
2	Poly	59,41%
3	Rbf	57,80%
3	Poly	96,70%
4	Rbf	36,25%
4	Poly	79,82%
5	Rbf	64,30%
5	Poly	352,34%

4.8.1.4 Matriz de Confusão-24 horas

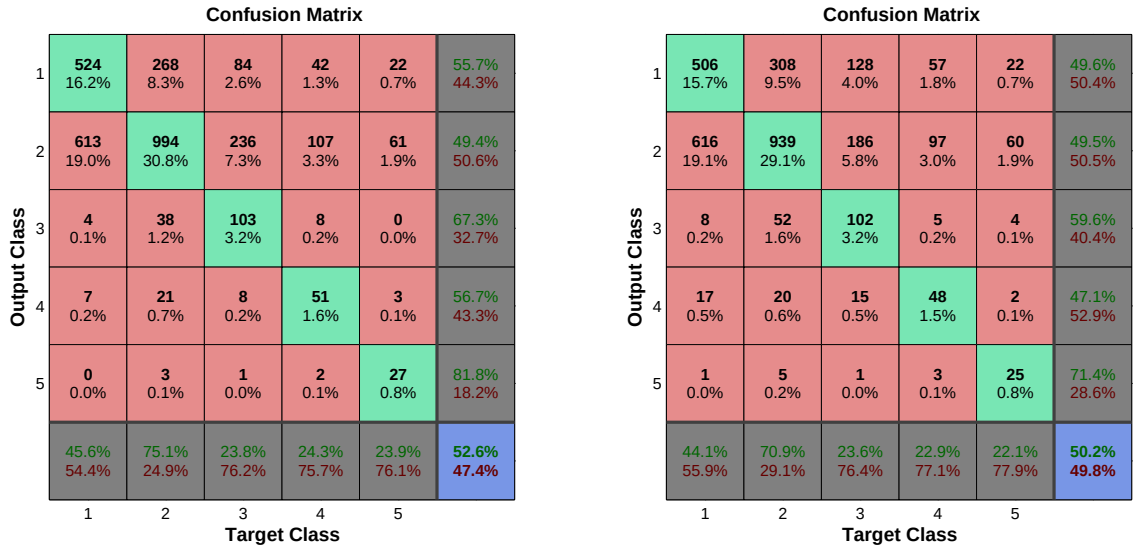
Foram utilizadas as técnicas de *matriz de confusão* conforme sugerido pela metodologia. As matrizes de confusão a seguir apresentadas, demostram os quantitativos e percentuais de acertos e erros da cada classe de precipitação, comparada com os dados reais, para o algorítimo

MOPSO-*Kernel* RBF. Esta forma de representação permite uma melhor avaliação qualitativa dos resultados.

No gráfico da matriz de confusão, as linhas correspondem à classe de saída da classe prevista e as colunas correspondem à classe verdadeira (classe de destino). As células diagonais correspondem a observações que são classificadas corretamente. As células fora da diagonal correspondem a observações classificadas incorretamente. Tanto o número de observações quanto a porcentagem do número total de observações são mostrados em cada célula.

A coluna na extremidade direita da plotagem mostra as porcentagens de todos os exemplos previstos pertencentes a cada classe que são classificados correta e incorretamente. Essas métricas geralmente são chamadas de precisão (ou valor previstivo positivo) e taxa de descoberta falsa, respectivamente. A linha na parte inferior do gráfico mostra as porcentagens de todos os exemplos pertencentes a cada classe que são classificados correta e incorretamente. Essas métricas geralmente são chamadas de *taxa recall* (ou taxa positiva verdadeira) e *taxa de falso negativo*, respectivamente. A célula no canto inferior direito do gráfico mostra a precisão geral.

A Figura 4.14 mostra a matriz de confusão para o algoritmo MODE usando o *Kernel* RBF e Polinomial. As Tabelas apresentam os rankings para as métricas de acurácia, recall e precisão extraídos das matrizes de confusão. Neste resultado, observa-se que as classes tiveram de uma maneira geral um comportamento semelhante, em termos de acurácia, variando em aproximadamente 22 % (classe 5) e 76 % (classe 2), para previsão de ocorrência de chuva. Significando uma acurácia maior que 50 %.



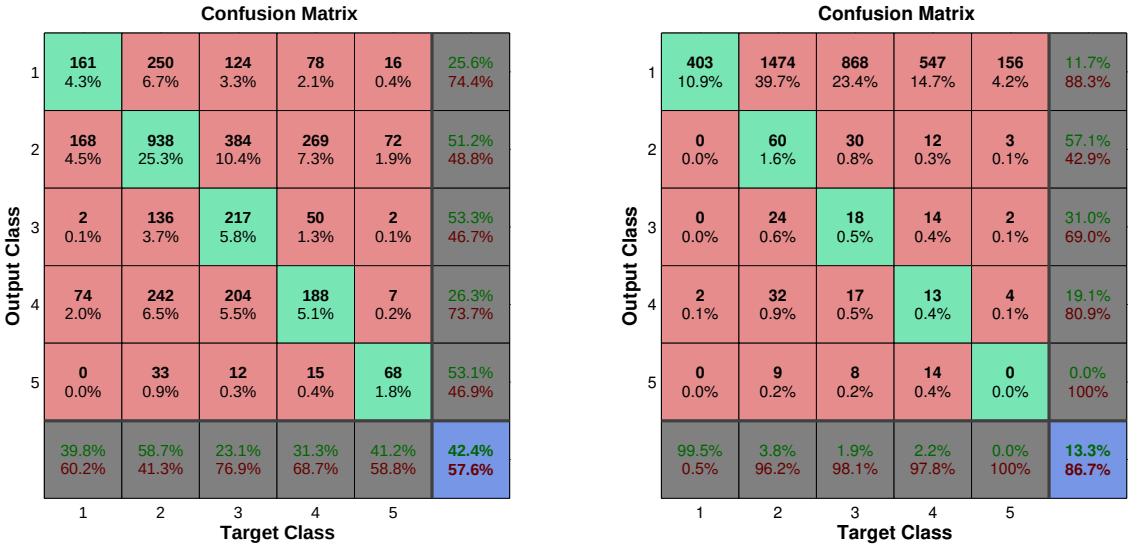
(a) Kernel RBF

(b) Kernel Polinomial

Figura 4.14 – Matriz de Confusão Algorítimo MODE-24 horas.

4.8.1.5 Matriz de Confusão-48 horas

A matriz de confusão para o algoritmo MODE usando o *Kernel* RBF e Polinomial é apresentada na Figura 4.15. Para este caso se observa um comportamento diferenciado entre as funções *kernel*, com acurácia de 42,4% para função RBF e 13,3% para função Polinomial. A função *kernel* Polinomial, apresenta recall menor 4% para as classes 2,3,4 e 5, e 99,5% para classe 1, motivado pela concentração das previsões nesta classe, como já apresentado na Figura 4.12. A função *Kernel* RBF, variou entre 23 % e 59%, como acurácia de 42,4%.



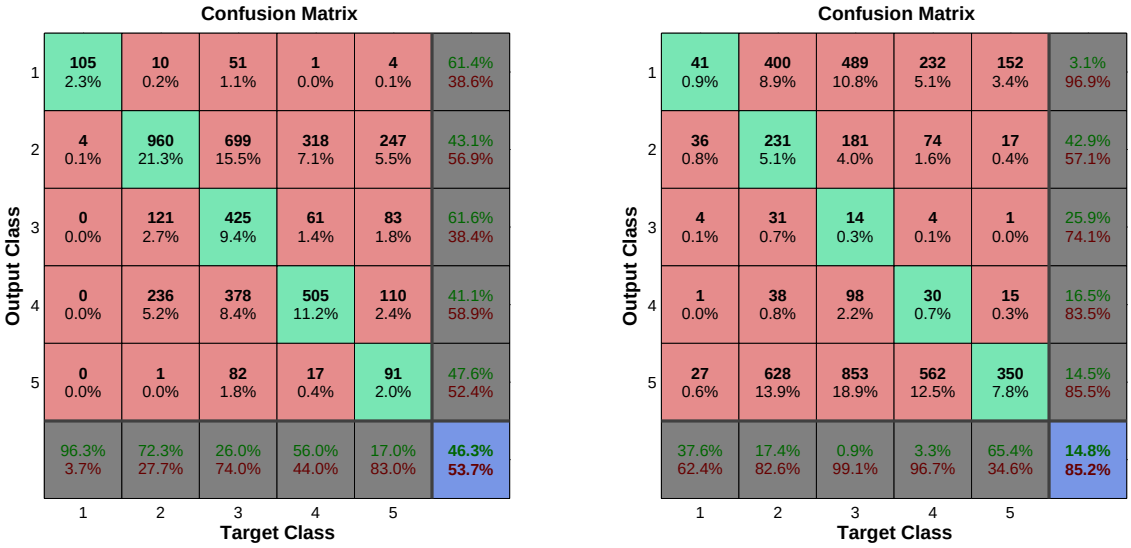
(a) Kernel RBF

(b) Kernel Polinomial

Figura 4.15 – Matriz de Confusão Algorítimo MODE-48 horas.

4.8.1.6 Matriz de Confusão-72 horas

Para 72 horas, a matriz de confusão do algoritmo MODE (usando o *Kernel* RBF e Polinomial) é apresentada na Figura 4.16. Neste caso, o *Kernel* RBF mostrou melhor acurácia (46,3%) do que o *kernel* Polinomial (14,8%). O *Kernel* RBF apresentou ganhos significativos para as classes 1 (96,3%), 2(72,3%) e 4(56%). A função Polinomial apresentou baixos valores de *recall*, principalmente para classes 1, 2, 3 e 4, variando de 0,9% à 37 %, apenas com acerto significativo para classe 5(65,4%).

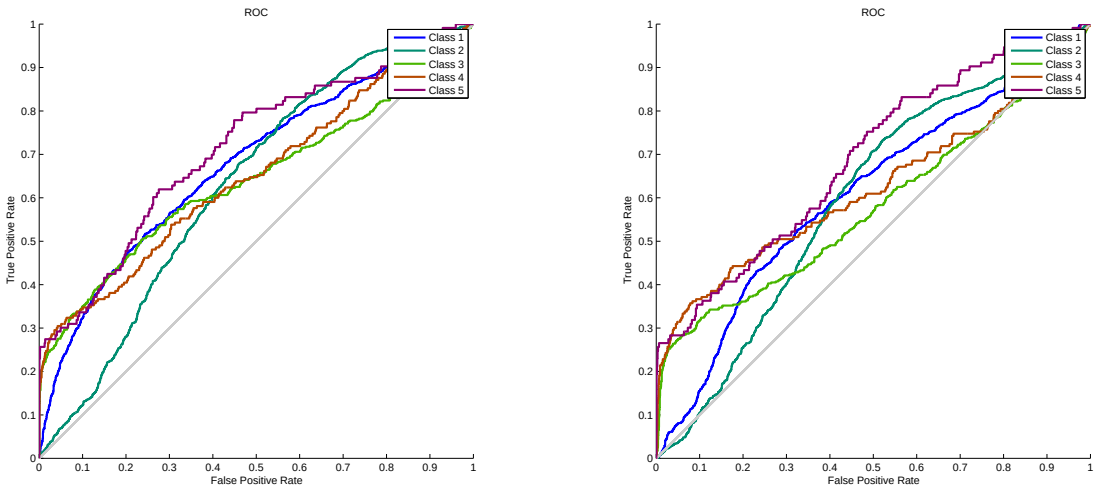


(a) Kernel RBF (b) Kernel Polinomial

Figura 4.16 – Matriz de Confusão Algorítimo MODE-72 horas.

4.8.1.7 Curvas ROC-24 horas

Foi utilizado a técnica de curvas ROC, para produzir graficamente o comportamento dos algorítimos em cada uma das classes de previsão pesquisada. A Figura 4.17 e a Tabela 4.13, apresentam, respectivamente as curvas ROC e os AUC, para previsão de cada uma das classes, utilizando o algoritmo MODE, com *Kernel* RBF e *Kernel* Polinomial.



(a) Kernel RBF (b) Kernel Polinomial

Figura 4.17 – Curvas ROC Algorítimo MODE-Previsão 24 hs.

Tabela 4.13 – AUC das Curvas ROC Algorítimo MODE-Previsão 24 hs.

(a) Kernel RBF

Classe	AUC
1	0,6759
2	0,6316
3	0,6455
4	0,6514
5	0,7134

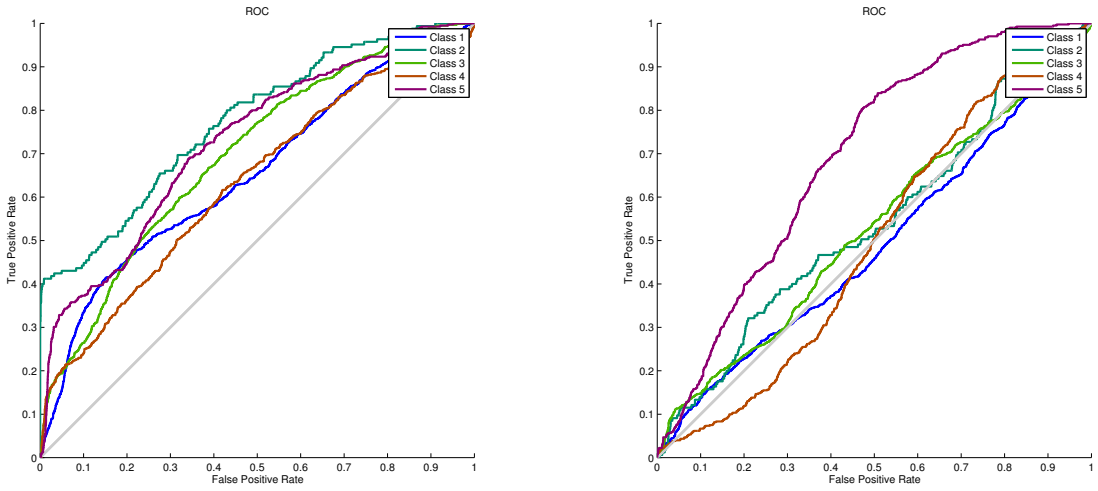
(b) Kernel Polinomial

Classe	AUC
1	0,607
2	0,5963
3	0,5775
4	0,6172
5	0,691

Se observa que as ocorrências, para todas as classes, se encontram na porção verdadeiro-positivo do gráfico. Pode ser visto também, que as classes 5 e 1 apresentam uma melhor performance para o *Kernel* RBF, e somente a classe 5 para o *Kernel* Polinomial. Somente a classe 3, para o *Kernel* Polinomial, tem valores menores que as demais classes. Desta maneira as funções Kernel, RBF e Polinomial, não afetaram significativamente os resultados de performance para a previsão 24 horas.

4.8.1.8 Curvas ROC-48 horas

Nas Figura 4.18 e Tabela 4.14 são mostradas as curvas e o AUC para previsão de 48 horas, para cada uma das classes de precipitação, utilizando o algoritmo MODE, com *Kernel* RBF e *Kernel* Polinomial. Para *Kernel* RBF todas as classes previstas se encontram na porção verdadeira-positivo.Destaque para a classe 2, chuva fraca com até 15,6 mm/24 h, seguida da classe 5 (evento extremo, chuva > 51,4 mm/24 h). O *Kernel* Polinomial teve comportamento diferente com duas classes (1-sem precipitação e 4-precipitação entre 32,5 mm/24 h e 50,4 mm/24 h), estando na porção inferior a linha diagonal do gráfico, e, somente, a classe 5 (evento extremo) apresentou melhor performance. Portanto o *Kernel* Polinomial não é recomendado para esta previsão.



(a) Kernel RBF

(b) Kernel Polinomial

Figura 4.18 – Curvas ROC Algorítimo MODE-Previsão 48 hs.

Tabela 4.14 – AUC das Curvas ROC Algorítimo MODE-Previsão 48 hs.

(a) Kernel RBF

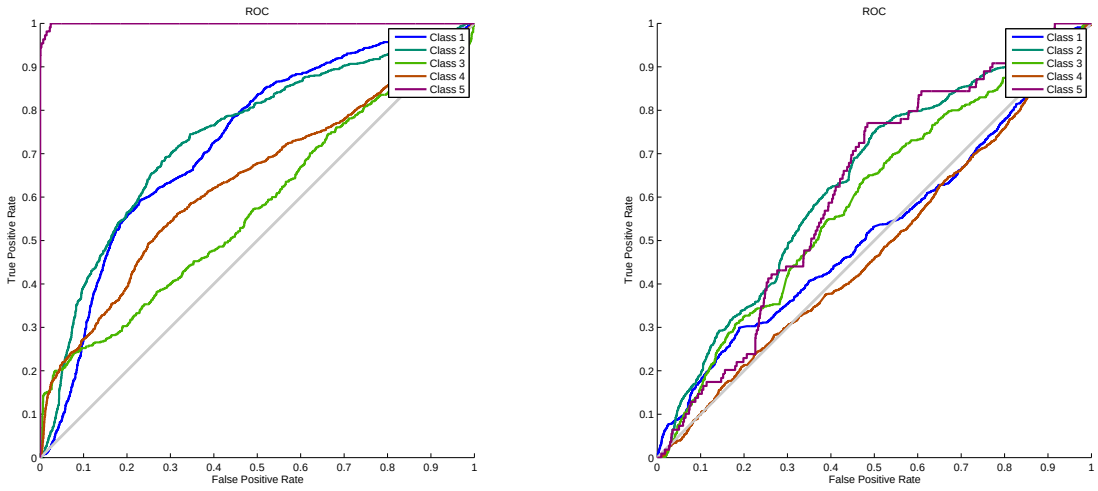
Classe	**AUC**
1	0,6529
2	0,4027
3	0,3706
4	0,4136
5	0,7288

(b) Kernel Polinomial

Classe	**AUC**
1	0,4913
2	0,4936
3	0,3998
4	0,3428
5	0,6938

4.8.1.9 Curvas ROC-72 horas

Para previsão de 72 horas o *Kernel* RBF apresenta boa performance com todas as classes na porção verdadeiro-positivo, destacando a classe 5 (evento extremo) com resultados próximos a 100% (Figura 4.19). Entretanto a previsão com *Kernel* Polinomial tem apenas três classes (2,3 e 5) na porção verdadeiro-positivo, com resultados menores que 65%, enquanto que as classes 1 e 4 estão na porção Falso-Positivo. Por conseguinte o kernel RBF se apresenta adequado para previsão de 72 horas.



(a) Kernel RBF (b) Kernel Polinomial

Figura 4.19 – Curvas ROC Algorítimo MODE-Previsão 72 hs.

Tabela 4.15 – AUC das Curvas ROC Algorítimo MODE-Previsão 72 hs.

(a) Kernel RBF

Classe	AUC
1	0,7257
2	0,1956
3	0,4668
4	0,3217
5	0,9992

(b) Kernel Polinomial

Classe	AUC
1	0,5219
2	0,3910
3	0,3757
4	0,5920
5	0,6235

4.8.2 Previsão com Algorítimo MOPSO

Semelhante ao apresentado para o Algorítimo MODE, são descritos a seguir os resultados obtidos com a utilização do algorítimo MOPSO, como otimizador, usando os *Kernel* RBF e Polinomial.

4.8.2.1 Histograma-24 horas

Comparando as previsões, para 24 horas, feitas com os dois *Kernel*, é possível notar pela Figura 4.20 e a Tabela 4.16, que o *Kernel* Polinomial tem valores muitos diferentes dos valores reais, onde a classe 2 tem quantitativos bem superiores aos reais (aproximadamente 80%). O *Kernel* RBF apresentou quantitativos mais próximos aos reais, com erro médio de aproximadamente 54,4%, mostrando assim uma melhor previsão do que o *Kernel* Polinomial.

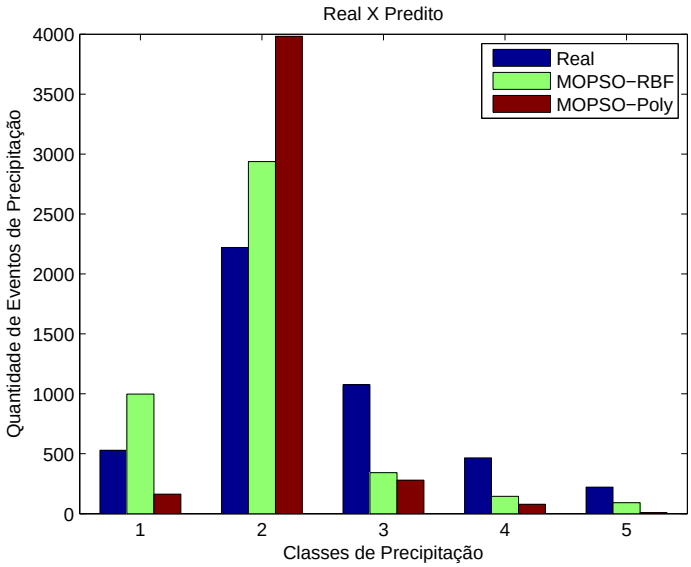


Figura 4.20 – Histograma de Previsão 24 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

Tabela 4.16 – Erro Relativo - Previsão 24 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

Classe	**Kernel**	**Erro**
1	Rbf	88,66%
1	Poly	69,19%
2	Rbf	32,24%
2	Poly	79,33%
3	Rbf	68,22%
3	Poly	73,98%
4	Rbf	68,82%
4	Poly	83,01%
5	Rbf	59,01%
5	Poly	96,40%

4.8.2.2 Histogramas-48 horas

As Figura 4.21 e a Tabela 4.17 apresentam, respectivamentes, o histograma e os valores de erro relativos para previsão de 48 horas, utilizando o algoritmo MOPSO como otimizador, para os *Kernel* RBF e Polinomial. Para ambos os *Kernel*, as classes 2,4 e 5 apresentaram melhores resultados, chegando a aproximadamente 100% de acerto nos quantitativos. Somente as classes 1 e 3 tem diferenças superiores a 50%, entre os quantitativos de precipitações dos valores reais e os previstos.

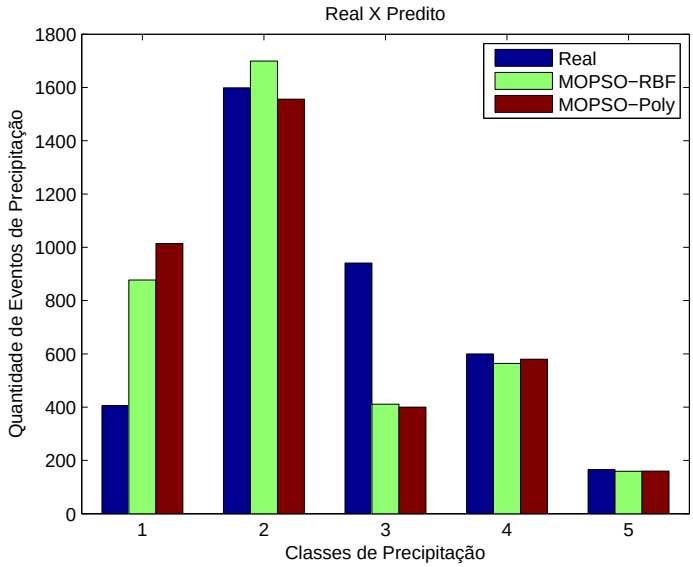


Figura 4.21 – Histograma de Previsão 48 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

Tabela 4.17 – Erro Relativo - Previsão 48 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

Classe	**Kernel**	**Erro**
1	Rbf	116,54%
1	Poly	150,37%
2	Rbf	6,25%
2	Poly	2,69%
3	Rbf	56,32%
3	Poly	57,49%
4	Rbf	6,00%
4	Poly	3,33%
5	Rbf	3,64%
5	Poly	3,03%

4.8.2.3 Histogramas-72 horas

Verificando a previsão de 72 horas utilizando o algorítimo MOPSO, como otimizador, e as funções *Kernel* RBF e Polinomial (Figura 4.22 e Tabela 4.18). Novamente pode ser verificado que o *Kernel* RBF apresenta melhores resultados quantitativos, com erro médio de aproximadamente 44%, diferente dos resultados do *Kernel* Polinomial, que tem erros superiores, o que demonstra uma melhor eficácia do *Kernel* RBF.

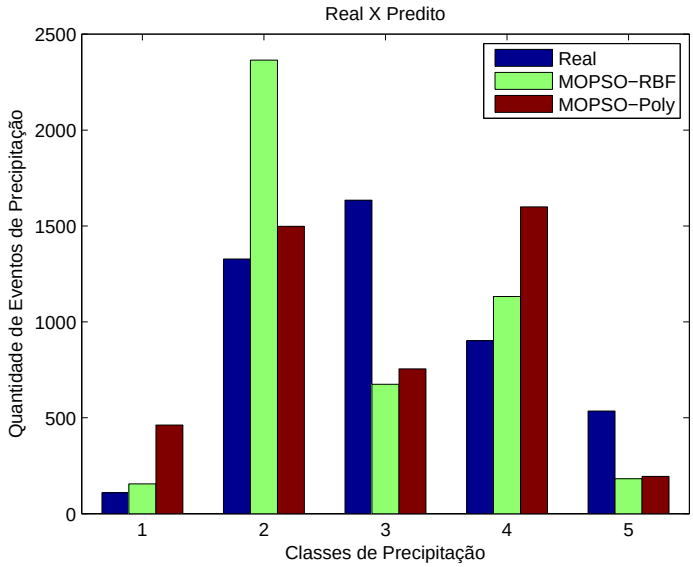


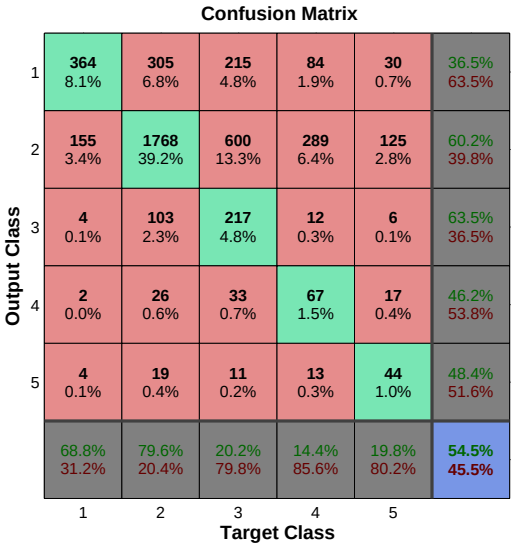
Figura 4.22 – Histograma de Previsão 72 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

Tabela 4.18 – Erro Relativo - Previsão 72 hs utilizando o algorítimo MOPSO-*Kernel* RBF e Polinomial

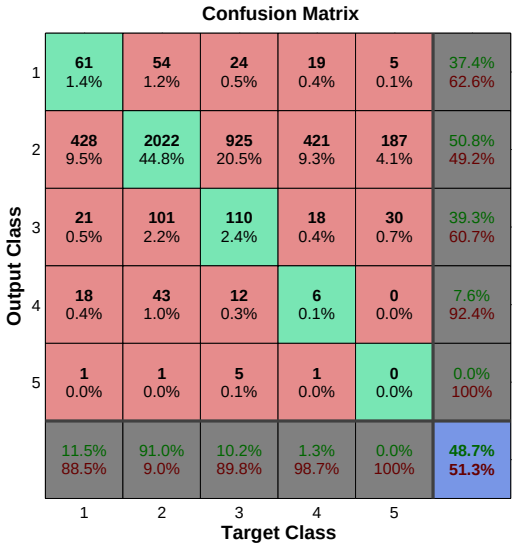
Classe	**Kernel**	**Erro**
1	Rbf	42,20%
1	Poly	323,85%
2	Rbf	78,09%
2	Poly	12,80%
3	Rbf	58,72%
3	Poly	53,82%
4	Rbf	25,50%
4	Poly	77,38%
5	Rbf	65,98%
5	Poly	63,74%

4.8.2.4 Matriz de Confusão-24 horas

A Figura 4.23 mostra a matriz de confusão para o algorítimo MOPSO usando os *Kernel* RBF e Polinomial. Neste resultado observa-se que as classes tiveram, de uma maneira geral, um comportamento semelhante em termos de acurácia, 54,5% para *Kernel* RBF e 48,7% para o *Kernel* Polinomial. As classes 1 e 2, apresentaram recall de 68,8% e 79,6% respectivamente, para o *Kernel* RBF enquanto que para o *Kernel* Polinomial a classe 2 apresentou recall de 91,0%. A precisão foi de 60,2% para classe 2 e 63,5% para classe 3, utilizando o *Kernel* RBF, ao passo que o *Kernel* polinomial teve a maior precisão para classe 2 (50,8%)



(a) Kernel RBF

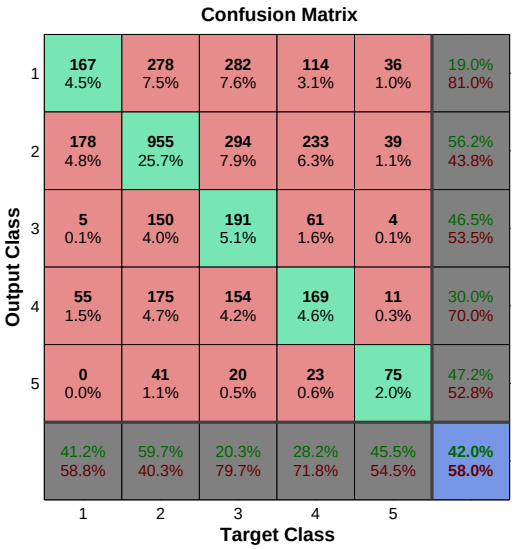


(b) Kernel Polinomial

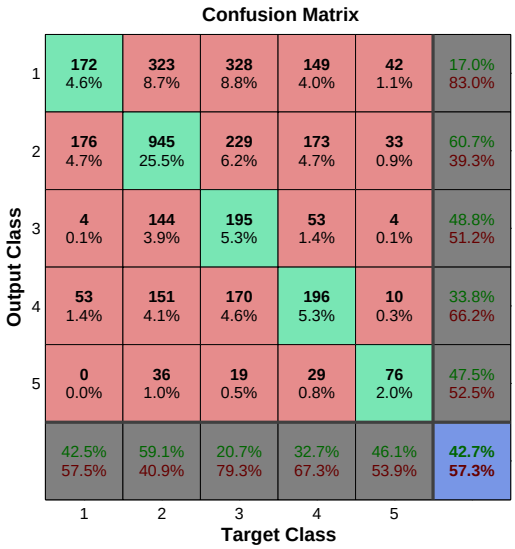
Figura 4.23 – Matriz de Confusão Algorítimo MOPSO-24 horas.

4.8.2.5 Matriz de Confusão-48 horas

Observa-se, nas matrizes de confusão (Figura 4.24), que não existem grandes diferenças entre os valores de acurácia para ambos os *Kernel* RBF (42,0%) e Polinomial (42,7%). Apresentando comportamento semelhante, para todas as classes, dos valores de recall e precisão. Destaque para a classe 2, que obtiveram 59,1% e 60%, para recall e precisão respectivamente para o *Polinomial*.



(a) Kernel RBF

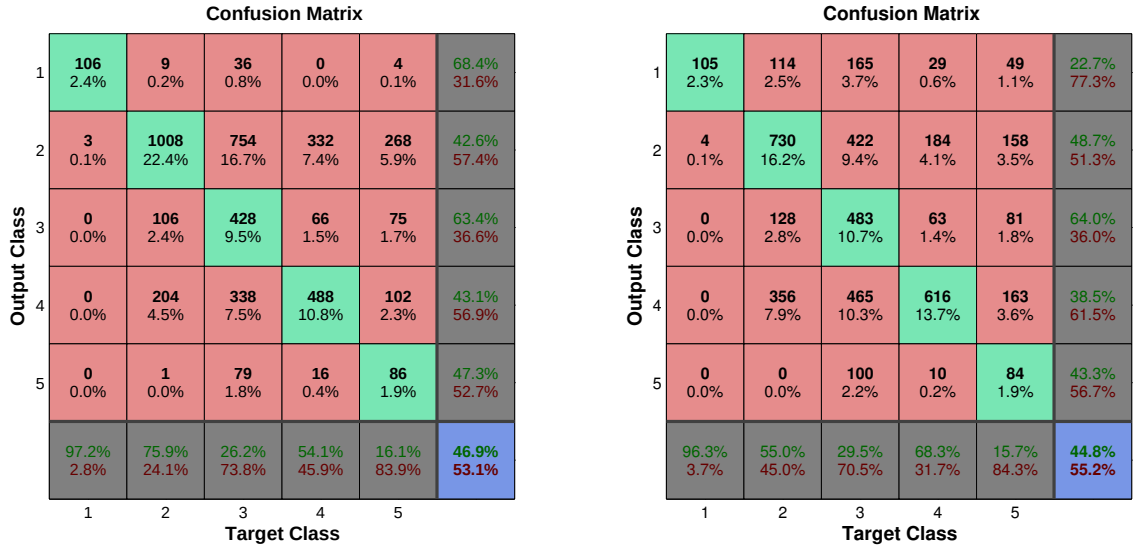


(b) Kernel Polinomial

Figura 4.24 – Matriz de Confusão Algorítimo MOPSO-48 horas.

4.8.2.6 Matriz de Confusão-72 horas

Na Figura 4.25 são apresentadas as matrizes de confusão para o algoritmo MOPSO usando o *Kernel* RBF e Polinomial. Pode ser observado que o *Kernel* RBF tem melhor acurácia (46,9%) do que o *Kernel* Polinomial (44,8%). O recall para o *Kernel* RBF tem melhores valores para classe 1 (97,2%) e classe 2 (75,9%), e uma precisão de 68,4%, para classe 1 e 63,4% para classe 3, enquanto que o *Kernel* Polinomial tem valores de racall de 96,3% e 683% (classes 1 e 4) e precisão de 64,0% (classe 3).



(a) Kernel RBF

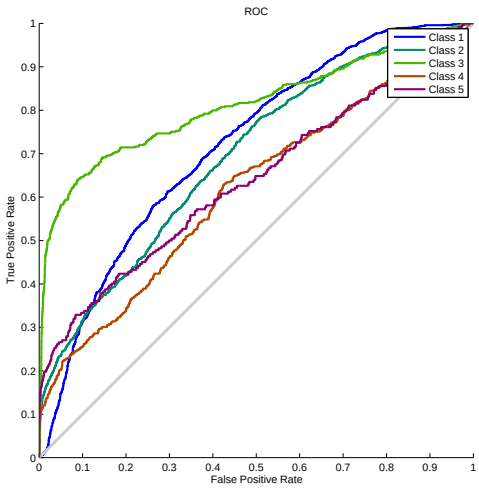
(b) Kernel Polinomial

Figura 4.25 – Matriz de Confusão Algorítimo MOPSO-72 horas.

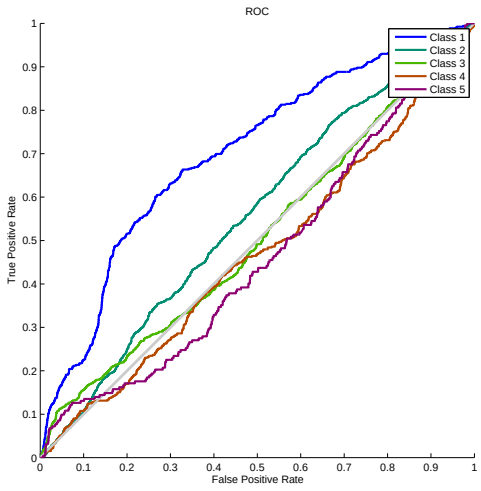
Observa-se, nestas matrizes, que as classes tiveram de uma maneira geral um comportamento semelhante, em termos de acurácia, precisão e recall.

4.8.2.7 Curvas ROC - 24 horas

As curvas ROC e seus respectivos AUC, para previsão de 24 horas, com os *Kernel* RBF e Polinomial, são apresentados na Figura 4.26 e Tabela 4.19 para o algorítimo MOPSO. A classe 3 tem uma performance superior as demais para o *Kernel* RBF, e todas as classes para estes *Kernel* se encontram na porção verdadeiro-positivo da curva ROC. Sob o aspecto do *Kernel* Polinomial apenas as classes 1 e 2, se encontram na porção verdadeiro-positivo da curva. Isto demostra que para previsão de 24 horas, o *Kernel* RBF tem o melhor desempenho



(a) Kernel RBF



(b) Kernel Polinomial

Figura 4.26 – Curvas ROC Algorítimo MOPSO-Previsão 24 hs.

Tabela 4.19 – AUC das Curvas ROC Algorítimo MOPSO-Previsão 24 hs.

(a) Kernel RBF

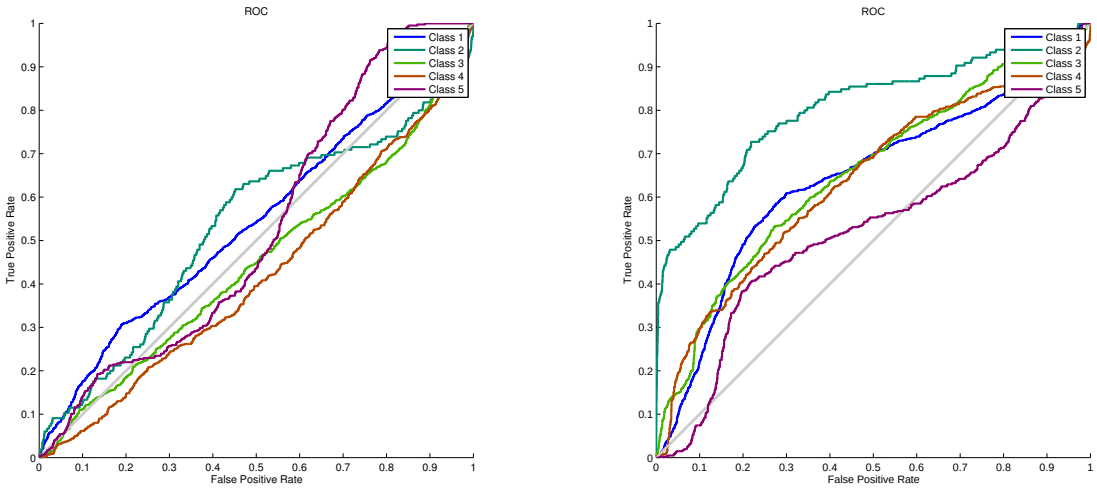
Classe	**AUC**
1	0,2472
2	0,4448
3	0,5029
4	0,621
5	0,6404

(b) Kernel Polinomial

Classe	**AUC**
1	0,701
2	0,5578
3	0,507
4	0,4688
5	0,4673

4.8.2.8 Curvas ROC - 48 horas

Para a previsão de 48 horas a Figura 4.27 e Tabela 4.20 apresentam os resultados utilizando o algorítimo MOSPO como otimizador. Para esta previsão o *Kernel* Polinomial exibe melhor performance que o *Kernel* RBF, com todas as classes na porção Verdadeiro-Positivo da curva ROC, destaque para a classe 3 com performance superior as demais classes.



(a) Kernel RBF

(b) Kernel Polinomial

Figura 4.27 – Curvas ROC Algorítimo MOPSO-Previsão 48 hs.

Tabela 4.20 – AUC das Curvas ROC Algorítimo MOPSO-Previsão 48 hs.

(a) Kernel RBF

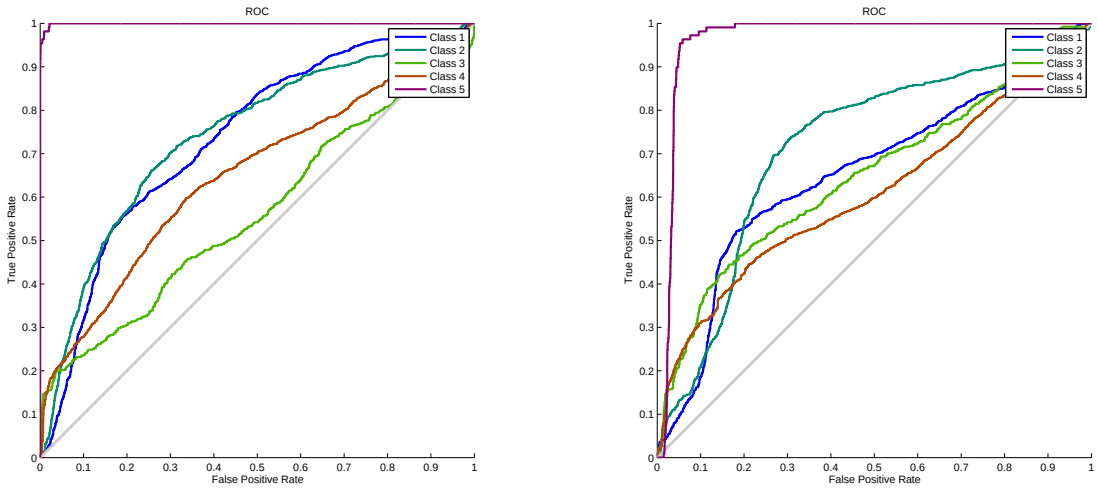
Classe	**AUC**
1	0,5439
2	0,4856
3	0,5604
4	0,6775
5	0,5279

(b) Kernel Polinomial

Classe	**AUC**
1	0,6455
2	0,4148
3	0,3041
4	0,3804
5	0,5235

4.8.2.9 Curvas ROC - 72 horas

A Figura 4.28 e a Tabela 4.21, apresentam as curvas ROC e seus respectivos AUC, para previsão de 72 horas de cada uma das classes, utilizando o algoritmo MOPSO, com *Kernel* RBF e *Kernel* Polinomial. Observa-se que todas as classes de ambos os *Kernel* se encontram na porção Verdadeiro-Positivo do gráfico, destaque para classe 5 (evento extremo) que apresentarem valores próximos ao ideal. Isto demonstra que ambos os *Kernel* tem desempenho satisfatório para previsão de 72 horas.



(a) Kernel RBF

(b) Kernel Polinomial

Figura 4.28 – Curvas ROC Algorítimo MOPSO-Previsão 72 hs.

Tabela 4.21 – AUC das Curvas ROC Algorítimo MOPSO-Previsão 72 hs.

(a) Kernel RBF

Classe	**AUC**
1	0,6565
2	0,259
3	0,4861
4	0,4763
5	0,9653

(b) Kernel Polinomial

Classe	**AUC**
Classe	AUC
1	0,7367
2	0,1674
3	0,4539
4	0,2588
5	0,9994

4.9 AVALIAÇÃO DOS RESULTADOS

Para a avaliação dos resultados, os mesmos foram agrupados em ranking de valores quantitativos e qualitativos. Para uma melhor visão e avaliação dos resultados obtidos, cada um dos grupos de resultados foi classificado da melhor para a pior performance, se estabelecendo rankings, os quais estão apresentados em tabelas descritas nas sub-seções a seguir.

4.9.1 Ranking dos Resultados Quantitativos - Histogramas

O ranking dos quantitativos tem com base os menores erros relativos a cada classes de precipitação. Estes valores foram calculados com base nos quantitativos apresentados nos histogramas. As Tabelas 4.22, 4.23 e 4.24 apresentam o ranking destes resultados. São apresentados também,

nestas tabelas, o Algorítimo e o Kernel utilizado, além dos valores do Erro, para cada classe de precipitação.

Tabela 4.22 – Ranking de Histogramas Previsão 24 hs

Rank	**Classe**	**Algoritimo**	**Kernel**	**Erro**
1º	1	MODE	Poly	11,06%
2º	1	MODE	Rbf	18,12%
3º	2	MOPSO	Rbf	32,24%
4º	2	MODE	Poly	43,35%
5º	4	MODE	Poly	51,43%
6º	2	MODE	Rbf	51,89%
7º	4	MODE	Rbf	57,14%
8º	5	MOPSO	Rbf	59,01%
9º	3	MODE	Poly	60,42%
10º	3	MODE	Rbf	64,58%
11º	3	MOPSO	Rbf	68,22%
12º	4	MOPSO	Rbf	68,82%
13º	5	MODE	Poly	69,03%
14º	1	MOPSO	Poly	69,19%
15º	5	MODE	Rbf	70,80%
16º	3	MOPSO	Poly	73,98%
17º	2	MOPSO	Poly	79,33%
18º	4	MOPSO	Poly	83,01%
19º	1	MOPSO	Rbf	88,66%
20º	5	MOPSO	Poly	96,40%

Tabela 4.23 – Ranking de Histogramas Previsão 48 hs

Rank	Classe	Algoritimo	Kernel	Erro
1º	2	MOPSO	Poly	2,69%
2º	5	MOPSO	Poly	3,03%
3º	4	MOPSO	Poly	3,33%
4º	5	MOPSO	Rbf	3,64%
5º	4	MOPSO	Rbf	6,00%
6º	2	MOPSO	Rbf	6,25%
7º	2	MODE	Rbf	14,51%
8º	4	MODE	Rbf	19,17%
9º	5	MODE	Rbf	22,42%
10º	1	MODE	Rbf	55,31%
11º	3	MOPSO	Rbf	56,32%
12º	3	MODE	Rbf	56,75%
13º	3	MOPSO	Poly	57,49%
14º	5	MODE	Poly	81,21%
15º	4	MODE	Poly	88,67%
16º	2	MODE	Poly	93,43%
17º	3	MODE	Poly	93,84%
18º	1	MOPSO	Rbf	116,54%
19º	1	MOPSO	Poly	150,37%
20º	1	MODE	Poly	751,36%

Tabela 4.24 – Ranking de Histogramas Previsão 72 hs

Rank	Algorítimo	Classe	Kernel	Erro
1º	MOPSO	2	Poly	12,80%
2º	MOPSO	4	Rbf	25,50%
3º	MODE	4	Rbf	36,25%
4º	MOPSO	1	Rbf	42,20%
5º	MOPSO	3	Poly	53,82%
6º	MODE	1	Rbf	56,88%
7º	MODE	3	Rbf	57,80%
8º	MOPSO	3	Rbf	58,72%
9º	MODE	2	Poly	59,41%
10º	MOPSO	5	Poly	63,74%
11º	MODE	5	Rbf	64,30%
12º	MOPSO	5	Rbf	65,98%
13º	MODE	2	Rbf	67,77%
14º	MOPSO	4	Poly	77,38%
15º	MOPSO	2	Rbf	78,09%
16º	MODE	4	Poly	79,82%
17º	MODE	3	Poly	96,70%
18º	MOPSO	1	Poly	323,85%
19º	MODE	5	Poly	352,34%
20º	MODE	1	Poly	1105,50%

Os resultados quantitativos apresentam, em sua maioria, erros relativos inferiores à 80% para números de eventos de cada uma das classes de previsão. Isto demonstra que a performance quantitativa dos algoritmos aplicados pode ser utilizada de forma adequada para qual período de precipitação se deseja prever, conforme atestam os histogramas relativos a cada previsão.

4.9.2 Ranking dos Resultados Qualitativos - Matriz de Confusão

Com base nos resultados obtidos pelas matrizes de confusão, foram criadas tabelas para cada uma das métricas destas matrizes (acurácia, precisão e recall).São apresentados também, nestas tabelas, qual o Algorítimo e o Kernel utilizado, além dos valores de cada métrica, dentro de cada uma das classes de precipitação.

4.9.2.1 Ranking dos Resultados Qualitativos - Matriz de Confusão - Métrica de Acurácia

As Tabelas 4.25, 4.26 e 4.27 apresentam o ranking dos resultados qualitativos para cada classe de precipitação, para os valores de acurácia. Como se observa o algorítimo MOSPO apresentou

melhores resultados que o MODE, entretanto com pequenas diferenças percentuais conforme atestam os valares apresentados nestas tabelas.

Tabela 4.25 – Ranking de Acurácia - Previsão 24 horas

Rank	**Algortimo/Kernel**	**Acurácia[%]**
1º	MOPSORBF	54,50
2º	MODERBF	52,60
3º	MODEPoly	50,20
4º	MOPSOPoly	48,70

Tabela 4.26 – Ranking de Acurácia - Previsão 48 horas

Rank	**Algortimo/Kernel**	**Acurácia[%]**
1º	MOPSOPoly	42,70
2º	MODERBF	42,40
3º	MOPSORBF	42,00
4º	MODEPoly	13,30

Tabela 4.27 – Ranking de Acurácia - Previsão 72 horas

Rank	**Algortimo/Kernel**	**Acurácia[%]**
1º	MOPSORBF	46,90
2º	MODERBF	46,30
3º	MOPSOPoly	44,80
4º	MODEPoly	14,80

4.9.2.2 Ranking dos Resultados Qualitativos - Matriz de Confusão - Métrica de Precisão

As Tabelas 4.28, 4.29 e 4.30 apresentam o ranking dos resultados qualitativos para cada classe de precipitação, para métrica de precisão, para os períodos de 24 horas, 48 horas e 72 horas. Como se observa as previsões para o período de 24 horas apresentaram resultados bastante expressivos, principalmente para classe 5 (eventos extremos).

Tabela 4.28 – Ranking de Precisão - Previsão 24 horas

Rank	**Algortimo/Kernel**	**Classe**	**Precisão[%]**
1º	MODERBF	5	81,80
2º	MODEPoly	5	71,40
3º	MODERBF	3	67,30
4º	MOPSORBF	3	63,50
5º	MOPSORBF	2	60,20
6º	MODEPoly	3	59,60
7º	MODERBF	4	56,70
8º	MODERBF	1	55,70
9º	MOPSOPoly	2	50,80
10º	MODEPoly	1	49,60
11º	MODEPoly	2	49,50
12º	MODERBF	2	49,40
13º	MOPSORBF	5	48,40
14º	MODEPoly	4	47,10
15º	MOPSORBF	4	46,20
16º	MOPSOPoly	3	39,30
17º	MOPSOPoly	1	37,40
18º	MOPSORBF	1	35,60
19º	MOPSOPoly	4	7,90
20º	MOPSOPoly	5	0,00

Tabela 4.29 – Ranking de Precisão - Previsão 48 horas

Rank	**Algortimo/Kernel**	**Classe**	**Precisão[%]**
1º	MOPSOPoly	2	60,70
2º	MODEPoly	2	57,10
3º	MOPSORBF	2	56,20
4º	MODERBF	3	53,30
5º	MODERBF	5	53,10
6º	MODERBF	2	51,20
7º	MOPSOPoly	3	48,50
8º	MOPSOPoly	5	47,50
9º	MOPSORBF	5	47,20
10º	MOPSORBF	3	46,50
11º	MOPSOPoly	4	33,80
12º	MODEPoly	3	31,00
13º	MOPSORBF	4	30,00
14º	MODERBF	4	26,30
15º	MODERBF	1	25,60
16º	MODEPoly	4	19,10
17º	MOPSORBF	1	19,00
18º	MOPSOPoly	1	17,00
19º	MODEPoly	1	11,70
20º	MODEPoly	5	0,00

Tabela 4.30 – Ranking de Precisão - Previsão 72 horas

Rank	Algortimo/Kernel	Classe	Precisão[%]
1º	MOPSORBF	1	68,40
2º	MOPSOPoly	3	64,00
3º	MOPSORBF	3	63,40
4º	MODEPoly	3	61,60
5º	MODEPoly	1	61,40
6º	MOPSOPoly	2	48,70
7º	MODEPoly	5	47,60
8º	MOPSORBF	5	47,30
9º	MOPSOPoly	5	43,30
10º	MODEPoly	2	43,10
11º	MOPSORBF	4	43,10
12º	MODERBF	2	42,90
13º	MOPSORBF	2	42,60
14º	MODEPoly	4	41,10
15º	MOPSOPoly	4	38,50
16º	MODERBF	3	25,90
17º	MOPSOPoly	1	22,70
18º	MODERBF	4	16,50
19º	MODERBF	5	14,50
20º	MODERBF	1	3,10

4.9.2.3 Ranking dos Resultados Qualitativos - Matriz de Confusão - Métrica de Recall

O ranking dos resultados qualitativos da Matriz de Confusão utilizando a métrica de recall são apresentados nas Tabelas 4.31, 4.32 e 4.33. Estes resultados demostram a existência de valores de recall superiores a 50% para classe 2 e 1, em previsão de 24 horas, classes 2 para previsão de 48 horas, e classes 1, 2, 4 e 5, para 72 horas. Não havendo grandes variações relativas aos mudanças do *Kernel* ou dos algorítimos de otimização (MODE ou MOSPO), demonstrando um bom comportamento do SVM como classificador de classes de precipitação.

Pode se observar também, que a técnica SVM utilizando o *Kernel* RBF tem melhores resultados que o *Kernel* Polinomial, para previsão de classes de precipitação. Pois, a utilização do *Kernel* Polinomial apresentou erros superiores a 50%, chegando a recall com erro de 100%, para algumas classes. A explicação seria porque o SVM usando *Kernel* Polinomial tem comportamento de concentrar valores em determinadas classes, diferente do *Kernel* RBF que distribui as classes.

Tabela 4.31 – Ranking de Recall - Previsão 24 horas

Rank	**Algortimo/Kernel**	**Classe**	**Recall[%]**
1º	MOPSOPoly	2	91,00
2º	MODERBF	2	75,10
3º	MODEPoly	2	70,90
4º	MOPSORBF	2	70,60
5º	MOPSORBF	1	68,80
6º	MODERBF	1	45,60
7º	MODEPoly	1	44,70
8º	MODERBF	4	24,30
9º	MODERBF	5	23,90
10º	MODERBF	3	23,80
11º	MODEPoly	3	23,60
12º	MODEPoly	4	22,90
13º	MODEPoly	5	22,10
14º	MOPSORBF	3	20,20
15º	MOPSORBF	5	19,80
16º	MOPSORBF	4	14,40
17º	MOPSOPoly	1	11,50
18º	MOPSOPoly	3	10,20
19º	MOPSOPoly	4	1,30
20º	MOPSOPoly	5	0,00

Tabela 4.32 – Ranking de Recall - Previsão 48 horas

Rank	**Algortimo/Kernel**	**Classe**	**Recall[%]**
1º	MODEPoly	1	99,50
2º	MOPSORBF	2	59,70
3º	MOPSOPoly	2	59,10
4º	MODERBF	2	58,70
6º	MOPSOPoly	5	46,10
5º	MOPSORBF	5	45,50
9º	MOPSOPoly	1	42,50
8º	MODERBF	5	41,20
7º	MOPSORBF	1	41,20
10º	MODERBF	1	39,80
13º	MOPSOPoly	4	32,70
12º	MODERBF	4	31,30
11º	MOPSORBF	4	28,20
14º	MODERBF	3	23,10
16º	MOPSOPoly	3	20,70
15º	MOPSORBF	3	20,30
17º	MODEPoly	2	3,80
18º	MODEPoly	4	2,20
19º	MODEPoly	3	1,90
20º	MODEPoly	5	0,00

Tabela 4.33 – Ranking de Recall - Previsão 72 horas

Rank	Algortimo/Kernel	Classe	Recall[%]
1º	MOPSORBF	1	97,20
2º	MODERBF	1	96,30
10º	MOPSOPoly	1	96,30
3º	MOPSORBF	2	75,90
4º	MODERBF	2	73,30
18º	MOPSOPoly	4	68,30
5º	MODEPoly	5	65,40
7º	MODERBF	4	56,00
14º	MOPSOPoly	2	55,00
8º	MOPSORBF	4	54,10
9º	MODEPoly	1	37,60
20º	MOPSOPoly	3	29,50
11º	MOPSORBF	3	26,20
12º	MODERBF	3	26,00
13º	MODEPoly	2	17,40
15º	MODERBF	5	17,00
16º	MOPSORBF	5	16,10
6º	MOPSOPoly	5	15,70
17º	MODEPoly	4	3,30
19º	MODEPoly	3	0,90

4.9.3 Ranking dos Resultados - Curvas ROC

A performance dos algoritmos foram boas, pois os mesmos possuem, na sua maioria, performance na porção do verdadeiro-positivo, conforme podem ser comprovadas na curvas ROC apresentadas anteriormente. Alem disto para uma melhor avaliação qualitativos destas curvas, foi utilizado os valores das AUC para ranking dos resultados. As Tabelas 4.34, 4.35 e 4.36 apresentam cada um destes rankings para os períodos utilizados para previsão - 24 horas, 48 horas e 72 horas.

Tabela 4.34 – Ranking de ROC (AUC) - Previsão 24 horas

Rank	**Algoritimo**	**Classe**	**AUC**
1º	MOPSORBF	1	0,2472
2º	MOPSORBF	2	0,4448
3º	MOPSOPoly	5	0,4673
4º	MOPSOPoly	4	0,4688
5º	MOPSORBF	3	0,5029
6º	MOPSOPoly	3	0,5070
7º	MOPSOPoly	2	0,5578
8º	MODEPoly	3	0,5775
9º	MODEPoly	2	0,5963
10º	MODEPoly	1	0,6070
11º	MODEPoly	4	0,6172
12º	MOPSORBF	4	0,6210
13º	MODERBF	2	0,6316
14º	MOPSORBF	5	0,6404
15º	MODERBF	3	0,6455
16º	MODERBF	4	0,6514
17º	MODERBF	1	0,6759
18º	MODEPoly	5	0,6910
19º	MOPSOPoly	1	0,7010
20º	MODERBF	5	0,7134

Tabela 4.35 – Ranking de ROC (AUC) - Previsão 48 horas

Rank	**Algoritimo**	**Classe**	**AUC**
1º	MODERBF	5	0,7288
2º	MODEPoly	5	0,6938
3º	MOPSORBF	4	0,6775
4º	MODERBF	1	0,6529
5º	MOPSOPoly	1	0,6455
6º	MOPSORBF	3	0,5604
7º	MOPSORBF	1	0,5439
8º	MOPSORBF	5	0,5279
9º	MOPSOPoly	5	0,5235
10º	MODEPoly	2	0,4936
11º	MODEPoly	1	0,4913
12º	MOPSORBF	2	0,4856
13º	MOPSOPoly	2	0,4148
14º	MODERBF	4	0,4136
15º	MODERBF	2	0,4027
16º	MODEPoly	3	0,3998
17º	MOPSOPoly	4	0,3804
18º	MODERBF	3	0,3706
19º	MODEPoly	4	0,3428
20º	MOPSOPoly	3	0,3041

Tabela 4.36 – Ranking de ROC (AUC) - Previsão 72 horas

Rank	Algoritimo	Classe	AUC
1º	MOPSORBF	5	0,9994
2º	MODERBF	5	0,9992
3º	MOPSOPoly	5	0,9653
4º	MOPSORBF	1	0,7367
5º	MODERBF	1	0,7257
6º	MOPSOPoly	1	0,6565
7º	MODEPoly	5	0,6235
8º	MODEPoly	4	0,5920
9º	MODEPoly	1	0,5219
10º	MOPSOPoly	3	0,4861
11º	MOPSOPoly	4	0,4763
12º	MODERBF	3	0,4668
13º	MOPSORBF	3	0,4539
14º	MODEPoly	2	0,3910
15º	MODEPoly	3	0,3757
16º	MODERBF	4	0,3217
17º	MOPSOPoly	2	0,2590
18º	MOPSORBF	4	0,2588
19º	MODERBF	2	0,1956
20º	MOPSORBF	2	0,1674

4.10 CONCLUSÕES DO CAPÍTULO

A metodologia proposta (capítulo 3) junto com o referência teórico e os trabalhos correlatos (capítulo 2), foram aplicados ao estudo de caso neste capítulo. Desta maneira, ficou comprovada a eficiência dos classificadores SVMs para previsão de classes de precipitação. A utilização de dados reais, de aproximadamente um ano de coleta, para treinamento e previsão, demostraram também que com baixo volume de dados pode-se fazer previsão de precipitação usando dispositivos inteligentes. Além disto, o estudo de caso, com dados reais de precipitação para Belém-Pa, foi adequado para comprovação dos objetivos desta pesquisa, ou seja, utilizar classificadores SVM como ferramenta de previsão de precipitação em sistemas embarcados.

Como em (CHEN; CHANG; LIN, 2004) a maioria dos trabalho disponíveis, são para utilização de regressores, o que criou uma dificuldade de comparação dos resultados com outras pesquisas. Assim, os resultados foram comparados diretamente com dados reais (3.327 amostras horárias, o que equivale a mais de 4 meses de dados reais, ou 134 dias). Os resultados demonstraram acertos quantitativos significativos, conforme demonstraram os histogramas apresentados.

Além de bom desempenho qualitativo conforme demonstraram os resultados apresentados nas matrizes de confusão e curvas ROC.

Ressalta-se que os algoritmos foram usados como classificador de classes de precipitação, diferentemente do que ocorreu em pesquisa realizadas com o SVMS como regressor ((CHEN; CHANG; LIN, 2004)),cujo os acertos foram melhores. Porém é importante destacar que o comportamento de classes de precipitação, possuem uma grande incerteza o que implica diretamente no balanceamento dos dados, visto que as variações podem ser grandes para curto intervalos de tempo. Por exemplo, a mudança de quantitativos de precipitação pode variar de 0 mm à 200 mm em um espaço de 1 hora. Outras questão, é que a maioria das propostas de utilização de SVMS como regressor, para realizar previsão, usam dados meteorológicos globais, e não somente informações locais da estação, como proposto em MESTRE et al.. Alem disto, a proposta desta pesquisa foi para o uso de baixo de baixo volume de dados.

No entanto, em termos qualitativos a performance do algorítimo foi boa, e demostrou que depende das reais necessidades do problema a ser aplicado. Ou seja, uma seleção de qual classe de precipitação se deseja destacar na classificação, o que implicar na escolha do tipo de algorítimo otimizador (MODE ou MOPSO) e qual a função *Kernel* (RBF ou Polinomial) será usada para previsão.

5 CONCLUSÃO

A proposta de utilizar classificadores com o algorítimo SVM, usando os algorítimos MODE e MOSPO, como otimizadores, com as funções *Kernel* RBF e polinomial, permitiu uma classificação direta e simples das classes de precipitações para períodos mínimos de 24 horas, com utilização de dados locais somente. Isto torna estes classificadores eficientes, principalmente, para previsão de eventos extremos, que são os que mais prejudicam a sociedade, pois, precipitações acima de 50 mm já provocam alagamentos, principalmente, nas regiões urbanas, podendo ocasionar quedas de árvores, fazendo com que a sociedade tenha perdas materiais ou mesmo, em casos extremos, a própria vida. Além disto, os modelos mostraram uma boa acurácia, e uma precisão e *recall* para classes de precipitação, podendo ser implementados para previsão de curto prazo.

Infere-se também, que os classificadores do tipo SVMs, treinados e otimizados com técnicas bio-inspiradas, possuem desempenhos apropriados à proposta deste estudo, pois, os mesmos apresentam características de convergência rápida no processo de treinamento.

Os modelos gerados pelo algorítimo SVM, com otimizadores MODE e MPSO e os *kernels* RBF e polinomial, demostraram possuir uma boa eficácia computacional, pois consumiram não mais de que 10 minutos de processamento para as tarefas de previsão, tempo este suficiente para atender os requisitos da janelas de previsão (mínimo de 1 hora), proposto por esta pesquisa. Portanto possíveis de se aplicar em sistemas embarcados, pois, não requerem grandes recursos de hardware e software, além de usar baixa quantidade de dados (curto período de tempo).

Os modelos obtidos nesta pesquisa podem ser implementados em dispositivos inteligentes, tais como drones, estações inteligentes e robôs, o que poderá ajudar nos monitoramentos de: Transporte Ferroviário de Minérios, Irrigação na Agricultura, Auxilio no Planejamento e Autonomia de Voos de Drones em ambientes externos, Auxilio no Planejamento e Autonomia de Trajetórias de Robôs em ambiente externo e, principalmente, para previsão de Eventos Extremos de precipitação que terá grande contribuição para a sociedade e o meio-ambiente.

Os experimentos e resultados demonstraram um baixo consumo computacional com uma utilização de um baixo volume de dados, reforçando e demonstrando assim as hipóteses propostas nesta pesquisa. Este tipo de resultado viabilizará a implementação dos classificadores em sistemas embarcados e/ou dispositivos inteligentes, os quais têm restrições na capacidade de processamento, tamanho de memória e consumo de energia elétrica.

Da mesma maneira, esta pesquisa serviu para validar a ferramenta NIOTs desenvolvida no LEIA (Laboratório de Sistemas Embarcados e Aplicações de Circuitos Integrados), a qual permite sintonizar os parâmetros dos kernels das SVMs usando otimização multi-objetivo e algoritmos bio-inspirados.

- Aplicabilidade

Estas técnicas demostrarem ter um vasto campo de pesquisa e aplicação, em diversas áreas, dentre elas podemos destacar as seguintes :

- Monitoramento de Transporte Ferroviário de Minérios
- Monitoramento de Irrigação na Agricultura
- Monitoramento de Eventos Extremos
- Auxílio no Planejamento e Autonomia de Voos de Drones em ambiente externos
- Auxílio no Planejamento e Autonomia de Trajetórias de Robôs em ambiente externos

- Trabalhos Futuros

Os próximos trabalhos devem estender a aplicação destes algoritmos para outras previsões, como por exemplo: temperatura e umidade. Além de outros trabalhos como :

- Ampliar o número de Variáveis Preditas
- Implementar em Dispositivos Inteligentes (Estações meteorológicas, drones e robôs)
- Melhorar as previsões para eventos extremos

REFERÊNCIAS BIBLIOGRÁFICAS

BASTOS, T. X. et al. Aspectos climáticos de belém nos últimos cem anos. *Embrapa Amazônia Oriental-Documentos (INFOTECA-E)*, Belém, PA: Embrapa Amazônia Oriental, 2002., 2002.

BERGH, F. Van den; ENGELBRECHT, A. P. A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation*, IEEE, v. 8, n. 3, p. 225–239, 2004.

BRERETON, R. G.; LLOYD, G. R. Support vector machines for classification and regression. *The Analyst*, v. 135, n. 2, p. 230–267, 2010. ISSN 0003-2654.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.

CARBONERO-RUZ, M. et al. A two dimensional accuracy-based measure for classification performance. *Information Sciences*, v. 382-383, p. 60 – 80, 2017. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025516319181>.

CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

CHEN, B.-J.; CHANG, M.-W.; LIN, C.-J. Load forecasting using support vector machines: A study on EUNITE competition 2001. *IEEE Transactions on Power Systems*, Institute of Electrical and Electronics Engineers (IEEE), v. 19, n. 4, p. 1821–1830, nov 2004.

COELLO, G. B. L. C. A. C.; VELDHUIZEN, D. A. V. *Evolutionary Algorithms for Solving Multi-Objective Problems*. USA: Springer Science, 2007. ISBN 978-0-387-33254-3.

CORTES, C.; VAPNIK, V. Support vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 08856125. Disponível em: <http://link.springer.com/10.1007/BF00994018>.

DAS, M.; GHOSH, S. K. A probabilistic approach for weather forecast using spatio-temporal inter-relationships among climate variables. In: IEEE. *2014 9th International Conference on Industrial and Information Systems (ICIIS)*. [S.l.], 2014. p. 1–6.

EIBEN, A. E.; SMITH, J. E. et al. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003. v. 53.

FAN, Q.; ZHANG, Y. Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation. *Chemometrics and Intelligent Laboratory Systems*, Elsevier, v. 151, p. 164–171, 2016.

FERREIRA, L. B. et al. Estimation of reference evapotranspiration in brazil with limited meteorological data using ANN and SVM – a new approach. *Journal of Hydrology*, Elsevier BV, v. 572, p. 556–570, may 2019.

FUENTES, M. et al. Design of an accurate, low-cost autonomous data logger for PV system monitoring using arduino^{TM} that complies with IEC standards. *Solar Energy Materials and Solar Cells*, Elsevier BV, v. 130, p. 529–543, nov 2014.

GONG, M. et al. *Bio-inspired Computing – Theories and Applications: 11th International Conference, BIC-TA 2016, Xi'an, China, October 28-30, 2016, Revised Selected Papers*. Springer Singapore, 2017. (Communications in Computer and Information Science, pt. 2). ISBN 9789811036149. Disponível em: <https://books.google.com.br/books?id=IXfbDQAAQBAJ>.

GRIVA, I.; NASH, S.; SOFER, A. *Linear and Nonlinear Optimization: Second Edition*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009. ISBN 9780898717730. Disponível em: <https://books.google.com.br/books?id=uOJ-Vg1BnKgC>.

GRIVA, I.; NASH, S. G.; SOFER, A. *Linear and Nonlinear Optimization (2. ed.).* [S.l.]: SIAM, 2008. I-XXII, 1-742 p. ISBN 978-0-89871-661-0.

HAMMER, Ø. et al. Past: paleontological statistics software package for education and data analysis. *Palaeontologia electronica*, California, v. 4, n. 1, p. 9, 2001.

HEMA, N.; KANT, K. Hourly real-time rainfall estimation for improved smart irrigation system using nearby automated weather station. *British Journal of Applied Science & Technology*, Sciencedomain International, v. 18, n. 5, p. 1–13, jan 2016.

HOENS, T. R.; CHAWLA, N. V. Imbalanced datasets: From sampling to classifiers. In: *Imbalanced Learning*. [S.l.]: John Wiley & Sons, Inc., 2013. p. 43–59.

HONG, W.-C. Rainfall forecasting by technological machine learning models. *Applied Mathematics and Computation*, Elsevier BV, v. 200, n. 1, p. 41–57, jun 2008.

IEEE. *IEEE is Fueling the Fourth Industrial Revolution*. 2018. Disponível em: <http://www.wpcentral.com/ie9-windows-phone-7-adobe-flash-demos-and-development-videos>.

INMET. *Consulta Dados da Estação Automática: Belém (PA)*. 2019. Disponível em: <http://www.inmet.gov.br/sonabra/pg_dspDadosCodigo_sim.php?QTIwMQ==>.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of ICNN 95 - International Conference on Neural Networks*. [S.l.]: IEEE.

KISI, O.; CIMEN, M. Precipitation forecasting by using wavelet-support vector machine conjunction model. *Engineering Applications of Artificial Intelligence*, Elsevier BV, v. 25, n. 4, p. 783–792, jun 2012.

KRASNOPOLSKY, V. M.; FOX-RABINOVITZ, M. S. Complex hybrid models combining deterministic and machine learning components as a new synergetic paradigm in numerical climate modeling and weather prediction. In: IEEE. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* [S.l.], 2005. v. 3, p. 1615–1620.

LIMA, C. A. de M. *Comitê de máquinas: uma abordagem unificada empregando máquinas de vetores-suporte*. Tese (Tese), 2004.

LIN, Q. et al. A novel multi-objective particle swarm optimization with multiple search strategies. *European Journal of Operational Research*, Elsevier, v. 247, n. 3, p. 732–744, 2015.

LLASAT, M.-C. An objective classification of rainfall events on the basis of their convective features: application to rainfall intensity in the northeast of spain. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, Wiley Online Library, v. 21, n. 11, p. 1385–1400, 2001.

LU, W. et al. Air pollutant parameter forecasting using support vector machines. In: IEEE. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. [S.l.], 2002. v. 1, p. 630–635.

MESTRE, G. et al. An intelligent weather station. *Sensors*, MDPI AG, v. 15, n. 12, p. 31005–31022, dec 2015.

NASSERI, M.; ASGHARI, K.; ABEDINI, M. Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network. *Expert Systems with Applications*, Elsevier BV, v. 35, n. 3, p. 1415–1421, oct 2008.

NOERGAARD, T. *Embedded systems architecture: a comprehensive guide for engineers and programmers*. [S.l.]: Newnes, 2012.

NOVAK, D. WEATHER FORECASTING | operational meteorology. In: *Encyclopedia of Atmospheric Sciences*. [S.l.]: Elsevier, 2015. p. 293–302.

NURCAHYO, S.; NHITA, F. et al. Rainfall prediction in kemayoran jakarta using hybrid genetic algorithm (ga) and partially connected feedforward neural network (pcfnn). In: IEEE. *2014 2nd International Conference on Information and Communication Technology (ICoICT)*. [S.l.], 2014. p. 166–171.

RASOULI, K.; HSIEH, W. W.; CANNON, A. J. Daily streamflow forecasting by machine learning methods with weather and climate inputs. *Journal of Hydrology*, Elsevier BV, v. 414-415, p. 284–293, jan 2012.

REN, Y.; ZHANG, L.; SUGANTHAN, P. N. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, IEEE, v. 11, n. 1, p. 41–53, 2016.

RIGET, J.; VESTERSTRØM, J. S. A diversity-guided particle swarm optimizer-the arpso. *Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep*, Citeseer, v. 2, p. 2002, 2002.

ROCCA, P.; OLIVERI, G.; MASSA, A. Differential evolution as applied to electromagnetics. *IEEE Antennas and Propagation Magazine*, IEEE, v. 53, n. 1, p. 38–49, 2011.

SAMPAIO, G.; SILVA DIAS, P. L. da. Evolução dos modelos climáticos e de previsão de tempo e clima. *Revista USP*, Universidade de Sao Paulo Sistema Integrado de Bibliotecas - SIBiUSP, n. 103, p. 41, nov 2014.

SANTOS, C. E. *SELEÇÃO DE PARÂMETROS DE MÁQUINAS DE VETORESDE SUPORTE USANDO OTIMIZAÇÃO MULTIOBJETIVOBASEADA EM META-HEURÍSTICAS*. Tese (Tese Doutorado) — Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Mecânica, mar 2019.

SANTOS, J. S. d. et al. Frequência de precipitação e impactos decorrentes associados à chuva na cidade de belém-pa. Universidade Federal do Pará, 2014.

SCHÖLKOPF, B.; BURGES, C.; VAPNIK, V. Incorporating invariances in support vector learning machines. In: SPRINGER. *International Conference on Artificial Neural Networks*. [S.l.], 1996. p. 47–52.

SEO, J.-H.; LEE, Y. H.; KIM, Y.-H. Feature selection for very short-term heavy rainfall prediction using evolutionary computation. *Advances in Meteorology*, Hindawi Limited, v. 2014, p. 1–15, 2014.

SFETSOS, A.; COONICK, A. Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. *Solar Energy*, Elsevier, v. 68, n. 2, p. 169–178, 2000.

SHARMA, N. et al. Predicting solar generation from weather forecasts using machine learning. In: IEEE. *2011 IEEE international conference on smart grid communications (SmartGridComm)*. [S.l.], 2011. p. 528–533.

SIVAPRAGASAM, C.; LIONG, S.-Y.; PASHA, M. Rainfall and runoff forecasting with ssa–svm approach. *Journal of Hydroinformatics*, IWA Publishing, v. 3, n. 3, p. 141–152, 2001.

SMOLA, A. J. et al. Advances in Large Margin Classi ers. *Advances*, 1999.

SMOLA, A. J. et al. *Regression estimation with support vector learning machines*. Tese (Doutorado) — Master's thesis, Technische Universität München, 1996.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and computing*, Springer, v. 14, n. 3, p. 199–222, 2004.

SOUZA, W. M.; AZEVEDO, P. V. de; ARAÚJO, L. E. de. Classificação da precipitação diária e impactos decorrentes dos desastres associados às chuvas na cidade do recife-PE. *Revista Brasileira de Geografia Física*, Revista Brasileira de Geografia Fisica, v. 5, n. 2, p. 250, oct 2012.

STORN, R.; PRICE, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, Springer, v. 11, n. 4, p. 341–359, 1997.

STORN, R.; PRICE, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, Springer, v. 11, n. 4, p. 341–359, 1997.

TECHSPARKS. *Confusion matrix*. 2019. Disponível em: <https://www.techsparks.co.in/hot-topic-for-project-and-thesis-machine-learning/>.

TING, K. M. Confusion matrix. In: ____. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 209–209. ISBN 978-0-387-30164-8. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8_157>.

TRAFALIS, T. B.; SANTOSA, B.; RICHMAN, M. B. Prediction of rainfall from WSR-88d radar using kernel-based methods. *International Journal of Smart Engineering System Design*, Informa UK Limited, v. 5, n. 4, p. 429–438, oct 2003.

VALHOULI, C. A. The internet of things: Networked objects and smart devices. *The hammersmith group research report*, v. 20, 2010.

WANG, W.; XU, Z.; LU, J. W. Three improved neural network models for air quality forecasting. *Engineering Computations*, Emerald, v. 20, n. 2, p. 192–210, mar 2003.

WEI, C.-C. Wavelet support vector machines for forecasting precipitation in tropical cyclones: Comparisons with GSVM, regression, and MM5. *Weather and Forecasting*, American Meteorological Society, v. 27, n. 2, p. 438–450, apr 2012.

WIKIPEDIA. *Smart Device*. Disponível em: <https://en.wikipedia.org/wiki/Smart_device>.

WIKIPEDIA. *Confusion matrix*. 2019. Disponível em: <https://en.wikipedia.org/wiki/Confusion_matrix>.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, Elsevier, v. 2, n. 1-3, p. 37–52, 1987.

ZHAO, Z. et al. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric latin hypercube design for unconstrained optimization problems. *European Journal of Operational Research*, v. 250, n. 1, p. 30 – 45, 2016. ISSN 0377-2217. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221715009698>.

ZHENG, Y. et al. Forecasting fine-grained air quality based on big data. In: ACM. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.], 2015. p. 2267–2276.

APÊNDICES

A MÁQUINAS DE VETORES DE SUPORTE PARA REGRESSÃO - SVR

Os *regressores* (ou aproximadores de funções) são modelos matemáticos utilizados para determinar o comportamento de um sistema do tipo *caixa preta*, simplificar modelos matemáticos complexos, eliminar informações indesejadas (ruídos e *outliers*); e podem ser empregados como preditores. As *Support Vector Regressors - SVR* são modelos matemáticos baseados na mesma teoria empregada nas SVMs como classificadores.

Os SVRs são obtidas a partir de um conjunto de treinamento $X = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N), \} \subset X \times \mathbb{R}$, em que $X = \mathbb{R}^d$ denota o espaço do padrão de entrada. Os SVRs são definidos de maneira semelhante às SVMs classificadoras, bastando para isso adequar a função de penalização dos erros cometidos no conjunto de treinamento.

Como no caso dos classificadores, inicialmente será tratado o caso linear, sendo a base para os demais conceitos desenvolvidos. A função $f : X \to \mathbb{R}$ é definida por:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b,$$

em que $\mathbf{w} \in X$, $b \in \mathbb{R}$.

As margens de um aproximador são dadas por:

$$\begin{aligned} H_1 : \quad & \mathbf{w} \cdot \mathbf{x} + b + \epsilon = 0 \\ H_2 : \quad & \mathbf{w} \cdot \mathbf{x} + b - \epsilon = 0, \end{aligned} \tag{A.1}$$

em que ϵ é o erro empírico permitido para cada dado do conjunto de treinamento. Nesse caso, a distância entre as margens é dada pela Equação A.2 e deve ser maximizada.

$$d(H_1, H_2) = \frac{2\epsilon}{\|\mathbf{w}\|}. \tag{A.2}$$

A Equação A.2 deve ser maximizada, para que as margens sejam o mais largas quanto possível e, assim, garantir uma boa generalização. O problema de otimização primal para o caso linear é dado pelo modelo A.3:

$$\begin{aligned} & \text{Min} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 \\ & s.a. \\ & y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) \geqslant \epsilon \qquad i = 1, ..., N \\ & (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i \geqslant \epsilon \qquad i = 1, ..., N, \end{aligned} \tag{A.3}$$

em que n é a cardinalidade do conjunto de treinamento.

O problema de aproximação de funções, modelado como em A.3, pode conter dados que não

são linearmente separáveis devido, por exemplo, a erros adicionados aos dados de treinamento, os quais não devem ser considerados. Nesse caso, permite-se erros no conjunto de treinamento que são penalizados por uma constante positiva. O modelo desse problema é conhecido como margens suaves, descrito pelo modelo A.4:

$$\begin{aligned}
&\text{Min} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\
&s.a. \\
&y_i - (\mathbf{w}\cdot\mathbf{x}_i + b) \geqslant \epsilon + \xi_i \qquad i = 1, ..., N \\
&(\mathbf{w}\cdot\mathbf{x}_i + b) - y_i \geqslant \epsilon + \xi_i^* \qquad i = 1, ..., N \\
&\xi_i, \quad \xi_i^* \geqslant 0.
\end{aligned} \tag{A.4}$$

O parâmetro de regularização C penaliza somente os dados que estão fora da região definida pelo parâmetro ϵ. Essa região é definida pela função de perda (*loss function*). Existem vários tipos de funções de perda, sendo as mais comuns a ϵ-insensitive, a Laplaciana, a Gaussiana, a perda robusta de Huber, a polinomial e polinomial por partes (SMOLA; SCHÖLKOPF, 2004; SMOLA et al., 1996). Neste trabalho, foi adotada a ϵ-*insensitive*, definida por Vapnik.

A função aproximadora é definida dentro do *tubo insensível*, conforme ilustrado na Figura A.1. Os dados são penalizados conforme a função de perda ϵ-*insensitive*. A Figura A.1 representa o gráfico da Equação A.5.

$$|\xi|_\epsilon = \begin{cases} 0, & \text{se } |\xi| \leq \epsilon \\ |\xi| - \epsilon, & \text{caso contrário .} \end{cases} \tag{A.5}$$

A função de perda ϵ-insensitive é atrativa, pois, ao contrário das funções perda quadrática e Huber, em que todos os pontos do conjunto de treinamento são vetores de suporte, a solução daquela função perda é esparsa. Devido a essa característica, as funções aproximadoras possuem maior eficiência computacional e uma boa aproximação das funções ideais (BRERETON; LLOYD, 2010).

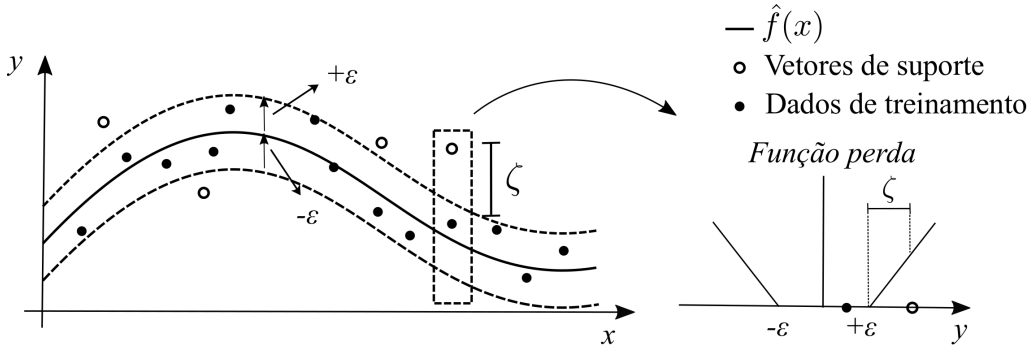


Figura A.1 – Representação da função ϵ-insensitive, função aproximadora $\hat{f}(x)$. Adaptado de (SMOLA; SCHÖLKOPF, 2004).

Na Figura A.1, a função $f(x)$ é a função ideal e a função $\hat{f}(x)$ é a função aproximadora.

B PROGRAMA GERADOR DE DADOS

Neste apêndice se apresenta o código B, em script Matlab, utilizado para o programas gerador de dados. Este programa realiza todas as tarefas relativas a preparação dos dados e descritas em 3.6.

```
%% Gera dados para experimentos
clc;
clear;
disp ('==> Start');
%% Parametres
win_delta = 24 * 1; % hours * day
win_lite = 6;
win_tx = win_delta/win_lite;
rel_level = 3; % relevancia level
%% Load Relevancia & INMET Data
Trel = readtable('Relevancia.csv');
[lrel crel] = size (Trel);
filename = 'Inmet20190207.csv';
Tgross = readtable(filename);
Tdata = Tgross;
clear Tgross;
header = Tdata.Properties.VariableNames;
% header = strrep(header,'_','-');
Tdata.data=datenum(Tdata.data,'dd/mm/yyyy');
[lin_org col_org]=size (Tdata);
Tdata = Tdata(1:(lin_org/1.),:);
[lin_org col_org]=size (Tdata);
Tinput = sortrows(Tdata);
%% Insert Mothly
mes = month(Tinput.data);
Tinput.mes = [mes];
clear Tdata;
%% Delete Column by Relevancia
disp ([{'==> Deleting Column by Relevancia'} i]);
col_del = col_org;
j=0;
for i=1:col_del
j=j+1;
if (table2array(Trel(i,3))>rel_level) % test for  relevancia level
Tinput (:,j)=[];
col_del = col_del - 1;
j=j-1;
end
end
%% Create Precipitacao Acumulada Columns
disp ('==> Create Accum Rain Colun');
```

```
[lin col] = size(Tinput);
Train = zeros (lin, 1);
for j=win_delta:1:lin
Tsumrain = zeros (win_delta,1);
disp ([{'==> Creating Accum Rain Colun ...'} j]);
for i=1:win_delta
lin_atual = (j-i+1);
Tsumrain(i,:) = table2array (Tinput(lin_atual,19));
end;
Train(j,1) = nansum(Tsumrain);
end;
Train = array2table(Train);
Tinput = [Tinput Train]; % Tmode];
%% Delete Invalid Input
Arrday = table2array(Tinput(:,(2:end)));
[lin col]=size (Arrday);
for i=1:col
disp ([{'==> Deleting Invalid Inputs...'} i]);
toDel = isnan (Arrday(:,(i)));
Arrday(toDel,:)=[];
end
[lin col]=size (Arrday);
disp ([lin_org-lin {'==> Deleted Invalid Inputs...'}]);
Tday = array2table(Arrday);
Tday.Properties.VariableNames = Tinput.Properties.VariableNames(2:end);
%% Quartil Definition
q_range = unique (Tday.Train);
[lq cq]=size(q_range);
q1 = min(q_range);
q2 = q_range(1:round(lq/2),:);
q2 = median (q2);
q3 = median(q_range);
q4 = q_range(round(lq/2):end,:);
q4 = median (q4);
q5 = max(q_range);
save ('quartil.mat', 'q1','q2','q3','q4','q5')
%% Rain coondition Column
Tchuva = ones (lin,1);%  {'sem chuva'};
Tchuva = array2table(Tchuva);
Tchuva.Tchuva (Tday.Train > q1) = 2 ;% {'chuva fraca'};
Tchuva.Tchuva (Tday.Train > q2) = 3 ;% {'chuva moderada'};
Tchuva.Tchuva (Tday.Train > q3) = 4 ;% {'chuva forte'};
Tchuva.Tchuva (Tday.Train > q4) = 5 ;% {'chuva muito forte'};
%% Create Mean Columns
disp ('==> Start Windows Mean Columns');
Tinput = [Tday];
Tday(:,19)  = []; % delete precipitacao
```

```
Tday(:,18)  = []; % delete mes
Tday(:,1)  = []; % delete hora
[lin col] = size(Tday);
Tmedia = zeros (lin, (col)*win_tx);
Tmode = zeros (lin, (col)*win_tx);
Tstd = zeros (lin, (col)*win_tx);
Tsum = zeros (lin, (col)*win_tx);
for j=win_delta:1:lin
Tw_media = zeros;
Tw_mode = zeros;
Tw_sum = zeros;
Tmeand = zeros (win_delta, col);
disp ([{'==> Creating Windows Mean Columns ...'} j]);
k_ini = 1;
k_fim = win_lite;
for i=1:win_tx
for k=k_ini:1:k_fim
Tmeand(k,:) = table2array(Tday(j-k+1,1:col));
end;
k_ini = k_ini + win_lite;
k_fim = k_fim + win_lite;
Tw_media = [Tw_media mean(Tmeand)];
Tw_mode  = [Tw_mode  mode(Tmeand)];
Tw_sum = [Tw_sum sum(Tmeand)];
end;
Tw_media = Tw_media (1,2:end);
Tmedia(j,:) = Tw_media;
Tw_mode = Tw_mode (1,2:end);
Tmode(j,:) = Tw_mode;
Tw_sum = Tw_sum (1,2:end);
Tsum(j,:) = Tw_sum;
end;

Tmedia = array2table(Tmedia);
Tmode = array2table(Tmode);
Tsum = array2table(Tsum);
Tinput = [Tinput Tmedia];% Tmode Tsum] ;
Tinput (:,1)  = []; % delete data
Tinput (:,1)  = []; % delete hora
%% Dist % & Plot Histogram
x = Tchuva.Tchuva;
x = x';
tabulate(x)
figure
hist(x)
%% Extract Data
win_start = win_delta+1; % window past start
```

```
[win_length  col] = size(Tinput);
win_length = win_length - win_delta - win_delta;
win_end = win_start + win_length; % window past end
%% verifica tamanho iguais
y_tr_un = 0;
y_val_un = 1;
y_un = 2;
while  ~((y_val_un ==y_tr_un) && ( 15  ==y_val_un)) % rating classes
x = table2array (Tinput(win_start:win_end-win_delta,:));
y = table2array (Tchuva(win_start+win_delta:win_end,:)); %chuva = com adianto
% group = table2array (Tmean(win_start:win_end,11:col));

l = size(x, 1);

%% Criando os dados de treinamento
aux_tr = l*0.7;
aux_tr = round (aux_tr);

ind_tr = randi(aux_tr,1,aux_tr);

x_tr = x(ind_tr, :);
y_tr = y(ind_tr, :);

x(ind_tr,:) = [];
y(ind_tr,:) = [];

%% Gerando os dados de validacaoo
l = size(x,1);
aux_val = round (l/2);
ind_val = randi(aux_val,1,aux_val);

x_val = x(ind_val, :);
y_val = y(ind_val, :);

%% Gerando os dados de teste
x(ind_val,:) = [];
y(ind_val,:) = [];
%% identifica classes geradss
y_tr_un = sum(unique (y_tr));
y_val_un = sum(unique (y_val));
y_un = sum(unique (y));
end;
escrever_dados([ num2str(win_delta) 'hr_chuva_treino_janelinha.txt'], x_tr, y_tr)
escrever_dados([ num2str(win_delta) 'hr_chuva_valida_janelinha.txt'], x_val, y_val)
escrever_dados([ num2str(win_delta) 'hr_chuva_teste_janelinha.txt'], x, y)

disp ('==> End');
```

C PROGRAMA RELATÓRIO DE RESULTADOS

Neste apêndice se apresentam o código, em script Matlab, utilizado para o programas gerador dos relatórios de resultados. Este programa gera os relatórios, gráficos e comparações, conforme descritas em 3.6

```
%% Metricas dos Resultados
clc;
clear;
disp ('==> Start');
%% Parametres
win_delta = 24 * 1; % hours * day
rel_level = 2; % relevancia level
%% Load Test Data
P_tst = dlmread('24hr_chuva_teste_janelinha.txt', ' ', 1);
P_tst = P_tst(:,87);
[l_tst c_tst] = size (P_tst);
%% Dist % & Plot Histogram
x = P_tst;
disp ({'==> Data Test(Real)'});
tabulate(x)
figure
hist(x)
title('Data Test (Real) Histogram');
%% Load Result Data
load('out_SVM_1_1.mat');
P_res = yf;
%% Dist % & Plot Histogram
x = P_res;
disp ({'==> Result (Predict) Test'});
tabulate(x)
figure
hist(x)
title('Data Result (Predict) Histogram');
%% Plot Histogram
y = [P_tst P_res];
figure
hist(y)
title('Real X Predict');
%% Compare Result
acertos = (P_res == P_tst);
tx_acertos= sum (acertos)/l_tst;
cl1_acertos = (P_tst == P_res == 1);
tx_cl1 = sum(cl1_acertos)/l_tst
%% Create Xls file
Tmatriz = [array2table(P_res) array2table(P_tst)];
writetable (Tmatriz,'Matriz.xls','Sheet','Dados');
```

```
%% Plot Confusion Matriz
targetsVector = P_tst'; % True classes
outputsVector = P_res'; % Predicted classes
% Convert this data to a [numClasses x 6] matrix
[lin col] = size (targetsVector);
[lin num_class] = size (unique(targetsVector));
targets = zeros(num_class,col);
outputs = zeros(num_class,col);
targetsIdx = sub2ind(size(targets), targetsVector, 1:col);
outputsIdx = sub2ind(size(outputs), outputsVector, 1:col);
targets(targetsIdx) = 1;
outputs(outputsIdx) = 1;
% Plot the confusion matrix for a 3-class problem
figure
plotconfusion(targets,outputs)
```