



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ  
CAMPUS LUIZ MENEGHEL - CENTRO DE CIÊNCIAS TECNOLÓGICAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS ROBERTO GARCIA JUNIOR

**PREDIZENDO A DEPOSIÇÃO DE PRODUTOS  
FITOFARMACÊUTICOS COM APRENDIZADO DE  
MÁQUINA**

**BANDEIRANTES-PR**

**2017**



CARLOS ROBERTO GARCIA JUNIOR

**PREDIZENDO A DEPOSIÇÃO DE PRODUTOS  
FITOFARMACÊUTICOS COM APRENDIZADO DE  
MÁQUINA**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Norte do Paraná para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Bruno Squizzato Faiçal

**BANDEIRANTES-PR**

**2017**

CARLOS ROBERTO GARCIA JUNIOR

**PREDIZENDO A DEPOSIÇÃO DE PRODUTOS  
FITOFARMACÊUTICOS COM APRENDIZADO DE  
MÁQUINA**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual do Norte do Paraná para obtenção do título de Bacharel em Ciência da Computação.

**BANCA EXAMINADORA**

---

Prof. Dr. Bruno Squizzato Faiçal  
Universidade Estadual do Norte do Paraná  
Orientador

---

Prof. Me. Glauco Carlos Silva  
Universidade Estadual do Norte do Paraná

---

Prof. Me. Wellington Aparecido Della Mura  
Universidade Estadual do Norte do Paraná

Bandeirantes-PR, 12 de dezembro de 2017

GARCIA, C. R.. **Predizendo a deposição de produtos fitofarmacêuticos com Aprendizado de Máquina**. 88 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual do Norte do Paraná, Bandeirantes–PR, 2017.

## RESUMO

O crescimento populacional impõe diversos desafios, entre os quais, aumentar a produção e fornecimento de alimentos. Para superar tal obstáculo é essencial a produção agrícola em grande escala. Além disso, a agricultura é uma área estratégica na economia de diversos países, impulsionada pelo aumento da produção. Nesse cenário, uma prática comum é o manejo de proteção com a pulverização de produtos fitofarmacêuticos a fim de resguardar a cultura da ação danosa de pragas agrícolas. Durante essa atividade é comum que ocorra deriva do produto pulverizado para fora da região alvo, contaminando regiões ambientais ou causando deposição excessiva em subáreas agrícolas devido a sobreposição do produto. A possibilidade do elemento pulverizador adaptar-se às condições meteorológicas é vista como uma alternativa para reduzir a deriva do produto pulverizado, aumentando a sua precisão na deposição do produto e, consecutivamente, proporcionando um ambiente mais adequado a cultura em desenvolvimento. Para isso, é necessário que o elemento pulverizador possa realizar a predição da deposição do produto em tempo de execução. Contudo, as abordagens atuais possuem alto custo computacional, fazendo com que este problema requirite um tempo exagerado para seu processamento. Nesse sentido, este trabalho propõe sistema como prova de conceito para uma nova abordagem baseada em aprendizado de máquina para a predição do produto pulverizado com intuito de representar de forma semelhante o comportamento real da deposição. As técnicas de rede neural artificial, árvore de regressão, máquinas de vetores de suporte e florestas aleatórias foram implantadas com tipos de configurações de parâmetros diferentes. Os resultados obtidos demonstram que tal abordagem permite alcançar uma representação satisfatória da deposição real e estimula novos estudos a fim de comparar o custo computacional da abordagem proposta com outras existentes.

**Palavras-chave:** Aprendizado de Máquina, Produtos Fitofarmacêuticos, Deposição e Pulverização.



GARCIA, C. R.. **Predicting the deposition of plant protection products with Machine Learning**. 88 p. Final Project (Bachelor of Science in Computer Science) – State University Northern of Paraná , Bandeirantes–PR, 2017.

## ABSTRACT

Population growth poses a number of challenges, including increasing food production and supply. To overcome such an obstacle, large-scale agricultural production is essential. In addition, agriculture is a strategic area in the economy of several countries, driven by increased production. In this scenario, a common practice is the management of protection with the spraying of plant protection products in order to protect the crop from the harmful action of agricultural pests. During this activity, it is common for the product to drift out of the target region, contaminating environmental regions or causing excessive deposition in agricultural subareas due to overlapping of the product. The possibility of the sprayer element adapting to the weather conditions is seen as an alternative to reduce the drift of the sprayed product, increasing its accuracy in the deposition of the product and, consequently, providing a more suitable environment for the growing crop. For this, it is necessary that the sprayer element can perform the prediction of the deposition of the product at run time. However, the current approaches have a high computational cost, causing this problem to require an exaggerated time for its processing. In this sense, this work proposes a system as proof of concept for a new approach based on machine learning for the prediction of the pulverized product in order to represent in a similar way the real behavior of the deposition. The techniques of artificial neural network, regression tree, support vector machines and random forests were deployed with different types of parameter settings. The results show that this approach allows to achieve a satisfactory representation of the actual deposition and stimulates new studies in order to compare the computational cost of the proposed approach with other existing ones.

**Keywords:** Machine Learning, Phytopharmaceutical Products, Deposition and Pulverization.





## LISTA DE ILUSTRAÇÕES

Figura 1 – Hierarquia de aprendizado de máquina. Adaptada de [FACELI et al., 2011]. . . . .	19
Figura 2 – Aproximação de uma função de valor real no conjunto de dados. Adaptada de [SUGIYAMA, 2015]. . . . .	20
Figura 3 – Arquitetura da RNA. Adaptada de [FACELI et al., 2011]. . . . .	22
Figura 4 – Arquitetura da rede perceptron simulando o fluxo de sinal. Adaptada de [HAYKIN, 2001]. . . . .	24
Figura 5 – Modelo de rede <i>feedforward</i> . . . . .	25
Figura 6 – Modelo de árvore de decisão e as áreas de decisão no espaço. Adaptada de [FACELI et al., 2011]. . . . .	28
Figura 7 – Exemplos de um conjunto de dados linearmente separável e não-linearmente separável. . . . .	35
Figura 8 – O modo simplificado do procedimento realizado por SVR. Adaptada de [FACELI et al., 2011]. . . . .	37
Figura 9 – Funcionamento do <i>random forests</i> . Adaptada de [GÓMEZ, 2012]. . . . .	41
Figura 10 – Variáveis do ambiente podem influenciar na deposição do produto pulverizado. Além destas, características do produto e configurações do sistema de pulverização também são fatores que influenciam na precisão da pulverização. . . . .	49
Figura 11 – Principais diferenças entre as abordagens existentes e a proposta neste trabalho para a previsão de deposição. . . . .	50
Figura 12 – Estrutura do sistema MoDeL e as respectivas interações dos seus componentes. . . . .	51
Figura 13 – Etapas realizadas no treinamento da técnica de Aprendizado de Máquina. . . . .	52
Figura 14 – Funcionamento do <i>cross-validation</i> , em 10 iterações de testes. . . . .	58
Figura 15 – Predições realizadas pelo algoritmo RNA na configuração empírica. . . . .	63
Figura 16 – Predições realizadas pelo algoritmo AR na configuração guiada. . . . .	65
Figura 17 – Variabilidade do algoritmo RNA na configuração empírica. . . . .	83
Figura 18 – Variabilidade do algoritmo AR na configuração guiada. . . . .	83
Figura 19 – Predições realizadas por cada algoritmo na configuração empírica. . . . .	85
Figura 20 – Variabilidade de cada algoritmo na configuração empírica. . . . .	86
Figura 21 – Predições realizadas por cada algoritmo na configuração guiada. . . . .	87
Figura 22 – Variabilidade de cada algoritmo na configuração guiada. . . . .	88



## LISTA DE TABELAS

Tabela 1 – Funções kernel . . . . .	36
Tabela 2 – Atributos da base de dados. . . . .	53
Tabela 3 – Intervalos de depósito. . . . .	54
Tabela 4 – Resultados alcançados utilizando a configuração empírica. . . . .	62
Tabela 5 – Teste estatístico <i>wilcoxon</i> na configuração empírica. . . . .	62
Tabela 6 – Resultados alcançados utilizando a configuração guiada. . . . .	64
Tabela 7 – Teste estatístico <i>wilcoxon</i> na configuração guiada. . . . .	64
Tabela 8 – Parâmetros empregados no algoritmo RNA. . . . .	77
Tabela 9 – Parâmetros empregados no algoritmo AR. . . . .	77
Tabela 10 – Parâmetros empregados no algoritmo SVM. . . . .	77
Tabela 11 – Parâmetros empregados no algoritmo FA. . . . .	77
Tabela 12 – Intervalos dos parâmetros aplicado na RNA. . . . .	79
Tabela 13 – Intervalos dos parâmetros aplicado na AR. . . . .	79
Tabela 14 – Intervalos dos parâmetros aplicado na SVM. . . . .	79
Tabela 15 – Intervalos dos parâmetros aplicado na FA. . . . .	80
Tabela 16 – Parâmetros empregados no algoritmo RNA. . . . .	81
Tabela 17 – Parâmetros empregados no algoritmo AR. . . . .	81
Tabela 18 – Parâmetros empregados no algoritmo SVM. . . . .	81
Tabela 19 – Parâmetros empregados no algoritmo FA. . . . .	82



## LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
AR	Árvore de Regressão
Com	<i>Communication</i>
FA	Florestas Aleatórias
GPS	Sistema de Posicionamento Global
KKT	<i>Krausk-Kuhn-Tucker</i>
MAD	<i>Mean Absolute Distance</i>
MSE	<i>Mean Squared Error</i>
MoDeL	<i>Model of <b>D</b>eposition in Machine <b>L</b>earning-Based</i>
Pp	<i>Preprocessing</i>
RNA	Rede Neural Artificial
SIG	Sistemas de Informação Geográfica
SVMs	<i>Support Vector Machines</i>
SVR	<i>Support Vector Regression</i>
TCC	Trabalho de Conclusão de Curso
TML	<i>Technique of <b>M</b>achine <b>L</b>earning</i>
VC	<i>Vapnik e Chervonenkis</i>



## SUMÁRIO

1	INTRODUÇÃO . . . . .	15
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	17
2.1	Aprendizado de Máquina . . . . .	17
2.1.1	Redes Neurais Artificiais . . . . .	21
2.1.2	Árvore de Regressão . . . . .	27
2.1.3	Máquinas de Vetores de Suporte . . . . .	31
2.1.4	Florestas Aleatórias . . . . .	38
2.2	Agricultura de Precisão . . . . .	42
3	TRABALHOS RELACIONADOS . . . . .	47
4	PROPOSTA . . . . .	49
5	METODOLOGIA . . . . .	53
5.1	Implementação do componente TML . . . . .	54
5.1.1	Configuração Empírica . . . . .	55
5.1.2	Configuração Guiada . . . . .	56
5.2	<i>Cross-Validation</i> . . . . .	58
5.3	Métricas de erro para regressão . . . . .	59
6	RESULTADOS E ANÁLISE . . . . .	61
6.1	Resultados da configuração empírica . . . . .	61
6.2	Resultados da configuração guiada . . . . .	63
7	CONCLUSÃO E TRABALHOS FUTUROS . . . . .	67
	REFERÊNCIAS . . . . .	69
	APÊNDICES . . . . .	75
	APÊNDICE A – PARÂMETROS DOS ALGORITMOS NA CONFIRURAÇÃO EMPÍRICA . . . . .	77
	APÊNDICE B – INTERVALOS DE BUSCA CONSIDERA- DOS PELA TÉCNICA <i>GRID-SEARCH</i> . . . . .	79

APÊNDICE C – PARÂMETROS DOS ALGORITMOS NA CONFIGURAÇÃO GUIADA . . . . .	81
APÊNDICE D – VARIABILIDADE DOS ALGORITMOS EM DIFERENTES CONFIGURAÇÕES . . . . .	83
APÊNDICE E – PREDIÇÕES E VARIABILIDADES DOS ALGORITMOS NA CONFIGURAÇÃO EMPÍRICA E GUIADA . . . . .	85



# 1 INTRODUÇÃO

Devido ao crescente aumento da população mundial, muitos desafios surgem no sentido de melhorar a qualidade de vida das pessoas, tais como: questões relacionadas à saúde, segurança e mobilidade. Entre os diversos desafios destaca-se a necessidade de produção e fornecimento de alimentos [JOHNSON, 2016]. Uma das alternativas é tornar as áreas de cultivo agrícola mais produtivas. Nesse sentido, produtos fitofarmacêuticos (também conhecidos como agrotóxicos) são empregados no processo de manejo agrícola com o objetivo de proporcionar um ambiente mais propenso ao desenvolvimento de culturas [DORNELLES et al., 2011].

No entanto, é importante ressaltar que o uso inadequado desses produtos pode causar sérios danos ao meio ambiente e a saúde dos seres humanos, como por exemplo doenças neurológicas, problemas respiratórios e cânceres [WEISENBURGER, 1993]. Estima-se que, cerca de 2,5 milhões de toneladas de produtos fitofarmacêuticos são aplicados na agricultura e há forte estimativa de crescimento de seu uso [FERREIRA; OLIVEIRA; PIETRO, 2009].

Tais produtos são aplicados na plantação por meio da pulverização em toda a área de cultivo. A pulverização agrícola pode ser definida como a ação de expelir/liberar produtos sobre uma cultura agrícola [FERREIRA; OLIVEIRA; PIETRO, 2009]. Normalmente, duas abordagens para a pulverização são amplamente investigadas na literatura científica, são elas: (i) terrestre e (ii) aérea [WANG et al., 2016; SAHA; PIPARIYA; BHADURI, 2016; GREGORIO et al., 2016; MINOV et al., 2016; JIAO et al., 2016; SALYANI; CROMWELL, 1992; GHATE; PERRY, 1994]. Tais métodos se diferenciam pelas características dos produtos empregados, o impacto ambiental causado, a efetividade dos equipamentos e os métodos de aplicação utilizados.

Apesar dos avanços científicos, em ambos os métodos de pulverização utilizados, o fator relacionado a deriva dos produtos fitofarmacêuticos para fora da região alvo permanece como um dos principais desafios encontrados na pulverização agrícola [SALYANI; CROMWELL, 1992; WANG et al., 2014; RU et al., 2014; DORUCHOWSKI et al., 2013; NUYTTENS et al., 2011], podendo ser influenciado por diversos fatores, como: distância, velocidade e volume pulverizado pelos equipamentos em relação a cultura. Além disso, questões relacionados a velocidade e direção do vento, umidade, temperatura do ambiente também exercem grande influência na precisão da aplicação.

Com o objetivo de reduzir os impactos causados pelo uso desses produtos e melhorar a precisão da pulverização, tem sido desenvolvido diversos trabalhos baseados em modelos matemáticos com o objetivo de estimar a deriva da pulverização [GRIFOLL;

ROSELL-LLOMPART, 2012; GUO; FLETCHER; LANGRISH, 2004; NIJDAM et al., 2006a; NIJDAM et al., 2006b]. No entanto, tais abordagens são computacionalmente custosas para serem executadas. Isso se deve ao fato desses modelos matemáticos realizarem cálculos complexos com diversas informações, tais como: (i) velocidade, (ii) direção do vento, (iii) umidade do ar, (iv) temperatura e (v) pressão atmosférica. Estes cálculos são realizados a cada instante de tempo para todas as gotas liberadas até que uma condição de finalização seja alcançada, como por exemplo a deposição do produto.

Assim, este trabalho faz o uso de técnicas de aprendizado de máquina com o objetivo de possibilitar uma rápida e eficiente predição da deposição de produtos fitofarmacêuticos.

Na realização do trabalho serão aplicados diversas técnicas de aprendizado de máquina, afim de comparar e determinar qual delas é melhor para estimar a o volume depositado na região alvo. Para cada técnica os parâmetros são configurados de duas formas diferentes, são elas: (i) empírica e (ii) guiada. O conceito *cross-validation* é utilizado para avaliar o protótipo implementado e possibilitar a análise de sua acurácia. É importante ressaltar que a análise realizada considera métricas de erro específicos para problemas de regressão, as quais consistem do (i) erro médio e o (ii) erro médio quadrático. Além dessas métricas outras medidas foram utilizadas, como a covariância e duas medidas estatísticas, sendo elas: (i) *Shapiro Wilk* e (ii) *Wilcoxon rank sum test*.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é apresentado os conceitos teóricos que são essenciais no desenvolvimento da proposta do modelo a qual será desenvolvido. A seção 2.1 é composta pela introdução sobre aprendizado de máquina, indução de hipóteses, viés indutivo, tarefas e hierarquia de aprendizado. A seção 2.1.1 apresenta a introdução sobre Rede Neurais Artificiais, arquitetura, aprendizagem e rede perceptron. A seção 2.1.2 faz uma introdução sobre Árvores de Regressão, indução da árvores de decisão e regressão, regras de divisão para regressão e estratégia de poda. A seção 2.1.3 demonstra os conceitos sobre Máquinas de Vetores de Suporte, além disso é introduzido a definição de *Support Vector Regression*. A seção 2.1.4 aborda os conceitos fundamentais sobre Florestas Aleatórias com foco na classificação e regressão. Por fim a seção 2.2 mostra os conceitos sobre à agricultura de precisão.

### 2.1 Aprendizado de Máquina

Com o passar dos anos, técnicas de inteligência artificial foram estudadas a fim de ajudar o ser humano na tomada de decisão em problemas reais. Algumas técnicas como: (i) Sistemas Especialista e (ii) Sistemas Baseados em Conhecimento são modelos que há necessidade de um especialista para adquirir conhecimento sobre o domínio da aplicação. Todo conhecimento do especialista é transformado em regras, a partir delas o sistema toma uma decisão. Contudo, essas técnicas tinham limitações na aquisição do conhecimento, pois depende do quanto o especialista compreende sobre o domínio em específico [FACELI et al., 2011].

Atualmente a quantidade de dados produzidos por diversas áreas houve um crescimento [BARBOSA, 2013], e com isso observou-se a dificuldade das técnicas na solução de problemas complexos. Com a necessidade de criar uma nova ferramenta que pudesse tomar decisões por conta e não dependesse de pessoas. Essa ferramenta deveria resolver um problema a partir dos seguintes itens, são eles: (i) experiência anteriores, (ii) por uma hipótese ou (iii) função [FACELI et al., 2011].

O conceito fundamental do aprendizado de máquina é sua indução que deve ser feita a partir de conhecimentos anteriores. AM para Mitchell [1997] “A capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência”. Seguindo esse raciocínio Real e Nicoletti [2014] descreve como uma subárea da inteligência artificial que pesquisa o desenvolvimento de formalismos e técnicas na qual possibilita a criação de sistemas automáticos de aprendizado.

Na construção do conhecimento, computadores são manipulados para aprender

sobre um determinado assunto com experiência passada. O princípio de inferência feito pelo AM é indutivo, o qual tira conclusões sobre um conjunto de dados. Algoritmos de AM podem induzir uma função ou hipótese que será capaz de solucionar um problema com informações específica. Essas informações devem representar um conjunto de dados que será instâncias para o problema a ser solucionado [FACELI et al., 2011; MONARD; BARANAUSKAS, 2003].

Na indução da hipótese, um grupo de dados é necessário para que o AM possa fazer a indução. Esse grupo de dados pode conter diversos atributos (conhecido como variáveis), demonstra as informações dentro dele. Na realização da indução um atributo é escolhido para ser o atributo de saída (conhecido como atributo alvo ou atributo meta), para estimar seu valor ele usa outros valores de atributo de entrada (conhecido como atributos previsoires) [FACELI et al., 2011].

Alguns grupos de dados não apresenta seus dados totalmente perfeito, sendo assim o algoritmo de AM deve ser capaz de suportar dados imperfeitos. Tipos de problemas encontrado em dados são: presença de ruídos, dados inconsistentes, dados ausentes e dados redundantes. Esses problemas pode afetar na indução do AM, técnicas de pré-processamento de dados é necessário, pois ajuda na diminuição de problemas. Com isso, é desejável que ao induzir uma hipótese ela seja capaz de induzir novamente para novos dados do mesmo domínio. Quando isto ocorre é chamado de generalização de hipótese, porém, uma hipótese tem que apresentar uma boa generalização [FACELI et al., 2011].

O *overfitting*, é quando a hipótese tem baixa capacidade de generalização, pode ser o superajustamento dos dados de treinamento. Isso acontece porque a hipótese memorizou ou especializou no conjunto de dado de treinamento. Já o *underfitting*, quando a indução teve baixa taxa de precisão no subconjunto de treinamento, é considerado subajustamento. Nessa caso, houve pouca amostra de dados para que o sistema de aprendizado pode-se induzir de forma ainda mais precisa [MONARD; BARANAUSKAS, 2003].

Aprendizado de máquina em seu aprendizado busca por um conjuntos de dados, que é chamado de conjunto de treinamento. Procurando assim uma hipótese, onde se adapta melhor aos objetos, e escolhe um conjunto de dados que mais adéque a hipótese. Hoje em dia existem vários tipos de algoritmos de AM, cada um apresenta formas diferentes de representar a hipótese induzida [FACELI et al., 2011].

A representação de uma hipótese induzida na literatura é chamada de viés(*bias*), é como o algoritmos de AM será induzido a partir da representação de um conjunto de hipótese. O AM também utiliza o conceito de viés de busca, é a forma que o algoritmo busca as hipóteses no conjunto de hipóteses, tentando achar o melhor conjunto de dados de treinamento que se ajusta a hipótese.

Os viés citados acima, é necessário para que o AM consiga obter aprendizado

e generalização [FACELI et al., 2011]. Sendo assim, algoritmos de AM sem viés não adquire a generalização para o conhecimento aprendido na fase de treinamento, com isso, o algoritmo tende a ter dificuldades a novos dados [MITCHELL, 1997].

Uso de algoritmos de AM tem sido utilizado frequentemente em diversas tarefas, pode ser organizadas por diferentes tipos de regras. Uma dessas regras diz a respeito do paradigma de aprendizado que ajuda a lidar com a tarefa. A regra da tarefa de aprendizado são divididas em duas partes: (i) Preditivas e (ii) Descritivas.

A hierarquia de aprendizado de máquina é representada na Figura 1, mostra a divisão das tarefas e seus paradigmas.

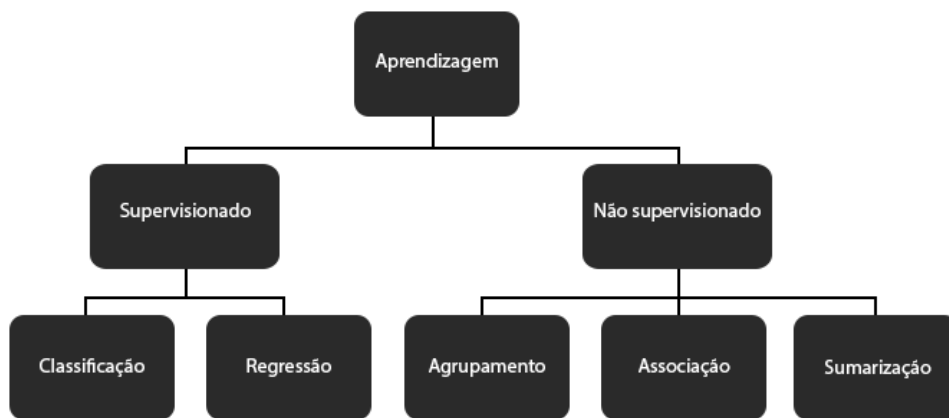


Figura 1 – Hierarquia de aprendizado de máquina. Adaptada de [FACELI et al., 2011].

Tarefas preditivas tem sua divisão com os seguintes itens, são eles: (i) Classificação e (ii) Regressão. No caso descritivas a divisão é: (i) Agrupamento, (ii) Associação e (iii) Sumarização.

Existem modelos de paradigmas diferentes para cada tarefa, modelo supervisionado para preditivas, modelo não supervisionado para descritivas e também á um outro modelo chamado semisupervisionado, está no meio termo do modelo supervisionado e não supervisionado. Todos conceitos apresentados anteriormente serão discutido detalhadamente a seguir.

Tarefas Preditivas é procurar uma função (conhecido como modelo ou hipótese), utilizando um conjunto de dados para prever um rótulo ou valor, na criação de novos exemplos, a partir dos valores de atributo de entrada. No conjunto de treinamento cada objeto (instância) deve conter um atributo de entrada e saída [FACELI et al., 2011]. Algoritmos de AM que usa esse tipo de tarefa ele tem sua indução de moledo preditivo. Então esse modelo segue o paradigma de aprendizado supervisionado [FACELI et al., 2011].

Tipos de tarefas que seguem o modelo de aprendizagem supervisionado é classificação e regressão. Na abordagem proposta o modelo de regressão será utilizado, pois o

estudo envolvido está relacionado a esse conceito. Para leitura complementar no caso de classificação pode ser encontrado em [NORVIG; RUSSELL, 2014; FACELI et al., 2011].

O modelo de aprendizagem supervisionado para regressão o objetivo é aproximar uma função de valor real no conjunto de dados (informações) [SUGIYAMA, 2015; JAMES et al., 2013]. A Figura 2 ilustra uma função dada a entrada pelo vetor  $d$ -dimensional  $x$ , e sua saída na escala real  $y$ , e a função alvo (meta) de aprendizagem é  $y = f(x)$ .

A função alvo (meta) de aprendizagem  $f$  é adotada como desconhecida, porém são observadas pelo conjunto de dados pareadas de entrada e saída  $\{(x_i, y_i)\}_{i=1}^n$ . O valor de saída observado  $y_i$  na prática pode acontecer que o valor esteja com algum ruído  $\epsilon_i$ , isto é,  $y_i = f(x_i) + \epsilon_i$ . Nessa configuração o  $x_i$  é correspondida a um questionamento que um aluno faz ao supervisor e  $y_i$  representa uma solução ao questionamento feito pelo aluno, na qual o supervisor oferece ao aluno. O ruído  $\epsilon_i$  corresponde ao erro do supervisor ou ao não entendimento do aluno em outras palavras ele não abstraio o que o supervisor queria dizer. A função alvo (meta) de aprendizagem  $f$  corresponde a inteligência do supervisor, o que lhe permite encontrar uma solução a quaisquer questionamento. O objetivo da regressão é fazer com que o aluno aprenda essa função, pelo qual também seja capaz de encontrar uma solução a quaisquer questionamento. O nível de generalização é calculado pela proximidade entre a função verdadeira  $f$  e sua aproximação  $\hat{f}$  [SUGIYAMA, 2015].

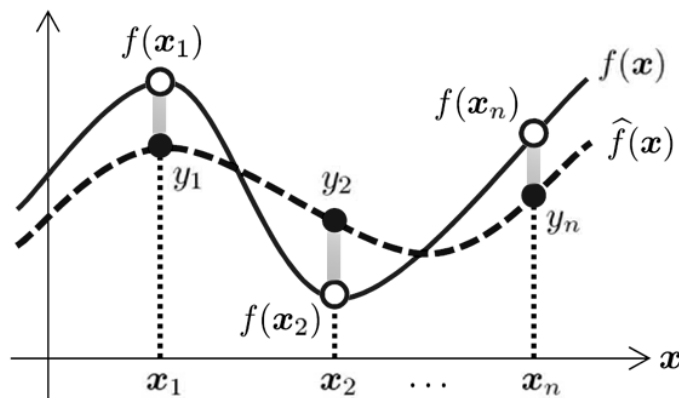


Figura 2 – Aproximação de uma função de valor real no conjunto de dados. Adaptada de [SUGIYAMA, 2015].

O paradigma supervisionado de acordo com Faceli et al. [2011] pode ser vista como uma simulação de um supervisor que conhece a saída, gerando um rótulo. Esse supervisor pode ser usado para avaliar a aptidão em induzir uma hipótese em novos exemplos.

Para Prati [2006] o paradigma supervisionado é observada como um conjunto de dado que são observações ou exemplos rotulados, essas observações também é conhecida como atributo meta ou atributo alvo. Partir desse atributo o objetivo é descobrir uma hipótese capaz de prever novas observações para o conjunto de dado já existentes.

Tarefas Descritivas (descrição), o objetivo é descobrir ou descrever um conjunto de

dado (informação). Os atributos de saída não são utilizados nos algoritmos de AM nessa tarefa. Por não usar os atributos de saída essa tarefa segue o paradigma de aprendizado não supervisionado [FACELI et al., 2011].

Tipos de tarefas que seguem o modelo de aprendizagem não supervisionado é agrupamento, associação e sumarização, esse conceito não será abordado no estudo e para leitura complementar pode ser encontrado em [NORVIG; RUSSELL, 2014; FACELI et al., 2011].

No paradigma não supervisionado a aprendizagem é partir de um conjunto de observações ou exemplos não rotulados, tem por meta encontrar similaridades entre grupos no conjunto de observações [PRATI, 2006; FACELI et al., 2011].

O paradigma semisupervisionado segundo Prati [2006], Faceli et al. [2011] o objetivo é usar dois conjuntos de dados para achar uma hipótese que seja capaz de classificar novos dados. Esses conjuntos de dados devem ser exemplos de dados rotulados e não rotulados. Os tipos de aprendizagem que fazem parte do paradigma semisupervisionado é agrupamento e classificação/regressão. O agrupamento é usado quando os exemplos de dados são não rotulados, já a classificação/regressão utilizado nos exemplos de dados que tenha rótulo.

### 2.1.1 Redes Neurais Artificiais

A rede neural artificial conhecida popularmente como rede neural, onde sua construção arquitetural é baseada no cérebro humano, na qual ele conhece agir de maneira inteligente. O cérebro humano pode ser visto na computação como um computador que processa dados complexo, não linear e paralelo. Na estrutura biológica do cérebro o componente principal encontrado ele são os neurônios. A forma como ele faz os processamentos são considerados o mais rápido que um computador digital atualmente [HAYKIN, 2001; FACELI et al., 2011].

Na década 1940 teve o início pela procura em desenvolver um sistema computacional que pode-se simular o sistema nervoso, o objetivo devia ser realizar ações do seres humanos como: a capacidade em aprender e adquirir conhecimento [FACELI et al., 2011].

Os primeiros estudos foram realizado por McCulloch e Pitts em 1943, suas proposta eram um modelo matemático de neurônio artificial. O conceito deles é executar funções lógicas simples, quem efetua essas funções são os neurônios. Cada neurônio podia executar uma função diferente dos outros neurônios. Hebb e Rosenblatt também colaboraram com importantes estudos, Hebb em 1949 contribuído com o estudo sobre o aprendizado da RNA. Já Rosenblatt em 1958 introduziu a teoria dos perceptrons, umas das redes mais conhecida na literatura [HAYKIN, 2001; FACELI et al., 2011].

Segundo Haykin [2001], Faceli et al. [2011] define a RNA como sistemas computa-

cionais distribuídos constituído de unidades de processamento simples, que são interconectadas. As unidades é chamada de neurônio artificial, e elas tem uma ou mais camadas no qual são interligadas por várias conexões na maioria das vezes unidirecionais. A unidade é quem computa funções matemática e as conexões geralmente possui pesos atrelado a elas. Os pesos podem ser números positivos ou negativos. No processo de aprendizagem os pesos são ajustados e codificam o conhecimento adquirido pela rede.

Um neurônio é a unidade de processamento de dados que é fundamental na operação da RNA [HAYKIN, 2001; FACELI et al., 2011]. Na Figura 3 é ilustrado uma arquitetura de um modelo simples de RNA.

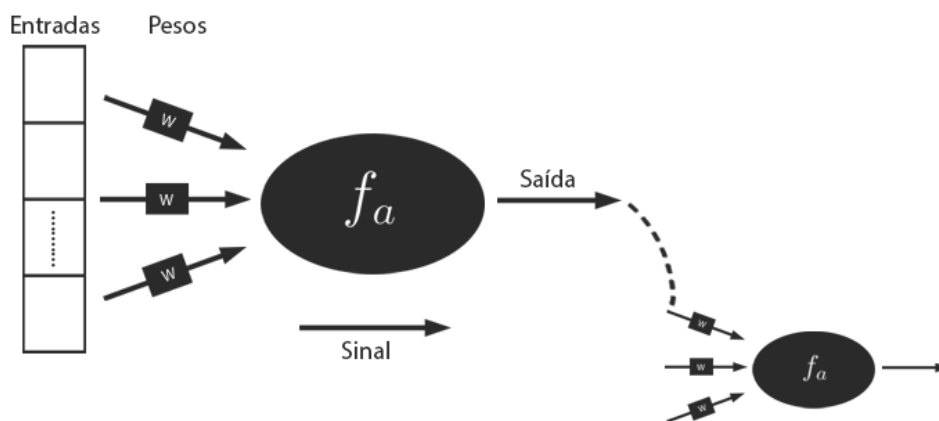


Figura 3 – Arquitetura da RNA. Adaptada de [FACELI et al., 2011].

Os neurônios (unidades de processamento) apresenta um comportamento muito simples. Todo terminal de entrada no neurônio recebe um valor que pode pertencer a um intervalo de valores positivos e negativos. Esses valores obtidos são ponderados e combinados, através de uma função matemática  $f_a$ , correspondendo ao processamento feito pelo soma. Está função gera uma saída que corresponde a resposta do neurônio para a entrada. Diferentes tipos de funções podem ser utilizadas [HAYKIN, 2001; FACELI et al., 2011].

Para especificar melhor a representação das funções, primeiramente devemos supor um conjunto de informações (dados)  $y$  com  $d$  sendo os atributos mostrado na forma de vetor como  $y = [y_1, y_2, \dots, y_d]^t$  e um neurônio com  $d$  terminais de entrada e seus pesos são  $w = [w_1, w_2, \dots, w_d]$ . Após isso podemos descrever o neurônio  $u$  sendo o total de entrada recebidas, em termos matemáticos pode ser definida pela equação 2.1, segundo Haykin [2001], Faceli et al. [2011].

$$u = \sum_{j=1}^d y_j w_j \quad (2.1)$$

Como dito anteriormente os neurônios pode ter conexões de entrada negativas



$w_j < 0$  ou positivas  $w_j > 0$ . Quando o valor do peso for igual a zero representa a inexistência da conexão associada [FACELI et al., 2011].

A saída de um neurônio é feita pela execução da função de ativação à entrada total [HAYKIN, 2001]. Existem várias funções de ativação propostas na literatura, os três tipos básicos de funções de ativação é apresentado a seguir:

1. Função linear: A função linear é a função de ativação mais simples, já que o valor retornado da função de saída é um número linear [HAYKIN, 2001; FACELI et al., 2011].
2. Função limiar: A função limiar é baseada no conceito do neurônio artificial de McCulloch e Pitts, no qual o resultado da função pode assumir 1 ou 0, isso ocorre a partir do valor do limiar. O valor 1 acontece se o neurônio induzido é não negativo, e 0 caso contrário [HAYKIN, 2001; FACELI et al., 2011].
3. Função sigmoidal: A função sigmoidal é mais comum função de ativação encontrada na construção de uma RNA. É definida como função estritamente crescente e exibe um balanceamento adequado entre comportamento linear e não-linear. Também à uma proximidade contínua e diferenciável da função limiar [HAYKIN, 2001; FACELI et al., 2011].

No aprendizado da RNA há vários algoritmos propostos na literatura para o ajuste dos parâmetros da RNA. O ajuste de parâmetros pode se entender como a escolha dos valores dos pesos relacionados às conexões da rede, eles fazem o modelo obter um melhor comportamento. O comportamento geralmente é medido pela acurácia preditiva da RNA. Esses algoritmos são chamados de algoritmo de treinamento, formados por um conjunto de regras específica para cada tipo de algoritmo, na qual indicam quando e como o valor de cada peso deve ser alterado [FACELI et al., 2011].

Vários autores apresentam formas diferentes para os algoritmos de treinamento da RNA, seguindo o paradigma de aprendizado. Esse trabalho têm como foco usar o paradigma supervisionado, em razão de apresentar um problema de regressão. A correção de erro é usada em aprendizado supervisionado, procuram diminuir os erros cometidos pela rede através do ajuste dos pesos da RNA [FACELI et al., 2011].

Em uma RNA existem alguns tipos de padrões para as conexões entre o neurônio, conforme isso a rede pode apresentar três padrão diferentes [FACELI et al., 2011; HAYKIN, 2001; LUGER, 2013], são eles:

1. Completamente conectada: quando os neurônios da rede encontram-se conectados a todos os neurônios da camada anterior ou seguinte, e também à possibilidade das duas camadas estarem conectadas.

2. Parcialmente conectada: quando os neurônios encontram-se conectados a apenas alguns neurônios da camada anterior ou seguinte, e também à possibilidade das duas camadas estarem conectadas.
3. Localmente conectada: é considerado redes parcialmente conectadas, quando os neurônios conectados a um neurônio se encontram em uma área bem definida.

A rede perceptron foi a primeira RNA a ser implementada, desenvolvida por Rosenblatt 1958 e dispõe do paradigma de aprendizagem supervisionada, além disso, a rede tinha uma única camada e ficou conhecida como perceptron. Essa rede utiliza o modelo de McCulloch-Pitts como neurônio, que é construída em torno de um neurônio não-linear. Por apresentar apenas uma camada de neurônio ela tem excelente acurácia preditiva em diferentes problemas de classificação [FACELI et al., 2011; HAYKIN, 2001; LUGER, 2013].

O modelo do perceptron constitui-se de um combinador linear seguido de um limitador abrupto (ele que realiza a função sinal). O nó adicional do modelo neural realiza o cálculo a partir da combinação linear das entradas colocadas no sinapses (pesos) e, também, possui um bias que é aplicado externamente [FACELI et al., 2011; HAYKIN, 2001].

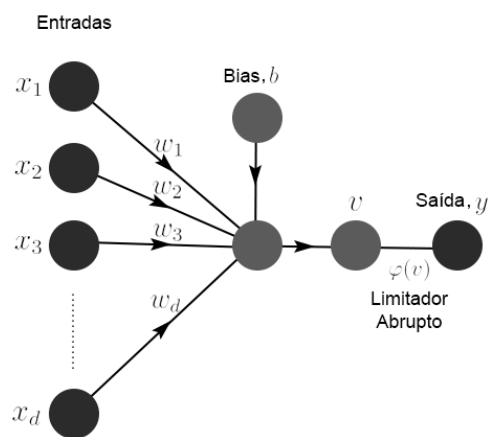


Figura 4 – Arquitetura da rede perceptron simulando o fluxo de sinal. Adaptada de [HAYKIN, 2001].

O resultado da somatória é o campo induzido, na qual é aplicado ao limitador abrupto. Proporcional, ao neurônio que gera uma saída igual a  $+1$  se a entrada do limitador abrupto é positiva e  $-1$  no caso negativa. Na Figura 4 os pesos da rede perceptron é exibido na forma de vetor  $w = [w_1, w_2, \dots, w_d]$ . As entradas atribuídas ao longo da rede perceptron é exibida matematicamente por  $x = [x_1, x_2, \dots, x_d]$ . O bias atribuído externamente é exibido por  $b$ . A equação 2.2 representa o campo induzido do neurônio [HAYKIN, 2001].

$$v = \sum_{i=1}^d w_i x_i + b \quad (2.2)$$

O propósito do perceptron é conseguir classificar de modo certo um conjunto de estímulos atribuídos externamente  $x = [x_1, x_2, \dots, x_d]$  em duas classes  $C_1$  ou  $C_2$ . Na classificação da rede utiliza-se a regra de decisão, na qual ele atribui o ponto exibida pelas entradas  $x = [x_1, x_2, \dots, x_d]$  à classe  $C_1$  quando a saída do perceptron  $y$  é  $+1$  e à classe  $C_2$  se ela for  $-1$  [HAYKIN, 2001].

Uma rede perceptron só consegue classificar objetos que são linearmente separáveis, como visto acima a classes  $C_1$  e  $C_2$  só pode ser considerada linearmente separáveis se houver um hiperplano que divide os dados das duas classes. Além disso os pesos  $w = [w_1, w_2, \dots, w_d]$  podem ser ajustados de interação para interação, e quem garante isto é a regra de correção de erro que é conhecida como teorema de convergência do perceptron [FACELI et al., 2011; HAYKIN, 2001].

A rede neural *feedforward* é considerada na literatura uma rede perceptron, na Figura 5 é ilustrado o modelo está rede.

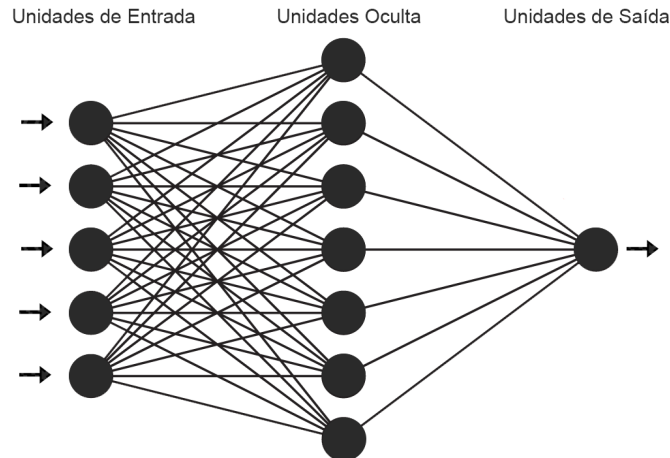


Figura 5 – Modelo de rede *feedforward*.

Uma rede neural *feedforward* sua arquitetura é dividida em três unidades (camadas):

1. Unidades de Entrada: Onde os valores são aplicados à rede.
2. Unidades Ocultas (Intermediária): onde é realizada a maioria do processamento, por meio das conexões ponderadas.
3. Unidades de Saída: onde o valor do resultado final é finalizado e exibido.

O algoritmo de treinamento da rede *feedforward* é o *back-propagation*, a função de ativação usada nesse algoritmo é do tipo sigmoidal por corresponder ao requisitos da função de ativação que precisa ser contínua, diferenciável e preferencialmente não decrescente [FACELI et al., 2011].

O algoritmo *back-propagation* usa o conceito de regra delta que é utilizada em RNA do tipo adaline, e também é chamado na literatura de regra delta generalizada. O *back-propagation* é composto da iteração de duas fases, a fase que vai para frente (*forward*) e outra fase que vai para trás (*backward*) [FACELI et al., 2011].

Na fase *forward*, todos objetos da entrada é mostrado à rede. Quando o objeto chega na rede ele é recebido por cada um dos neurônios da camada (unidade) intermediária (oculta) da rede, quando é ponderado pelo peso agregado a suas conexões de entrada correspondentes. Os neurônios dessa camada aplica a função de ativação para sua entrada total e gera um valor de saída, que é usado como valor de entrada para outros neurônios da camada (unidade) posterior. O processo deve continuar até que todos neurônios da camada de saída gerem o seu valor de saída, na qual é comparado ao valor pretendido para a saída desse neurônio. A diferença dos valores de saída gerados e pretendidos para cada um dos neurônios da camada de saída define o erro cometido pela rede para o objeto mostrado [FACELI et al., 2011].

Na fase *backward* ele utiliza o valor do erro de cada neurônio da camada de saída para acertar seus pesos de entrada. Esse acerto (ajuste) continua da camada de saída até a camada intermediária [FACELI et al., 2011]. A equação 2.3 mostra como é feito o acerto (ajuste) dos pesos de uma RNA através do algoritmo *back-propagation*.

$$w_{jl}(t + 1) = w_{jl}(t) + nx^j \delta_l \quad (2.3)$$

Na equação apresentada o  $w_{jl}$  corresponde o peso entre um neurônio  $l$  e o  $j$ -ésimo atributo de entrada ou da saída do  $j$ -ésimo neurônio da camada antecedente,  $\delta_l$  aponta o erro agregado ao  $l$ -ésimo neurônio e  $x^j$  mostra a entrada recebida por esse neurônio [FACELI et al., 2011].

Os erros da camada intermediária não é conhecido então é preciso estimar o seu valor, o algoritmo de *back-propagation* estima o erro desse neurônios usando os erros observados de outros neurônios da camada seguinte. O erro da camada intermediária (oculta) é estimado a partir da soma dos erros dos neurônio da camada posterior, onde os terminais de entrada (unidades de entrada) estão completamente conectados a ele, ponderados ao valor do peso agregado a essas conexões [FACELI et al., 2011].

Na equação 2.4 o  $e_l$  representa o erro quadrático realizado pela saída do neurônio quando sua resposta é comparada ao pretendido, de acordo com Faceli et al. [2011].

$$e_l = \frac{1}{2} \sum_{q=1}^k (y_q - \hat{f}_q)^2 \quad (2.4)$$

O acerto (ajuste) dos pesos é dada pela derivada parcial  $f'_a e_l$  usando o gradiente proveniente da função de ativação. A derivada mede a colaboração de cada peso no erro da rede para a classificação de um dado objeto. Derivando essa função dado um peso e se o resultado for positivo isso implica que o peso está causando um aumento na diferença entre a saída da rede e a saída pretendida. Dessa forma, a magnitude deve ser diminuída para baixar o erro. Se o resultado da derivada der negativo o peso está colaborando para que a saída gerada pela rede seja mais perto da saída pretendida. Assim, seu valor deve ser ampliado [FACELI et al., 2011].

### 2.1.2 Árvore de Regressão

Árvore de regressão é uma variação da árvore de decisão, os algoritmos de indução da duas são muito idênticas. Árvore de decisão utiliza o método de dividir para conquistar na resolução de problemas de decisão. Em problema complexo é feito a divisão em problemas mais simples, no qual utiliza o conceito de recursividade para aplicar a mesma técnica. Todas soluções gerada dos subproblemas podem ser combinadas, em formato de árvore, para alcançar uma solução em problemas complexos [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

Uma árvore de decisão pode ser representada matematicamente como um grafo acíclico direcionado, em outras palavras é grafo dirigido sem ciclo. Cada vértice não há nenhuma ligação iniciando e finalizando para ele mesmo [FERREIRA, 2013]. Um nó da árvore de decisão ou ele é nó de divisão com dois ou mais sucessores, ou um nó folha [FACELI et al., 2011; NORVIG; RUSSELL, 2014; FERREIRA, 1999].

Nó folha é designado por uma função. Os valores que chegam a este nó são da variável alvo. Há também caso que a função é uma constante que diminui a função de custo, quando isso ocorre é chamado de caso simples. No problema de regressão, para diminuir a função de custo do erro do médio quadrático é utilizado a média, para a para a função de custo do desvio absoluto adota-se a mediana [FACELI et al., 2011; FERREIRA, 1999].

Nó de divisão inclui um teste condicional baseado nos valores do atributo. Por padrão os teste são univariados: as condições contêm um único atributo e valores no domínio desse atributo [FACELI et al., 2011; FERREIRA, 1999].

Na Figura 6 é representado um modelo de árvore de decisão e sua divisão proporcional no espaço definido pelos atributos  $y_1$  e  $y_2$  [FACELI et al., 2011].

Os nós da árvore são representado por uma área nesse espaço. Essa área é definida

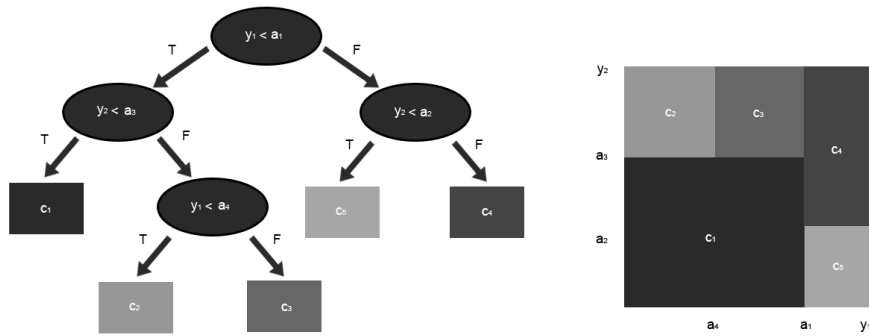


Figura 6 – Modelo de árvore de decisão e as áreas de decisão no espaço. Adaptada de [FACELI et al., 2011].

pelos nós da árvore são mutuamente exclusivos, e o agrupamento dessas áreas cobre todo espaço que é estabelecido pelos atributos [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

Árvore de decisão pode atingir todo o espaço de instâncias. Além disso uma árvore de decisão tem a capacidade de fazer previsões em qualquer conjunto de informações (dados). Isso é possível pois o fato de uma árvore de decisão conseguir abranger um dado espaço de instâncias [FACELI et al., 2011].

No espaço de hipótese de uma árvore de decisão pode-se incluir dentro do formalismo da forma normal disjuntiva. A forma normal disjuntiva na árvore de decisão são condições ao longo do ramo (filho) é o caminho entre a raiz e uma folha são conjunções de condições e os ramos específico é disjunções. Cada ramo na árvore forma uma regra que gera uma parte condicional e uma conclusão. Na parte condicional é representada por uma conjunção de condições, essas condições envolvem um atributo específico e um operador do tipo ( $>$ ,  $<$ ,  $=$  entre outros.) e um valor (número) do domínio do atributo [FACELI et al., 2011].

O algoritmo que constrói as árvores de decisão e regressão são parecidas, elas conseguem gerar uma árvore a partir de dados considerados simples. Um dos problemas mais conhecido na literatura na construção de uma árvore é o número de nós que quando condizente com o conjunto de dados isso considera-se um problema NP completo [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

Geralmente os algoritmos investigam heurísticas que localmente realizam uma pesquisa olha para a frente um passo. Quando tomado uma decisão ela não será mais reconsiderada. A pesquisa de subida de encosta do inglês *hill-climbing* sem *back-tracking* é vulnerável a riscos usuais de convergência a uma solução agradável localmente que não é agradável globalmente. Porém, esse método permite a geração de árvores de decisão em tempo linear no número de exemplares [FACELI et al., 2011].

Na regras de divisão existe um conceito fundamental chamado de ganho de informações, na qual utiliza o conceito de entropia que tem como base medir a aleatoriedade de uma variável aleatória. Árvore de decisão ou regressão usa a entropia para medir a aleatoriedade (complexabilidade em fazer predições) na variável alvo (meta). Todos nós de decisão, o atributo que mais reduz a aleatoriedade da variável meta é nomeado para dividir os dados [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

O ganho de informações mede a redução da entropia para cada atributo, a partir das partições adquirido de acordo com os valores do atributo. Na literatura o ganho de informações é calculado entre a diferença da entropia do conjunto de informações (dados) e a soma ponderada da entropia nas partições. A construção de uma árvore de decisão (regressão) é guiada pelo propósito de minimizar a entropia, que refere-se a complexidade em fazer predições da variável alvo (meta) [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

De acordo com Norvig e Russell [2014] o tipo de medida da entropia é *bits* utilizando o logaritmo na base 2. Normalmente a entropia de uma variável aleatória  $v$  com valores  $v = [v_1, v_2, v_3, \dots, v_k]$  cada um com probabilidade  $P(v_k)$ , é definida pela equação 2.5.

$$H(v) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = - \sum_k P(v_k) \log_2 P(v_k) \quad (2.5)$$

Segundo Norvig e Russell [2014] a equação 2.6 representa a entropia do atributo alvo em todo o conjunto de aprendizagem da árvore de decisão (regressão). Onde  $p$  representa os exemplos positivos e  $n$  exemplos negativos.

$$H(alvo) = B\left(\frac{p}{p+n}\right) \quad (2.6)$$

Geralmente é medido a entropia remanescente após testar o atributo. Para um melhor entendimento vamos supor um atributo  $a$  com valores  $a = [a_1, a_2, a_3, \dots, a_d]$  que divide um conjunto de treinamento  $e$  em subconjuntos  $e = [e_1, e_2, e_3, \dots, e_d]$ . Para cada subconjunto  $e_k$  tem  $p_k$  que representa os exemplos positivos e  $n_k$  os exemplos negativos. Se continuar ao longo dessa ramificação, precisamos de  $B(p_k/(p_k + n_k))$  *bits* complementares de informações para responder à pergunta. Os exemplos nomeado aleatoriamente do conjunto de treinamento tem seu valor  $k$ -ésimo do atributo, com probabilidade  $(p_k + n_k)/(p + n)$ , então a equação 2.7 refere-se a entropia remanescente após testar o atributo  $a$  [NORVIG; RUSSELL, 2014].

$$Resto(a) = \sum_{k=1}^d \frac{p_k + n_k}{p + n} B\left(\frac{p_k}{p_k + n_k}\right) \quad (2.7)$$

O ganho de informações do teste aplicado no atributo  $a$  pode ser visto na equação 2.8, representa a redução prevista na entropia [NORVIG; RUSSELL, 2014].

$$Ganho(a) = B\left(\frac{p}{p+n}\right) - \sum_{k=1}^d \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k + n_k}\right) \quad (2.8)$$

Nos problemas de regressão, a função de custo que reduz é, geralmente, o erro quadrático. Como citado no começo da seção sobre árvore de decisão, a média é a constante que reduz o erro quadrático. Mediante disso, a constante relacionada às folhas de uma árvore de regressão é a média dos valores do atributo alvo (meta) das amostras de treinamento que caem na folha [FACELI et al., 2011].

Na criação de uma árvore de regressão ela baseá-se na árvore de decisão e praticamente tudo é semelhante, o que diferencia é a função de custo referida. Em 1984 Breiman propôs uma métrica chamada SDR (*Standard Deriation Reduction*) para estimar o mérito da partição adquirido por um experimento no valor de uma variável. A equação 2.9 representa a variância da variável  $x$  em um conjunto de exemplares  $E$ , com  $n$  exemplos [FACELI et al., 2011].

$$sd(E, x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.9)$$

Considerando um teste teórico  $h_c$  sobre o atributo  $C$ , exemplificando  $C \leq c_1$ . O conjunto de dados (informações)  $E$  será dividido em dois subconjuntos,  $E_l$  e  $E_r$ , com tamanhos  $n_l$  e  $n_r$ , onde  $n = n_l + n_r$ . Dado  $x$  sendo a variável alvo (meta) sua variância em cada subconjunto  $E_l$  e  $E_r$  tende a ser sempre menor ou igual à variância de  $x$  anterior a divisão. Na equação 2.10 ilustra a função que estima a redução em variância adquirida pela execução do teste  $h_c$  [FACELI et al., 2011].

$$SDR(h_c) = sd(E, x) - \frac{n_l}{n} \times sd(E_l, x) - \frac{n_r}{n} \times sd(E_r, x) \quad (2.10)$$

Em um conjunto de atributo, para cada atributo e para cada realização de teste no valor do atributo, é feito o cálculo da redução da variância relacionada a esse teste. O teste que gera a maior redução em variância é nomeado como teste para o nó [FACELI et al., 2011].

Um desafio na construção da árvore de decisão (regressão) é os dados ruidosos que levantam dois problemas, são eles: (i) A indução na árvore é afeta, assim novos objetos serão classificado de maneira não confiável. As estatísticas predita nos nós mais profundos de uma árvore tem baixíssimo nível de relevância devido ao pouco número de exemplares que alcançam nesses nós. Os nós que estão mais profundo impacta no conjunto de treinamento (superajustamento) e crescem o erro à variância do classificador



e (ii) A árvore induzida geralmente tende a ser grande e por isto há uma dificuldade na compreensão [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

A forma de ajudar a resolver o problema de dados ruidosos na construção da árvore é podar ela, porém isso acarreta na classificação incorreta em pequeno número de exemplos do conjunto de treinamento. Uma vantagem da poda fica visível quando se classificam novos exemplos não utilizado no modo de criação da árvore. A poda geralmente contribuem com os erros de generalização menores [FACELI et al., 2011; NORVIG; RUSSELL, 2014]. Na literatura dois métodos de podas é proposto, o método pré-poda e pós-poda, esses métodos são apresentado a seguir:

1. Pré-poda: Esse método tem como base as regras de parada que intervêm na criação daqueles ramos que não contribuem na melhora da precisão preditiva pela árvore. A sua vantagem é o tempo não gasto na construção de uma estrutura que não será utilizada na árvore final [FACELI et al., 2011; NORVIG; RUSSELL, 2014].
2. Pós-poda: Esse método é usualmente utilizado na construção da árvore de decisão. A árvore é criada por completa, mesmo com o superajustamento dos dados de treinamento e depois será podada. O ponto chave de podar a árvore de decisão é saber quando podar uma subárvore [FACELI et al., 2011; NORVIG; RUSSELL, 2014].

### 2.1.3 Máquinas de Vetores de Suporte

No decorrer das décadas, o número de pesquisas e projetos que fazem o uso da técnica de *support vector machines* (SVMs) obteve um aumento e com isso a comunidade de aprendizado de máquina passou a lidar mais importância. Está técnica em alguns casos obtêm resultados que muitas vezes são superiores a outros algoritmos famosos de aprendizado, tal como a rede neural artificial [FACELI et al., 2011; SANTOS, 2002].

As máquinas de vetores de suporte é baseado pela teoria de aprendizado estatístico, criada por Vapnik em 1995. SVM apresenta duas características que sobressai, são elas: (i) Contém uma fundamentação teórica sólida e (ii) Alcança grande desempenho em aplicações práticas. A teoria de aprendizagem é capaz de reconhecer precisamente os aspectos na qual precisam ser considerados para a aprendizagem ser bem sucedida e desenvolver modelos que são muito complexos [SANTOS, 2002; BISHOP, 2006; JAMES et al., 2013].

Na fase de treinamento do SVM é utilizado uma função quadrática convexa, na qual se caracteriza um problema de otimização matemática. Os parâmetros do SVM é ajustado pelo usuário e também não há nenhuma dependência entre eles, esse ajuste são alterados dependendo do contexto em que será usado [SANTOS, 2002; KWOK, 1998].

O conceito da teoria de aprendizagem estatística no SVM tem por objetivo controlar, em definições matemáticas, é a capacidade de generalização da técnica. Na literatura existem diversas teorias que permite representar precisamente os fatores que têm necessidade de ser controlados na garantia de uma boa performance na generalização. Os princípios básicos empregue no SVM vêm da teoria de Vapnik e Chervonenkis (VC). Essa teoria é a mais propícia na descrição de um algoritmo SVM, além dela pode-se usar a interpretação bayesiana, entre outras [KWOK, 1998; SANTOS, 2002; BISHOP, 2006].

Considerando uma tarefa de aprendizagem, com um volume de informações (dados) de treinamento finita, a maior performance na generalização ocorre no momento em que é alcançado um equilíbrio entre a precisão concebida no conjunto de treinamento específico e a capacidade do sistema, isto é, a competência do sistema aprender em qualquer conjunto de treinamento sem erro [SANTOS, 2002].

A otimização é uma parte da matemática que inclui a caracterização de resultados para classes de problemas formados por funções que devem ser selecionados para minimizar ou maximizar uma certa função custo, e algumas restrições podem ser impostas. Otimização matemática da mesma forma se preocupa com a performance dos algoritmos na procura por tais soluções. Além disso a teoria de otimização oferece as técnicas algorítmicas e inclusive define requisitos necessários e eficazes para um função dada ser uma solução [SANTOS, 2002; FACELI et al., 2011].

As questões (problemas) normalmente iniciam a partir de um conjunto de variáveis ou parâmetros separados e regularmente acrescentam condições ou restrições que especificam valores aceitáveis das variáveis. Os resultados de uma questão de otimização é vista como um conjunto dessas variáveis que atendem as restrições (requisitos) e minimizam ou maximizam a função de custo [SANTOS, 2002].

Na literatura existem alguns tipos de restrições que pode ser impostas às funções, a seguir é apresentado duas definições relacionados as classes de problemas de otimização [SANTOS, 2002; FACELI et al., 2011]:

1. Otimização Irrestrita: Engloba os problemas (questões) em que as variáveis podem apropriá-se qualquer valor.
2. Otimização Restrita: Engloba a classe de problemas (questões) em que as variáveis podem apropriá-se valores retidos (condicionados) quer dizer, estão sujeitos a restrições. Essas restrições pode assumir dois tipos, são eles: (i) Igualdade, representam regularmente a operação do sistema que está sendo apontado e (ii) Desigualdade, determinam limites inferior e/ou superior das variáveis.

Considerando um problema geral de otimização dadas as funções  $f, g_i, i = 1, \dots, k$ , e  $h_i, i = 1, \dots, m$ , estabelecida em um domínio  $\Omega \subseteq \mathfrak{R}^n$ , onde a função de minimização

é  $f(w)$  tal que  $w \in \Omega$ , com restrições  $g_i(w) \leq 0, i = 1, \dots, k$  e  $h_i(w) = 0, i = 1, \dots, m$ . A função  $f(w)$  é denominado função objetivo e as outras relações é denominado, respectivamente, restrições de desigualdade e de igualdade. Quando o valor é ótimo a função objetivo é denominado valor do problema de otimização. É chamado região viável onde a função objetivo é estabelecido e suas restrições são satisfeitas [SANTOS, 2002].

A restrição de desigualdade  $g_i(w) \leq 0$  é ativa se um resultado  $w^*$  satisfaz  $g_i(w^*) = 0$  e inativa se  $g_i(w^*) < 0$ . Essa restrição é dita satisfeita se estiver ativa ou inativa e caso contrário é dita violada se  $g_i(w^*) > 0$  [SANTOS, 2002; FACELI et al., 2011].

Encontra-se também caso que é preciso transformar uma restrição de desigualdade para igualdade, quando isso ocorre é introduzido o conceito de variáveis de folga  $\xi$ . O exemplo a seguir demonstra como isso acontece, dada a função  $g_i(w) \leq 0 \Leftrightarrow g_i(w) + \xi_i = 0$ , com  $\xi_i \geq 0$ . Sempre que essas variáveis estiverem associadas com restrições ativas, serão iguais a zero e no caso de estarem associadas com restrições inativa especificam a quantidade de folga na restrição [SANTOS, 2002; FACELI et al., 2011].

Logo abaixo é exibido as definições existentes para otimização matemática por programação [SANTOS, 2002].

- Programação Linear: Um programa linear é considerado um problema de otimização quando a função objetivo e todas as outras funções de restrição são lineares.
- Programação Não-Linear: Quando a programação não-linear satisfaz os seguintes critérios, a função objetivo e todas as demais funções de restrição são não-lineares, então é considerado um problema de otimização.
- Programação Quadrática: É considerado problema de otimização quando a função objetivo é quadrática e as funções de restrição é linear.
- Programação Convexa: Engloba uma função convexa minimizada sobre um conjunto de restrições. De acordo com Santos [2002] o principal teorema dessa programação é: “Qualquer mínimo local de um problema de programação convexa é um mínimo global”.
- Programação Quadrática Convexa: Um problema de otimização na qual a função objetivo deve ser convexa e quadrática. Esse tipo de programação é utilizado pelo SVM, pois suas restrições são lineares e a função objetivo é convexa e quadrática.

Na realização da busca de máximos e mínimos condicionados, em situações de problemas quadráticos convexos e de outros problemas de otimização matemática, são usado o método de Lagrange [FACELI et al., 2011; BISHOP, 2006; SANTOS, 2002].

O objetivo do teorema de Lagrange é solucionar o problema de otimização em condições quando não há restrições de desigualdade. O método desenvolvido em 1797 por

Lagrange teve sua versão atualizada em 1951 por Kuhn e Tucker e passou a permitir restrições de desigualdade [SANTOS, 2002; FACELI et al., 2011; GONÇALVES, 2015].

A teoria de Kuhn e Tucker por ser representado no seguinte problema, dado a função minimizadora  $f(w)$ , tal que  $w \in \Omega$  e o problema de otimização pertence ao domínio convexo  $\Omega \subseteq \mathbb{R}^n$  com  $f \in C^1$  convexa. As funções  $g_i(w) \leq 0, i = 1, \dots, k$  e  $h_i(w) = 0, i = 1, \dots, m$  são as condições de restrições. Essas restrições são chamadas de funções afins e apresenta a seguinte forma  $f(x) = kx + p$ , onde  $k$  é um valor fixo (constante) que proporciona o declive da reta e  $p$  é ordenada na origem ou também pode ser um ponto da interseção na reta com o eixo das ordenadas [SANTOS, 2002; FACELI et al., 2011; GONÇALVES, 2015].

As condições fundamentais e suficientes dado um ponto  $w^*$  para representar o ponto excelente são a existência de  $\alpha^*, \beta^*$  (atribuindo o multiplicador de Lagrange em cada restrição) tem-se a seguinte equação 2.11, [SANTOS, 2002; FACELI et al., 2011; GONÇALVES, 2015].

$$\begin{aligned} \frac{\partial L(w^*, \alpha^* \beta^*)}{\partial w} &= 0 \\ \frac{\partial L(w^*, \alpha^* \beta^*)}{\partial \beta} &= 0 \\ \alpha_i^* g_i(w^*) &= 0, i = 1, \dots, k \\ g_i(w^*) &\leq 0, i = 1, \dots, k \\ (\alpha_i^*) &\geq 0, i = 1, \dots, k \end{aligned} \tag{2.11}$$

A função  $(\alpha_i^*) \geq 0, i = 1, \dots, k$  é conhecida na literatura como condição complementar Krausk-Kuhn-Tucker (KKT). Esta função implica em  $\alpha_i^* \geq 0$  quando as restrições é ativas e  $\alpha_i^* = 0$  em caso de restrições inativas. Além disso é possível que um ponto de solução possa localizar-se em uma de duas posições com ligação à uma restrição de desigualdade. Desse modo, a condição KKT referem-se que ou uma restrição é ativa sendo  $g_i(w^*) = 0$  ou os multiplicadores de Lagrange coincide e atendem a  $\alpha_i^* = 0$  [SANTOS, 2002; FACELI et al., 2011; GONÇALVES, 2015].

Na construção do SVM é utilizado o conceito de dualidade, considerando o problema de otimização P, este problema é primal, então pode-se definir um problema relacionado D com o mesmo formato, o problema dual, em que são utilizado a multiplicação de Lagrange de P na qual fazem parte do resultado em D, da mesma forma os multiplicadores de Lagrange de D encontram-se compostos no resultado em P. Dado  $y^*$  sendo o resultado do problema dual D, o resultado do problema primal P é capaz de ser encontrado através

de  $y^*$  [SANTOS, 2002].

Existem várias teorias para descrever um problema de otimização convexos que permitem utilizar o conceito de dualidade, a teoria de Lagrange é uma delas. Usualmente esta descrição é mais fácil ser solucionada por uso computacional do que o problema primal, pois a manipulação direta nas restrições de desigualdade são difíceis. Através da indução dos multiplicadores de Lagrange se obtém um problema dual, essa indução é conhecida como variáveis duais. A metodologia usada no conceito de variáveis duais é que essas variáveis não são conhecidas e necessitam ser encontradas para resolver o problema [SANTOS, 2002].

Em situações reais SVM com padrões linearmente separáveis são difíceis de ser encontrados, pois a maior parte trabalha com resolução de problemas não-lineares. O conjunto de informações (dados) é dito como não-linear, caso não possa ser dividido em um hiperplano. O espaço de características (*feature space*) é o nome dado quando as SVMs fazem o mapeamento em um conjunto de dados não-lineares onde passam a transformar o espaço original em outro com maior dimensão [FACELI et al., 2011; GONÇALVES, 2015].

A Figura 7 apresenta dois casos em que os conjuntos de dados são linearmente separável e não-linearmente separável.

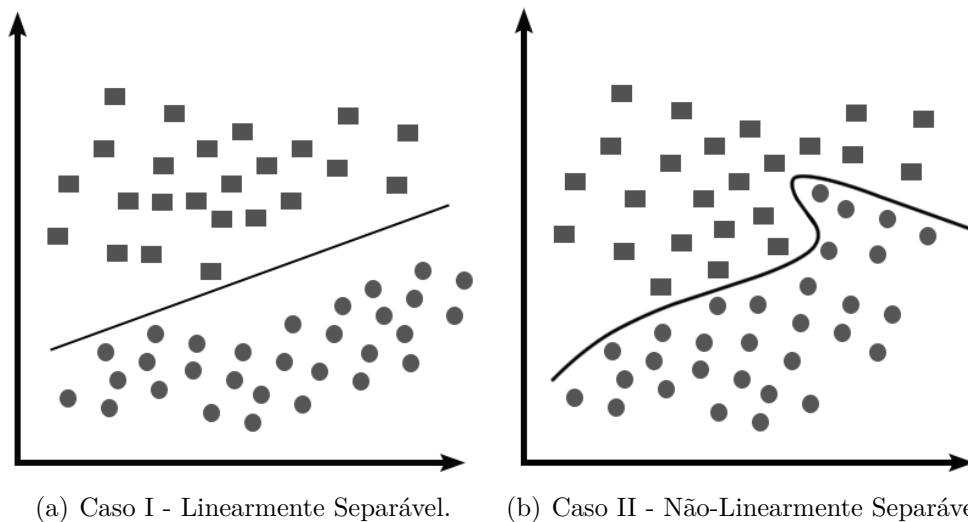


Figura 7 – Exemplos de um conjunto de dados linearmente separável e não-linearmente separável.

O conceito de mapeamento faz o uso do teorema de Cover, afirma que dado um conjunto de dados não-linear ele apresenta maior probabilidade de ser linearmente separável, em um espaço com maior dimensão. As transformações de dimensões é feita através das funções Kernel, assim consegue-se um problema linear, podendo então aproveitar do hiperplano ótimo [FACELI et al., 2011; GONÇALVES, 2015].

A função Kernel tem como entrada dois dados  $x_i$  e  $x_j$  onde é calculado o produto escalar destes dados no espaço de características [FACELI et al., 2011; GONÇALVES, 2015], como mostrado na equação 2.12.

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.12)$$

Frequentemente é utilizado a função Kernel sem conhecer o mapeamento  $\Phi$ , que é criado implicitamente. O principal benefício do Kernel é poder representar espaços abstratos, pois seu cálculo é simples e baixo custo computacional [FACELI et al., 2011; GONÇALVES, 2015].

Funções que satisfazem as condições do Teorema de Mercer é considerada função kernel. De modo resumido, uma função kernel que corresponde as condições de Mercer é identificado como uma matriz de valores positivos e semidefinidas  $K$ , onde cada elemento  $K_{ij}$  é definido pela equação 2.13, segundo Faceli et al. [2011], Gonçalves [2015].

$$K_{ij} = K(x_i, x_j), i, j = 1, \dots, n \quad (2.13)$$

A Tabela 1 mostra os kernels que satisfaz as condições de Mercer e que são frequentemente utilizado em SVM, tais kernels podem ser chamados de Kernels de Mercer [FACELI et al., 2011; GONÇALVES, 2015].

Tabela 1 – Funções kernel

Tipo de kernel	Função $K(x_i, x_j)$	Parâmetros
Polinomial	$(\delta (x_i \cdot x_j) + k)^d$	$\delta, k$ e $d$
RBF	$\exp(-\sigma \ x_i - x_j\ ^2)$	$\sigma$
Sigmoidal	$\tanh(\delta (x_i \cdot x_j) + k)$	$\delta$ e $k$

Na criação do SVM alguns itens precisam ser determinado pelo usuário como por exemplo a função kernel e os parâmetros da função, essas configurações são necessárias para buscar o hiperplano ótimo (excelente).

Os SVMs também lidam com problemas de regressão, em 1995 Vapnik introduziu o conceito do algoritmo  $\varepsilon$ -SVR (*support vector regression*) no qual o objetivo é buscar uma função  $h(x)$  que gere saídas contínuas para o conjunto de dados de treinamento que afastem no máximo de  $\varepsilon$  de seu rótulo pretendido. Essa função deve apresentar duas características, são elas: (i) Uniforme e (ii) Regular. É desejável que essas características apresentem o melhor comportamento possível [FACELI et al., 2011].

Considerando-se o uso da função linear  $h(x) = w \cdot x + b$ , essa função procura por uma função com um pequeno  $w$ , isso é alcançado através da minimização da norma  $\|w\|$ . Dado um problema de otimização apresenta-se então as seguintes equações, conforme [FACELI et al., 2011]:

Minimizar:

$$\frac{1}{2} \|w\|^2 \quad (2.14)$$

Restrições:

$$\begin{cases} y_i - w \cdot x_i - b \leq \varepsilon_i \\ w \cdot x_i + b - y_i \leq \varepsilon_i \end{cases} \quad (2.15)$$

Busca-se a função linear que alcance os pares  $(x_i, y_i)$  de um conjunto de treinamento com um acerto de  $\varepsilon$ . Na Figura 8 é exibido a forma como é realizado o procedimento. Procura-se por uma função linear na qual o conjunto de dados de treinamento deve ficar dentro de um local (região) em torno de  $h$ , isto é encenado pela sombra na Figura [FACELI et al., 2011].

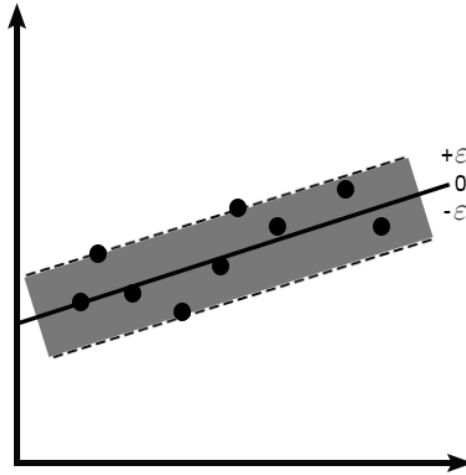


Figura 8 – O modo simplificado do procedimento realizado por SVR. Adaptada de [FACELI et al., 2011].

Da mesma forma que as SVMs de margens suaves, o problema consegue ser relaxado a partir do conceito de variáveis de folga, desta forma é possível lidar com ruídos e *outliers* nos objetos. Tais variáveis de folga possibilita que alguns exemplos (instâncias) estejam fora do local entre  $-\varepsilon$  e  $+\varepsilon$ . Dado o problema de otimização, com a introdução do conceito de variáveis de folga tem-se as seguintes equações, segundo Faceli et al. [2011]:

Minimizar:

$$\frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^n \xi_i + \bar{\xi}_i \right) \quad (2.16)$$

Restrições:

$$\begin{cases} y_i - w \cdot x_i - b \leq \varepsilon + \xi \\ w \cdot x_i + b - y_i \leq \varepsilon + \bar{\xi} \\ \xi, \bar{\xi}_i \geq 0 \end{cases} \quad (2.17)$$

Nas equações exibidas,  $\xi$  e  $\bar{\xi}$  significam as variáveis de folga e  $C$  é um valor constante que determina um *trade-off* entre a regularidade de  $h$  e a quantidade de desvios

que são permitidos. As SVMs trabalha com o problema dual e faz o uso do conceito de Lagrange, assim a solução das derivações parciais se torna nulo e substituí as expressões resultantes na equação lagrangiana inicial [FACELI et al., 2011].

O problema dual alcançado é representado matematicamente como produtos internos (o valor real que associa o comprimento entre vetores e o ângulo formado por eles) entre os objetos. Assim pode-se então utilizar a definição de kernels para executar regressões não lineares. A utilização do kernel envolve o mapeamento dos objetos em espaço de características, na qual a função linear mais regular e que apresenta baixo erro no conjunto de treinamento é descoberta. Então o problema de otimização final é dado pelas equações, de acordo com Faceli et al. [2011]:

Minimizar:

$$-\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \bar{\alpha}_i) (\alpha_j - \bar{\alpha}_j) K(x_i, x_j) - \varepsilon \sum_{i=1}^n (\alpha_i + \bar{\alpha}_i) + \sum_{i=1}^n y_i (\alpha_i - \bar{\alpha}_i) \quad (2.18)$$

Restrições:

$$\begin{cases} \sum_{i=1}^n (\alpha_i - \bar{\alpha}_i) = 0 \\ \alpha_i, \bar{\alpha}_i \in [0, C] \end{cases} \quad (2.19)$$

Nas equações exibidas,  $\alpha_i$  e  $\bar{\alpha}_i$  significam as variáveis de Lagrange e  $K$  é a função kernel. As variáveis de Lagrange incluso na margem entre  $-\varepsilon$  e  $+\varepsilon$  são nulas, vale ressaltar que tais variáveis estão associados aos exemplos.

#### 2.1.4 Florestas Aleatórias

Inicialmente a florestas aleatórias (*random forests*) surgiu da proposta de Breiman [BREIMAN, 2001]. O algoritmo florestas aleatórias é baseado em um conjunto de combinação de classificadores ou regressões idealizado especialmente para árvores de decisão. Constitui-se de um conjunto de classificadores organizados em árvores  $\{h(X, \Theta_k), k = 1, \dots\}$ , em que  $\{\Theta_k\}$  é vetores separados e similarmente distribuídos, e todas árvores fazem uma votação para determinar a classe mais comum em  $X$ . Um conjunto de vetores aleatórios é criado baseado em uma distribuição de probabilidade definida em cima de um vetor de entrada inicial [GÓMEZ, 2012; BREIMAN, 2001].

A medida da precisão do algoritmo floretas aleatórias é calculado em termos probabilístico sobre a margem do classificador, suponha-se um conjunto de classificadores  $\{h_1(X), h_2(X), \dots, h_k(X)\}$ , e um conjunto de treinamento aleatório a começar do vetor  $Y, X$ . A equação 2.20 representa a medida da margem do classificador [GÓMEZ, 2012].

$$mg(X, Y) = av_k I(h_k(X) = Y) - max_{j \neq y} av_k I(h_k(X) = j) \quad (2.20)$$



Na classe  $h_k(X)$  são predita de  $X$  em conformidade com um classificador desenvolvido utilizando o vetor randômico  $K$ . A exatidão do classificador na inferência de um determinado exemplo de  $X$  cresce conforme o crescimento da margem. A proporção em que a correlação de todas árvores cresce, ou a rigidez do conjunto decresce, o intervalo do erro de generalização propende ao crescimento. Na equação 2.21 exibe como o intervalo do erro de generalização converge na ocasião em que o número de árvores estiver suficientemente grande [GÓMEZ, 2012; BREIMAN, 2001].

$$PE^* = P_{x,y}(mg(X, Y) < 0) \quad (2.21)$$

Quando há um número grande de árvores é preciso recorrer a *Strong Law of Large Numbers* e da estrutura da árvore que é expressa pela equação 2.22, segundo Gómez [2012].

$$P_{x,y}(P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq y} P_{\Theta}(h(X, \Theta) = j) < 0) \quad (2.22)$$

Florestas Aleatórias (*Random Forests*) não apresenta *overfitting* toda a vez que mais árvores são atribuídas, porém gera um valor do intervalo de erro de generalização. O intervalo superior no erro de generalização consegue ser derivado da agregação entre dois parâmetros: um é a proporção individual da precisão para cada classificador, prevista durante a equação da margem do classificador exibida acima, e da subordinação entre eles [GÓMEZ, 2012; BIAU, 2012; LIAW; WIENER et al., 2002]. A equação 2.23 apresenta como é medida a rigidez do conjunto de classificadores  $\{h(x, \Theta)\}$ , de acordo com Gómez [2012].

$$S = E_{x,y}mr(X, Y) \quad (2.23)$$

Com o objetivo de ampliar à aleatoriedade, Bagging é uma alternativa para ser usado em conjunto com a escolha aleatória de características. Todo novo conjunto de treinamento passa a ser formado pela substituição com base em um novo vetor de entrada inicial. Qualquer árvore nova faz sua indução a partir de um novo conjunto de treinamento utilizando a escolha aleatória de características. Todas árvores alcançam o seu comprimento máximo e não é processada a poda sobre as mesmas [BREIMAN, 2001; GÓMEZ, 2012]. É possível ser enumeradas por dois motivos para a utilização de Bagging, segundo Gómez [2012] são elas: (i) quando é usado há uma melhora na precisão nos casos em que são utilizados as características aleatórias e (ii) Bagging é capaz de ser utilizado no sentido de proporcionar inferências no erro de generalização da combinação de vários conjunto de árvore bem como as inferências para rigidez (força) e correlação no decorrer da aplicação.

A forma como é realizadas as inferências no conjunto de treinamento necessita-se da técnica *Out-of-Bag* e tem o seu funcionamento no seguinte modo. Dado um conjunto de treinamento a partir do método de construção de um classificador aleatório, e um conjunto de treinamento  $T$  estabelecido. A começar realizastes, são feitos os conjuntos de treinamento de *Bootstrap*  $T_k$ . Seguidamente é criado os classificadores  $h(x, T_x)$ . A votação é utilizado na formação do *Bagged Predictor*. Em cada  $y, x$  sobre o conjunto de treinamento pode haver o incorporamento de votos naqueles em que os classificadores de  $T_k$  não inclui  $y, x$ . Tal classificador é chamado de *Out-of-Bag* [GÓMEZ, 2012].

Para todo conjunto de treinamento que faz o uso da técnica *Bootstrap* tem-se um terço das instâncias deixado de lado. Consequentemente as inferências de *Out-of-Bag* baseá-se na combinação unicamente em um terço dos classificadores. Conforme a medida da taxa de erro decresce e o número de combinações cresce, as inferências de *Out-of-Bag* propendem a superestimar o valor do erro corrente. Além disso, o método de *Out-of-Bag* consegue fazer a estimativa da força e correlação existente entre as árvores [GÓMEZ, 2012]. De acordo com Breiman [2001] essas inferências (estimativas) internas auxiliam a compreender a precisão de qualquer classificador e também como aperfeiçoar de maneira mais adequada.

Todas árvores de decisão usam um conjunto de valores aleatórios, independentemente do tipo da técnica utilizado para árvore ser construída. Tal conjunto é gerado a partir da distribuição de probabilidade fixa. O vetor aleatório em alguns casos é incorporado no processo de elaboração da árvore de várias maneiras [BREIMAN, 2001; BIAU, 2012; GÓMEZ, 2012; LIAW; WIENER et al., 2002].

Na literatura científica a técnica Floresta-RI é frequentemente aplicada, a qual RI remete à escolha aleatória de entrada, está abordagem procura selecionar aleatoriamente  $F$  características de entrada com objetivo de realizar a divisão em todo nodo da árvore de decisão. Ao contrário de outras técnicas que examinam todas as características acessíveis a Floresta-RI tem o nodo dividido a partir das  $F$  características selecionadas. A técnica CART é utilizada no desenvolvimento da árvore que possui a propriedade de permitir o crescimento total das árvores sem poda. Posteriormente a construção das árvores, o esquema de votação por maioria é o método usado para combinar as previsões. A rigidez (força) e a correlação das Florestas Aleatórias em algumas situações há dependência do tamanho de  $F$ . Quando  $F$  estiver extremamente pequeno todas árvores propendem a se tornar menos correlacionadas. Com isto, a força do classificador tende a aprimorar por meio de um número maior de características  $F$ . No balanceamento do número de características é escolhido uma função, usualmente utiliza-se a  $F = \log_2 d + 1$ , em que  $d$  representa o número de características de entrada. Se porventura o número de  $d$  com características originais apresentar um valor muito pequeno, será difícil a escolha de um conjunto independente de características aleatórias para o desenvolvimento da árvore de

decisão [GÓMEZ, 2012].

O espaço de características pode ser aumentado a partir da geração de combinações lineares das características de entrada. Breiman propôs esta abordagem e a chamou de Floresta-RC [BREIMAN, 2001]. Essa abordagem todo nodo gera uma nova característica após selecionar aleatoriamente  $L$  características de entrada. Usam a combinação linear, utilizando os coeficientes construídos com base na distribuição uniforme com intervalo de  $[-1, 1]$ . Em todo nodo é criado  $F$  novas características agrupadas aleatoriamente, desse modo a melhor é escolhida como divisor de nodo. Dado  $M$  sendo variáveis de entrada. Posteriormente a criação de todas as árvores, o número da variável  $m$ th na amostra de *Out-of-Bag* é randomicamente permutado, e tem o conjunto de dados (informações) classificado pela árvore equivalente. Sendo que a classificação concebida por cada  $X_n$  que não localiza-se dentro do *Out-of-Bag* é guardada. O procedimento faz sua interação até que a condição seja satisfeita, nesse caso o  $m = 1, 2, \dots, M$ . Ao fim da execução todas as classes diferentes do *Out-of-Bag* faz a votação de  $X_n$ , a variável  $m$ th é confrontada com o rótulo da classe verdadeira de  $X_n$  com intuito de calcular o valor de erro da classificação. O resultado é o crescimento do percentual no valor da classificação incorreta vinculada ao valor de *Out-of-Bag* [GÓMEZ, 2012]. A Figura 9 apresenta o modelo de funcionamento em um algoritmo *random forests*. A variável  $D$  é o conjunto de dados de treinamento inicial, e a partir dele o passo 1 gera os vetores randômicos (aleatórios), desse modo o passo 2 utiliza os vetores para construir as árvores de decisão e por fim o passo 3 faz a combinação delas.

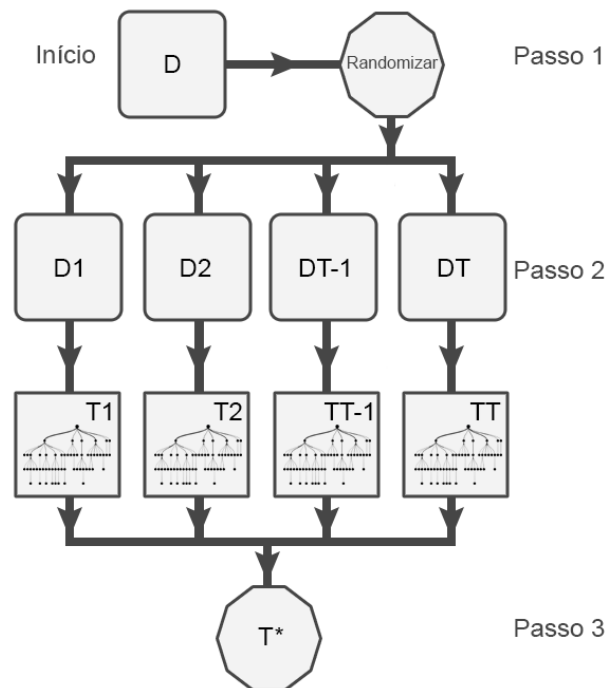


Figura 9 – Funcionamento do *random forests*. Adaptada de [GÓMEZ, 2012].

## 2.2 Agricultura de Precisão

Atualmente fala-se muito sobre agricultura de precisão, mas são poucas pessoas que sabem o real valor dessa tecnologia. Ao longo da história os agricultores procuravam potencializar a produção física e econômica das culturas, diferenciando a aplicação de insumos de acordo com os diversos tipos de solo e desempenho das culturas. Os antigos agricultores sabiam a utilidade e os benéficos de aplicar o esterco e material calcário em tipo de solo específico [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007; TSCHIEDEL; FERREIRA, 2002; JAKOB et al., 1999].

A agricultura de precisão com o passar dos anos obteve aumento na procura para utilização precisa de materiais no solo, vários estudos tem sido propostos. Em 1980 foram usado um grande número de termos para representar o conceito de agricultura de precisão, de acordo com Coelho [2005] são eles: (i) agricultura por metro quadrado, (ii) agricultura com base em tipos de solos, (iii) aplicação de insumos a taxas variáveis, (iv) variável espacial, precisão, prescrição, ou manejo específico de culturas e (v) manejo por zonas uniformes.

O manejo de solos e culturas por zonas uniformes é definida pelas propriedades de identificação e análise das características do solo, histórico de cultivo, climas e outras possíveis variáveis do sistema de produção, para diferentes localização dentro do campo. Conceito de agricultura de precisão sem a variabilidade do princípio fundamental que é o manejo da variabilidade dos solos e culturas no espaço não teria evoluído [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007; TSCHIEDEL; FERREIRA, 2002].

A agricultura de precisão tem por base o gerenciamento localizado de sistemas agrícolas, usando os recursos como: mapeamento dos elementos de produção, ferramentas de ajuda na tomada de decisão e aplicação localizada de insumos [ANTUNIASSI; BAIIO; SHARP, 2007].

Para o aspecto econômico na utilização da agricultura de precisão representa uma restituição maior na economia, pois ela possibilita ao agricultor uma melhor escolha nos investimentos em regiões onde o potencial de produção seja mais satisfatório. Essa tecnologia permite a geração de dados que podem ser analisados e transformados em informação práticas, para posteriormente utiliza-la de forma correta. [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007; TSCHIEDEL; FERREIRA, 2002].

Em relação ao aspecto ambiental a utilização da agricultura de precisão reflete em redução de produtos fitofarmacêuticos, melhor eficiência de uso dos nutrientes, ampliação na eficiência dos insumos injetados e melhoria na proteção dos solos em combate a degradação (erosão) são geralmente citados como benefícios potenciais na redução do impacto da agricultura ao meio ambiente [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007; TSCHIEDEL; FERREIRA, 2002].

Segundo Antuniassi, Baio e Sharp [2007] define a agricultura de precisão como uma tecnologia avançada para o manejo do solo, dos insumos e das culturas, de forma adequada e considerando as modificações espaciais e temporais dos elementos que afetam a produtividade.

Agricultura de precisão para Coelho [2005] é a aplicação de conceitos e tecnologias no manejo de variabilidade espacial e temporal, relacionada a todos os aspectos da produção agrícola, com finalidade em ampliar a produtividade na agricultura e melhorar qualidade ambiental.

O uso maciço de produtos químicos (fitofarmacêuticos) na agricultura de forma não controlada (precisão) tem trazido preocupações quando a saúde humana e ao meio ambiente. Esta preocupação é recorrente aos números de casos de doenças registradas na população e transformações ambientais [BORGUINI; TORRES, 2006; COLLA et al., 2008]. Agricultura de precisão ajuda a combater o uso incorreto de produtos químicos, já que, ela usa tecnologia ao seu favor para usar esse produtos de forma mais precisa [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007].

O conceito de agricultura de precisão usam algumas tecnologias ao seu favor e essas são divididas em seis categorias:

1. Computadores e programas: A agricultura de precisão precisa de computadores com alto poder de processamento e programas para o manejo de dados e criação de gráficos e mapas, essas tecnologias é quem possibilita a aplicação do conceito de agricultura de precisão. Algumas atividades como mapeamento de colheita, levantamento de informações (dados) de culturas e amostras sistematizada do solos proporcionam informações sobre a variabilidade das culturas e solos em uma determinada região. A agricultura de precisão no manejo há necessidade de processamento e análise de grande volume de informações que variam no espaço e no tempo. No campo o sistema móveis é utilizado por causa da portabilidade e, também, há necessidade de um sistema móvel com um microprocessador com alta velocidade de processamento e memória expansiva que possa guarda um grande volume de informações (dados) [COELHO, 2005].
2. Sistema de posicionamento global (GPS): Utilizado no controle local que é significativo para a agricultura de precisão, na qual possibilita o acesso a variabilidade e na aplicação localizada e específica de insumos. Essa tecnologia tem grande importância para execução dos conceitos de manejo localizados. Antigamente o agricultor para determinar a posição relativa do campo usava cálculos matemáticos (estimativas), o método era simples, no qual a posição era tomada a partir da relação de um ponto previamente fixado, determinado as distâncias contando com a ajuda do radar, ultra-som, entre outros. O sistema de posicionamento global - GPS um sis-

tema que tem como finalidade determinar a posição em qualquer parte do mundo e, também, contribui com o desenvolvimento da agricultura de precisão a ser tornar uma realidade [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007].

3. Sistemas de informação geográfica (SIG): Sistemas de informações geográficas é definida como um agrupamento de programas, equipamentos, métodos, dados, totalmente integrados, isso tornar-se possível a coleta, o armazenamento de informações (dados), o processamento e a análise de um conjunto de informações (dados) georreferenciados [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007].
4. Sensoriamento remoto: Sensoriamento remoto pode ser definido como a aquisição de um conjunto de informações (dados) a respeito de algum objeto (utensílio) que não está em contato físico com ele. Na agricultura o principal objetivo do sensoriamento remoto é a interação de solos e plantas que tenha energia eletromagnética. Os sensores mais utilizados na agricultura de precisão pode ser divididos em duas categorias: fotográfico ou não fotográfico. Os dois coletam um conjunto de informações (dados) sobre a energia eletromagnética e como ela interage na superfície sendo vista. Essa tecnologia tem sido usado em várias aplicações no campo, por fornecer um método rápida e eficiente no acesso da variabilidade espacial e temporal dentro de uma região em uma propriedade [COELHO, 2005].
5. Sensores: Hoje em dia existem uma gama de sensores que pode ser definido como instrumentos que transportam impulsos elétricos em *feedback* a estímulos físicos como calor, luz, magnetismo, movimento, pressão e som. Os sensores são os elementos que menos se desenvolveram, a falta de sensores tem sido apontando como o fator crítico na implementação da agricultura de precisão [COELHO, 2005].
6. Controladores eletrônicos de aplicação: É fortemente ligada com a tecnologia da informação, a agricultura de precisão dispõe, como sendo os objetivos essenciais, para potencializar a eficiência no uso dos insumos agrícolas, executando-os diferencialmente ao longo da lavoura, conforme a real utilidade de cada zona uniforme de manejo. No gerenciamento localizado o auxílio de programas específico de sistemas de informações geográfica com foco na agricultura são cruciais, pois eles fornecem um mapa que mostra as regiões onde foram feita as aplicações de insumos. Assim outros equipamentos podem ler esse mapa que ajuda na aplicação correta de insumos em determinado local [COELHO, 2005; ANTUNIASSI; BAIIO; SHARP, 2007].

Na agricultura de precisão à aplicação baseada em mapeamento é a mais comum e são utilizadas por diversos agricultores. O mapeamento é feito usando o sistema de posicionamento global diferencial que faz o levantamento de informações em diferentes fases da cultura, no qual são extraídos dados sobre infestação e suas características. O mapa é processado e transformado em mapas de tratamentos, os quais consistem em recomendações

de controle distribuídas espacialmente no terreno (campo). Na aplicação o computador de controle do pulverizador determina qual é a posição atual da máquina via sistema de posicionamento global diferencial e localiza o mapa que tem o tratamento adequado para esta posição, fazendo o controle e diferenciando a dose ao longo da locomoção pelo terreno (campo) [ANTUNIASSI; BAIIO; SHARP, 2007].

O processo de tomada de decisão na geração do mapa de aplicação são levado em conta os erros que o sistema de posicionamento pode apresentar, a precisão do mapeamento dos alvos, as características de performance do equipamento de aplicação como tempo de resposta, velocidade, tamanho da barra e entre outros [ANTUNIASSI; BAIIO; SHARP, 2007].

Os equipamentos usados na aplicação localizada de defensivos são considerados sofisticados. O sistema de pulverização é o equipamento mais utilizado na agricultura de precisão na aplicação de produtos químicos (fitofarmacêuticos). Esse sistema é controlado por um computador central que guarda um conjunto de informações (dados) sobre as regiões onde devem ser efetuadas as aplicações, além das doses recomendadas (mapas de tratamentos). A central de controle deve ser capaz de interpretar o mapa e processar ele em tempo real, as informações de posicionamento geográfico é adquirido a partir do sistema de posicionamento global diferencial. Com isso permite localizar a posição atual do pulverizador no campo, na qual auxilia a aplicação localizada e a alternância de doses ou produtos [ANTUNIASSI; BAIIO; SHARP, 2007].

Como pode ser visto, a abordagem que é frequentemente utilizada na agricultura de precisão não leva em conta alguns fatores como pressão do ar, velocidade do vento, o volume depositado em um determinado local. Com o conceito de aprendizado de máquina com essa tecnologia de pulverizar com a atribuição da abordagem que está sendo proposta pode ajudar a responder algumas questões como: qual a quantidade de produto foi aplicado em determinado local, qual a melhor configuração para determinada previsão do tempo entre outras.





### 3 TRABALHOS RELACIONADOS

Na agricultura de precisão a aplicação produtos fitofarmacêuticos (pesticida) em uma área é feita por um pulverizador. A função de um pulverizador é aplicar produtos em um dado terreno de forma localizada, ele é constituído por uma barra com um conjunto de bicos injetores. Os bicos injetores soltam pequenas gotículas de produtos em um determinado local, porém essas gotas sofrem com a dispersão durante o trajeto entre o bico e o local previsto. Alguns fatores como a velocidade do vento e tamanho das gotas injetadas estão diretamente relacionada com a dispersão das gotículas.

Na agricultura existe duas abordagens para o cálculo da deposição de produtos fitofarmacêuticos, são elas: (i) abordagem real e (ii) abordagem simulada. Na abordagem real é feito ensaio em campo e utiliza diversos equipamentos e recursos humanos na coleta de informações que posteriormente será calculada a deposição. Apesar da abordagem real mostrar ótimos resultados a sua desvantagem é a utilização de vários equipamentos e recursos humanos. Neste trabalho não serão tratado o tipos de cálculos e técnicas na coleta de informações da abordagem real, para uma leitura complementar são indicado os trabalhos [DONKERSLEY; NUYTENS, 2011; KASIOTIS et al., 2014]. Na abordagem simulada, são utilizado computadores que fazem a simulação de um ambiente de pulverização. A seguir é apresentado trabalhos que fazem simulações de pulverização.

O trabalho proposto por Duga et al. [2013] foi realizar uma análise dos diversos sistemas de pulverização para tipos diferentes de arquitetura de árvores frutíferas. Visto que, o padrão com que os produtos fitofarmacêuticos são depositados influenciam diretamente na qualidade da produção. Na pesquisa foram necessário medições de deposição em três árvores no experimento de campo. Cada árvore tinha um espaçamento entre a outra de 1,75 metros, já que cada sistema de treinamento tem seu modelo arquitetural. Para analisar o impacto nas folhas e ramos das árvores no ar utilizou-se um perfil vertical da densidade da área foliar e coeficiente de arrasto. O sistema de treinamento usou o perfil vertical, na qual o método de cálculo é através da fração de folga fotográfica. Na simulação a árvore e o simulador de pulverização depende do tipo arquitetural, os pulverizadores Duoprop, CondorV e Tango são representados pelo o domínio de pulverizador. Na representação do efeito do vento na deposição necessitou-se de um especialista com experiência em campo. O modelo de deposição concentrou-se em folha, caule e solo na avaliação dos resultados fazendo o comparativo dos diferentes sistema de treinamento, em alguns tipos de pulverizador o sistema de treino não sobe ajustar as posições dos valores máximos. Os três tipos de pulverizadores apresentaram uma desigualdade, visto que, ao ser simulado cada um se apresentou melhor em algumas situações e em outras piores.

No trabalho proposto por Nijdam et al. [2006c] é discutido modelagem *lagrangian*

e *eulerian* para simulação de dispersão turbulenta e coalescência de gotas dentro de um *spray*. Por falta de um modelo onde a dispersão turbulenta e a aglomeração não são tratados simultâneos, é a razão da ausência de um secador por pulverização, dado que os fabricantes de secadores fazem os modelos empíricos simples ou de uma abordagem de ensaio e erro no qual trabalha para aperfeiçoar os desenhos e condições de funcionamento. No modelo *lagrangian* utiliza-se uma matemática contínua é representada por um fluxo de gás, na qual as parcelas de gotas são fisicamente de tamanho similar, no modelo *eulerian* as fases de fluxo de ar e gotículas são tratadas como interpenetrantes, interagindo continuamente. Ambos os modelos mostra que a velocidade de espalhamento da gota tem diferentes tamanho e são prevista semelhantes, isto implica que as gotas menores se dispersam mais rapidamente, já as maiores tem sua dispersão mais lentas. As abordagens proposta são capazes de prever uma pulverização turbulenta, podendo até incluir a decadência da velocidade da linha central e a dispersão de gotículas com distância axial do bico, entretanto é necessário um cálculo pesado para chegar nessa previsão. O tempo computacional preciso para completar uma simulação tem uma ordem grande em ambas as abordagens, na *eulerian* é mais limitada em cálculos bidimensionais para discretizar corretamente a distribuição de tamanho de gotícula. Na prática o modelo *eulerian* é limitado por motivo de não conseguir realizar cálculos grandes, já na abordagem *lagrangian* precisa-se supor que o tamanho da gota seja semelhante e provenientes de diferentes bicos.

A proposta do trabalho de Baetens et al. [2006] é apresentar o quanto a deriva e aplicações de forma não precisa dos pesticidas no campo podem trazer malefícios ao meio. Este problema ocorre pela inexistência de uma ferramenta que proporciona ao agricultor o quando a deriva está influenciando no campo. Na criação de um modelo de estado estacionário forjaram uma simulação de um pulverizador de campo, onde foram utilizadas algumas variáveis para determinar o tamanho do campo, comprimento do pulverizador e quantidade de bicos, o limite atmosférico, a velocidade na qual o trator anda, a injeção de gotículas e turbulência. A geração dessas variáveis tem custo computacional grande, é indispensável alguns cálculos para chegar aos resultados, isto pode elevar tempo. Com o estudo na amostragem notou-se que a direção do vento influência diretamente na deriva, e quando o trator se depara com ventos vindo a sua transversal é onde os valores de deriva cresce consideravelmente. Apesar de apresentar ótimos resultados o modelo de dinâmica dos fluidos computacionais de estado estacionário pode ser considerado robusto por causa dos grandes cálculos envolvidos.

As abordagens de Duga et al. [2013], Nijdam et al. [2006c], Baetens et al. [2006] apresentam bons resultados mas elas não permitem que a estimativa seja feita em tempo de execução. Essas técnicas utilizam cálculos matemáticos para simular o comportamento das gotas (transformação e movimento). Além disso não consideram o padrão existente na pulverização, de acordo com sua configuração.

## 4 PROPOSTA

Os produtos fitofarmacêuticos são pulverizados durante o manejo de proteção (agricultura de precisão) de culturas agrícolas, afim de reduzir os danos causados por pragas. Porém, não foi possível encontrar na literatura científica uma abordagem que permita mensurar a deposição destes produtos com baixo custo computacional, o que permitiria: (i) em laboratório, avaliar uma maior quantidade de situações e técnicas de aplicação para a pulverização do produto na cultura; e (ii) durante a pulverização, quantificar e avaliar se o produto está sendo depositado na região alvo na proporção adequada. Ainda, a segunda possibilidade pode ser explorada para possibilitar que um sistema de pulverização se adapte ao ambiente em que está atuando, afim de proporcionar menor erro na pulverização.

É conhecido que existem diversas variáveis que influenciam a deposição de um produto pulverizado, tal como: características do produto utilizado, pressão do sistema de pulverização, bico de pulverização utilizado, modelo de pulverização (terrestre/aéreo) e as características meteorológicas no momento da liberação do produto até o estado final de cada gota (deposição ou total evaporação). Como exemplo, pode ser citado a variação apenas do vento no momento da pulverização do produto fitofarmacêutico (veja a figura 10). Um aumento na velocidade do vento pode resultar uma deriva mais distante do ponto de origem da pulverização, reduzindo o volume depositado na região a baixo do bico de pulverização. Por outro lado, se a velocidade do vento for baixa no mesmo instante, o volume depositado é maior do que a situação anterior.



(a) Caso I - *Vento fraco.*



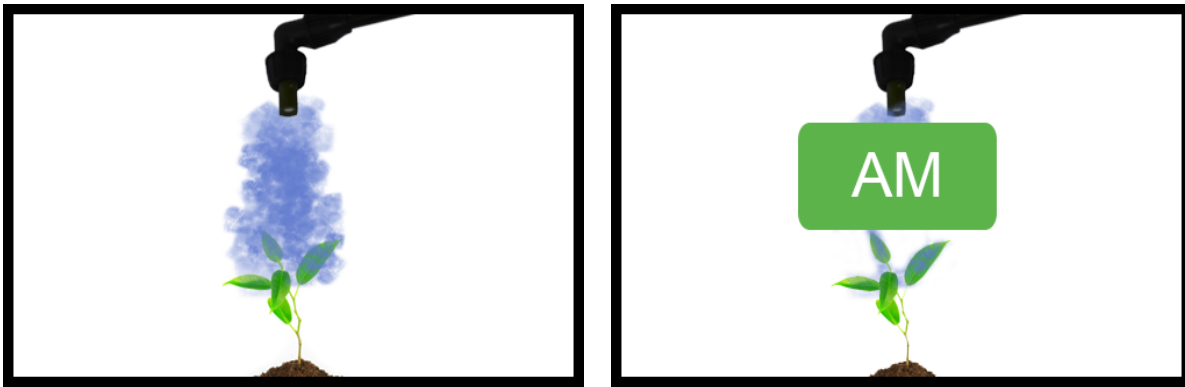
(b) Caso II - *Vento forte.*

Figura 10 – Variáveis do ambiente podem influenciar na deposição do produto pulverizado. Além destas, características do produto e configurações do sistema de pulverização também são fatores que influenciam na precisão da pulverização.

O cenário em que existem variáveis com alguma relação entre si (ou dependências) caracteriza um problema de regressão. Assim, como o sistema proposto tem o objetivo de

estimar a deposição do produto em uma determinada região sem representar a transformação física ou movimentação do produto até a cultura, é possível abordá-lo como um problema de regressão.

Nesse sentido, este trabalho visa contribuir com a proposta de um sistema de baixo custo computacional para a predição do volume de produtos fitofarmacêuticos depositado em uma determinada região da cultura agrícola. Técnicas de AM são exploradas para compôr o módulo principal deste sistema, o qual permita mapear uma função  $f(x)$  que trate deposição do produto pulverizado como um problema de regressão. Nesta abordagem (veja figura 11), o movimento e a transformação é vista como uma caixa preta (não é explicitamente descrita) e a variável resultado é o volume depositado em uma região especificada (por exemplo, no  $\text{cm}^2$  localizado logo a baixo do bico de pulverização).



(a) As abordagens existentes calculam o movimento e a transformação de cada partícula pulverizada (gota) até atingir uma condição de parada (evaporação total ou deposição em alguma superfície). O resultado dessa abordagem depende da análise realizada no conjunto de resultados gerados.

(b) A abordagem proposta estima a deposição da partícula (gota) usando a técnica de AM. O resultado dessa abordagem é específico e depende do atributo alvo escolhido na fase de treinamento.

Figura 11 – Principais diferenças entre as abordagens existentes e a proposta neste trabalho para a previsão de deposição.

Durante a pulverização agrícola o produto é depositado de acordo com as condições com que é realizado a aplicação, incluindo a técnica de aplicação e as condições meteorológicas. Se uma das inúmeras características for alterada (por exemplo, o bico de pulverização) isso influenciará o volume que é depositado em um respectivo local. Esse mesmo comportamento ocorre para qualquer outra característica. No entanto, é necessário destacar que, diferentes fatores possuem influência diferente sobre a pulverização. As respectivas influências podem ser maiores ou menores, de acordo com as condições no momento da pulverização.

É possível notar que este problema não possui resolução trivial. Nesse sentido, é empregado técnicas de aprendizado de máquina para estimar a deposição do produto e generalizar o conhecimento (adquirido a partir de casos conhecidos) para prever a

deposição em situações não conhecidas. Nota-se que, dado as características do problema, as técnicas utilizadas precisam suportar problemas de regressão.

O sistema proposto é denominado *Model of Deposition in Machine Learning-Based* (MoDeL) e sua estrutura está apresentada na figura 12. Este sistema é composto por três componentes essenciais: (i) *Communication*, (ii) *Preprocessing*, e (iii) *Technique of Machine Learning*. O componente Com é responsável em interagir com elemento externos. É este componente que irá receber as informações sobre as características em que a pulverização está sendo realizada e, posteriormente, enviar para a predição realizada para o requisitante. O componente Pp trata as informações recebidas de maneira que sejam repassadas com valores coerentes (compatíveis) para a técnica de AM utilizada. Por exemplo, caso a técnica utilize algum atributo em escala diferente de que é coletada, este componente realizará tal processamento. Dessa forma, o processamento realizado pelo MoDeL se mantém transparente ao requisitante e garante que os dados sejam compatíveis com a entrada esperada pela técnica de AM. Por fim, o componente TML é a técnica de aprendizado de máquina empregada para realizar a predição da deposição. A técnica por ser composta por técnicas diferentes ou até mesmo por múltiplas técnicas.

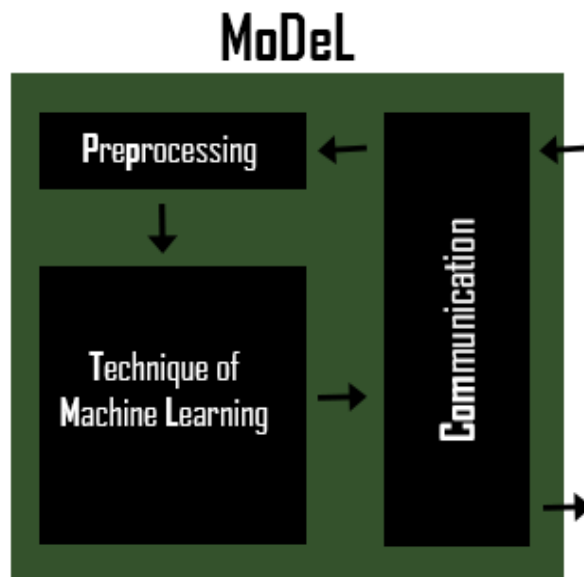


Figura 12 – Estrutura do sistema MoDeL e as respectivas interações dos seus componentes.

Vale ressaltar que o componente TML é composto apenas pela técnica de AM já treinada. Essa técnica treinada é alcançada a partir dessas etapas, são elas: (i) leitura da base de dados, (ii) *cross-validation* e (iii) saída do treinamento. A Figura 13 mostra o diagrama de como é feita a interação entre as etapas. A função da leitura da base de dados é receber os dados que já estão processados, pois ele não faz o pré-processamento de uma base de dados. Após recebido a base o *cross-validation* faz a função de separar os

conjunto de teste e treinamento. A função do *cross-validation* é detalhada na seção 5.2. Dentro dessa etapa é aplicado as técnicas de aprendizado de máquina, posteriormente a execução do treino a etapa de saída do treinamento salva todas informações necessárias para posteriormente ser analisadas.

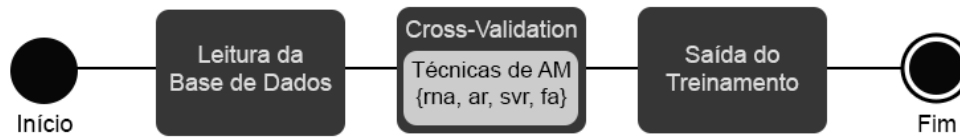


Figura 13 – Etapas realizadas no treinamento da técnica de Aprendizado de Máquina.

Dado que o sistema proposto mapeia uma função para representar o conhecimento adquirido com base em ocorrência conhecidas, é evidente que existe uma fase de treinamento do sistema com a base de dados de amostras previamente conhecidas. Afim de simplificar a fase de treinamento, a base de dados utilizada pode já estar pré-processada.

## 5 METODOLOGIA

Na realização dos experimentos foi utilizado uma base de dados com amostras reais da deposição realizada na escola de agronomia da Universidade Estadual do Norte do Paraná. As pulverizações foram realizadas utilizando um sistema de pulverização com 15 metro de comprimento para deslocamento, equipado com uma barra de três metros contendo quatro bocais, espaçados 0,5 metro por peça e a uma altura de 0,5 metro do topo das plantas. O deslocamento do pulverizador foi ajustado com diferentes velocidades e taxas de aplicação. O sistema hidráulico consistiu em um controlador de pressão manual e uma bomba hidráulica de três pistões, alimentada por um motor elétrico de 1,5 quilowatt.

A base de dados é composta por 214 amostras de deposição com os seguintes atributos apresentado na Tabela 2.

Tabela 2 – Atributos da base de dados.

Breve descrição da pulverização realizada
Número da replicação referente a coleta
Código da técnica
Parte de coleta da amostra de deposição na planta
Código da espécie da planta
Volume depositado

Os atributos descritos na Tabela 2 é detalhado a seguir: (i) breve descrição da pulverização realizada: pressão e bocal usados; (ii) número da replicação referente a coleta: número para identificar a coleta de dados em uma única configuração de pulverização; (iii) código da técnica: número para identificar cada técnica de pulverização utilizada; (iv) parte de coleta da amostra de deposição na planta: altura da planta em que a superfície de coleta está posicionada (alta, média ou baixa); (v) código da espécie da planta: número para identificar a cultura em que a pulverização foi realizada. O atributo alvo é o volume depositado: quantidade do produto depositado na superfície de coleta. Seu valor está muito desigual, variando de 0.0000 a 1.2290. Tabela 3 ilustra a quantidade de valores depois de fragmentar os valores do atributo alvo em cinco intervalos. De acordo com a Tabela 3, o valor alvo da maioria dos exemplos está nos primeiros intervalos. Os valores-alvo são distribuídos de forma desigual porque as repetições da pulverização geralmente têm diferentes volumes depositados dentro de um intervalo adequado para o cenário atual.

Outra informação relevante sobre a base de dados utilizada é que a quantidade de amostras para cada instância é desbalanceada. Tal situação adiciona um nível de complexidade no treinamento do componente TML. Apesar de existirem abordagens para minimizar o impacto dessa característica, tal complexidade permanece presente nesta etapa e requer atenção em seu processamento.

Tabela 3 – Intervalos de depósito.

Intervalo de valor da etiqueta ( $\mu L/cm^2$ )	Número de valores
[0.0000, 0.2478)	64
[0.2478, 0.4956)	96
[0.4956, 0.7434)	36
[0.7434, 0.9912)	05
[0.9912, 1.2390)	09

A qualidade e organização das informações pode afetar o desempenho das técnicas de AM [FACELI et al., 2011]. A base de dados utilizada neste experimentos foi submetida a um pré-processamento para que as informações contidas fossem adequadas e integras. Este pré-processamento se difere do realizado pelo módulo Pp por se tratar da adequação da base de dados para a realização da análise de precisão do componente TML. Enquanto que o pré-processamento realizado pelo elemento Pp é direcionado a adequação das informações para a execução correta do componente TML durante a sua execução em um ambiente real.

O pré-processamento realizado consiste em remover as instâncias que, por algum erro durante a construção da base de dados, não possuíam o atributo alvo (volume depositado) preenchido. Além disso, os atributos com informações redundantes também foram removidos da base de dados, este é o caso da “breve descrição da pulverização realizada”. Este atributo possui exatamente a mesma variação que o atributo “código da técnica”, pois este é basicamente um código para a descrição. Adicionalmente, os atributos “código da técnica” e “parte de coleta da amostra de deposição na planta” tiveram seu formato transformado de simbólico para numérico mantendo a mesma variação, conseqüentemente as mesmas informações. Com a conclusão da fase de pré-processamento, a base de dados passou a ser formada por 210 instâncias de exemplos.

## 5.1 Implementação do componente TML

Para simplificar a etapa de implementação e possibilitar a prova de conceito, o componente TML foi implementado com o linguagem R<sup>1</sup> com pacotes que implementam técnicas de AM de forma otimizada, as quais permitem avaliar a abordagem proposta e fornecer subsídios suficientes para futuros trabalhos.

Quatro técnicas de AM foram avaliadas neste trabalho, são elas: (i) Rede Neural Artificial, (ii) Árvore de Regressão, (iii) Máquinas de Vetores de Suporte e (iv) Floretas Aleatórias.

Na implementação da RNA foi utilizado o pacote *Feed-forward neural networks*

<sup>1</sup> <https://cran.r-project.org/>



*and multinomial log-Linear models* (nnet<sup>2</sup>) da linguagem R. Esse pacote se baseia em uma RNA do tipo *feed-forward* com três camadas: camada de entrada, camada oculta e camada de saída. Cada camada é totalmente conectada a sua camada subsequente. Em outras palavras, todos os neurônios da camada de entrada são conectados a todos os neurônios da camada oculta, a qual possui seus respectivos neurônios conectados a todos os neurônios que compõem a camada de saída.

A implementação da árvore de regressão foi utilizado o pacote (biblioteca) *Recursive partitioning for classification, regression and survival trees* (rpart<sup>3</sup>) da linguagem R que baseia-se no modelo CART, o qual implementa diversas funcionalidades descritas em [BREIMAN et al., 1984].

Na implementação da máquinas de vetores de suporte foi utilizado o pacote *Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien* (e1071<sup>4</sup>) da linguagem R, na qual as funcionalidades principais podem ser encontrada em [FACELI et al., 2011].

As florestas aleatórias é implementada usando o pacote *Breiman and Cutler's Random Forests for Classification and Regression* (randomForest<sup>5</sup>) da linguagem R, onde as principais funções baseia-se nos métodos proposto por Breiman [2001].

É importante ressaltar que foram aplicados dois tipos de configuração de parâmetros para cada técnica em específico, são elas: (i) configuração empírica e (ii) configuração guiada. Tais configurações são utilizadas com o propósito de analisar qual configuração demonstra melhor o comportamento dos dados reais. Nas seções 5.1.1 e 5.1.2 são detalhados como é feito cada configuração de parâmetros para empírica e guiada respectivamente. Todas Tabelas que será citado a seguir é encontrado no apêndice.

### 5.1.1 Configuração Empírica

Cada técnica de aprendizado de máquina teve seus parâmetros configurados de forma a demonstrar uma alta variabilidade nos valores preditos. Neste tipo de configuração diversos valores são testados para cada parâmetro em específico, após a execução da fase de treinamento os resultados obtidos passavam por uma análise. Na análise era verificado o comportamento dos valores preditos em relação aos dados reais, com a finalidade de analisar a semelhança.

A Tabela 8 apresenta os valores dos parâmetros usado na RNA com configuração empírica. O parâmetro da função de ativação é único que não pode ser configurado, porque o pacote (nnet) só implementa a função de ativação do neurônio do tipo *sigmoid*.

<sup>2</sup> <https://cran.r-project.org/web/packages/nnet/>

<sup>3</sup> <https://cran.r-project.org/web/packages/rpart/>

<sup>4</sup> <https://cran.r-project.org/web/packages/e1071/>

<sup>5</sup> <https://cran.r-project.org/web/packages/randomForest/>

Além disso, o pacote garante que nunca haverá *overfitting*, isso ocorre em razão de uma implementação dentro no algoritmo que ao verificar tal ocorrência ele converge, assim garantindo a não existência do *overfitting*.

Na Tabela 9 exibe os valores dos parâmetros utilizado no treinamento da AR para esse tipo de configuração. Apenas dois parâmetros foram modificados, são eles: (i) tipo da árvore e (ii) número de poda, os demais parâmetros teve seus valores padrões. Opto-se por não realizar a poda na árvore já que os resultados obtidos não apresentava alta variabilidade, apesar do erro médio ser relativamente satisfatório. O tipo da árvore é basicamente qual o modelo da árvore, nesse caso o parâmetro anova indica que deve-se construir uma árvore de regressão.

A Tabela 10 demonstra os valores dos parâmetros que foram aplicado no treinamento da SVM. Diferente das outras técnicas mostradas anteriormente todos os parâmetros teve os valores modificados, tais como: (i) constante de Lagrange, (ii) kernel, (iii) gamma, (iv) epsilon e (v) tipo da máquina. O parâmetro do kernel é o mais importante pois ele influencia diretamente em outros parâmetros, a escolha do radial para o kernel deu-se com base nos resultados que apresentou alta variabilidade e baixo erro. Além disso, o tipo da máquina determina o modelo do SVM, nesse caso é utilizado *eps-regression* indicando o modelo SVR na criação.

Na Tabela 11 exibe os valores dos parâmetros usados na fase de treinamento da floretas aleatórias. Somente dois parâmetros teve os valores alterados, são eles: (i) número de árvores e (ii) proximidade. O número de árvores refere-se a quantidade de árvore de regressão que são construídas, determinou-se o valor mil após diversos testes. O parâmetro proximidade basicamente determina a frequência em que dois pontos de dados caem no mesmo nó da folha para diferentes árvores.

### 5.1.2 Configuração Guiada

A configuração guiada permite alcançar os melhores valores de cada parâmetro com maior performance e baixo erro. A técnica de hiper-parâmetro determina os valores dos parâmetros em que são escolhidos antes do início da fase de treinamento sobre uma técnica de aprendizado de máquina [BERGSTRA et al., 2011].

A otimização de um hiper-parâmetro é a dificuldade em definir o conjunto de hiper-parâmetros ideais para uma técnica de AM. Modelos semelhantes de aprendizado de máquinas pode requerer restrições como pesos, taxas de aprendizado diversos na generalização de diferentes padrões de um conjunto de informações (dados). Tais medidas é conhecida como hiper-parâmetros e precisam ser ajustadas para obter um melhor modelo que possa solucionar um problema de AM. Usualmente, uma métrica é estabelecida para mensurar o desempenho do algoritmo em um conjunto de informações independente e são empregues hiper-parâmetros que maximizam essa medida. O *cross-validation* pode

ser utilizado para estimar esse desempenho de generalização [BERGSTRA et al., 2011].

Na otimização do hiper-parâmetro colaciona com os problemas reais de aprendizagem, que do mesmo modo são usualmente apresentados como sendo problemas de otimização, porém melhoram uma função de perda no conjunto de treinamento sozinho. Com consequência, os algoritmos de aprendizado de máquina assimilam parâmetros o qual modelam ou refazem satisfatoriamente as entradas, já a otimização do hiper-parâmetro é assegurar que o modelo gerado não manifeste *overfitting* [BERGSTRA et al., 2011].

O *grid-search* é uma técnica de procura por hiper-parâmetros, essa é a maneira tradicional de processar a otimização de hiper-parâmetro, além disso é frequentemente usada em trabalhos científicos. Basicamente ele faz uma busca exaustiva através de um subconjunto de dados especificado pelo usuário em um espaço de hiper-parâmetro para uma técnica de aprendizado de máquina [LERMAN, 1980; HSU et al., 2003; BERGSTRA; BENGIO, 2012].

Um grande problema do *grid-search* é a dimensionalidade, no entanto é embaraçosamente paralela, porque usualmente as configurações do hiper-parâmetro que analisa são independentes uma da outra [BERGSTRA et al., 2011].

Diferente da configuração empírica a guiada foi empregue a técnica de hiper-parâmetros. O pacote e1071 da linguagem R implementa uma função chamada *tune* o qual usa a técnica de otimização de parâmetros *grid-search*. Na escolha dos intervalos de cada parâmetro aproveitou-se de experiências passadas aplicado na configuração empírica para todas técnicas de aprendizado de máquinas.

É importante ressaltar que a técnica *grid-search* tem um custo computacional grande, isso se deve ao fato de ela realizar combinações entre os parâmetros. Por causa desse problema foi usado o *cluster Euler* da Universidade de São Paulo localizado no campus de São Carlos. O recurso mais utilizado é o hardware em específico a memória, pois a técnica aloca um número considerável desse recurso por exemplo na árvore de regressão era alocado 16 gigabytes.

Nas Tabelas 12, 13, 14 e 15 demonstra os intervalos dos parâmetros para rede neural artificial, árvore de regressão, máquinas de vetores de suporte e florestas aleatórias respectivamente. Parâmetros que não apresenta mais de um intervalo deve-se pelo motivo de demonstrar que o modelo a ser criado baseá em regressão por exemplo no caso do parâmetro tamanho mínimo dos nós folhas da técnica FA.

Ao fim da execução do *grid-search* os modelos de cada técnica de aprendizado de máquina foi obtido e aplicado. As Tabelas 16, 17, 18 e 19 representa os modelos que foram aplicados na fase de treinamento na configuração guiada para RNA, AR, SVM e FA.

## 5.2 *Cross-Validation*

A técnica *cross-validation* Faceli et al. [2011], Kohavi et al. [1995], Monard e Baranauskas [2003] foi empregada nestes experimentos para organizar a base de dados nos grupos de dados de treinamento e teste, respectivamente. Esta técnica também define a dinâmica dos experimentos, a qual é configurada para ser executada com  $k = 10$  sem reposição de instâncias. Em outras palavras, são gerados 10 grupos com sorteios aleatórios das instâncias na base de dados. As instâncias sorteadas são retiradas da base de dados para não serem novamente escolhidas como membro de algum grupo, evitando a presença de elementos repetidos entre os grupos. Por fim, é importante ressaltar que, dado o valor definido para  $k$  e a quantidade de instâncias da base de dados, em cada execução desta técnica são criados 10 grupos com 21 instâncias. Na Figura 14 apresenta o funcionamento do *cross-validation*, o objetivo desse módulo aplicado dentro do componente TML é criar grupos de teste e treinamento.

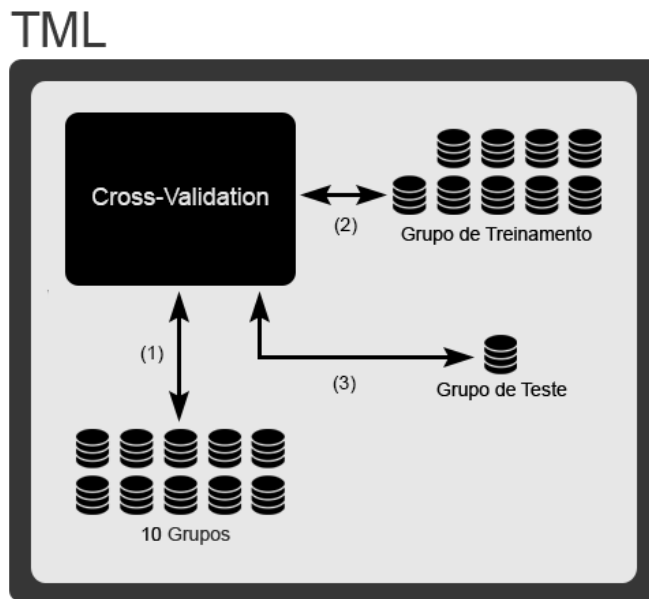


Figura 14 – Funcionamento do *cross-validation*, em 10 iterações de testes.

Os grupos gerados (1) são empregados em 10 iterações de testes, onde são utilizados 9 grupos (2) para treinar o componente TML e o teste (3) é realizado com o grupo não utilizado durante o treinamento. As iterações são realizadas variando o grupo empregado na fase de teste do componente TML até que todos os grupos sejam utilizados. É importante destacar que esse comportamento detalhado se refere a uma iteração do *cross-validation*. Como os sorteios das instâncias que fazem parte do grupo de treinamento ou de teste são realizados de maneira aleatória e a grupos diferentes (mesmo que com pouca diferença - apenas uma instância diferente, por exemplo) os experimentos foram replicados por 10 iterações completas do *cross-validation*. Acredita-se que essa abordagem permite

obter valores de resposta com comportamento próximos do esperado em um ambiente real.

### 5.3 Métricas de erro para regressão

Na análise de um algoritmo de aprendizado de máquina de modelo supervisionado é geralmente realizada através da verificação do desempenho do atributo preditor por ele na identificação de novos objetos, não exibidos antecipadamente em seu treinamento [FACELI et al., 2011; MONARD; BARANAUSKAS, 2003].

Nos problemas de regressão, o erro de hipótese  $\hat{f}$  pode ser previsto calculando-se a distância entre o valor  $y_i$  real e com o atributo predito pela hipótese induzida, ou seja,  $\hat{f}(x_i)$  [FACELI et al., 2011; MONARD; BARANAUSKAS, 2003].

Geralmente, duas medidas de erro são utilizadas: (i) o erro médio quadrático (MSE - *mean squared error*) e (ii) a distância absoluta média (MAD - *mean absolute distance*), representadas por 5.1 e 5.2, respectivamente.

$$MSE(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2 \quad (5.1)$$

$$MAD(\hat{f}) = \frac{1}{n} \sum_{i=1}^n |(y_i - \hat{f}(x_i))| \quad (5.2)$$

As funções 5.1 e 5.2 resultam sempre em valores positivos. Os resultados desses valores para ambas as medidas, é preferível apresentar valores baixos que correspondem a melhores modelos, isto é, melhores estimativas dos rótulos verdadeiros do objeto [FACELI et al., 2011].

Na literatura científica ambas métricas são amplamente utilizadas para avaliar o desempenho do preditor. Alguns trabalhos como [DOSUALDO; REZENDE, 2003; BARDAL; ENEMBRECK, 2013; LIAW; WIENER et al., 2002] usam tais medidas nas avaliações dos algoritmos para problemas de regressão.



## 6 RESULTADOS E ANÁLISE

Como dito anteriormente, os resultados analisados nesta seção refletem o comportamento dos resultados obtidos durante dez execuções da técnica *cross-validation*, mais precisamente aos resultados das etapas de teste de cada técnica com configurações diferentes. Porém, é importante ressaltar que no final de cada execução é gerado um valor médio para o *MAD* e *MSE*, também um gráfico dos valores pontuais das predições (e os respectivos valores esperados). Estas informações são apresentados e analisados nesta seção. Adicionalmente, informações sobre variabilidade das predições e dos erros são apresentados e utilizados na análise.

Dado o comportamento do *cross-validation* definido na seção 5.2, para cada tipo de configuração as técnicas avaliadas realizaram uma predição para cada instância que compõem a base de dados utilizada (sem repetição), totalizando 210 predições. Considerando que qualquer alteração realizada no conjunto de treinamento e consecutivamente nas predições posteriores, o *cross-validation* foi executado 10 vezes (totalizando 2100 predições por cada técnica avaliada), já que a divisão dos grupos de treinamento e de testes são realizados de maneira aleatória. Portanto, as execuções da técnica *cross-validation* são diferentes entre si.

As seções 6.1 e 6.2 apresenta os resultados obtidos pelas configurações empírica e guiada respectivamente. Todas Figuras citadas a seguir são encontradas no apêndice.

### 6.1 Resultados da configuração empírica

A configuração empírica apresentou resultados bastantes satisfatórios, lembrando que nesta configuração o foco é atingir maior variabilidade nos dados preditos. Dessa maneira, é possível simular um ambiente real em que as variáveis pode demonstrar um padrão estocástico. Além das métricas que fazem a verificação do desempenho do atributo preditor exibida na seção 5.3, outras medidas foram usadas afim de obter diversos tipos de análises. Como a covariância que neste estudo é usada para medir a variabilidade conjunta da variável alvo em relação aos valores preditos. Duas análises estatísticas foram utilizadas, são elas: (i) o teste de *Shapiro Wilk* e (ii) conseqüentemente o *Wilcoxon rank sum test*.

Na Tabela 4 exhibe o valor obtido por todas técnicas. O valor demonstrado para cada medida é o resultado do cálculo considerando todas as predições realizadas por cada técnica durante todas as execuções do *cross-validation*. É importante ressaltar, que na configuração empírica algumas técnicas não conseguiram um comportamento semelhante ao ambiente real. Esses casos é visto nas técnicas FA e AR mesmo apresentado um valor

baixo no erro médio. O motivo do baixo valor é porque a predição feita está dentro da faixa de intervalo aceitável.

Tabela 4 – Resultados alcançados utilizando a configuração empírica.

Técnicas	Erro Médio	Erro Médio Quadrático	Covariância	P-valor Shapiro Wilk
<i>Support Vector Machine</i>	0.1503145	0.04125272	0.02992697	6.674134e-30
<i>Random Forests</i>	0.1184432	0.02464298	0.01945926	3.524556e-32
<i>Artificial Neural Networks</i>	0.1138986	0.0239372	0.03349775	5.311755e-40
<i>Regression Tree</i>	0.1063276	0.02053883	0.03268355	2.461403e-41

Considerando os dados apresentados, foram realizados testes estatísticos a fim de aprofundar a análise dos resultados obtidos. Inicialmente foi verificado a hipótese dos valores de predição resultante de cada técnica para testar a aderência em uma distribuição normal. Tal abordagem permite direcionar as próximas etapas na análise estatística entre o uso de técnicas paramétricas ou não-paramétricas, com objetivo de proporcionar maior confiabilidade nos seus resultados.

O teste de *Shapiro Wilk* foi utilizado para avaliar a hipótese, o valor retornado pelo p-valor é apresentado na Tabela 4 para cada técnica (SVM, FA, RNA e AR) da hipótese de valores de predição. Portanto, a hipótese para todas técnicas foi consideradas recusadas e as próximas etapas da análise estatística utilizam o *Wilcoxon rank sum test* (não-paramétrico).

O *Wilcoxon rank sum test* foi utilizado para calcular se existe diferença estatística significativa entre os valores das predições das técnicas SVM, FA, RNA e AR. Adicionalmente, foi analisado se existe diferença estatística significativa no valor resultante das predições realizadas pelas técnicas citadas. Existem dois critérios para analisar o p-valor, são eles: (i) p-valor  $> 0.05$  sem diferença e (ii) p-valor  $< 0.05$  com diferença. O resultado do p-valor obtido por cada técnica é exibida na Tabela 5.

Tabela 5 – Teste estatístico *wilcoxon* na configuração empírica.

	<i>Support Vector Machine</i>	<i>Random Forests</i>	<i>Artificial Neural Networks</i>
<i>Random Forests</i>	0.0265	-	-
<i>Artificial Neural Networks</i>	1.0000	0.0012	-
<i>Regression Tree</i>	1.0000	0.0012	1.0000

Como visto na Tabela 5 o valor decorrente do p-valor da técnica *random forests* mostra que há diferença estatística em relação a outras técnicas. Apesar do valor resultante do erro médio ser aproxima das outras técnicas, um fator relevante que possivelmente fez com que tal técnica apresenta-se diferença é a baixa variabilidade na parte superior e inferior nas predições realizadas.

A Figura 15 exemplifica as execuções feitas pelo *cross-validation* para o algoritmo RNA. A linha azul traçada no gráfico representar os dados reais e cada linha vermelha



equivale aos valores das predições realizadas durante uma execução, que corresponde a 210 valores preditos. O algoritmo RNA apresentou a melhor variabilidade em relação aos outros algoritmos. Comportamento dessa técnica reflete ao ambiente real em que os valores podem apresentar características estocásticas. Vale ressaltar, que a técnica RNA não teve diferença estatística em comparação a SVM e AR, mesmo com alta variabilidade o erro médio dela ainda se compara com outras técnicas.

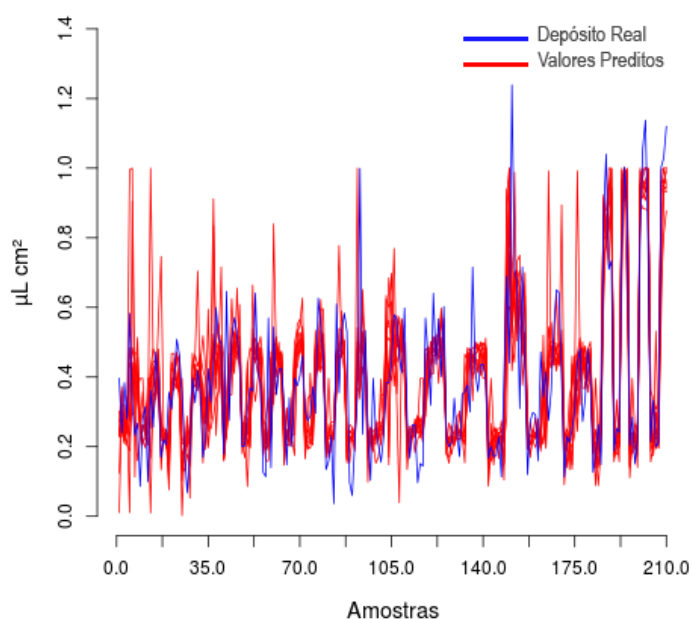


Figura 15 – Predições realizadas pelo algoritmo RNA na configuração empírica.

Para melhor visualização da variabilidade do algoritmo RNA a Figura 17 exhibe os dados reais e as 2100 predições realizadas pelo *cross-validation*. Tais informações também apresentam indícios de que a variabilidade da predição realizada pela RNA é mais próxima do comportamento esperado, sendo mais representativa do comportamento apresentado pelos dados reais.

## 6.2 Resultados da configuração guiada

Ao contrário da configuração empírica em que os parâmetros foi definido a mão, a configuração guiada utilizou da técnica *grid-search*. O *grid-search* tem seus parâmetros configurado a partir de um conjunto de dados (intervalos), na qual faz uma busca exaustiva para encontrar o melhor modelo com o menor erro em uma determinada base de dados. Os intervalos dos parâmetros para cada técnica está definido na seção 5.1.2, os resultados alcançados baseá-se nos parâmetros com o melhor valor encontrado pela técnica *grid-search*.

Todos conceitos utilizados na configuração empírica também foi aplicado nesta configuração. A Tabela 6 mostra os resultados obtidos após a execução das 10 vezes do *cross-validation*. É notável que o erro médio dos algoritmos SVM, FA, RNA e AR apresentam valores menores que da configuração empírica, isso acontece por causa do *grid-search*. Ao procurar um modelo o fator primordial é o erro médio. Conseqüentemente a variabilidade é afetada, nesse sentido a reprodução de um ambiente real pode não ser satisfatório. Dado que o ambiente real em algumas condições apresentam eventos estocásticas. Em algumas técnicas a variabilidade aumentou em outras diminuiu em comparação entre as configurações empírica e guiada.

Tabela 6 – Resultados alcançados utilizando a configuração guiada.

Técnicas	Erro Médio	Erro Médio Quadrático	Covariância	P-valor Shapiro Wilk
<i>Support Vector Machine</i>	0.1034247	0.01939825	0.03131878	4.376545e-37
<i>Random Forests</i>	0.1054316	0.02095754	0.03219461	6.284324e-39
<i>Artificial Neural Networks</i>	0.1120831	0.02249108	0.03423133	8.204795e-39
<i>Regression Tree</i>	0.1016021	0.01863527	0.03194168	1.30244e-43

Como dito na seção anterior o teste estatístico *Shapiro Wilk* também é empregue na configuração guiada. Todos algoritmos tiveram o teste da distribuição normal recusado, o p-valor de cada técnica é exibido na Tabela 6. Como consequência o teste estatístico *Wilcoxon rank sum test* foi aplicado e seus resultados é visto na Tabela 7.

Tabela 7 – Teste estatístico *wilcoxon* na configuração guiada.

	<i>Support Vector Machine</i>	<i>Random Forests</i>	<i>Artificial Neural Networks</i>
<i>Random Forests</i>	0.00255	-	-
<i>Artificial Neural Networks</i>	0.00097	1.00000	-
<i>Regression Tree</i>	0.00063	1.00000	1.00000

No caso da configuração guiada o algoritmo SVM apresentou diferença estatisticamente para as demais técnicas.

A Figura 16 exemplifica as 10 vezes em que o algoritmo AR foi executado na configuração guiada. A linha azul representa os dados reais (depósito real) e as linhas vermelha refere-se aos valores previstos durante cada execução do *cross-validation*. O algoritmo AR nesta configuração mostrou uma menor variabilidade nas partes superior e inferior. Nesse caso a árvore de regressão passou a encontrar um padrão, porém a análise estatística mostrou que ela não teve diferença estatisticamente com outras técnicas com exceção da SVM.

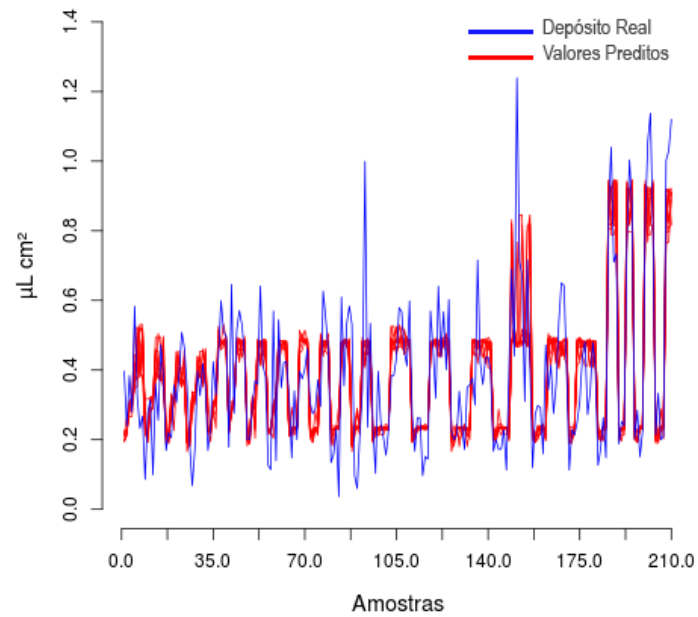


Figura 16 – Predições realizadas pelo algoritmo AR na configuração guiada.

Variabilidade da árvore de regressão é demonstrada na Figura 18, nessa situação os dados previstos não estão variando conforme os dados reais. Os *outliers* identificado no conjunto de dados previstos não assemelha-se aos dados reais. A maior parte da dispersão do conjunto de dados previstos é encontrado nos menores valores dispersos.



## 7 CONCLUSÃO E TRABALHOS FUTUROS

Atualmente, existem abordagens que se baseiam em métodos matemáticos para calcular a movimentação e a transformação ocorrida em cada gota para estimar o volume de deposição em uma área alvo. Tal abordagem realiza os cálculos necessários em cada espaço de tempo até que a respectiva gota satisfaça uma condição de parada, ou seja, sua total evaporação ou alcance alguma superfície. Dado o grande volume de cálculos necessário para estimar o volume depositado em uma região alvo, sendo necessário calcular toda a pulverização e com base em seus resultados, mensurar o volume de produto que atingiu a respectiva área alvo.

Apesar dos bons resultados que podem ser obtidos por essa abordagem, é de conhecimento que técnicas de mecânica de fluídos são computacionalmente custosas [BAETENS et al., 2006]. Tal comportamento dificulta seu uso durante a realização do manejo agrícola de proteção da cultura para fazer ajustes finos na configuração do maquinário. Nesse sentido, este trabalho investigou e propôs um sistema denominado *Model of Deposition in Machine Learning-Based* (MoDeL) como prova de conceito de uma nova abordagem baseada em aprendizado de máquina. Até o momento foi desenvolvido o componente central, denominado TML, que emprega o conceito de uso das técnicas de aprendizado de máquina.

As técnicas de rede neural artificial, árvore de regressão, máquinas de vetores de suporte e florestas aleatórias foi implementadas e testadas. Os testes para as quatro técnicas é realizado dez vezes para cada tipo de configuração de parâmetros, a fim de obter diversos resultados para posteriormente fazer a comparação. Vale ressaltar, que cada teste equivale a dez execução, isso acontece por causa do uso da técnica de *cross-validation*, com isto, todas técnicas é rodada em um total de cem vezes. Neste trabalho foi aplicado dois tipos de configurações nos parâmetros para cada técnica, são eles: (i) empírica os parâmetros são definidos a partir de testes, a qual é analisado a variabilidade dos dados previstos em relação ao depósito real e (ii) guiada os parâmetros é definido com o resultado da execução da técnica *grid-search*, em que cada parâmetro recebe um conjunto de valores para procurar o melhor modelo que apresente o menor erro.

Os resultados obtidos de cada configuração são satisfatório, na empírica os resultados da técnica florestas aleatórias apresentou baixa variabilidade nos dados previstos. Todas técnicas teve os resultados do teste estatístico *Shapiro Wilk* para distribuição normal recusado, e posteriormente foi verificado se havia diferença estatisticamente entres elas. O teste *Wilcoxon rank sum test* mostrou que a floretas aleatórias possui diferença estatística em comparação as técnicas RNA, SVM e AR. Na guiada também utiliza os testes feito na empírica, os resultados de todas técnicas para a distribuição normal foi re-

cusado. Em seguida o algoritmo SVM apontou que tem diferença estatística para outras técnicas.

Tais resultados mostrou que as técnicas podem ser utilizadas dependendo do objetivo almejado, como exemplo pode-se citar (i) o uso de técnicas que modelem a alta variabilidade da deposição para o sistema atuar como simulador da deposição resultante de uma determina configuração ou (ii) encontre o padrão de deposição esperado para determinadas condições de pulverização. Porém a indício que o algoritmo RNA na configuração empírica apresenta o comportamento semelhante do ambiente real, demonstrando ser capaz de manter informações de variáveis estocásticas do ambiente.

Diante disso, é possível vislumbrar novos estudos a partir deste, tais como: (i) investigar novas técnicas de aprendizado de máquina, (ii) aprofundar os experimentos analisando variações dos conceitos estudados até o momento (por exemplo, RNA com mais de um camada oculta), (iii) validar a eficiência do sistema proposto em novos dados reais que apresentem atributos distintos, (iv) verificar a possibilidade deste sistema modelar a influência de condições meteorológicas na deposição do produto fitofarmacêutico e (v) mensurar e comparar o custo computacional da abordagem atual em relação a abordagem matemática existente.

## REFERÊNCIAS

- ANTUNIASSI, U. R.; BAILO, F. H.; SHARP, T. C. Agricultura de precisão. In: *CONGRESSO BRASILEIRO DO ALGODÃO*. [S.l.: s.n.], 2007. v. 6, p. 11–21.
- BAETENS, K. et al. The relative importance of environmental and field sprayer parameters for reducing drift: a cfd sensitivity study. *Aspects of Applied Biology*, AAB; 1999, v. 77, n. 2, p. 303, 2006.
- BARBOSA, J. P. M. *UPBox: Armazenamento na Nuvem para Dados de Investigação da U. Porto*. Tese (Doutorado) — UNIVERSIDADE DO PORTO, 2013.
- BARDDAL, J. P.; ENEMBRECK, F. *Detecção de Mudança de Conceito em Problemas de Regressão Utilizando a Teoria de Redes Sociais*. [S.l.]: ENIAC, 2013.
- BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, n. Feb, p. 281–305, 2012.
- BERGSTRA, J. S. et al. Algorithms for hyper-parameter optimization. p. 2546–2554, 2011.
- BIAU, G. Analysis of a random forests model. *Journal of Machine Learning Research*, v. 13, n. Apr, p. 1063–1095, 2012.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006.
- BORGUINI, R. G.; TORRES, E. A. F. da S. Alimentos orgânicos: qualidade nutritiva e segurança do alimento. *Segurança alimentar e Nutricional*, v. 13, n. 2, p. 64–75, 2006.
- BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.
- BREIMAN, L. et al. *Classification and regression trees*. [S.l.]: CRC press, 1984.
- COELHO, A. M. Agricultura de precisão: manejo da variabilidade espacial e temporal dos solos e culturas. *Agricultura*, v. 1518, n. 4277, p. 46, 2005.
- COLLA, L. M. et al. Isolamento e seleção de fungos para biorremediação a partir de solo contaminado com herbicidas triáziniacos. 2008.
- DONKERSLEY, P.; NUYTENS, D. A meta analysis of spray drift sampling. *Crop Protection*, Elsevier, v. 30, n. 7, p. 931–936, 2011.
- DORNELLES, M. E. et al. Inspeção técnica de tratores e pulverizadores utilizados em pulverização agrícola. *Engenharia na Agricultura*, v. 19, n. 1, p. 36–43, 2011.
- DORUCHOWSKI, G. et al. Drift evaluation tool to raise awareness and support training on the sustainable use of pesticides by drift mitigation. *Computers and Electronics in Agriculture*, v. 97, p. 27 – 34, 2013. ISSN 01681699. Adjacent structures;Application parameters;Application softwares;Decision supports;Pesticide applications;Practical recommendation;Risk mitigation;Sustainable use of pesticides;. Disponível em: <<http://dx.doi.org/10.1016/j.compag.2013.06.006>>.

- DOSUALDO, D. G.; REZENDE, S. O. Relatorios tecnicos. 2003.
- DUGA, A. T. et al. Sprayer-canopy characterization using field experiments and cfd modelling. In: *12th Int Workshop on Sustainable Plant Protection Techniques in Fruit Growing, Valencia (Spain), Jul.* [S.l.: s.n.], 2013. p. 26–28.
- FACELI, K. et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, v. 2, p. 192, 2011.
- FERREIRA, M. d. C.; OLIVEIRA, J. R. D.; PIETRO, I. R. d. S. D. Distribuição da calda herbicida por pontas de pulverização agrícola utilizadas em áreas de reflorestamento com eucalipto. *Engenharia Agrícola*, Associação Brasileira de Engenharia Agrícola (SBEA), p. 267–276, 2009.
- FERREIRA, M. d. F. M. Árvores de regressão e generalizações: Aplicações. Universidade do Porto. Reitoria, 1999.
- FERREIRA, S. d. R. I. Aplicações de matrizes no ensino médio. 2013.
- GHATE, S.; PERRY, C. Ground speed control of pesticide application rates in a compressed air direct injection sprayer. *Transactions of the American Society of Agricultural Engineers*, v. 37, n. 1, p. 33 – 38, 1994. ISSN 00012351. Direct injection sprayer;Ground speed control;.
- GÓMEZ, S. N. *Random forests estocástico*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2012.
- GONÇALVES, A. R. Máquina de vetores suporte. *Acesso em*, v. 21, 2015.
- GREGORIO, E. et al. Measurement of spray drift with a specifically designed lidar system. *Sensors (Switzerland)*, v. 16, n. 4, 2016. ISSN 14248220. Erbium doped glass;Experimental campaign;Integrated measurements;Lidar measurements;Light detection and ranging techniques;Real time measurements;Spray drift;Temporal and spatial;. Disponível em: <<http://dx.doi.org/10.3390/s16040499>>.
- GRIFOLL, J.; ROSELL-LLOMPART, J. Efficient lagrangian simulation of electrospray droplets dynamics. *Journal of Aerosol Science*, v. 47, p. 78 – 93, 2012. ISSN 00218502. Disponível em: <<http://dx.doi.org/10.1016/j.jaerosci.2012.01.001>>.
- GUO, B.; FLETCHER, D. F.; LANGRISH, T. A. Simulation of the agglomeration in a spray using lagrangian particle tracking. *Applied Mathematical Modelling*, v. 28, n. 3, p. 273 – 290, 2004. ISSN 0307904X. Eulerian-Lagrangian approach;. Disponível em: <[http://dx.doi.org/10.1016/S0307-904X\(03\)00133-1](http://dx.doi.org/10.1016/S0307-904X(03)00133-1)>.
- HAYKIN, S. S. *Redes neurais*. [S.l.]: Bookman, 2001.
- HSU, C.-W. et al. A practical guide to support vector classification. Taipei, Taiwan, 2003.
- JAKOB, A. A. E. et al. Estudo da correlação entre mapas de variabilidade de propriedades do solo e mapas de produtividade para fins de agricultura de precisão. [sn], 1999.
- JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 6.



- JIAO, L. et al. Monitoring spray drift in aerial spray application based on infrared thermal imaging technology. *Computers and Electronics in Agriculture*, v. 121, p. 135 – 140, 2016. ISSN 01681699. Concentration distributions;Image processing algorithm;Infrared thermal imaging;Infrared thermal imaging systems;Real time measurements;Spray application;Spray drift;Water sensitive paper;. Disponível em: <<http://dx.doi.org/10.1016/j.compag.2015.12.006>>.
- JOHNSON, A. T. Food for the future [state of the art]. *IEEE Pulse*, v. 7, n. 3, p. 48 – 49 and 57, 2016. ISSN 21542287. Food production;Political impact;Population growth;State of the art;. Disponível em: <<http://dx.doi.org/10.1109/MPUL.2016.2538483>>.
- KASIOTIS, K. M. et al. Spray drift reduction under southern european conditions: A pilot study in the ecopest project in greece. *Science of the Total Environment*, Elsevier, v. 479, p. 132–137, 2014.
- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.
- KWOK, J. T.-Y. Support vector mixture for classification and regression problems. In: IEEE. *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*. [S.l.], 1998. v. 1, p. 255–258.
- LERMAN, P. Fitting segmented regression models by grid search. *Applied Statistics*, JSTOR, p. 77–84, 1980.
- LIAW, A.; WIENER, M. et al. Classification and regression by randomforest. *R news*, v. 2, n. 3, p. 18–22, 2002.
- LUGER, F. G. *Inteligência Artificial*. 6. ed. [S.l.]: Person, 2013.
- MINOV, S. V. et al. Spray droplet characterization from a single nozzle by high speed image analysis using an infocus droplet criterion. *Sensors (Switzerland)*, v. 16, n. 2, 2016. ISSN 14248220. Droplet generators;Focus characteristic;High speed image analysis;Image acquisition and analysis;Phase Doppler particle analyzer;Piezoelectric generators;Proof of principles;Spray characterizations;. Disponível em: <<http://dx.doi.org/10.3390/s16020218>>.
- MITCHELL, T. M. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas Inteligentes-Fundamentos e Aplicações*, v. 1, n. 1, 2003.
- NIJDAM, J. et al. Validation of the lagrangian approach for predicting turbulent dispersion and evaporation of droplets within a spray. *Drying Technology*, v. 24, n. 11, p. 1373 – 1379, 2006. ISSN 07373937. Lagrangian approach;Particle drying;Spray drying;. Disponível em: <<http://dx.doi.org/10.1080/07373930600952412>>.
- NIJDAM, J. J. et al. Lagrangian and eulerian models for simulating turbulent dispersion and coalescence of droplets within a spray. *Applied Mathematical Modelling*, v. 30, n. 11, p. 1196 – 1211, 2006. ISSN 0307904X. Droplets;Eulerian models;Lagrangian models;Spray;. Disponível em: <<http://dx.doi.org/10.1016/j.apm.2006.02.001>>.

- NIJDAM, J. J. et al. Lagrangian and eulerian models for simulating turbulent dispersion and coalescence of droplets within a spray. *Applied mathematical modelling*, Elsevier, v. 30, n. 11, p. 1196–1211, 2006.
- NORVIG, P.; RUSSELL, S. *Inteligência Artificial, 3a Edição*. [S.l.]: Elsevier Brasil, 2014.
- NUYTTENS, D. et al. Drift from field crop sprayers using an integrated approach: Results of a five-year study. *Transactions of the ASABE*, v. 54, n. 2, p. 403 – 408, 2011. ISSN 21510032. Drift mitigation; Drift model; Field crop sprayers; Spray application technology; Spray drift;.
- PRATI, R. C. *Novas abordagens em aprendizado de máquina para a geração de regras, classes desbalanceadas e ordenação de casos*. Tese (Doutorado) — Universidade de São Paulo, 2006.
- REAL, E. M.; NICOLETTI, M. do C. *Investigação de algoritmos sequenciais de agrupamento com pré-processamento de dados em aprendizado de máquina*. [S.l.]: FACCAMP, 2014.
- RU, Y. et al. Performance experiment of rotary hydraulic atomizing nozzle for aerial spraying application. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, v. 30, n. 3, p. 50 – 55, 2014. ISSN 10026819. Comprehensive performance; Deposition distribution; Nozzle atomizations; Operating parameters; Optimum parameters; Performance experiment; Structure parameter; Unmanned helicopter;. Disponível em: <<http://dx.doi.org/10.3969/j.issn.1002-6819.2014.03.007>>.
- SAHA, A.; PIPARIYA, A.; BHADURI, D. Enzymatic activities and microbial biomass in peanut field soil as affected by the foliar application of tebuconazole. *Environmental Earth Sciences*, v. 75, n. 7, 2016. ISSN 18666280. Nitrogen transformations; Peanut; Soil health; Soil microbial biomass; Tebuconazole;. Disponível em: <<http://dx.doi.org/10.1007/s12665-015-5116-x>>.
- SALYANI, M.; CROMWELL, R. Spray drift from ground and aerial applications. *Transactions of the American Society of Agricultural Engineers*, v. 35, n. 4, p. 1113 – 1120, 1992. ISSN 00012351. Aerial applications; Citrus; Florida; Ground applications; Spray drift; Spray solutions;.
- SANTOS, E. M. dos. *Teoria e aplicação de support vector machines à aprendizagem e reconhecimento de objetos baseado na aparência*. Tese (Doutorado) — Universidade Federal da Paraíba, 2002.
- SUGIYAMA, M. *Introduction to Statistical Machine Learning*. [S.l.]: Morgan Kaufmann, 2015.
- TSCHIEDEL, M.; FERREIRA, M. F. Introdução à agricultura de precisão: conceitos e vantagens. *Ciência Rural*, v. 32, n. 1, p. 159–163, 2002.
- WANG, R. et al. Impacts of polymer/surfactant interactions on spray drift. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, v. 500, p. 88 – 97, 2016. ISSN 09277757. Disponível em: <<http://dx.doi.org/10.1016/j.colsurfa.2016.03.056>>.

WANG, X. et al. Development and performance test of spray drift test system for sprayer with bar. *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering*, v. 30, n. 18, p. 55 – 62, 2014. ISSN 10026819. Disponível em: <<http://dx.doi.org/10.3969/j.issn.1002-6819.2014.18.007>>.

WEISENBURGER, D. D. Human health effects of agrichemical use. *Human Pathology*, v. 24, n. 6, p. 571 – 576, 1993. ISSN 0046-8177. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0046817793902348>>.



## Apêndices



## APÊNDICE A – PARÂMETROS DOS ALGORITMOS NA CONFIRURAÇÃO EMPÍRICA

Tabela 8 – Parâmetros empregados no algoritmo RNA.

Parâmetros	Valores
Iterações Máximas (Treinamento)	1000
Neurônios da camada de entrada	4
Neurônios da camada oculta	6
Neurônios da camada de saída	1
Decadência do peso	0.00001
Função de ativação	sigmoid

Tabela 9 – Parâmetros empregados no algoritmo AR.

Parâmetros	Valores
Tipo da árvore	anova
Número de poda	0

Tabela 10 – Parâmetros empregados no algoritmo SVM.

Parâmetros	Valores
Constante de Lagrange	10
Kernel	radial
Gamma	1.5
Epsilon	00.1
Tipo da máquina	eps-regression

Tabela 11 – Parâmetros empregados no algoritmo FA.

Parâmetros	Valores
Número de árvores	1000
Proximidade	true





## APÊNDICE B – INTERVALOS DE BUSCA CONSIDERADOS PELA TÉCNICA *GRID-SEARCH*

Tabela 12 – Intervalos dos parâmetros aplicado na RNA.

Parâmetros	Intervalos
Iterações Máximas (Treinamento)	100 * (1 : 10)
Neurônios da camada oculta	1 * (1 : 15)
Decadência do peso	$10^{(-6:0)}$
Peso inicial	$10^{(-6:0)}$
Saída linear	( <i>true</i> : <i>false</i> )
Otimização	( <i>true</i> : <i>false</i> )
Valor máximo dos pesos	100 * (10 : 100)

Tabela 13 – Intervalos dos parâmetros aplicado na AR.

Parâmetros	Intervalos
Tipo da árvore	anova
Número de poda	$10^{(-6:0)}$
Número mínimo para divisão do nó	1 * (0 : 50)
Número mínimo de observação em nó folha	1 * (0 : 50)
Número de divisões retidos na saída	1 * (0 : 50)
Número de divisões de substituição	1 * (0 : 50)
Substitutos no processo de divisão	1 * (0 : 2)
Seleção do melhor substituto	1 * (0 : 1)
Profundidade máxima de um nó	1 * (0 : 30)

Tabela 14 – Intervalos dos parâmetros aplicado na SVM.

Parâmetros	Intervalos
Constante de Lagrange	1 * (1 : 30)
Kernel	<i>radial</i>
Gamma	(0 : 100/8)
Epsilon	$10^{(-6:-1)}$
Tipo da máquina	<i>eps - regression</i>

Tabela 15 – Intervalos dos parâmetros aplicado na FA.

Parâmetros	Intervalos
Número de árvores	1 * (1 : 1000)
Proximidade	( <i>true</i> : <i>false</i> )
Tamanho mínimo dos nós folhas	5
Número de variáveis candidata a divisão	1 * (1 : 100)

## APÊNDICE C – PARÂMETROS DOS ALGORITMOS NA CONFIGURAÇÃO GUIADA

Tabela 16 – Parâmetros empregados no algoritmo RNA.

Parâmetros	Valores
Iterações Máximas (Treinamento)	1000
Neurônios da camada de entrada	4
Neurônios da camada oculta	6
Neurônios da camada de saída	1
Decadência do peso	0.00005
Peso inicial	0.1
Saída linear	true
Otimização	false
Valor máximo dos pesos	10000
Função de ativação	sigmoid

Tabela 17 – Parâmetros empregados no algoritmo AR.

Parâmetros	Valores
Tipo da árvore	anova
Número de poda	0
Número mínimo para divisão do nó	0
Número mínimo de observação em nó folha	12
Número de divisões retidos na saída	0
Número de divisões de substituição	0
Substitutos no processo de divisão	0
Seleção do melhor substituto	0
Profundidade máxima de um nó	3

Tabela 18 – Parâmetros empregados no algoritmo SVM.

Parâmetros	Valores
Constante de Lagrange	2
Kernel	radial
Gamma	0.125
Epsilon	0.0000001
Tipo da máquina	eps-regression

Tabela 19 – Parâmetros empregados no algoritmo FA.

Parâmetros	Valores
Número de árvores	1000
Proximidade	true
Tamanho mínimo dos nós folhas	5
Número de variáveis candidata a divisão	3

## APÊNDICE D – VARIABILIDADE DOS ALGORITMOS EM DIFERENTES CONFIGURAÇÕES

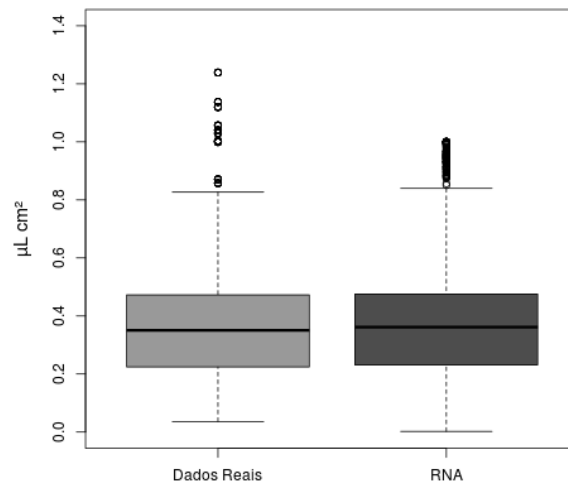


Figura 17 – Variabilidade do algoritmo RNA na configuração empírica.

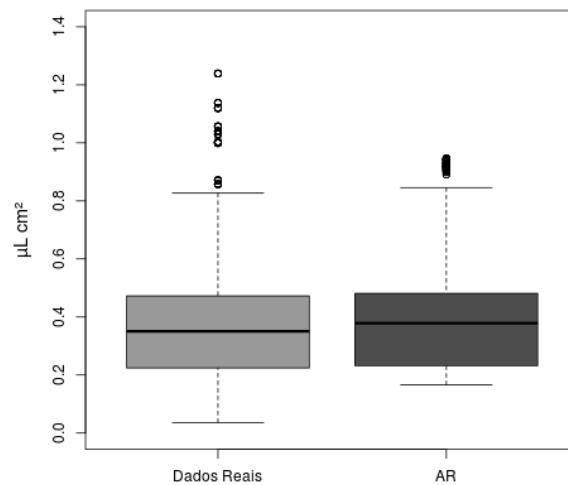


Figura 18 – Variabilidade do algoritmo AR na configuração guiada.



## APÊNDICE E – PREDIÇÕES E VARIABILIDADES DOS ALGORITMOS NA CONFIGURAÇÃO EMPÍRICA E GUIADA

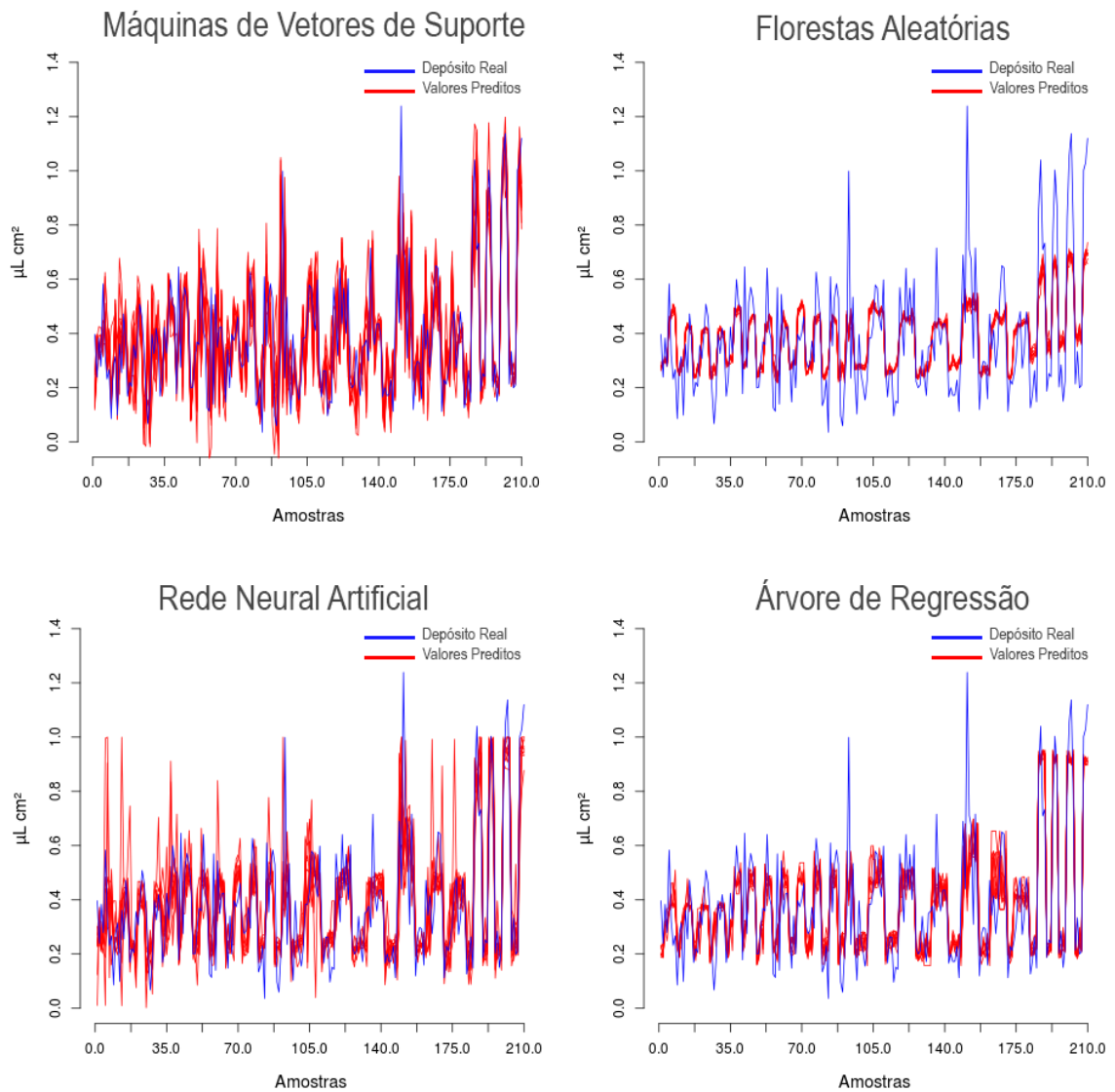


Figura 19 – Predições realizadas por cada algoritmo na configuração empírica.

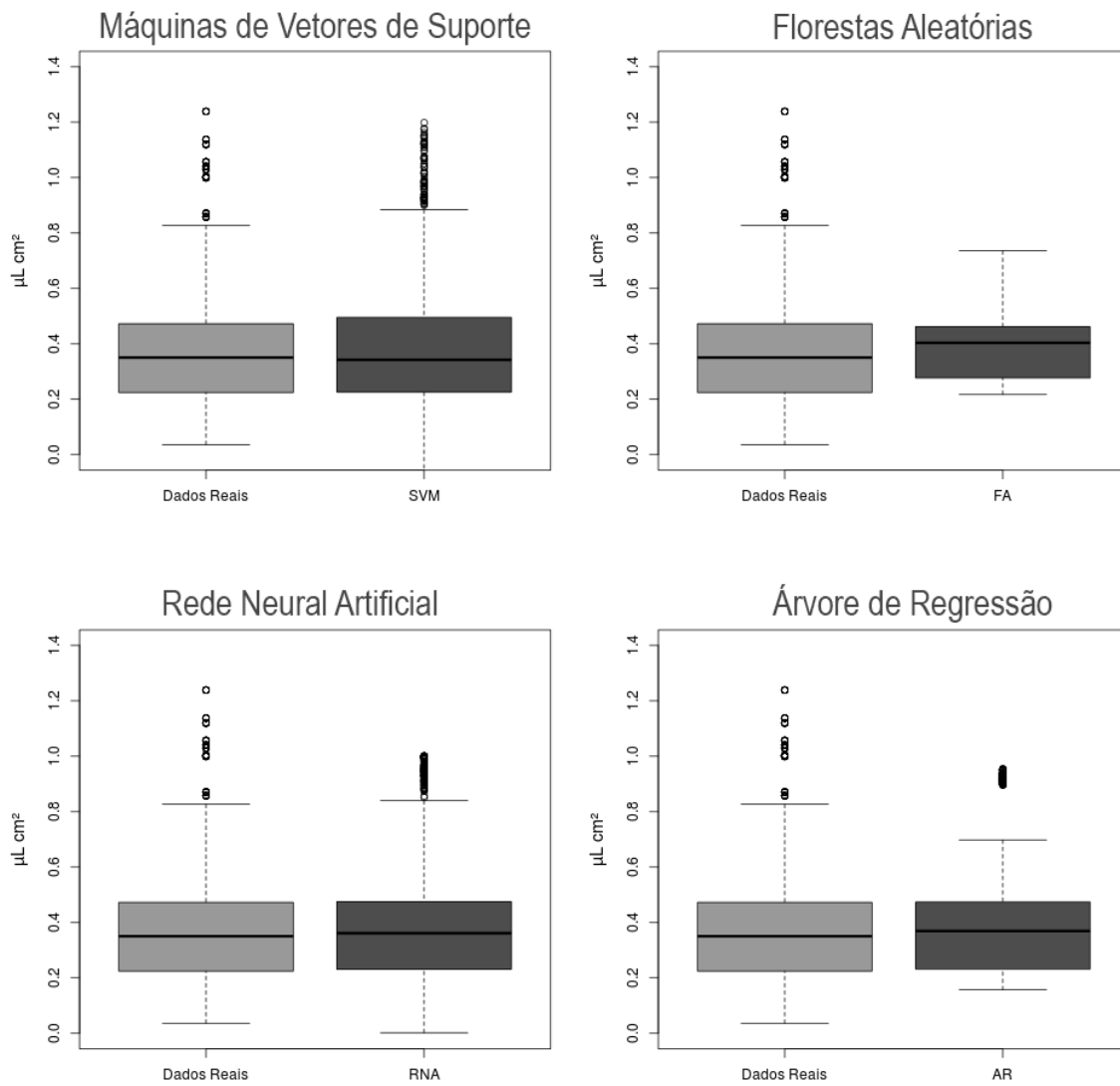


Figura 20 – Variabilidade de cada algoritmo na configuração empírica.



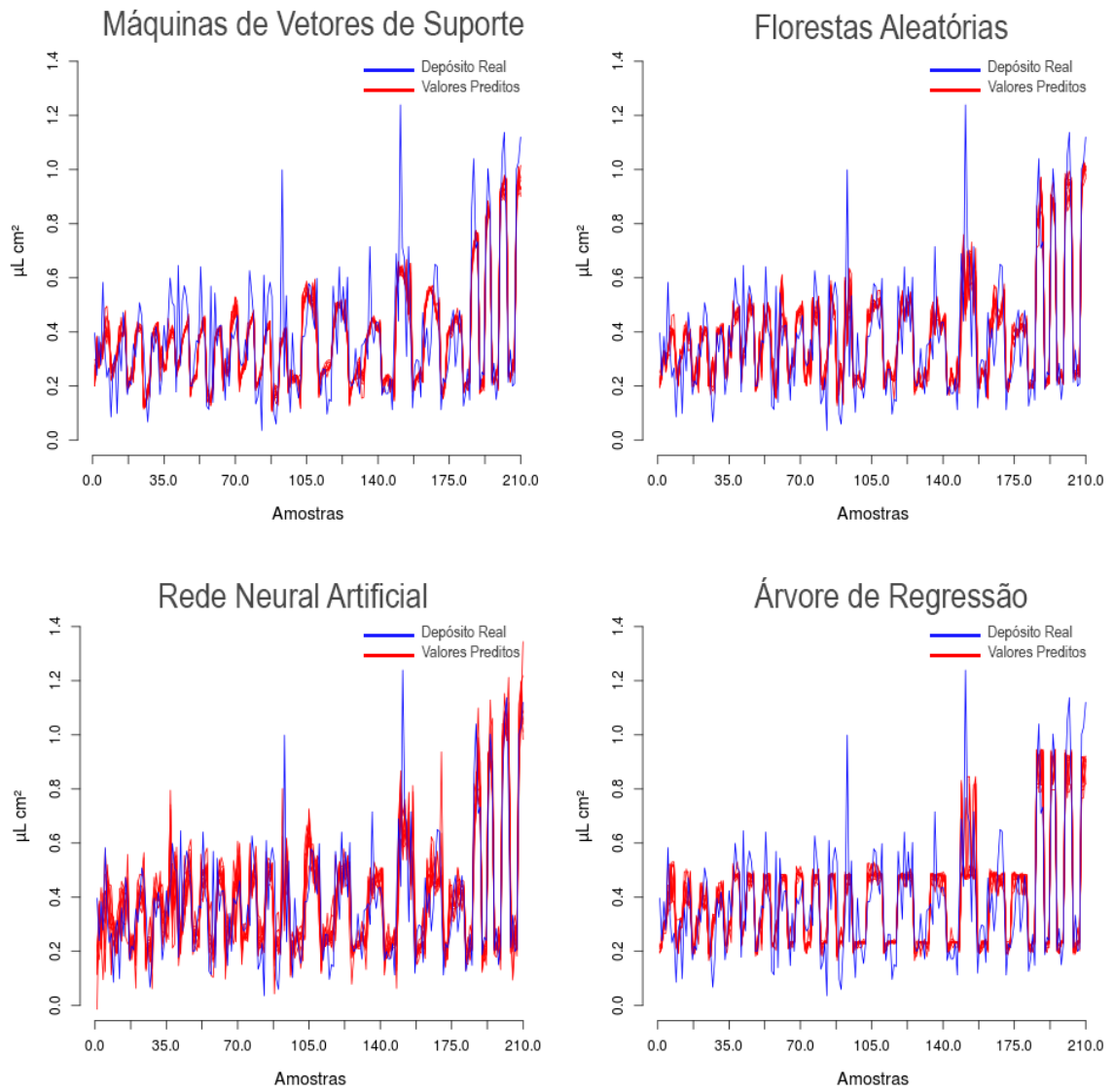


Figura 21 – Predições realizadas por cada algoritmo na configuração guiada.

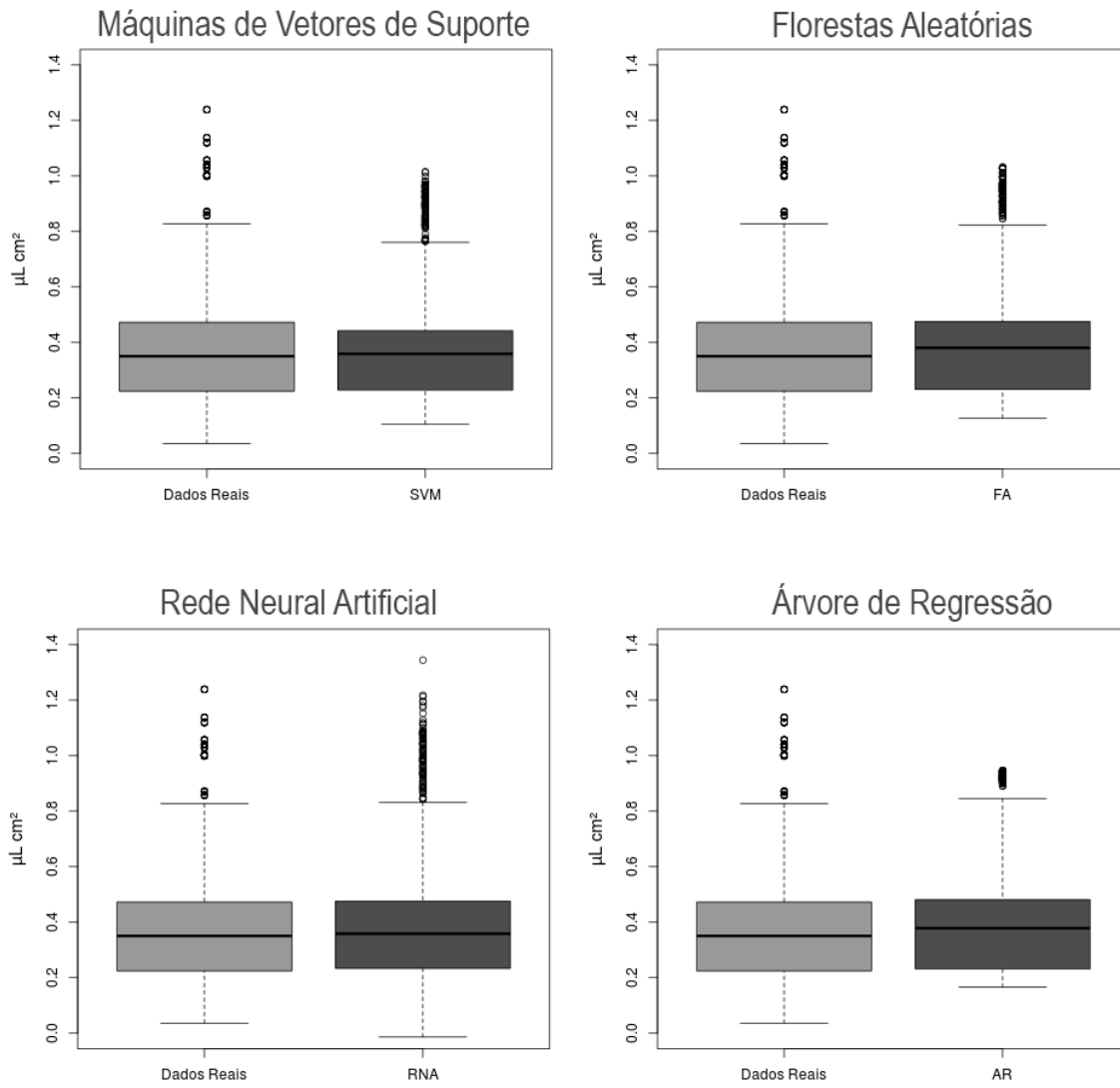


Figura 22 – Variabilidade de cada algoritmo na configuração guiada.