UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAFAEL DE BONA FERRO

ARQUITETURA HÍBRIDA DE MÁQUINAS DE VETORES DE SUPORTE E REDES NEURAIS ARTIFICIAIS APLICADA A CLASSIFICAÇÃO DOS SOLOS

CRICIÚMA
2018

RAFAEL DE BONA FERRO

ARQUITETURA HÍBRIDA DE MÁQUINAS DE VETORES DE SUPORTE E REDES NEURAIS ARTIFICIAIS APLICADA A CLASSIFICAÇÃO DOS SOLOS

Projeto de Pesquisa do Trabalho de Conclusão de Curso em Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientadora: Profª. Dra. Merisandra Côrtes de Mattos Garcia

CRICIÚMA

2018

RAFAEL DE BONA FERRO

ARQUITETURA HÍBRIDA DE MÁQUINAS DE VETORES DE SUPORTE E REDES NEURAIS ARTIFICIAIS APLICADA A CLASSIFICAÇÃO DOS SOLOS

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Inteligência Computacional.

Criciúma, 28 de novembro de 2018.

BANCA EXAMINADORA

Prof. Merisandra Côrtes de Matos Garcia - Doutora - (UNESC) - Orientadora

Prof. Gustavo Bisognin - Mestre - (UNESC)

Prof. Kristian Madeira - Doutor - (UNESC)

Aos meus pais Isauro e Edite, meus irmãos e toda minha família.

AGRADECIMENTOS

A Deus por ter me dado saúde e força para superar as dificuldades.

Gostaria de agradecer aos meus pais Isauro e Edite por terem me dado a chance de uma educação de qualidade, sempre me incentivaram e não mediram esforços para que eu concluísse a graduação.

Agradeço a minha família, minha irmã Gisele e meu irmão André, por sempre me apoiarem. Gostaria de agradecer especialmente meu cunhado Marcelo, pelos seus conselhos.

Agradeço a minha namorada Ana pela paciência e pelo seu apoio nessa minha jornada.

Agradeço ao meu amigo Adão que esteve presente nos momentos de dificuldades e que me apoiou durante todo o curso.

Agradeço a minha orientadora, professora Merisandra, por ter me sugerido esta pesquisa que foi de um grande aprendizado para minha vida acadêmica e profissional, por ter me acalmado nos momentos de nervosismo e também por sempre me incentivar a fazer o meu melhor.

Agradeço aos membros da banca que avaliaram este trabalho, os professores Gustavo e Kristian.

Agradeço ao Fernando Basquiroto que me auxiliou com o conhecimento dele sobre o domínio de aplicação da pesquisa.

Agradeço a todos que direta e indiretamente fizeram parte da minha formação.

RESUMO

O aprendizado de máquina é uma área da inteligência artificial que tem como objetivo desenvolver aplicações que possuem a característica de aprender com suas experiências. Para isso, podem ser utilizadas diversas técnicas entre elas destacam-se as redes neurais artificiais e as máquinas de vetores de suporte. Redes neurais artificiais são modelos computacionais inspirados no cérebro, que se originaram dos estudos sobre a teoria psicológica do aprendizado em animais. As máquinas de vetor de suporte utilizam medidas estatísticas para traçar retas que são empregadas para separar conjuntos de dados com a maior distância entre si. Esta técnica vem sendo aplicada com sucesso em várias áreas, como por exemplo, em Engenharia Ambiental e Geologia. A classificação dos solos é um processo com alto custo e trabalhoso, na qual algumas vezes é necessário a predição de alguns dados. A predição das propriedades do solo quando realizada por meio da inteligência artificial possui resultados melhores do que quando empregada a forma tradicional. As máquinas de vetores de suporte apesar de serem eficientes nas resoluções de problemas de classificação, possui uma desvantagem que é o alto tempo despendido nas fases de treinamento e execução. Esta pesquisa teve como objetivo o desenvolvimento de uma arquitetura híbrida de máquinas de vetores de suporte e redes neurais artificiais aplicada a classificação dos solos. Foi utilizada uma base de dados sobre a classificação dos solos do estado de Santa Catarina, aplicando primeiramente as máquinas de vetores de suporte de forma isolada e posteriormente foi desenvolvida a arquitetura híbrida com redes neurais artificiais do tipo *Kohonen*. O hibridismo foi realizado empregando-se o modelo de rede neural artificial *Kohonen* para o agrupamento dos dados e aplicando-se uma máquina de vetores de suporte para cada um dos grupos gerados a fim de se classificar os solos. Posteriormente, por meio de métodos estatísticos avaliou-se o desempenho da aplicação das máquinas de vetores de suporte e da arquitetura híbrida, considerando-se os parâmetros de taxa de erro, tempo de treinamento, tempo de execução, acurácia, entre outras medidas de avaliação de um classificador. Foram realizados testes com diferentes quantidades de grupos gerados pela rede neural artificial, sendo elas dois, quatro, cinco, oito e quinze grupos. Ocorreram pequenas melhoras na arquitetura com dois grupos em relação a qualidade do classificador. Aumentando-se a quantidade de grupos para maior que dois, o tempo de execução apresentou melhoras significativas quando comparado com a execução das máquinas de vetores de suporte isoladamente, no entanto observou-se uma diminuição na precisão -da classificação. Nesta pesquisa, a arquitetura híbrida teve seu tempo de execução otimizado, no entanto, não foi significativo estatisticamente. Em relação a acurácia dos modelos gerados, esta se manteve a mesma nos modelos híbridos e de máquinas de vetores de suporte, que foi de aproximadamente 77,5%.

Palavras-chave: Aprendizado de máquina**.** Redes Neurais Artificiais. Máquinas de Vetores de Suporte. Arquitetura Híbrida. Classificação dos Solos.

ABSTRACT

Machine learning is a field of artificial intelligence that has the objective to develop applications that have the ability to learn from past experiences. For this, several techniques can be used, among them the most common techniques are artificial neural networks and support vector machines. Artificial neural networks are brain-based computational models originated from studies of the psychological theory of animal learning. Support vector machines use statistical measures to draw lines that are employed to separate data sets with the longest distance from each other. This technique has been applied successfully in several areas, such as Environmental Engineering and Geology. Soil classification is a costly and labor-intensive process which sometimes data needs to be predicted. The prediction of soil properties when performed through artificial intelligence has better results than when performed by traditional way. Support vector machines, although efficient in resolving classification problems, has a disadvantage which is the high amount of time spent in the training and execution phases. This research proposed to develop a hybrid architecture of support vector machines and artificial neural networks applied to soil classification. A database was used on the classification of soils of the state of Santa Catarina, first applying the support vector machines and later the hybrid architecture was developed with Kohonen's artificial neural networks. The hybrid architecture was developed using the Kohonen's artificial neural network to group the data and applying a support vector machine to each group generated in order to classify the soils. Afterward, through statistical methods, the performance of the support vector machines and the hybrid architecture was evaluated, regarding the parameters of error rate, training time, execution time, accuracy and others evaluation measures of a classifier. Tests were executed with different amounts of groups generated by the artificial neural network, using two, four, five, eight and fifteen groups. There were small improvements in the architecture with two groups regarding the quality of the classifier, also was identified that within the increase of the number of groups for more than two, the execution time presented significant improvements when compared to the execution of the support vector machines, however this improvement cost accuracy in the classification. In this research, the hybrid architecture has improved the execution time, however the improvement was not significant, regard the accuracy of the hybrid architecture it remained the same as the support vector machines, which were approximately 77.5%.

Keywords: Machine Learning. Artificial Neural Networks. Support Vector Machines. Hybrid Architecture. Soil Classification.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
AM	Aprendizado de Máquina
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
IA	Inteligência Artificial
RBF	*Radial Basis Function*
RNA	Redes Neurais Artificiais
SiBCS	Sistema Brasileiro de Classificação de Solos
SVM	*Support Vector Machines*
TAE	Teoria do Aprendizado Estatístico

SUMÁRIO

1 INTRODUÇÃO

O avanço da tecnologia com o aumento significativo da capacidade de armazenamento e processamento dos dados tem possibilitado a utilização de sofisticados métodos estatísticos e de aprendizado de máquina (KOVAČEVIĆ et al., 2010, tradução nossa).

O aprendizado de máquina é uma área da inteligência artificial que tem como objetivo desenvolver aplicações que possuem a característica de aprender com suas experiências. Para isso, podem ser utilizadas diversas técnicas entre elas destacam-se as redes neurais artificiais e as máquinas de vetores de suporte (LUGER, 2013).

Redes neurais artificiais são modelos computacionais inspirados no cérebro, que se originaram do estudo de Hebb (1949) sobre a teoria psicológica do aprendizado em animais. Estas redes são estruturadas por componentes simples e interativos que por meio de processos de aprendizagem têm ajustadas as conexões entre si. Esta técnica baseia-se no funcionamento do neurônio artificial em que existe um vetor de entradas e uma função que determina a saída do neurônio (LUGER, 2013).

As máquinas de vetor de suporte utilizam medidas estatísticas para traçar retas que são utilizadas para separar conjuntos de dados com a maior distância entre si (LUGER, 2013). Esta técnica vem sendo aplicada com sucesso em várias áreas, como por exemplo, Engenharia Ambiental (SALVADOR; CHOU, 2014, tradução nossa), Química (BASSBASI et al., 2014, tradução nossa), Biologia (NATH; SUBBIAH, 2016, tradução nossa), assim como na área da Geologia (CONSENZA et al., 2015; XUE; YANG, 2015, tradução nossa).

Na área da geologia tem-se a classificação dos solos que é um processo laboratorial trabalhoso e com alto custo, que por vezes devido à falta de amostras suficientes é necessária a utilização de técnicas de predição (KOVAČEVIĆ et al., 2010, tradução nossa). A predição das propriedades do solo quando realizada por meio da inteligência computacional possui resultados melhores do que quando empregada a forma tradicional (SOARES et al., 2014).

As máquinas de vetores de suporte apesar de serem eficientes nas resoluções de problemas de classificação, possui uma desvantagem que é o alto

tempo despendido nas fases de aprendizagem e testes (KANG; CHO, 2014, tradução nossa). Com o intuito de otimizar a aplicação desta técnica algumas abordagens híbridas com outras técnicas de inteligência computacional já foram aplicadas, como por exemplo, com algoritmos genéticos (CHOU et al., 2014, tradução nossa) e com redes neurais artificiais (KANG; CHO, 2014, tradução nossa; NIU; SUEN, 2011, tradução nossa).

Desta forma, nesta pesquisa utilizou-se uma arquitetura híbrida entre redes neurais artificiais e máquinas de vetores de suporte a fim de melhorar sua performance, para isso a técnica foi aplicada em uma base de dados de classificação do solo da região de Santa Catarina, em seguida por meio de métodos estatísticos foi avaliado o desempenho da arquitetura híbrida.

1.1 OBJETIVO GERAL

Desenvolver uma arquitetura híbrida de Máquinas de Vetores de Suporte e Redes Neurais Artificiais aplicada a classificação dos solos.

1.2 OBJETIVO ESPECÍFICO

Os objetivos específicos da pesquisa consistem em:

a) descrever as máquinas de vetores de suporte, redes neurais artificiais e arquitetura híbrida;
b) aplicar a técnica de máquinas de vetores de suporte para classificação dos solos;
c) aplicar o hibridismo entre máquinas de vetores de suporte e rede neural artificial para classificação dos solos;
d) empregar métodos estatísticos para avaliação do desempenho do hibridismo proposto para classificação dos solos;
e) analisar se o hibridismo proposto melhorou o tempo de aprendizagem e teste das máquinas de vetores de suporte na problemática de classificação dos solos.

1.3 JUSTIFICATIVA

O aprendizado de máquina atualmente é uma das áreas que mais cresce, seja pelo desenvolvimento de novas técnicas e teorias de aprendizado ou pela disponibilidade de grande quantidade de dados online e baixo custo da computação. O emprego do aprendizado de máquina pode ser visto em diversas áreas, como por exemplo, ciência, tecnologia e comércio (JORDAN; MITCHELL, 2015, tradução nossa). Dentre as técnicas de aprendizado de máquina, as máquinas de vetores de suporte e as redes neurais artificiais são empregadas nesta pesquisa.

As Redes Neurais Artificiais (RNA), modelos computacionais que compartilham algumas propriedades do cérebro humano, conseguem obter performances aceitáveis em tarefas que quando aplicadas outras técnicas da inteligência computacional exigem complexidades maiores para obter resultados semelhantes (RUSSEL; NORVIG, 2013).

As Máquinas de Vetores de Suporte, do inglês *Support Vector Machines* (SVM), são um dos métodos de aprendizado de máquina mais utilizados, conseguindo alcançar uma performance superior em generalização tanto para problemas de classificação quanto para regressão (GOPAL; KUMAR, 2011, tradução nossa; KANG; CHO, 2014, tradução nossa). SVM tem sido utilizado com sucesso em várias aplicações, como por exemplo, no reconhecimento de escrita à mão, previsão financeira (BURGES, 1998; SMOLA; SCHÖLKOPF, 2004, tradução nossa), predição de estruturas terciárias de proteínas (BISOGNIN, 2007), classificador de qualidade de energia (YONG et al., 2015), predição de tendência no mercado de ações (BALLINGS et al., 2015), como também na classificação e estimativas de propriedades dos solos (DEBAENE et al., 2014, tradução nossa; KOVAČEVIĆ et al., 2010, tradução nossa; PANDEY et al., 2010, tradução nossa) .

No que se refere as arquiteturas híbridas, estas visam compensar as limitações das técnicas envolvidas, sejam melhorias em sua precisão ou em seu desempenho. Recentemente, vários estudos relacionados a aplicação de hibridismo entre técnicas de inteligência computacional vem sendo realizados, como por exemplo, hibridismo de SVM e RNA aplicado ao reconhecimento de dígitos escritos à mão (NIU; SUEN, 2014).

De acordo com os estudos de Debaene et al. (2014, tradução nossa), Kovačević et al. (2010, tradução nossa) e Pandey et al. (2010, tradução nossa) o SVM vem sendo aplicado com sucesso na classificação dos solos. No entanto, devido ao SVM possuir uma desvantagem que é o alto tempo consumido nas fases de aprendizagem e testes (KANG; CHO, 2014, tradução nossa), busca-se nesta pesquisa disponibilizar uma alternativa para esta problemática com a aplicação de modelos de inteligência computacional de arquitetura híbrida, no caso entre o SVM e as RNA, pois pode-se agregar as vantagens das duas técnicas, como a capacidade superior de generalização do SVM e o menor tempo consumido nas fases de aprendizado e testes da RNA.

Dentre os fatores que levaram a escolha da classificação dos solos, está o fato de que a região catarinense passou por atividades mineradoras o que trouxe pesquisas sobre o solo na região (TREIN, 2008), sendo assim a região dispõe de dados referente a classificação dos solos.

1.4 ESTRUTURA DO TRABALHO

Esta pesquisa é dividida em cinco capítulos, sendo que o primeiro capítulo possui uma introdução, os objetivos gerais e específicos e a justificativa da pesquisa.

O capítulo dois trata do aprendizado de máquina com uma breve introdução das formas de aprendizado, são apresentadas as técnicas de máquinas de vetores de suporte e seu funcionamento. Redes neurais artificiais englobando seu conceito, suas aplicações e suas divisões, também são demonstradas algumas formas de hibridismo entre técnicas de inteligência computacional. Ainda neste capitulo são demonstradas as diferentes medidas de qualidade que podem ser empregadas para se avaliar um classificador.

O terceiro capítulo traz os trabalhos que influenciaram na fundamentação desta pesquisa. No quarto capitulo é descrito o trabalho desenvolvido nesta pesquisa, na qual conta com uma breve explicação sobre a classificação dos solos e também a metodologia utilizada para o desenvolvimento, pré-processamento dos dados, a aplicação das máquinas de vetores suporte, desenvolvimento e aplicação da arquitetura híbrida, desenvolvimento de uma interface gráfica, resultados obtidos e discussão dos mesmos.

No quinto e último capítulo é apresentada a conclusão do trabalho e sugestões de trabalhos futuros.

2 APRENDIZADO DE MÁQUINA

A Inteligência Artificial (IA), de acordo com Coppin (2010), consiste na aplicação de métodos que modelam o comportamento dos seres inteligentes, sejam humanos ou animais, na resolução de problemas complexos. Inicialmente constituía-se em um domínio de conhecimento massivamente teórico da computação (FACELI et al., 2011; RICH; KNIGHT; NAIR, 2009), no entanto, devido ao aumento significativo na quantidade de dados a ser analisada e também na capacidade de processamento dos computadores, originaram-se os sistemas autônomos, como os que empregam aprendizado de máquina, que diminuem a dependência de especialistas humanos (FACELI et al., 2011).

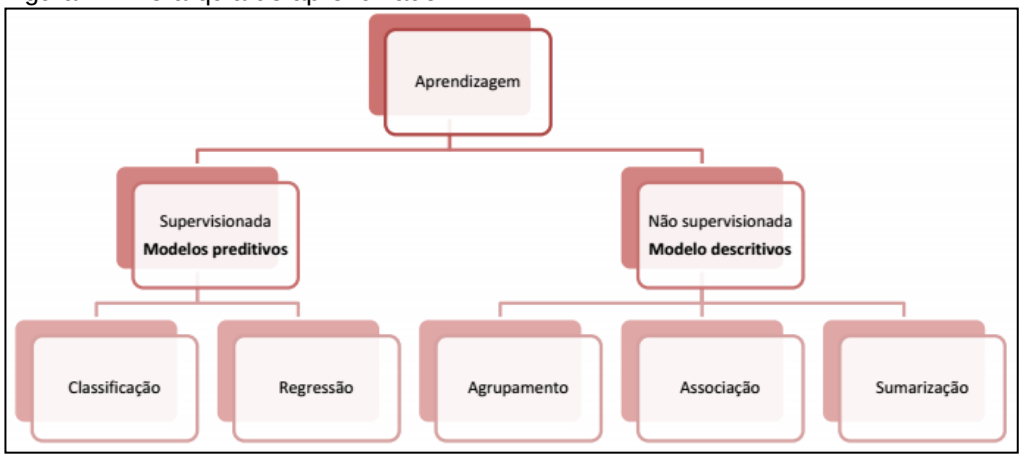
Aprendizado de máquina é uma solução computacional que se aperfeiçoa com suas próprias experiências. A utilização dessas aplicações torna-se mais eficiente do que a codificação humana, pois mesmo demandando tempo na aplicação, não se consegue prever todas as entradas e cenários. Assim, o sistema pode não se adaptar a entradas diferentes e possíveis mutações, problema esse que é minimizado em máquinas inteligentes (LUGER, 2013; MITCHELL, 1997, tradução nossa).

O emprego do Aprendizado de Máquina (AM) pode ser visto em diversas áreas, da ciência, tecnologia e comércio (JORDAN; MITCHELL, 2015, tradução nossa). É utilizado em segurança computacional, como proteção de redes, reconhecimento facial e impressões digitais, entre outros (LOURIDAS; EBERT, 2016, tradução nossa). A sua aplicação ocorre em outros domínios de estudo, como por exemplo, para a predição de estruturas terciárias de proteínas (BISOGNIN, 2007); classificador de qualidade de energia (YONG; BHOWMIK; MAGNAGO, 2015, tradução nossa); predição de tendência no mercado de ações (BALLINGS et al., 2015); classificação e estimativas de propriedades dos solos (DEBAENE et al., 2014, tradução nossa; KOVAČEVIĆ et al., 2010, tradução nossa; PANDEY et al., 2010, tradução nossa).

O AM tem sido empregado em diferentes tarefas como as preditivas e descritivas. O modelo preditivo busca encontrar uma hipótese, a partir dos pares de entrada e saída, que consiga prever um novo resultado, seja por classificação ou regressão. Já o modelo descritivo tem como objetivo descrever um grupo de dados, relacionando-os entre si, para isto pode utilizar técnicas de agrupamento, associação

e sumarização (FACELI et al., 2011). Segundo Coppin (2010), estas tarefas compõem as duas principais abordagens de aprendizado de máquina. Os modelos preditivos referem-se ao aprendizado supervisionado, enquanto os descritivos são conhecidos como não supervisionado (figura 1).

Figura 1 - Hierarquia de aprendizado.



Fonte: Adaptado de Faceli et al. (2011, p. 6).

Algumas tarefas não correspondem a hierarquia da figura 1, como por exemplo, os aprendizados semissupervisionado, ativo e por reforço (FACELI et al., 2011).

O aprendizado semissupervisionado consiste na aplicação de uma abordagem intermediária entre o aprendizado supervisionado e o não supervisionado (ZHU et al., 2009, tradução nossa). Possibilitando assim a aplicação de aprendizado supervisionado nos casos em que há uma pequena quantidade de elementos de saída já rotulados e uma grande quantia de elementos ainda sem classificação (SANCHES, 2003).

Segundo Faceli et al. (2011) a técnica de aprendizado semissupervisionado pode ser utilizada tanto para o agrupamento, quanto para a classificação ou regressão. Aplica-se o agrupamento quando se tem o conhecimento de que alguns dados devem pertencer a um determinado grupo, já para a classificação é aplicado quando uma pequena porção dos dados possuem as classes definidas.

No aprendizado ativo o sistema desenvolve novas hipóteses para serem testadas, tornando assim o processo de aprendizagem iterativo (SETTLES, 2012, tradução nossa). Nesta abordagem o sistema encontra uma função no conjunto de dados já rotulados e a aplica nos dados não rotulados, buscando assim otimizar a função a fim de minimizar erros (TONG; KOLLER, 2001).

No aprendizado por reforço, o sistema não recebe instruções de quais ações deve tomar, descobrindo qual a melhor ação de acordo com o *feedback* (positivo ou negativo) recebido a cada ação (SUTTON; BARTO, 1998, tradução nossa). Esta abordagem é comum em sistemas classificatórios, sendo também aplicada em algumas redes neurais artificiais (COPPIN, 2010). As abordagens de aprendizado supervisionado e não supervisionado são as mais utilizadas.

No aprendizado não supervisionado, a aplicação geralmente realiza uma espécie de união por semelhança, sem que seja fornecida anteriormente a saída (RUSSEL; NORVIG, 2013). A aplicação constrói uma representação considerando padrões e tendências, sem que seja guiado por elementos externos ao sistema (FACELI et al., 2011). Nesta abordagem, pode-se utilizar tarefas de agrupamento, associação e sumarização.

O agrupamento visa unir os dados conforme suas similaridades, isto resulta em grupos de dados ou *clusters*. Nesta técnica cada algoritmo analisa as características para definir uma estrutura que descreva os dados da melhor forma (FACELI et al., 2011). Para Luger (2013) ao utilizar o agrupamento é importante atribuir pesos diferentes para as características de modo que aquelas mais importantes para a aplicação em questão terão maiores pesos. O agrupamento é utilizado em diversos domínios, como por exemplo, análise de dados em redes sociais.

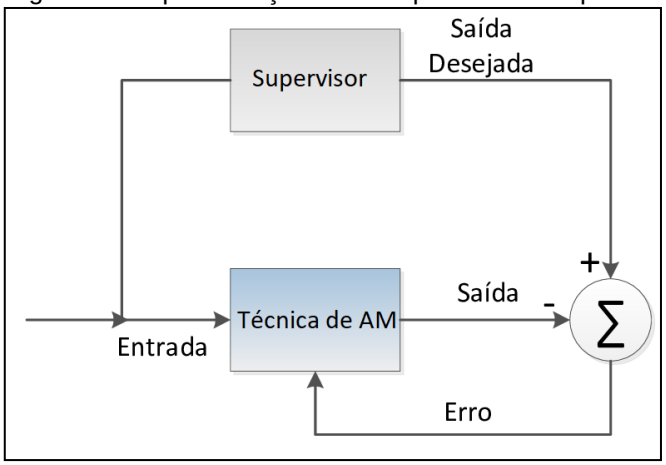
A associação consiste em encontrar regras que acontecem frequentemente em uma base de dados. Formalmente pode-se representar essa técnica de acordo com a fórmula $X \rightarrow Y$, sendo X e Y conjuntos de dados, tal que $X \cap Y = \emptyset$. A aplicação dessa técnica pode ser vista na descoberta de produtos que geralmente são vendidos na mesma transação em um supermercado (GOLDSCHIMIDT; PASSOS, 2005).

A sumarização busca resumir e generalizar os dados dos conjuntos de forma clara, identificando os seus principais atributos. Toda via é importante ressaltar que este processo não é apenas uma enumeração de dados, pois permite comparar e discriminar os mesmos. Um exemplo de aplicação dessa técnica é a busca do perfil de assinantes de uma revista (GOLDSCHIMIDT; PASSOS, 2005).

No aprendizado não supervisionado por meio da tarefa de agrupamento pode-se empregar dentre outras técnicas, as redes neurais artificiais, as quais são aplicadas nesta pesquisa. As técnicas de redes neurais artificiais podem ser aplicadas tanto no aprendizado não supervisionado quanto no aprendizado supervisionado.

No aprendizado supervisionado a aplicação em fase de treinamento recebe um conjunto de dados que já possui uma saída conhecida, tendo como objetivo encontrar uma função que relacione os dados de entrada com a saída, que posteriormente será aplicada a novos dados. A representação dessa abordagem pode ser vista na figura 2 (RUSSEL; NORVIG, 2013).

Figura 2 - Representação de um aprendizado supervisionado



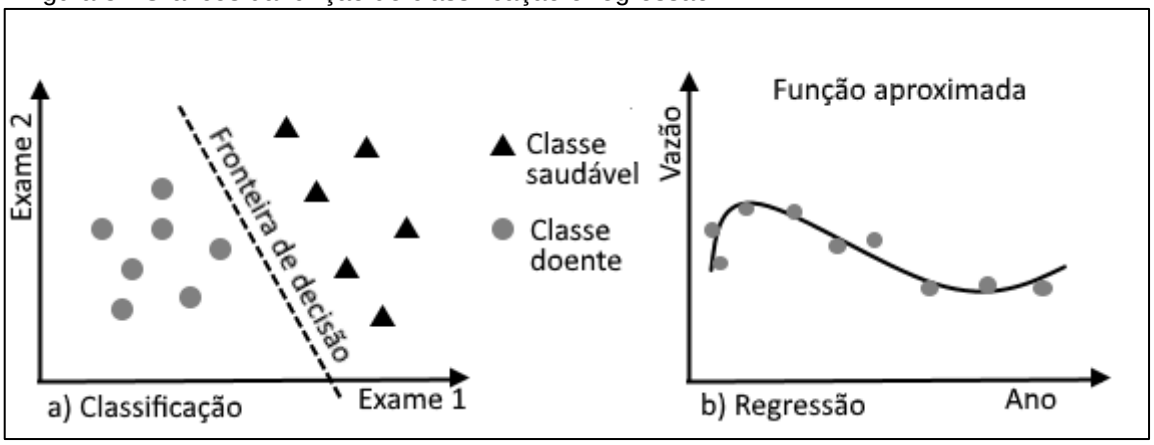
Fonte: Adaptado de Braga, Ludemir e Carvalho (2000).

Entre as tarefas de aprendizado supervisionado destacam-se a regressão e a classificação. A regressão tem como meta encontrar uma função estimada que se aproxime o máximo possível dos valores reais, assumindo assim valores de um conjunto infinito, um exemplo da aplicação dessa técnica é aprender uma função que estime o índice de uma de bolsa de valores baseando-se em indicadores econômicos (FLACH, 2012, tradução nossa).

A classificação, segundo Goldschimidt e Passos (2005), busca encontrar elementos discretos. Esta é a técnica mais comum de aprendizado de máquina supervisionado e tem como objetivo categorizar as entradas em grupos finitos de classes. Algumas vezes é aplicada a classificação binária, onde o resultado pode ser sim ou não, zero ou um. Em outras aplicações são utilizadas classificações com várias classes, podendo ter vários resultados discretos (FABRIS; MAGALHÃES; FREITAS, 2017, tradução nossa). Atualmente a tarefa de classificação pode ser aplicada em vários domínios, como por exemplo, detecção facial.

Na figura 3, pode-se visualizar a diferença entre a função de classificação e regressão, onde a classificação visa separar os dados em duas classes distintas e a regressão busca uma função que aproxime os valores.

Figura 3 - Gráficos da função de classificação e regressão



Fonte: Faceli et al. (2011).

No aprendizado supervisionado por meio da tarefa de classificação pode-se empregar dentre outras técnicas, as máquinas de vetores de suporte e as redes neurais artificiais, as quais são aplicadas nesta pesquisa, no entanto a técnica de rede neural artificial empregada nesta pesquisa utiliza a tarefa de agrupamento do aprendizado não supervisionado.

2.1 MÁQUINAS DE VETORES DE SUPORTE

As máquinas de vetores de suporte são uma técnica de aprendizado de máquina utilizada para a classificação de padrões e regressão linear. Esta técnica tem como objetivo traçar retas que separem dois conjuntos com a maior distância possível (HAYKIN, 2001).

Atualmente vêm sendo aplicadas com sucesso em diferentes áreas, como por exemplo, na medicina, auxiliando nos diagnósticos do câncer de mama (WANG et. al., 2018, tradução nossa) e da doença de Alzheimer (BI et. al., 2018, tradução nossa); na tecnologia, para a detecção de atividade por voz (WU; ZHANG, 2011, tradução nossa), no reconhecimento facial (OWUSU; ZHAN; MAO, 2014, tradução nossa), em sistema gerenciador de relação com o cliente (FARQUAD; RAVI; RAJU, 2014, tradução nossa) e na detecção de intrusão em redes de computadores (ANBAR et al., 2018, tradução nossa; KABIR et al., 2018, tradução nossa); na geologia, por meio de modelo de classificação da qualidade do solo (LIU et. al., 2015, tradução nossa), e de classificação dos solos (GUANG et. al., 2015, tradução nossa).

As SVM são fundamentadas na Teoria do Aprendizado Estatístico (TAE), esta teoria estabelece princípios matemáticos para encontrar classificadores com boa capacidade de generalização (LUGER, 2013).

Na TAE sendo um classificador h e o conjunto de todos os classificadores que podem ser gerados representado por H, obtém-se um classificador particular que pode ser denominado por $\hat{h}$ pertencente a H. A escolha desse $\hat{h}$ é baseada no desempenho do classificador no treinamento e na complexidade (LORENA, CARVALHO, 2007).

Ao se escolher o classificador deve-se considerar o risco esperado e empírico, sendo o primeiro a probabilidade de ocorrer um erro na classificação de acordo com o total de conjuntos, ou seja, considerando Y um número finito de conjuntos e X um dado pertencente a um desses conjuntos então $P(x, y)$. Enquanto o risco empírico é a probabilidade de o classificador errar na classificação dos dados de treinamento (FACELI et al., 2011).

O risco empírico é importante, pois é uma das formas de medir o desempenho do classificador, para calcular este risco são considerados alguns valores como o número de vezes que o classificador foi executado e o somatório da função de custo (equação 1)**.** Uma das funções de custo mais aplicadas em tarefas de classificação é a 0-1, nesta o resultado será 0, caso tenha sido classificado corretamente e 1 se incorreto (LORENA, CARVALHO, 2007).

$$R_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n} c(h(x_i), y_i) \qquad (1)$$

Por exemplo, sendo um classificador de duas classes, que armazene todos os dados de treinamento e gere classificação para novos dados, o seu risco empírico é zero, pois foram armazenados todos os dados de treinamento e seu risco esperado é 0,5 pois trata-se da probabilidade de uma classificação incorreta. O risco esperado não pode ser minimizado, toda via o risco empírico pode ser minimizado utilizando o princípio de indução. Desta forma a TAE estipula limites no risco esperado de uma função que é utilizada na escolha do classificador (LORENA, CARVALHO, 2007).

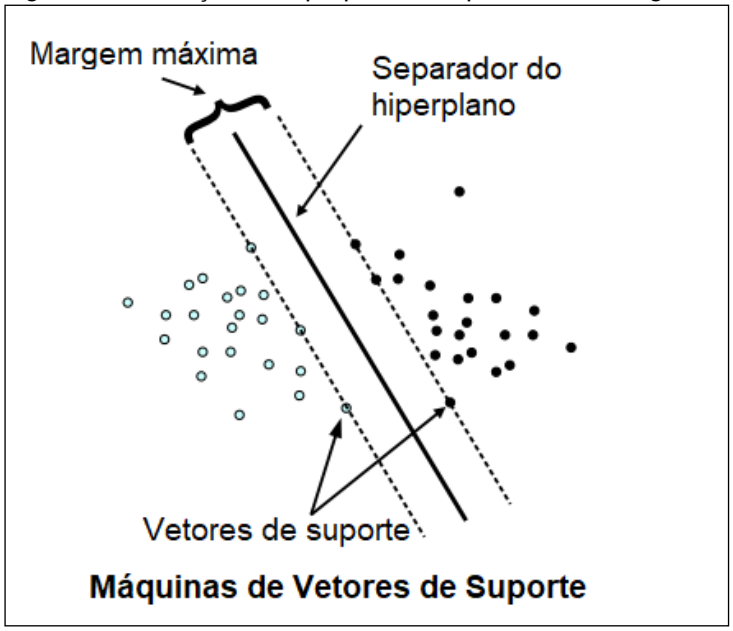
Para que um classificador seja considerado ótimo, o hiperplano deve manter um equilíbrio entre a maximização da margem e a obtenção de um erro

marginal baixo. Uma vez que uma máquina muito simples não memorizará o conjunto de treinamento, por outro lado uma máquina muito complexa, não conseguirá generalizar os dados novos (VAPNIK, 1998, tradução nossa).

Segundo Russell e Norvig (2013), esse método tornou-se um dos mais empregados na abordagem de aprendizado supervisionado, sendo indicado quando há pouco ou nenhum conhecimento sobre o domínio em que será aplicado. O crescente uso deste método deve-se também a três propriedades: o separador de margem máxima, a função de *kernel* e o fato dele ser um método não paramétrico.

O separador de margem máxima auxilia na generalização fornecendo um limite com a maior distância entre os conjuntos. A função de *kernel* é uma separação linear em hiperplano utilizado quando a separação dos dados no plano de entrada se torna impossível, porém esses dados podem ser separados em uma outra dimensão (figura 4). O método não paramétrico refere-se ao SVM guardar apenas uma pequena parte de exemplos, podendo assim representar funções complexas sem superadaptar os dados de treinamento (RUSSELL; NORVIG, 2013).

Figura 4 - Ilustração do hiperplano e separador de margem máxima do SVM



Fonte: Adaptado de Faceli et al. (2011, p. 127).

As SVM encontram melhores resultados quando aplicadas em problemas com dados numéricos, pelo fato de serem construídas com base em princípios matemáticos (LUGER, 2013).

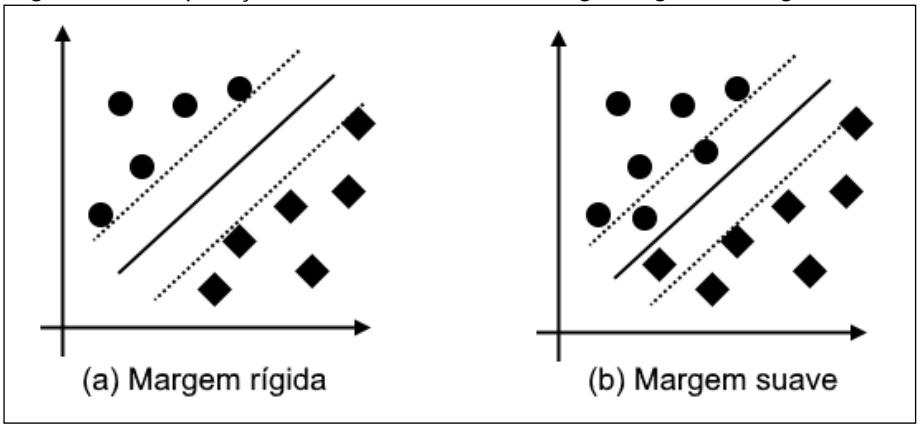
As SVM podem ser divididas em dois grupos distintos, as lineares e não lineares, diferenciando-as pelo limite de decisão gerado, sendo que a primeira faz a separação utilizando uma reta e a segunda realiza a distinção entre os conjuntos por meio de uma curva (FACELI et al., 2011).

2.1.1 SVM linear

As SVM lineares buscam desenhar uma fronteira entre os conjuntos de dados, com um hiperplano de margem máxima. Por isso é chamado por muitos autores de classificador de margem máxima (TAN; STEINBACH; KUMAR, 2009).

O grupo das SVM lineares é subdividido entre as de margens rígidas e de margens suave (figura 5). Nas de margem rígida os conjuntos ficam claramente separados e são impostas restrições que garantem que não existam dados entre a margem de separação das classes. Porém, é comum que os dados não possam ser separados de forma linear devido a presença de ruído, para estes casos são aplicadas SVM com margem suave (FACELI et al., 2011).

Figura 5 - Comparação entre SVM linear de margem rígida e margem suave



Fonte: Adaptado de Faceli et al. (2011).

Estas SVM lineares entregam bons resultados de classificação para conjuntos de dados linearmente separáveis ou com a presença de pouco ruído, no entanto para outros conjuntos de treinamento em que essa separação não é ideal, as SVM lineares tornam-se ineficazes, sendo assim necessário utilizar limites curvos, estas são chamadas de SVM não lineares (FACELI et al., 2011).

2.1.2 SVM não linear

As SVM não lineares realizam um mapeamento do espaço original, utilizando-os como conjunto de entrada para um espaço com uma dimensão maior. A aplicação deste conceito é baseada no Teorema de Cover, este teorema afirma que o conjunto de dados denominado X, quando transformado em um espaço de características, aumenta a probabilidade de separação linear dos dados. Para isso, a transformação deve ser não linear e a dimensão do espaço de características alta (HAYKIN, 2001).

Como este espaço pode ser de grande dimensão ou infinita, a aplicação do mapeamento pode tornar-se computacionalmente inviável ou de alto custo. No entanto, como a SVM apenas armazena a forma de realização do cálculo de produtos escalares entre os objetos no espaço, não é necessário realizar o mapeamento completo dos dados, para isso utilizam-se as funções de *kernel* (FACELI et al., 2011).

Entre as funções de *kernel* utilizadas nos SVM não lineares estão a polinomial, função de base radial, do inglês *Radial Basis Function* (RBF), e a sigmoidal (tabela 1). Estas funções diferem na forma de aplicação, porém todas têm como objetivo atender as exigências do teorema de Mercer, o qual estabelece que o

resultado da função de *kernel* devem ser matrizes positivas semidefinidas[1], sendo assim possível o cálculo de produtos escalares [2](HAYKIN, 2001).

Tabela 1 - Funções de *kernel*

Tipo de *kernel*	Função
Polinomial	$(\gamma(x_i \,.\, x_j) + r)^d$
RBF	$\exp(-\gamma \,\|\|x_i - x_j\|\|^2)$
Sigmoidal	$\tanh(\gamma(x_i . x_j) + r)$

Fonte: Adaptado de Faceli et al. (2011, p. 132).

A escolha dessa função juntamente com os parâmetros determina o limite de decisão induzido, sendo assim afeta diretamente o desempenho do classificador (FACELI et al., 2011).

Em casos de dados de alta dimensão, a precisão das SVM lineares e SVM não lineares são semelhantes, porém com a aplicação da primeira técnica há um ganho de tempo considerável em seu treinamento, tendo em vista que o treinamento é mais simples em comparação com as não lineares (CHAUHAN; DAHIYA; SHARMAN, 2018, tradução nossa).

Segundo Haykin (2001) as SVM destacam-se pelo desempenho em casos em que há pouco ou nenhum conhecimento do domínio ao qual será aplicada. Outras características são a sua capacidade de generalização e robustez diante de objetos de grande dimensão (CRISTIANINI, SHAWE-TAYLOR, 2000). Entretanto, este desempenho está diretamente relacionado a sua complexidade computacional (HAYKIN, 2001). Outra técnica de aprendizado de máquina que pode ser empregada são as redes neurais artificias.

[1] Matriz positiva semidefinida é quando a matriz se destaca como matriz de Gram de vetores fixos, diferente da definida positiva, os vetores não são linearmente independentes (TEIXEIRA; PIETROBOM; ASSUNÇÃO, 2000).
[2] Produto escalar é uma função binária aplicada entre dois vetores que resulta em um número real (FRANCO, 2006).

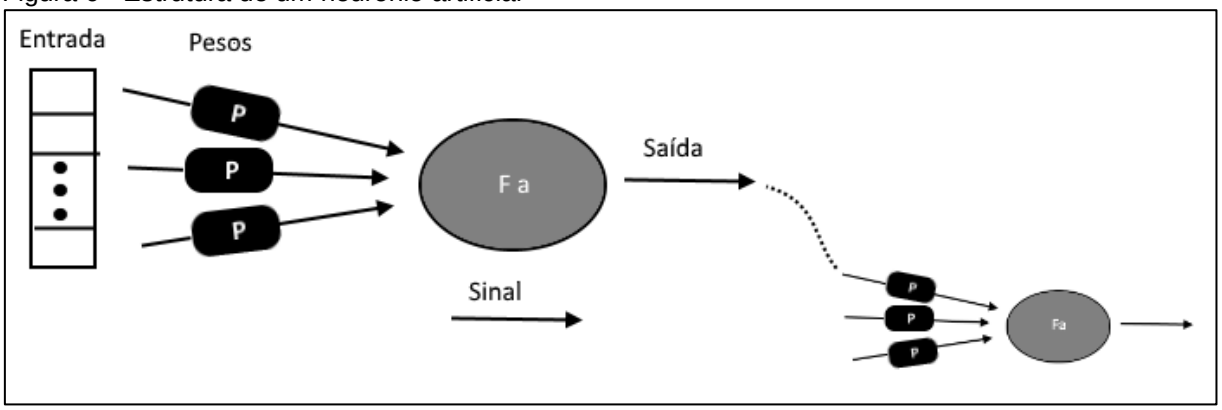
2.2 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais são fundamentadas no sistema nervoso humano, sendo assim é formada por nós simples e fortemente interconectados, lembrando os neurônios e suas ligações (TAN; STEINBACH; KUMAR, 2009). Os nós das RNA conhecidos como neurônios artificiais, podem possuir várias camadas e grande quantidade de conexões, na maioria das vezes unidirecionais e que têm o papel de realizar pequenas funções matemáticas (FACELI et al., 2011).

As RNA são principalmente utilizadas na solução de problemas de classificação e regressão. Atualmente esta técnica vem sendo aplicada em diversas áreas, como por exemplo, na área da saúde, com a predição do resultado do tratamento de doenças cardiovasculares em pacientes com diabetes (SERGEEV; WECKMAN, 2015, tradução nossa); na tecnologia, no reconhecimento de voz (DESAI et al., 2010, tradução nossa; SILVA, 2017) e reconhecimento facial (OMIDVAR; DEZFOULI; RAHMANI, 2010, tradução nossa); na área de engenharia, para a predição de geração de energia solar com base em previsões meteorológicas (RODRÍGUEZ et al., 2018, tradução nossa).

As RNA são estruturadas por unidade de processamento conectadas entre si, essas ligações tem um peso que é representado por W_{xy}, que define o sinal e a força da conexão (figura 6) (RUSSELL; NORVIG, 2013).

Figura 6 - Estrutura de um neurônio artificial



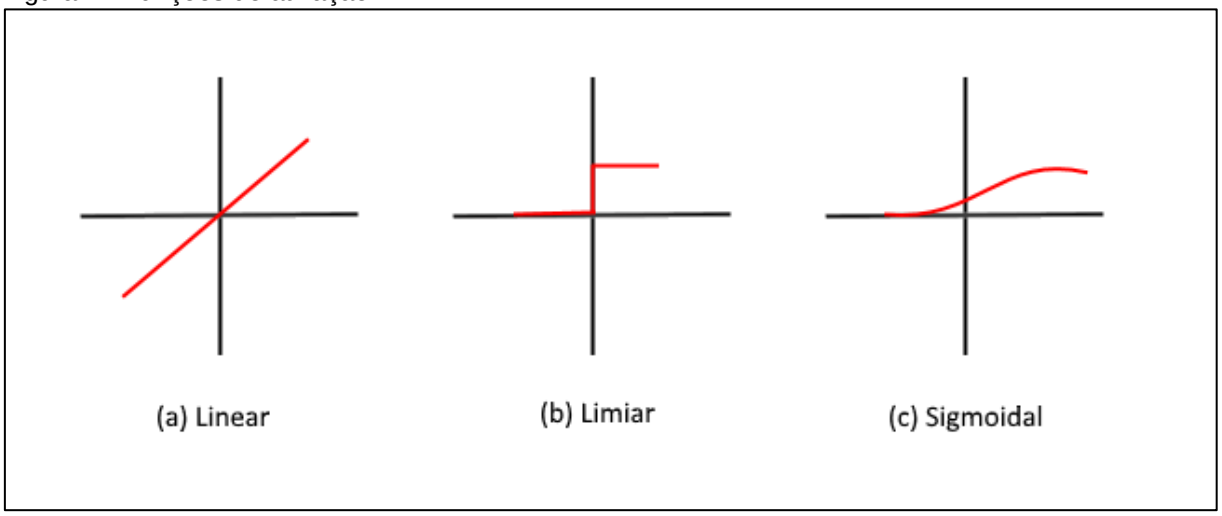
Fonte: Faceli et al. (2011).

As conexões podem ser positivas representadas por $W_j > 0$, negativas sendo $W_j < 0$, caso o peso W_j for 0 tem-se uma ausência de conexão (RUSSELL; NORVIG, 2013). Um dos elementos principais de um neurônio é o somador, que

realiza a junção dos sinais de entrada e uma função de ativação, que tem como objetivo limitar o valor de saída em um intervalo aceitável (HAYKIN, 2001).

As funções de ativação podem ser divididas em parcialmente e totalmente diferenciáveis, que se distinguem pela existência ou não da derivada de primeira ordem[3] de seus pontos. Três dessas funções são linear, limiar e sigmoidal (SILVA; SPATTI; FLAUZINO, 2010).

Figura 7 - Funções de ativação



Fonte: Faceli et al. (2011).

A função limiar, segue a definição tudo ou nada do modelo proposto por McCulloch-Pitts, nesta função caso o valor do neurônio seja negativo retornará 0, caso contrário retorna 1. Diferentemente, na função linear, pode ser retornado um valor entre 0 e 1. A função sigmoidal é a função mais utilizada nas RNA, retornando valores contínuos entre 0 e 1 (HAYKIN, 2001).

A forma mais simples de configuração de uma RNA é o *perceptron* podendo ser de múltiplas camadas ou de camada única. Sendo que a segunda tem um funcionamento mais simples, nela todas as entradas estão ligadas de forma direta com a saída. Enquanto no *perceptron* de múltiplas camadas podem existir várias camadas intermediárias entre a de entrada e saída (RUSSEL; NORVIG, 2013).

A arquitetura de única camada é formada por neurônios de entrada e um neurônio de saída, todos os nós de entrada são ligados ao de saída por conexões ponderadas. No treinamento deste tipo de *perceptron* os valores do peso

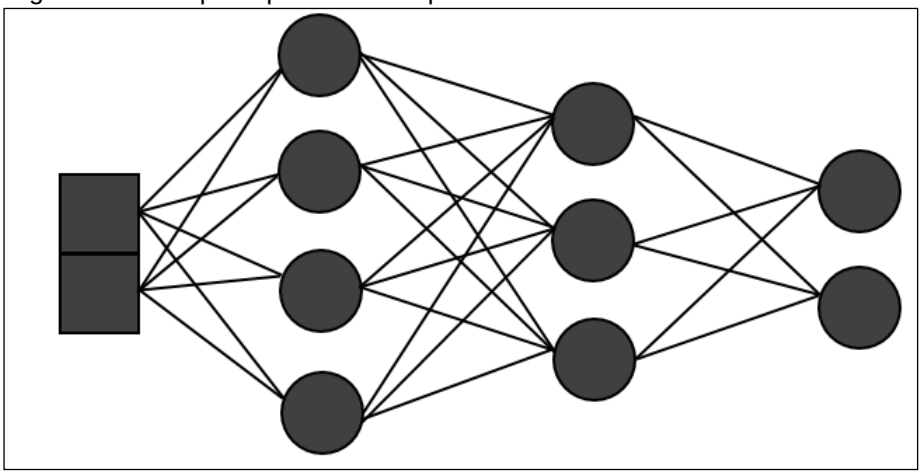
[3] As derivadas de primeira ordem são derivadas de uma determinada função utilizando apenas uma variável (CRISOSTOMO; LOPES, 2017).

representado por W, são ajustados para que se aproximem dos valores de saída do treinamento (TAN; STEINBACH; KUMAR, 2009).

Este tipo de *perceptron* mesmo apresentando uma estrutura simples tem uma boa precisão na sua classificação, porém possui algumas limitações como a incapacidade de resolver problemas que não são linearmente separáveis (FACELI et al., 2011). Para esses casos devem ser utilizados os *perceptrons* de múltiplas camadas, modelo explicado nesta pesquisa.

Os *perceptrons* de múltiplas camadas são utilizados para a resolução de problemas que não são linearmente separáveis, eles possuem uma estrutura mais complexa e apresentam uma ou mais camadas entre as de entrada e saída, essas camadas são chamadas de ocultas (figura 8) (TAN, STEINBACH, KUMAR, 2009). Para Cybenko (1989), esse tipo de *perceptron* com uma camada oculta permite a resolução de qualquer função linear, já a aplicação de duas ou mais camadas ocultas possibilita a aproximação de qualquer função.

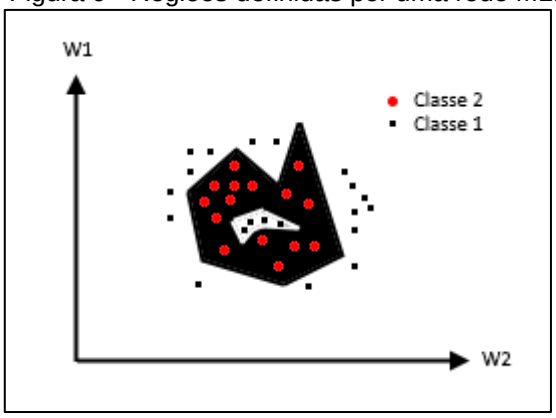
Figura 8 - Rede perceptron de múltiplas camadas.



Fonte: Adaptado de Faceli et al. (2011).

Neste *perceptron*, o funcionamento de cada rede é definido pela saída dos nós da camada anterior. Considerando o *perceptron* de pelo menos duas camadas ocultas tem-se o seguinte comportamento de cada camada: a primeira camada traça retas de padrão de treinamento, a segunda camada intermediária forma regiões convexas ligando as retas traçadas pela camada anterior a ele. A camada de saída combina as retas traçadas e as áreas formadas por todas as camadas anteriores, formando regiões convexas (figura 9) (BRAGA; CARVALHO; LUDERMIR, 2000).

Figura 9 - Regiões definidas por uma rede MLP.



Fonte: Braga, Carvalho e Ludermir (2000).

Uma das dificuldades na utilização deste *perceptron* é a complexidade de implementar o seu treinamento, como alternativa pode-se dividir o *perceptron* em várias sub-redes e realizar um treinamento de forma separada, porém algumas vezes o problema não pode ser dividido em subproblemas, tornando a aplicação desta alternativa inviável ou complexa. Os algoritmos de treinamento de redes MLP são classificados em estáticos e dinâmicos, sendo que o primeiro altera apenas os pesos e o outro pode alterar a estrutura da RNA, adicionando camadas, nós e conexões (BRAGA; CARVALHO; LUDERMIR, 2000).

Neste tipo de *perceptron* cada neurônio está ligado com todos os neurônios da próxima camada, porém a camada oculta torna essa RNA mais complexa. O erro da camada de saída é visível, enquanto o erro da camada intermediária não fica claro. Para resolver este problema é utilizada a técnica de retropropagar o erro da saída nas camadas ocultas (RUSSELL; NORVIG, 2013).

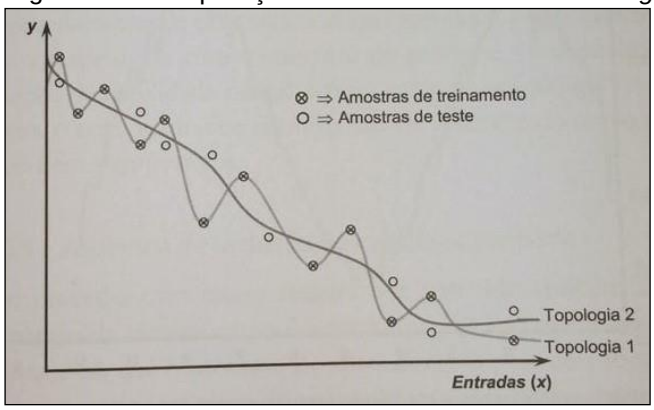
Esta técnica de retropropagar o erro, ou do inglês *backpropagation*, consiste em duas fases: a primeira trata de informar as entradas e obter o resultado da classificação de acordo com os pesos atuais, nesta fase nada é alterado na rede; na segunda é realizada a comparação do resultado obtido na primeira fase com o resultado esperado. Esta comparação gera um sinal de erro, que é retropropagado na rede para que os pesos dos neurônios artificias sejam reajustados de acordo com a classificação correta (SILVA; SPATTI; FLAUZINO, 2010).

Sendo assim, considerando C uma camada da rede, na propagação os resultados obtidos em C são utilizados em C+1 e na retropropagação os pesos encontrados em C+1 são utilizados em C, assim realiza-se o treinamento e ajuste para

que os valores sejam consistentes com a saída correta (TAN; STEINBACH; KUMAR, 2009).

Neste treinamento normalmente são definidos alguns critérios de parada como o número máximo de erros ou de ciclos realizados, a fim de evitar o *overfitting*, que ocorre quando a rede perde a capacidade de generalização, por ter memorizado os conjuntos de treinamento. Pode-se também separar alguns dados do treinamento para formar um conjunto validador e realizar um teste com esse conjunto, conforme o número de ciclos definido. A taxa de erros de validação quando começar a subir significa que a mesma deixou de aprender e se tornou superajustada ao conjunto de treinamento (figura 10) (FACELI et al., 2011).

Figura 10 - Comparação da rede com e sem *overfitting*.



Fonte: Silva, Spatti e Flauzino (2010).

As RNA destacam-se pela sua tolerância a ruído e facilidade na resolução de problemas práticos, tornando-se popular a sua aplicação em tarefas de controle e percepção na robótica, bem como em outros problemas de visão computacional (FACELI et al., 2011). Todavia devido a quantidade de parâmetros e a aplicação de complexas fórmulas matemáticas, o entendimento de como a RNA toma as decisões, é comumente chamado de caixa preta (HAYKIN, 2001). Entre as RNA de múltiplas camadas, encontra-se a rede neural do tipo *Kohonen*.

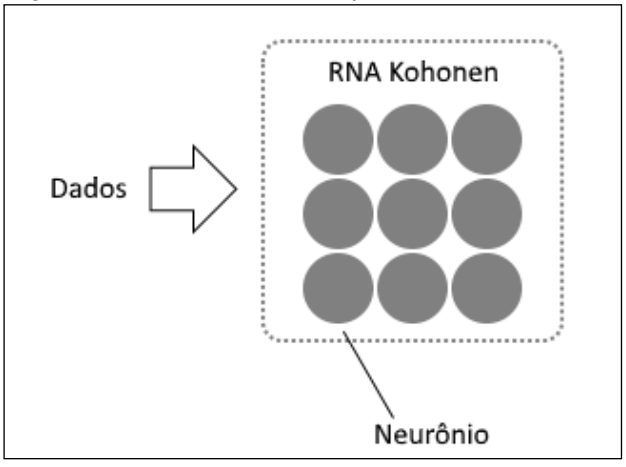
2.2.1 Redes neurais artificiais do tipo *Kohonen*

Kohonen (1990, tradução nossa) propôs uma RNA que é chamada de *Self-organizing Map*, em outras literaturas também é conhecida por rede neural artificial do tipo *Kohonen*. Esta RNA pode ser aplicada tanto para a tarefa de classificação quanto para a de agrupamento.

Esta técnica vem sendo empregrada em diversas áreas do conhecimento, como por exemplo, na engenharia com estimativa de carga de energia elétrica para comunidades isoladas (LLANOS et al., 2017, tradução nossa), na biologia para explorar a relação entre espécies de peixe e a qualidade da água (TSAI et al., 2017, tradução nossa).

Nesta RNA são utilizados parâmetros para a largura e altura, estes definirão a quantidade de neurônios da rede. Os dados são mapeados na área criada pelos mesmos, como por exemplo, caso tenha sido informado nos parâmetros, três para altura e três para largura, o algoritmo cria nove neurônios e mapeia os dados entre eles, gerando assim nove grupos de dados (figura 11).

Figura 11 - Modelo RNA do tipo *Kohonen*.



Fonte: Do autor.

Esta RNA destaca-se por permitir uma melhor interpretação dos resultados (KOHONEN, 1990, tradução nossa). Para isso são utilizados métodos estatísticos, como esta RNA pode ser utilizada tanto para a tarefa de agrupamento quanto para classificação, devem ser selecionadas as medidas estatísticas conforme a tarefa escolhida. Nesta pesquisa foi empregada a mesma para tarefa de agrupamento a fim de realizar o hibridismo com as máquinas de vetores de suporte.

2.3 ARQUITETURA HÍBRIDA EM INTELIGÊNCIA ARTIFICIAL

Na inteligência artificial, uma arquitetura hibrida é a mescla de técnicas de aprendizado de máquina, visando manter as melhores características de cada técnica. O hibridismo pode ser utilizado nos casos em que a aplicação de apenas uma técnica não resolve o problema de uma forma eficiente, por suas limitações ou deficiências. Desta forma, a mescla de técnicas pode gerar uma solução mais eficiente e robusta, porém algumas vezes o resultado obtido não é melhor do que o resultado da aplicação das técnicas que compõem o sistema híbrido isoladamente (BRAGA; CARVALHO; LUDERMIR, 2000).

Atualmente as técnicas de hibridismo vêm sendo aplicadas em diferentes áreas, como por exemplo: na medicina com classificação de tumores cerebrais (SACHDEVA et al., 2016, tradução nossa) e na classificação de tumores de câncer de mama (ROUHI; JAFARI, 2016, tradução nossa); nas engenharias como o diagnóstico de defeitos em turbinas (SEO; ROH; CHOI, 2009, tradução nossa) e na previsão de cargas elétricas (SILVA, 2012); na geologia com a classificação do relacionamento entre as propriedades do solo e a distribuição de plantas em uma área protegida no Irã (HOMAYOUN et al., 2015, tradução nossa).

Na escolha das técnicas a serem utilizada no hibridismo deve-se considerar as características de cada técnica, tendo em vista que caso utilize métodos com características semelhantes os resultados podem não ser satisfatórios. Assim, deve-se escolher técnicas que tenham boas características e que sejam distintas (FACELI et al., 2011).

O hibridismo pode ocorrer de diversas formas, como por exemplo, RNA com Algoritmos Genéticos[4] (AG), SVM com AG, RNA com SVM, entre outros (BRAGA; CARVALHO; LUDERMIR, 2000; HAN et al., 2011, tradução nossa; LIU; XIE; WU, 2007, tradução nossa).

Na técnica de hibridismo entre RNA e AG, o AG gera várias redes aleatórias válidas, que são treinadas com o mesmo conjunto de dados, em seguida cada uma tem seu desempenho avaliado, as que obtiverem os melhores resultados passam para

[4] Os algoritmos genéticos são algoritmos de otimização e busca que têm como base o princípio da teoria de seleção natural e genética (FACELI et al., 2011).

a fase de reprodução que gerará novas redes para próxima geração, este processo termina quando é encontrada uma rede com resultados satisfatórios ou chega ao limite do número de gerações (BRAGA; CARVALHO; LUDERMIR, 2000).

No hibridismo de AG e SVM, o AG gera um grupo de parâmetros aleatórios para o SVM, cada parâmetro é testado pelo SVM que irá determinar os que têm melhor performance, em seguida é analisado se os critérios do classificador foram atingidos, se sim, pode-se dizer que o classificador ótimo foi obtido. Caso não foram atingidos, o processo continua gerando novos parâmetros a partir dos que obtiveram melhores resultados na fase de teste (LI; KONG, 2014, tradução nossa).

O hibridismo entre RNA e SVM, pode ser realizado de formas distintas, sendo o sistema separado em duas camadas. A primeira é encarregada de realizar um pré-processamento dos dados, enquanto a segunda utiliza o resultado da primeira camada para determinar a resposta final do sistema. Em alguns casos, a RNA faz o pré-processamento dos dados e o SVM realiza a classificação final, em outros, esse processo acontece inversamente (LEE; ROH; CHOI, 2009, tradução nossa; LIU; XIE; WU, 2007, tradução nossa).

Estudos como o de Liu, Xie e Wu (2007, tradução nossa) apontam que o hibridismo possui como vantagem a redução do conjunto de dados de treinamento, maior precisão e redução no tempo computacional ao realizar a predição, enquanto como desvantagem tem a complexidade de implementação.

Para a avaliação do desempenho da arquitetura híbrida é necessário aplicar medidas estatísticas de avaliação da qualidade do modelo.

2.4 METODOLOGIA DE AVALIAÇÃO DE DESEMPENHO EM APRENDIZADO DE MÁQUINA

No aprendizado de máquina não é possível definir uma técnica que tenha melhor performance em todos os domínios, por isso muitas vezes é necessário mensurar o desempenho de diferentes técnicas aplicadas ao mesmo problema (TAN; STEINBACH; KUMAR, 2009). A avaliação pode ser efetuada considerando diferentes características, tais como, acurácia, tempo de aprendizado, complexidade do conhecimento, entre outras (FACELI et al., 2011).

Uma das maneiras mais comum de avaliar um classificador é por meio de sua taxa de erro e acurácia. Para medir a taxa de erro, divide-se 1 pelo número de dados no conjunto de treinamento, em seguida multiplica-se pelo somatório da função de custo. Nesta função retorna-se 1 para classificações incorretas e 0 para corretas, conforme a equação (2):

$$Erro = \frac{1}{n} \sum_{i=1}^{n} I\ (y_i\ \neq \hat{y}_i\ ) \tag{2}$$

Os resultados desta função variam de 0 a 1, sendo que quanto mais próximo de 0 melhor é considerado o classificador. Após isso, pode-se calcular a acurácia deste classificador, diminuindo a taxa de erro de 1 (equação 3).

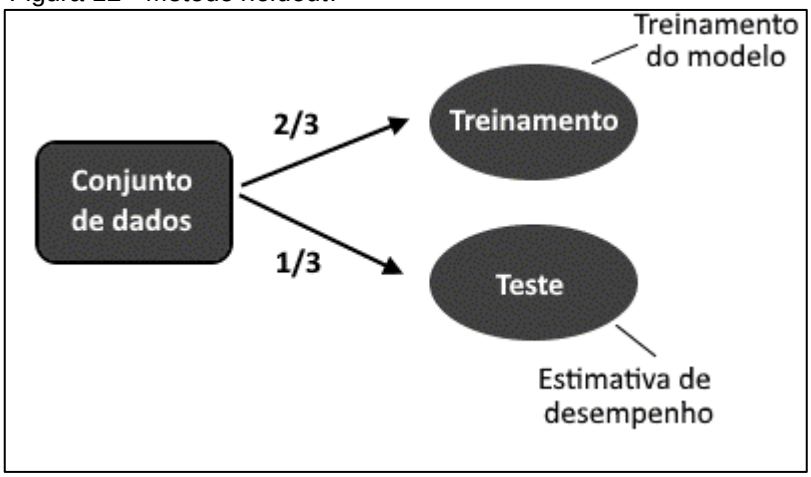
$$Acc(f) = 1 - Erro \tag{3}$$

Diferentemente da taxa de erro, na função de acurácia, quanto mais próximo de 1 melhor será considerado o classificador. Para aplicar estas fórmulas é necessário separar os conjuntos de dados em subconjuntos, de treinamento e de validação. Sendo que o primeiro será utilizado na indução e no ajuste do modelo, enquanto o segundo simula a apresentação de novos dados ao classificador, ou seja, dados que não foram vistos no treinamento (FACELI et al., 2011).

Existem vários métodos que definem como deve ser realizada a divisão do conjunto nestes subconjuntos, entre os mais comuns estão o *holdout*, amostragem aleatória, validação cruzada e *bootstrap* (FACELI et al., 2011).

O método *holdout* consiste em dividir os dados originais em dois subconjuntos, de teste e de treinamento, sendo que a proporção destes dados é 2/3 para treinamento e 1/3 para validação (figura 12) (TAN; STEINBACH; KUMAR, 2009).

Figura 12 - Método *holdout*.



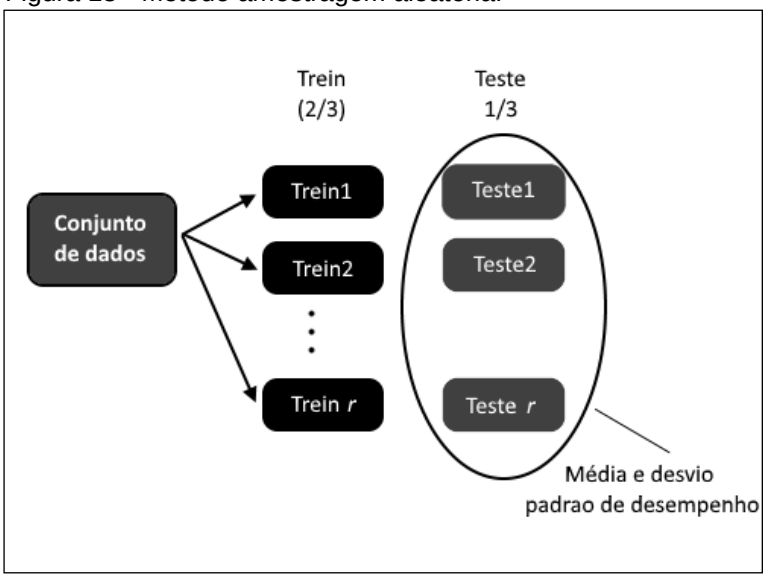
Fonte: Faceli et al. (2011).

Este método possui algumas limitações, como (TAN; STEINBACH; KUMAR, 2009):

a) subestimar a taxa de acerto, pois esta quando gerada considerando-se o conjunto original será maior do que as originadas pelos subconjuntos;
b) não permitir avaliar a variação da técnica quando são apresentados novos dados;
c) considerar que na divisão dos subconjuntos, uma classe pode estar demasiadamente representada em um conjunto do que no outro.

Em algumas situações, com o objetivo de tornar este método menos dependente do conjunto de treinamento, a técnica de *holdout* é aplicada diversas vezes, formando subconjuntos aleatórios, para esta técnica é dado o nome de amostragem aleatória (figura 13). Na amostragem aleatória persiste a limitação de que não são utilizados todos os dados para treinamento (FACELI et al., 2011).

Figura 13 - Método amostragem aleatória.



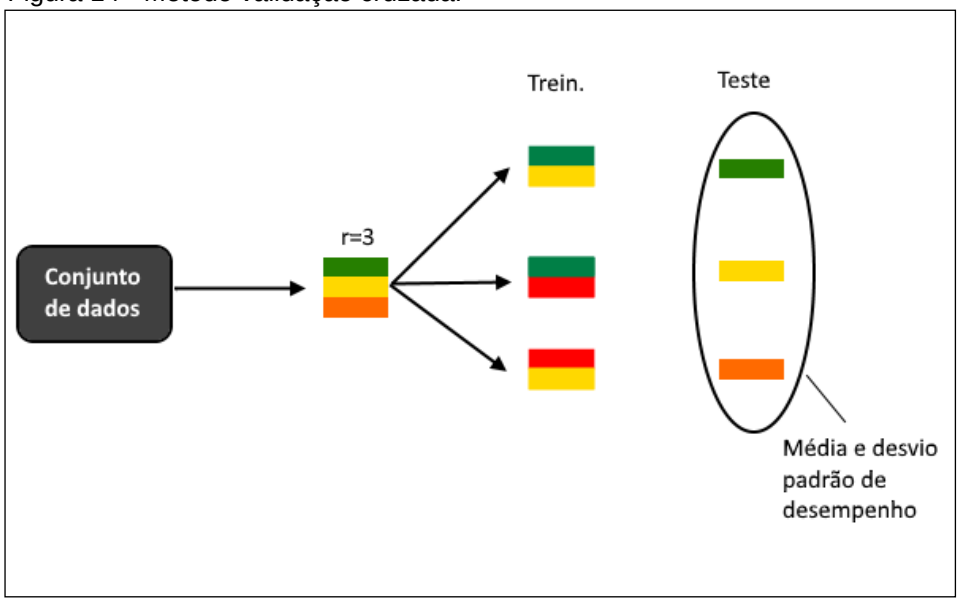
Fonte: Faceli et al. (2011).

Para calcular a precisão neste método deve-se considerar o somatório das precisões de cada iteração e após, obter uma média aritmética (equação 4) (TAN; STEINBACH; KUMAR, 2009).

$$prec_m = \sum_{i=1}^{k} prec_i / k \quad (4)$$

A validação cruzada aparece como uma alternativa a amostragem aleatória, neste método cada registro é usado para treinamento pelo mesmo número de vezes e apenas uma vez para teste. A amostragem é dividida em *K* partes (figura 14), após o método será iterado por K vezes, trocando sempre o conjunto que é utilizado para testes até que todos os subconjuntos tenham sido utilizados uma vez para testes. Este método é conhecido também por validação cruzada de K partes (TAN; STEINBACH; KUMAR, 2009).

Figura 14 - Método validação cruzada.

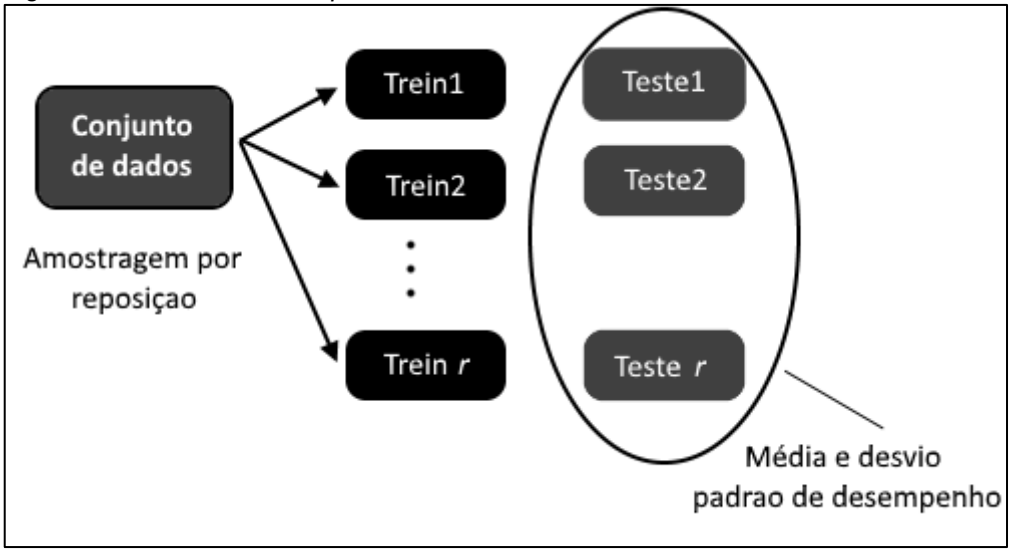


Fonte: Faceli et al. (2011).

Uma variação deste método é o *Leave-one-out*, que é utilizado quando o número de ciclos que são executados tem a mesma quantidade dos dados disponíveis. A cada iteração, um elemento diferente é separado e utilizado para teste, enquanto os demais são utilizados para treinamento, o processo se repete até que todos os registros sejam utilizados uma vez para teste. Apesar desta técnica apresentar um resultado fiel ao desempenho do modelo, deve ser aplicada a amostras pequenas, pois o seu custo computacional se torna muito elevado, devido ao alto número de iterações que são realizadas (THEODORIDIS; KOUTROUMBAS, 2009, tradução nossa). Outra vantagem desta técnica é que nela os dados são exclusivos e todos os dados da amostragem são utilizados para treinamento (TAN; STEINBACH; KUMAR, 2009).

No método *bootstrap* são criados subconjuntos a partir da amostragem original, neste método é possível que o mesmo dado esteja presente nos subconjuntos de treinamento e de teste (figura 15) (TAN; STEINBACH; KUMAR, 2009).

Figura 15 - Método *bootstrap*.



Fonte: Faceli et al. (2011).

A probabilidade de um registro do subconjunto de teste estar presente no de treinamento é de 63,2%, os registros que não estão presentes no *bootstrap* são utilizados no conjunto de testes (equação 5), este processo é repetido diversas vezes, normalmente adota-se um número de repetições maior ou igual a 100 (FACELI et al., 2011).

$$prob = (1 - 1/n)^n \quad (5)$$

A característica mais utilizada para comparar o desempenho de classificadores é por meio da taxa de erro ou acerto. Para que seja obtida uma taxa de erro mais próxima do real é necessário utilizar os métodos de amostragens apresentados acima, um dos métodos mais comuns é a validação cruzada (FACELI et al., 2011). Existem outras formas de avaliar o desempenho do classificador que são as medidas estatísticas que podem ser obtidas por meio da matriz de confusão, estas outras formas são a quantidade de Verdadeiro Positivo (VP) que é o número de classificações corretas, Verdadeiros Negativos (VN) os quais são as classificações negativas corretas, Falso Positivo (FP) que são as classes que foram classificadas como positiva incorretamente e por último o Falso Negativo (FN) em que são as quantidades de classificações que foram consideradas incorretamente negativas (LAROSE; LAROSE, 2014, tradução nossa).

Com base nos valores básicos obtidos com o VP, VN, FP e FN, pode-se então calcular a Taxa de Falsos Negativos (TFN), a Taxa de Falsos Positivos (TFP), a sensibilidade, que também pode ser chamada de *recall* e a especificidade (LAROSE; LAROSE, 2014, tradução nossa).

A taxa de falsos negativos é a relação entre FN e o total de FN e VN (equação 6), a taxa de falsos positivos é a relação de FP com o total de FP e VP (equação 7).

$$PFN = \frac{FN}{FN + VN} \tag{6}$$

$$PFP = \frac{FP}{FP + VP} \tag{7}$$

A sensibilidade pode-se definir como a proporção dos valores que o classificador definiu como verdadeiros positivos considerando o total de valores que realmente são verdadeiros positivos. A especificidade é a formula que relaciona os valores que realmente são negativos com os valores que o classificador indicou como negativa (SOKOLOVA; LAPALME, 2009, tradução nossa; WITTEN; FRANK; HALL, 2011, tradução nossa).

A capacidade de o classificador conseguir uma classificação correta pode ser medida pela área abaixo da curva ROC, quanto mais perto de um esse valor, melhor é considerado o classificador (equação 8).

$$Área\ ROC = \frac{1}{2}\left(\frac{VP}{VP + FN} + \frac{VN}{VN + FP}\right) \tag{8}$$

Estas são as medidas mais comuns em avaliação de desempenho do classificador e serão empregadas nesta pesquisa para avaliar a aplicação do SVM isolado e após avaliada a arquitetura híbrida.

3 TRABALHOS CORRELATOS

São abordadas seis produções cientificas que influenciaram no desenvolvimento desta pesquisa, sendo que três delas sobre hibridismo em inteligência artificial e dois sobre classificação dos solos.

3.1 PREDIÇÃO DE TEOR DE ÁGUA NO SOLO POR MEIO DE UMA ARQUITETURA HÍBRIDA ENTRE RNA E SVM

Neste artigo publicado na revista *Springer Science* por Hongbin Liu, Deti Xie e Wei Wu no ano 2007, foi proposta uma arquitetura híbrida entre redes neurais artificiais e máquinas de vetores de suporte para prever o teor de água no solo.

Para aplicar a arquitetura híbrida foi utilizado o software MATLAB 6.5, nesta arquitetura utilizando o princípio de dividir para conquistar, primeiramente uma RNA *Kohonen* é aplicada sobre a entrada para separar em áreas, para cada área é utilizado um SVM com diferentes funções de *kernel* para encontrar a melhor predição. Para treinamento deste sistema foi utilizado um conjunto de dados de 170 amostras e outras 20 amostras para validação.

Como forma de avaliação da performance do sistema foram definidos os parâmetros de erro médio, erro quadrático médio e coeficiente de variação.

Neste artigo concluíram que a aplicação desta arquitetura minimizou os erros e melhorou a precisão quando comparado com a aplicação das duas técnicas em separado.

3.2 CLASSIFICAÇÃO DO TIPO DO SOLO E ESTIMATIVA DE SUAS PROPRIEDADES UTILIZANDO MÁQUINAS DE VETORES DE SUPORTE

Este artigo publicado na revista *Geoderma* por Miloš Kovačević, Branislav Bajat e Boško Gajić, no ano de 2010, utilizou as máquinas de vetores de suporte para classificar tipos de solos e estimar suas propriedades, a escolha desse modelo deu-se por vários estudos apontarem que esta técnica tem mostrado bons resultados para estimar propriedades físicas e pH quando utilizado dados químicos como entrada.

Foi visto que para este caso, as SVM lineares apresentam melhores resultados se comparado as SVM de regressão para domínios com poucos dados de treinamento, porém quando os dados para o treinamento são suficientes a performance da SVM de regressão é superior. Como trabalhos futuros foi sugerida a comparação do uso de SVM e das técnicas estatísticas convencionais no mapeamento do solo.

3.3 REDES NEUROFUZZY PARA AVALIAÇÃO DE APARTAMENTOS EM CRICIÚMA

Este trabalho de conclusão de curso, escrito por Niria Borges Ferreira em 2006 para obtenção do Grau de Bacharel em Ciência da Computação pela Universidade do Extremo Sul Catarinense utilizou o hibridismo entre duas técnicas de IA, redes neurais e lógica *fuzzy* para avaliação de apartamentos na cidade de Criciúma.

Neste trabalho foi desenvolvida a *Shell Ícaro*, ferramenta *NeuroFuzzy*. O hibridismo destas duas técnicas de IA são interessantes, pois a lógica *fuzzy* é capaz de fazer a incorporação do sistema especialista, enquanto a rede neural é importante para o aprendizado do sistema.

No desenvolvimento deste trabalho foram encontrados alguns problemas para treinamento do sistema devido à baixa quantidade de dados amostrais, mesmo com este problema foi desenvolvido um sistema com boa taxa de acerto. Como trabalhos futuros foi sugerido que sejam aplicadas a técnica *NeuroFuzzy* a outros domínios de conhecimento.

3.4 CLASSIFICAÇÃO DE SOLOS USANDO-SE REDES NEURAIS ARTIFICIAIS

Este artigo publicado no Simpósio Brasileiro de Pesquisa Operacional em 2007, escrito por Luiz Biondi Neto et al., utilizou redes neurais na área de geotécnica no segmento de classificação de solos.

Para treinamento da RNA foram utilizados dados advindos de situações reais do comportamento dos solos no teste de penetração do cone. Foi implementada uma rede de múltiplas camadas com as etapas de *forward* e *backward,* a rede que

apresentou melhor desempenho foi desenvolvida com duas camadas ocultas, com 14 neurônios na primeira camada e 120 na segunda camada oculta, a camada de saída possui 12 neurônios, sendo uma para cada tipo de solo possível.

A RNA apresentou um resultado considerado bom em comparação a retro propagação tradicional, tendo taxa de erro inferior a 5% enquanto a tradicional apresenta resultados perto dos 10%. Como trabalho futuro foi sugerido o treinamento de uma RNA utilizando o método de propagação resiliente para dados advindos de uma medição da poropressão do solo.

3.5 DIAGNÓSTICO DE DEFEITOS DO MOTOR DE TURBINAS A GÁS SUAV UTILIZANDO MÉTODO HÍBRIDO ENTRE SVM-REDES NEURAIS ARTIFICIAIS

Este artigo publicado no jornal de ciência e tecnologia mecânica por Sang-Myeong Lee, Tae-Seong Roh e Dong-Whan Choi no ano de 2009, utilizou uma arquitetura híbrida entre máquinas de vetores de suporte e redes neurais artificias para o diagnóstico de defeitos em turbinas.

A arquitetura desenvolvida utilizava um SVM para classificar as entradas para apontar em qual componente estava o defeito, em seguida foi utilizada uma RNA MLP para cada componente que identificava a magnitude do defeito, com esta arquitetura a RNA era treinada apenas com os dados de um componente.

Foi avaliado que a arquitetura híbrida obteve uma melhor acurácia na predição da magnitude do defeito quando comparado a aplicação de apenas a RNA, também houve melhoras nos tempos de treinamento e execução.

A arquitetura híbrida conforme avaliada por meio de medidas de qualidade apresentou-se mais confiável.

3.6 RECONHECIMENTO DE PADRÕES DE DEFEITOS ESPACIAIS UTILIZANDO UMA ABORDAGEM HÍBRIDA DE SOM-SVM NA FABRICAÇÃO DE SEMICONDUTORES

Neste artigo publicado na revista de Sistemas Especialistas com Aplicação em 2009, por Te-Sheng Li e Cheng-Lung Huang, utilizou uma arquitetura híbrida com RNA do tipo *Kohonen* e máquinas de vetores de suporte para o reconhecimento de padrões de defeitos espaciais na fabricação de semicondutores.

Na pesquisa foi implementada uma arquitetura em que a RNA do tipo *Kohonen* agrupa as entradas, em seguida é aplicado um SVM para cada grupo para realizar a classificação. Foram realizados diversos testes alterando os parâmetros da RNA e também alterando os parâmetros de cada SVM.

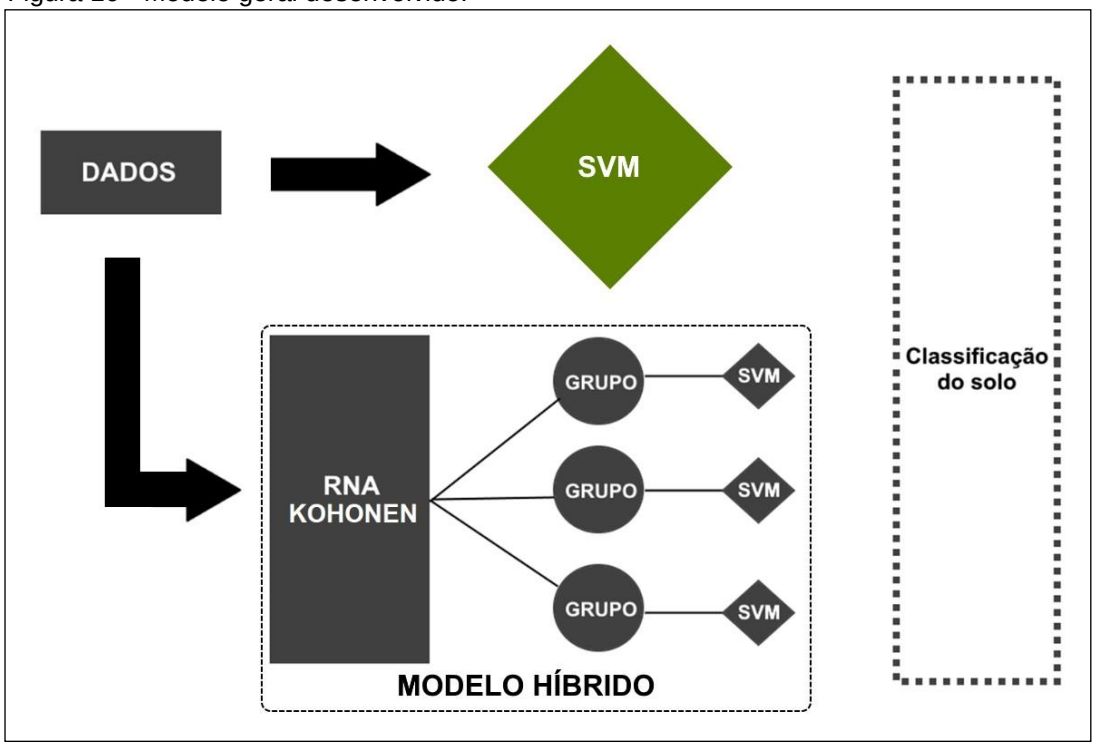
A medida de qualidade para avaliação do modelo foi especificamente a acurácia, em que apresentou uma precisão do modelo híbrido acima de 90% em alguns conjuntos, enquanto que quando não utilizado o hibridismo a precisão foi de aproximadamente 85%. Nesta pesquisa concluíram que o modelo híbrido apresentou uma melhor acurácia.

4 ARQUITETURA HÍBRIDA DE MÁQUINAS DE VETORES DE SUPORTE E REDES NEURAIS ARTIFICIAIS APLICADA A CLASSIFIÇÃO DOS SOLOS

Nesta pesquisa foi desenvolvido um modelo de máquinas de vetores de suporte para a classificação dos solos. Visando otimizar este modelo foi implementada uma arquitetura híbrida, unindo as técnicas de redes neurais artificiais e o SVM, esta arquitetura busca melhorar os tempos despendidos nas fases de treinamento e de execução. As RNA e SVM são comumente utilizadas para resolução de problemas em IA e vêm sendo utilizadas atualmente em arquiteturas híbridas em diferentes domínios de aplicação, conforme os estudos de Lee, Roh e Choi (2009, tradução nossa), Li e Huang (2009, tradução nossa) e Liu, Xie e Wu (2007, tradução nossa).

Os modelos desenvolvidos nesta pesquisa podem ser visualizados na figura 16 que demonstra a aplicação do SVM isoladamente e outra aplicação na qual é empregado o hibridismo entre uma RNA do tipo *Kohonen* e o SVM. Este tipo de RNA foi escolhido pois foi aplicada em um domínio de conhecimento semelhante e apresentou melhores resultados.

Figura 16 - Modelo geral desenvolvido.



Fonte: Do autor.

Este trabalho contribui para o conhecimento e abre possibilidades de trabalhos futuros com este mesmo tema na universidade, sendo que o hibridismo entre estas técnicas de IA não havia sido aplicado em trabalhos na universidade até o presente momento.

4.1 CLASSIFICAÇÃO DOS SOLOS

A classificação do solo é assunto recorrente em trabalhos acadêmicos devido a necessidade de levantamentos pedológicos, essa classificação é realizada muitas vezes por indução, levando um tempo considerável para ser realizada (SANTOS et al., 2006).

Este é um processo trabalhoso e com alto custo, muitas vezes devido à falta de dados há necessidade de utilizar técnicas de predição, para isso pode ser empregado o uso de inteligência artificial (KOVAČEVIĆ et al., 2010, tradução nossa).

Nas pesquisas de Brungard et al. (2015, tradução nossa) foram utilizadas técnicas de aprendizado de máquina, como por exemplo, SVM para a classificação do solo em regiões semiáridas. Enquanto Heung et al. (2016, tradução nossa) comparou a aplicação de diversas técnicas de aprendizado de máquina, como por exemplo, as redes neurais artificiais, máquinas de vetores de suporte e florestas de decisão para o mapeamento digital do solo.

O processo de classificação do solo busca encontrar as similaridades e diferenças entre os solos para que se possa armazenar e recuperar as informações com maior eficiência. Existem vários sistemas para classificar o solo, cada sistema é preparado de forma diferente, porém todos possuem dois objetivos principais: agrupar as informações para que o processo de recuperação destas seja eficiente e cada classe de solo seja distinta entre si (BUOL et al., 2011, tradução nossa).

Existem diversos sistemas de classificação de solo, no Brasil é utilizado o Sistema Brasileiro de Classificação de Solos (SiBCS), criado em 1999 pela EMBRAPA com parceria de instituições de ensino, e desde então tem recebido atualizações. Este sistema possui treze classes de solos no nível mais alto de categorização e seus quatro primeiros níveis de categorização foram definidos como ordem, subordem, grandes grupos e subgrupos, enquanto que os demais níveis ainda estão em discussão (SANTOS et al., 2006).

Para a divisão do primeiro nível categórico, denominado ordem, são considerados os sinais deixados no solo pelo principal processo que atuou no seu desenvolvimento. No segundo nível categórico, subordem, são consideradas as características deixadas pelos processos coadjuvantes e variações dentro da ordem (SANTOS et al., 2006).

A classificação dentro dos grandes grupos se dá através das características e de propriedades do solo, como por exemplo, infiltração de água e desenvolvimento de raízes. Para a divisão no quarto nível categórico, subgrupos, são utilizadas características detalhadas, abrangendo os solos que estejam no limite entre as divisões dos níveis categóricos superiores (SANTOS et al., 2006).

Dentro da ordem existem diferentes classificações sendo elas (SANTOS et al., 2006):

a) argissolos possuem uma separação visível entre suas camadas, podem ser arenosos ou argilosos. Apresentando uma cor amarelada ou avermelhada e possuem maior concentração de argila nas suas camadas inferiores;
b) cambissolos englobam solos minerais com diferentes características, possuem fragmentos de rochas em sua composição. Apresentam cores mais vivas se comparado a solos exclusivamente rochosos. Este solo disponibiliza maior quantidade de nutrientes para plantas, também pode ser utilizado para a agricultura após eliminação de algumas restrições;
c) chernossolos são originados de rochas, possuem uma alta concentração de cálcio e magnésio. Caracteriza-se por uma cor escura e presença de matéria orgânica, devido a esta característica é considerado um solo fértil;
d) espodossolos são solos arenosos com uma separação definida entre as camadas. Sua cor diversifica entre cinza e avermelhada. Em sua composição apresenta alumínio e húmus ácido, por este motivo estes solos têm baixa fertilidade e não são indicados para cultivos agrícolas;
e) gleissolos são solos com alta concentração de água, formados geralmente em planícies inundadas. Possuem cores acinzentadas, sua fertilidade pode variar entre alta e baixa, pois este solo depende dos solos que estão em seu entorno;

f) latossolos têm baixa diferenciação entre as camadas, com uma cor homogênea em toda a sua profundidade. Possuem textura argilosa e são pouco férteis;
g) luvissolos são solos com pouca espessura com pedras em sua superfície, apresentam cores claras e diferenciação entre as suas duas camadas. Apesar de possuírem alto teor de nutrientes, o fato de ocorrer em ambientes secos limita o seu uso agrícola;
h) neossolos são rasos com camadas distintas com baixa retenção de água, para o uso agrícola é necessário a utilização de técnicas que evitem a sua degradação;
i) nitossolos têm baixa diferenciação conforme sua profundidade, caracteriza-se pela retenção de água e cor avermelhada. Devido a estas características mantém sua fertilidade por vários anos desde que sejam utilizadas técnicas a fim de evitar erosão;
j) organossolos são solos orgânicos, possuem em sua composição matérias vegetais não decompostas. Apresenta cor escura, alta acidez e alto teor de água. Devido as suas características tem limitação no seu uso agrícola;
k) planossolos são solos rasos, possuem textura arenosa na camada superior e argilosa na inferior. Possuem uma coloração acinzentada e diferenciação entre suas camadas. Estes solos são pouco permeáveis e podem fornecer nutrientes as plantas, seu relevo plano permite o uso de máquinas agrícolas;
l) plintossolos são solos minerais com presença de ferro, possuem uma acidez excessiva, por isso têm uma baixa fertilidade. Necessita de técnicas de manejo para que se torne apto ao uso agrícola;
m) vertissolos são solos pouco permeáveis, apresentam rachaduras em grande extensão e profundidade nos períodos de seca. Nos períodos de chuva apresenta nutrientes para as plantas. O cultivo nestes solos é dificultado, pois quando úmido sua textura fica pegajosa e quando seco muito rígido.

4.1.1 Base de dados

A base de dados utilizada neste trabalho foi obtida através do site da Empresa Brasileira de Pesquisa em Agropecuária (EMBRAPA)[5], nesta base constam os dados do solo de pesquisas realizadas no país, nesta pesquisa a base de dados é referente a apenas dados do estado de Santa Catarina.

Após obter a base de dados o especialista do domínio de aplicação, Fernando Basquiroto de Souza, selecionou os atributos mais relevantes para a classificação do solo. Esta seleção foi realizada, pois a base possuía 274 colunas, ou seja, 274 atributos. Segundo Weston et al. (2000, tradução nossa) a seleção de atributos é importante para melhorar o tempo de treinamento e execução, assim como também o SVM pode ter uma performance ruim quando existem muitos atributos irrelevantes.

Após realizada a seleção permaneceram 71 atributos, como por exemplo, profundidade, cor da amostra, textura, classe de drenagem, entre outros. No apêndice A pode ser visualizada a tabela com todos os atributos e valores que os mesmos possam assumir.

Esta base de dados contém um total de 450 registros que englobam oito das treze classes de solo do Sistema Brasileiro de Classificação dos Solos, sendo elas, argissolo, cambissolo, gleissolo, latossolo, neossolo, nitossolo, organossolo e planossolo (tabela 2).

Tabela 2 - Quantidade de registros por classe de solo

Classe do solo	Quantidade
Argissolo	74
Cambissolo	95
Gleissolo	9
Latossolo	102
Neossolo	26
Nitossolo	135
Organossolo	5
Planossolo	4
Total	450

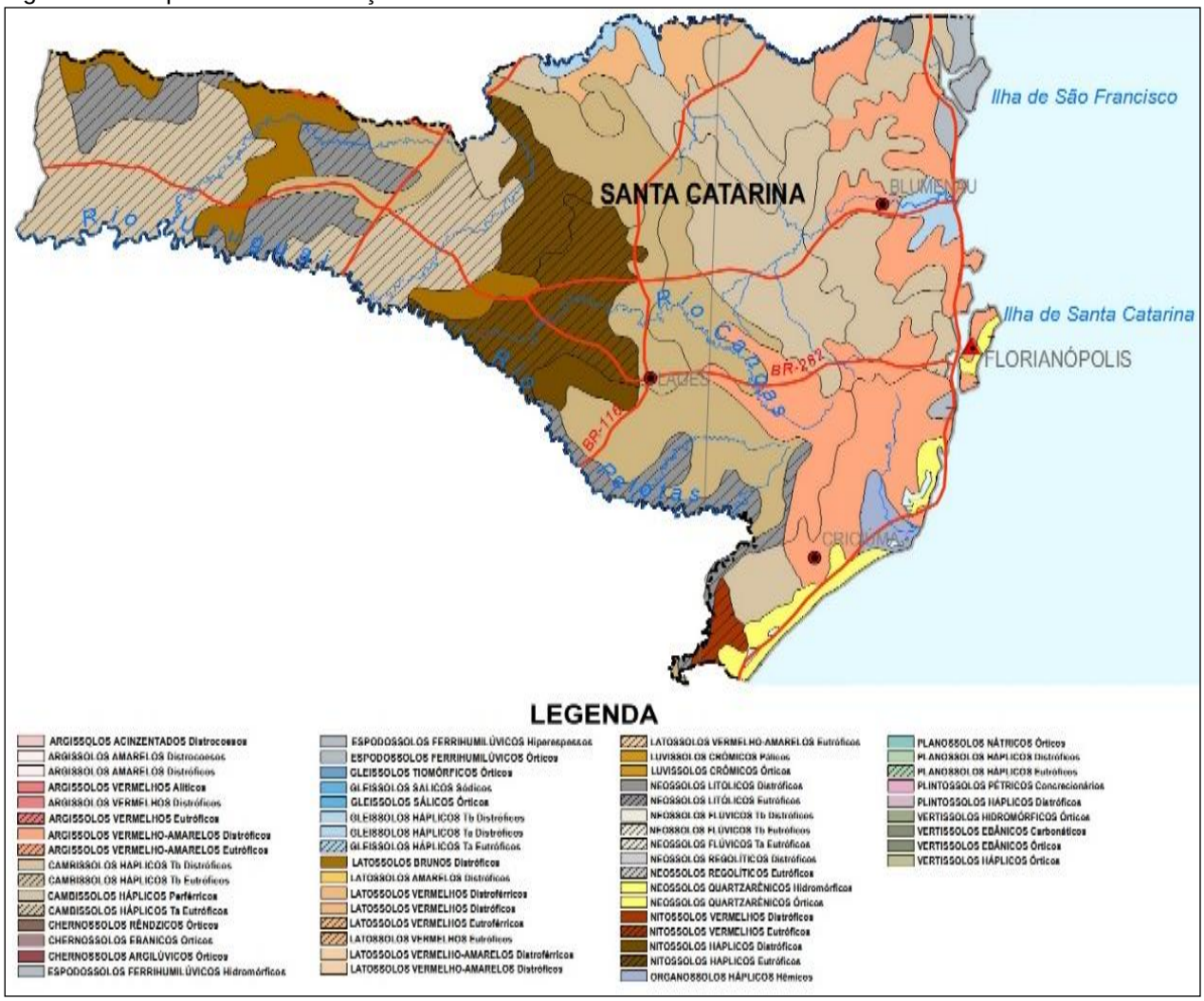
Fonte: Do autor.

Como a base de dados é do estado de Santa Catarina alguns solos são menos predominantes no estado, como por exemplo, as classes gleissolo, neossolo,

[5] A base de dados é disponibilizada gratuitamente em (https://www.sisolos.cnptia.embrapa.br/)

organossolo e planossolo, enquanto as classes argissolo, cambissolo, latossolo e nitossolo são mais frequentes, a distribuição de classes[6] de solo em Santa Catarina pode ser visualizada na figura 17.

Figura 17 - Mapa com a distribuição de classes de solos no estado de Santa Catarina.



Fonte: Adaptado de Embrapa (2011).

4.2 METODOLOGIA

Esta pesquisa metodologicamente classifica-se como aplicada e de base tecnológica, tendo-se desenvolvido as seguintes etapas principais: levantamento bibliográfico, obtenção e preparação da base de dados, aplicação da técnica de máquinas de vetores de suporte, desenvolvimento e aplicação da arquitetura híbrida

[6] Mapas com a distribuição das classes do solo são disponibilizados gratuitamente no site http://mapoteca.cnps.embrapa.br/geoacervo/det_mapa.aspx

entre RNA e SVM, avaliação dos modelos por meio de métodos estatísticos e análise dos resultados.

Os materiais empregados no desenvolvimento desta pesquisa em termos de recursos de dados foram os disponíveis no site da Embrapa sobre a classificação do solo no estado de Santa Catarina; em termos de *software* empregou-se o ambiente de desenvolvimento *Netbeans*[7] na versão 8.2, biblioteca em Java do *LibSVM*[8] na versão 3.23 e a biblioteca *Weka API*[9], sendo que nesta última está a implementação da rede neural artificial do tipo *Kohonen*.

Para a comparação do tempo de execução os dados coletados foram organizados e analisados pelo programa *IBM Statiscal Package for the Social Sciencies* (SPSS) versão 23.0. As variáveis quantitativas foram expressas por média e desvio padrão.

As análises estatísticas inferenciais foram realizadas com nível de significância alfa = 0,05, isto é, confiança de 95%. Investigou-se a variância das variáveis quanto a homogeneidade por meio da aplicação do teste de Levene.

A comparação das médias das variáveis quantitativas entre as categorias das variáveis qualitativas dicotômicas[10] foi realizada por meio da aplicação do teste U de Mann-Whitney.

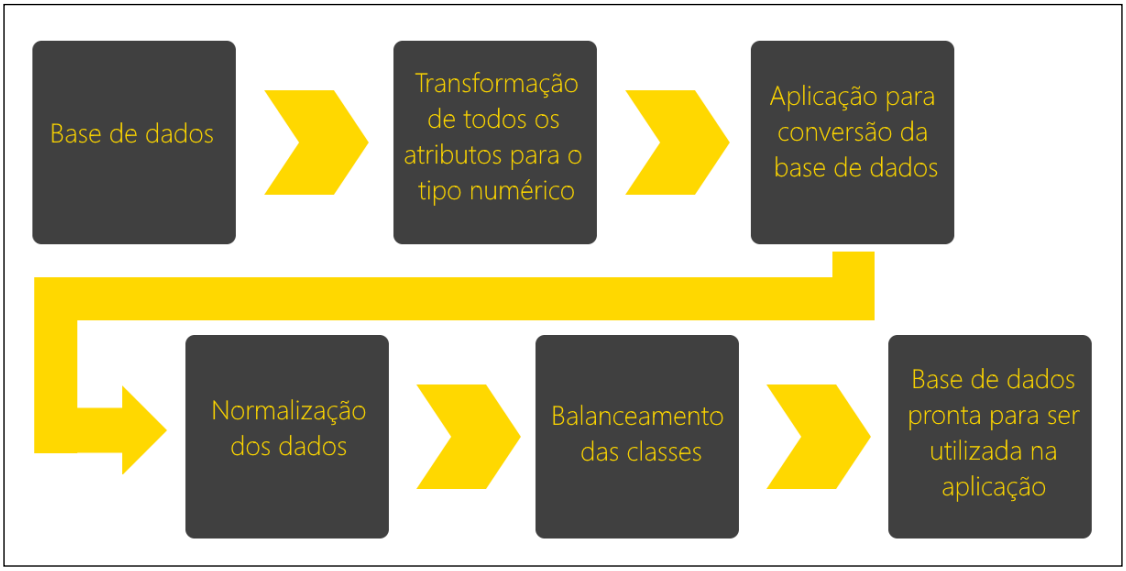
A comparação das médias das variáveis quantitativas entre as categorias das variáveis qualitativas politômicas[11] foi realizada por meio da aplicação do teste H de Kruskal-Wallis seguido do *post hoc* de Dunn.

4.2.1 Pré-processamento dos dados

No pré-processamento foram realizadas a transformação dos atributos para o tipo numérico, conversão da base de dados para o formato *libsvm* utilizado na biblioteca *LibSVM*, normalização dos atributos e balanceamento de classes (figura 18).

Figura 18 - Diagrama de etapas do pré-processamento.

[7] Disponível em: https://netbeans.org/downloads/
[8] Disponível de forma gratuita em: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
[9] Disponível em: https://www.cs.waikato.ac.nz/ml/weka/downloading.html
[10] Variáveis que podem assumir apenas dois valores.
[11] Variáveis que podem assumir mais que dois valores



Fonte: Do autor.

A base de dados possui uma grande quantidade de atributos categóricos, para que as técnicas de aprendizado de máquina pudessem ser aplicadas foi realizada a transformação de dados categóricos para numéricos, isto devido ao fato do SVM aceitar apenas valores numéricos nos atributos dos dados. Na realização deste procedimento obteve-se para cada atributo categórico os possíveis valores, substituindo-se cada valor por um número inteiro iniciando em um, este procedimento foi aplicado para 25 atributos categóricos encontrados na base de dados.

Após a transformação dos atributos para numérico, foi também realizada a normalização. A normalização consiste em colocar os atributos dentro de uma escala, para que cada atributo não domine os outros, ou seja, para que não seja considerado apenas um atributo como relevante, isso pode ocorrer quando um atributo possui valores muito alto e outro muito baixo. Conforme Wang et al. (2009, tradução nossa) demonstra em sua pesquisa, as técnicas de SVM e RNA obtém melhores performances com os atributos normalizados, o que também ocorre nessa pesquisa.

Para a normalização dos atributos foi utilizada a ferramenta que está disponível junto com a biblioteca *LibSVM*, porém a base de dados obtida está no formato CSV, este formato não é aceito pela biblioteca. Assim, foi necessária a implementação de uma aplicação para conversão dos dados para o formato *libsvm*. A aplicação desenvolvida realiza a leitura de um arquivo de texto com separador e converte para o formato *libsvm*.

O arquivo do *LibSVM* é um arquivo de texto, em que a primeira coluna representa a classe e as demais colunas são os atributos que devem estar no formato

de índice do atributo, dois pontos (:) e o seu valor. Todos os dados devem ser numéricos (figura 19).

Figura 19 - Exemplo de arquivo do *LibSVM*.

```
1.0 1:-1.0 2:-1.0 3:-0.777778 4:-0.954545 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
1.0 1:-1.0 2:-1.0 3:0.333333 4:-0.515152 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
1.0 1:-1.0 2:-1.0 3:-0.555556 4:-0.409091 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
1.0 1:-1.0 2:-1.0 3:-0.111111 4:-0.533333 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
1.0 1:-1.0 2:-1.0 3:-0.111111 4:-0.954545 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
1.0 1:-1.0 2:-1.0 3:-0.777778 4:-0.618182 5:-1.0 6:-1.0 7:-1.0 8:-1.0 9:-1.0 10:-1.0 11:-1.0
```

Fonte: Do autor.

A ferramenta *svm-scale* normaliza os atributos nas escalas de 0 e 1 ou -1 e 1, devendo ser executada por linha de comando, na qual podem ser passados os parâmetros da escala a ser utilizada e o arquivo que deve salvar os dados normalizados. Neste trabalho os dados foram normalizados entre -1 e 1.

O *svm-scale* utiliza os valores mínimo e máximo de cada atributo para colocar dentro da escala informada nos parâmetros, como por exemplo, no atributo profundidade o maior valor encontrado no conjunto é 620 que no conjunto normalizado tornou-se 1 já o valor mínimo que é 0 tornou-se -1 e os valores intermediários ficaram entre esse intervalo. Para executar o *svm-scale* a linha de comando: `svm-scale.exe -l -1 -u 1 -s "dataset.range" ".\dataset" > "dataset.scale"`

Os parâmetros *l* e *u* indicam os valores mínimos e máximos da escala, respectivamente. O parâmetro *s* indica onde salvar as regras de cada escala dos atributos para que seja possível utilizar a mesma escala posteriormente caso sejam inseridos registros novos, após é indicado o caminho do conjunto de dados para ser escalado e o símbolo de > indica em qual arquivo a ferramenta salvará os dados normalizados.

Ainda foi avaliado o desbalanceamento de classes devido ao fato da base de dados ser do estado de Santa Catarina em que predominam algumas classes do solo. Dessa forma, algumas classes possuem menor frequência em relação as demais, o que pode ocasionar classificações incorretas para as classes de menor frequência, já que a técnica aprenderá a classificar melhor os exemplos mais vistos no treinamento.

Para este problema algumas técnicas foram estudadas, desde adicionar cópias dos registros até um número satisfatório para cada classe, porém esta solução pode apresentar o *overfitting* do modelo. Para evitar isto pode ser utilizada a técnica

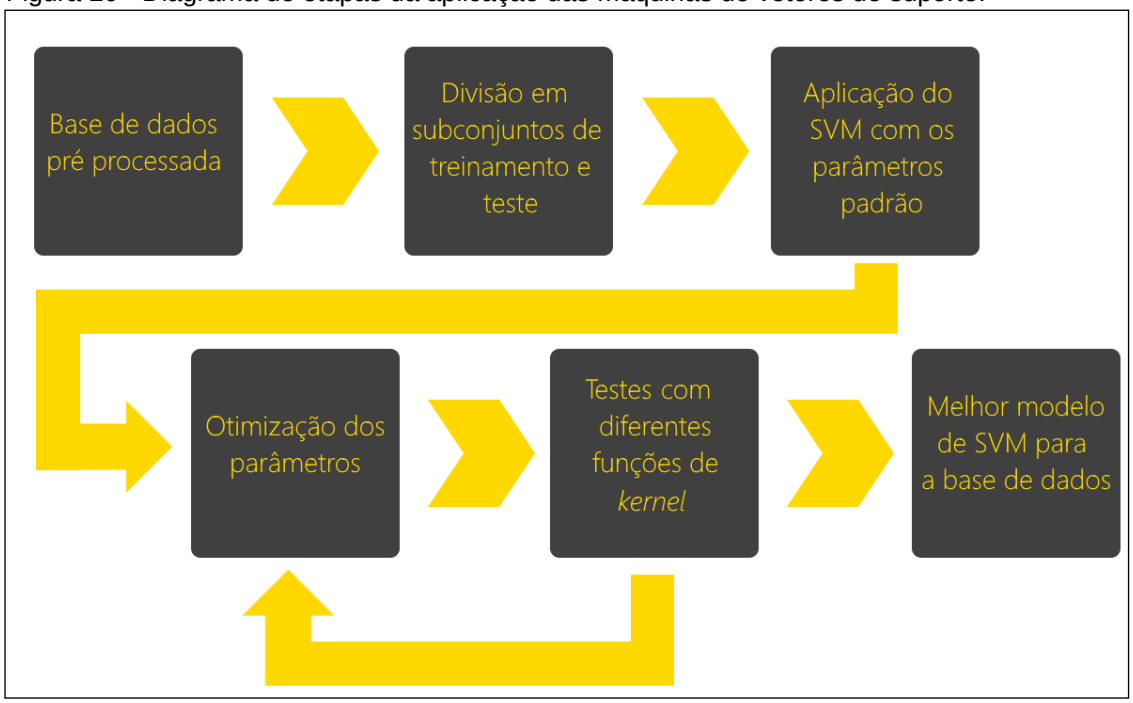
Synthetic Minority Over-sampling Technique (SMOTE). No SMOTE o algoritmo por meio de cálculos matemáticos cria registros sintéticos a partir dos dados originais (CHAWLA et al., 2002, tradução nossa). Alguns classificadores ainda permitem a utilização de pesos no seu treinamento, assim pode-se informar pesos diferentes para cada classe, neste caso os pesos serão considerados para a penalização por classificação incorreta, a técnica de SVM permite utilizar pesos.

Nesta pesquisa como a técnica do SVM permite utilizar pesos foram geradas duas bases de dados para a aplicação, uma em que foi aplicada a técnica SMOTE e uma sem a aplicação de nenhuma técnica de balanceamento.

4.2.2 Aplicação das máquinas de vetores de suporte

Para a aplicação foram realizadas as etapas de separar a base de dados em subconjuntos de treinamento e de teste, aplicação do SVM com os parâmetros sugeridos, otimização dos parâmetros e aplicação das diferentes funções de *kernel*, r*adial basis function*, polinomial e sigmoidal (figura 20).

Figura 20 - Diagrama de etapas da aplicação das máquinas de vetores de suporte.



Fonte: Do autor.

Antes da aplicação das máquinas de vetores de suporte separou-se a base de dados em subconjuntos de treinamento e teste, para a validação do modelo e avaliação do mesmo por meio de métodos estatísticos. O primeiro subconjunto foi utilizado para o aprendizado do SVM, já o segundo simulou novos dados sendo apresentados ao modelo.

O método escolhido para a separação do conjunto de dados em subconjuntos de treinamento e de teste foi o *holdout*, no qual dois terços dos dados foram utilizados no treinamento e um terço para teste. A distribuição de registros nos subconjuntos foi de 303 para treinamento e 147 para teste, a distribuição de cada classe nos subconjuntos seguiu a mesma proporção de dois terços para treinamento e um terço para teste (tabela 3).

Tabela 3 - Quantidade de registros por classe de solo nos subconjuntos.

Classe do solo	Treinamento	Teste
Argissolo	50	24
Cambissolo	64	31
Gleissolo	6	3
Latossolo	68	34
Neossolo	18	8
Nitossolo	90	45
Organossolo	4	1
Planossolo	3	1
Total	303	147

Fonte: Do autor.

Para a implementação do SVM foi utilizada a biblioteca do Java *LibSVM*, a mesma possui as principais funções de *kernel* para serem utilizadas, as quais são linear, RBF, polinomial e sigmoidal. A função linear trata apenas de classificações linearmente separáveis, como o problema tratado nesta pesquisa não é linearmente separável esta função não será utilizada. Cada função de *kernel* utiliza diferentes quantidades de parâmetros, a RBF utiliza o parâmetro *gamma* (*γ*), a sigmoidal os parâmetros *γ* e coef0 (*r*) e por último o polinomial, que utiliza a maior quantidade de parâmetros, *γ, r e degree* (*d*) (tabela 4). Alguns parâmetros são gerais para o SVM que é o caso do parâmetro de custo (*C*), este parâmetro é utilizado para indicar a penalização por uma classificação incorreta, os demais parâmetros não interferem na precisão do modelo, pois tratam da quantidade de memória a ser utilizada, entre outros que não são utilizados nesta pesquisa.

Tabela 4 - Parâmetros de cada função de *kernel*.

Tipo de *kernel*	Função	Parâmetros
RBF	$\exp(-\gamma \lVert x_i - x_j \rVert^2)$	γ
Sigmoidal	$\tanh(\gamma(x_i . x_j) + r)$	γ e r
Polinomial	$(\gamma(x_i \ . \ x_j) + r)^d$	γ, r e d

Fonte: Adaptado de Lorena e Carvalho (2007).

Conforme Hsu, Chang e Lin (2016, tradução nossa), inicialmente é sugerida a escolha da função de *kernel* RBF, pois esta utiliza a menor quantidade de

parâmetros, o que torna a otimização mais fácil. Para encontrar os melhores valores para estes parâmetros foi empregada a técnica *Gridsearch,* que consiste em utilizar vetores para cada parâmetro, combinar os valores dos vetores e testar o SVM com a base de dados. Nesta técnica são selecionados os melhores parâmetros da rodada e alteram-se os valores dos vetores para valores mais próximos aos encontrados na rodada anterior e o processo se repete até que se encontre os melhores parâmetros.

Como a função de *kernel* sugerida inicialmente é a RBF, foi então implementado o *Gridsearch* para esta função. Para isso, foram criados dois vetores, um com os valores de *C* e outro com os valores de *γ,* utilizando dois laços de repetição os vetores foram percorridos realizando a combinação entre si. Ao executar o SVM os seus resultados eram comparados, caso fossem melhores que o anterior ambos os parâmetros eram armazenados para que se pudesse identificar qual obteve uma melhor performance. Os valores iniciais para a técnica, conforme sugerido por Hsu, Chang e Lin (2016, tradução nossa), são parâmetros em base 10, pois geralmente são os melhores. Assim, nesta pesquisa foram utilizados entre 10^{-7} até 10^{-1} para *γ* e entre 10^{-2} até 10^{4} para *C* (figura 21).

Figura 21 - Código fonte Java do método de *Gridsearch*.

```
public void otimizarTeste() {
    Thread t1 = new Thread(new Runnable() {
        @Override
        public void run() {
            treinamento.setClassIndex(-1);
            teste.setClassIndex(-1);

            double[] gridC = {0.01, 0.1, 1.0, 10, 100, 1000};
            double[] gridGamma = {0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1};
            Instanciar SVM e seta variáveis
            for (int j = 0; j < gridC.length; j++) {
                for (int k = 0; k < gridGamma.length; k++) {
                    options = "-S 0 -K 2 -D 3 -G " + gridGamma[k] + " "
                            + "-R 0.0 -N 0.5 -M 40.0 -C " + gridC[j] + " -E 0.001 -P 0.1";

                    dto = svm.executaSVM(true, i, options);
                    if (dto.getPrecisao() > melhorPrecisao) {
                        Armazena melhores parâmetros
                    }
                }
            }
```

Fonte: Do autor.

Para busca dos melhores parâmetros das diferentes funções de *kernel*, não foi implementado o *Gridsearch*. Para isso, em uma aplicação do SVM separada foram utilizados laços de repetição para testar os parâmetros que eram informados manualmente no código.

Na primeira rodada de execução do SVM foi utilizada a base de dados sem o emprego de técnicas de balanceamento e com os parâmetros padrão da biblioteca. Foi então obtida uma quantidade de erros de 59 do total de registros de 147, a precisão do modelo ficou em aproximadamente 59,8%, pode-se notar que as classes com menores quantidades de registros no treinamento tiveram praticamente 100% de erro, com exceção da classe neossolo a qual obteve 1 classificação correta de 8 no total (tabela 5). O tempo de treinamento e execução foi em média de 71,4 e 19,5 milissegundos respectivamente.

Tabela 5 - Primeira rodada da aplicação do SVM.

Classe do solo	Total no treinamento	Erros
Argissolo	24	12
Cambissolo	31	9
Gleissolo	3	3
Latossolo	34	14
Neossolo	8	7
Nitossolo	45	12
Organossolo	1	1
Planossolo	1	1
Total	147	59

Fonte: Do autor.

A segunda rodada de execução utilizou a técnica *Gridsearch*, a qual encontrou os melhores parâmetros para *C* e para *γ*, que foram 14,835 e 0,027 respectivamente. A quantidade de erro do modelo foi de 33 de um total de 147, sua precisão foi de aproximadamente 77,5%, nesta rodada com o ajuste dos parâmetros as classes que foram vistas menos vezes no treinamento apresentaram uma precisão melhor, onde apenas uma continuou com 100% de erros que foi a classe organossolo (tabela 6). Os tempos se mantiveram os mesmos.

Tabela 6 - Rodada do SVM com os melhores parâmetros.

Classe do solo	Total no treinamento	Erros
Argissolo	24	8
Cambissolo	31	3
Gleissolo	3	1
Latossolo	34	9
Neossolo	8	3
Nitossolo	45	8
Organossolo	1	1
Planossolo	1	0
Total	147	33

Fonte: Do autor.

Foram realizadas rodadas com o conjunto que foi aplicada a técnica SMOTE e também foram utilizados os pesos no treinamento do SVM. Uma abordagem comum para o cálculo do peso a ser utilizado é dividir o número total de registros (*N*) no conjunto de dados pelo número de classes (*Nº classes*), conforme equação 6. Para este conjunto de dados o peso resultante do cálculo é 37,875.

$$Peso = \frac{N}{Nº\ classes} \tag{9}$$

A aplicação da técnica SMOTE ou pesos para este conjunto de dados não se mostrou eficiente. Utilizando o SMOTE a quantidade de erros e a precisão se mantiveram as mesmas do que quando não utilizado.

Com a utilização dos pesos a precisão piorou, com uma quantidade de erros de 42 de um total de 147, a precisão ficou em aproximadamente 71,4%. Além de não melhorar a precisão ela também não otimizou a quantidade de acertos das classes que estão desbalanceadas, mantendo as mesmas quantidades de erros e aumentando a quantidade de erros das demais classes (tabela 7).

Os tempos de execução se mantiveram os mesmos, porém o tempo de treinamento aumentou quando utilizada a técnica SMOTE, pois nesta foram gerados registros sintéticos para o treinamento.

Tabela 7 - Rodadas do SVM com SMOTE e pesos

Classe do solo	Total no treinamento	Erros sem balancear	Erros SMOTE	Erros pesos
Argissolo	24	8	8	9
Cambissolo	31	3	3	3
Gleissolo	3	1	1	1
Latossolo	34	9	9	11
Neossolo	8	3	3	3
Nitossolo	45	8	8	14
Organossolo	1	1	1	1
Planossolo	1	0	0	0
Total	147	33	33	42

Fonte: Do autor.

Para este conjunto de dados os testes com diferentes funções de *kernel* não se mostraram eficientes, o melhor modelo obtido foi com a função de *kernel* RBF com os parâmetros de 14,835 para *C* e 0,027 para *γ* com o conjunto de dados sem aplicação da técnica SMOTE ou utilização de pesos. Neste modelo a precisão foi de aproximadamente 77,5% com os tempos de treinamento e execução em média de 71,4ms e 19,5ms respectivamente. Com melhor modelo também foram obtidos os dados das medidas estatísticas de qualidade para posterior análise e comparação na discussão dos resultados.

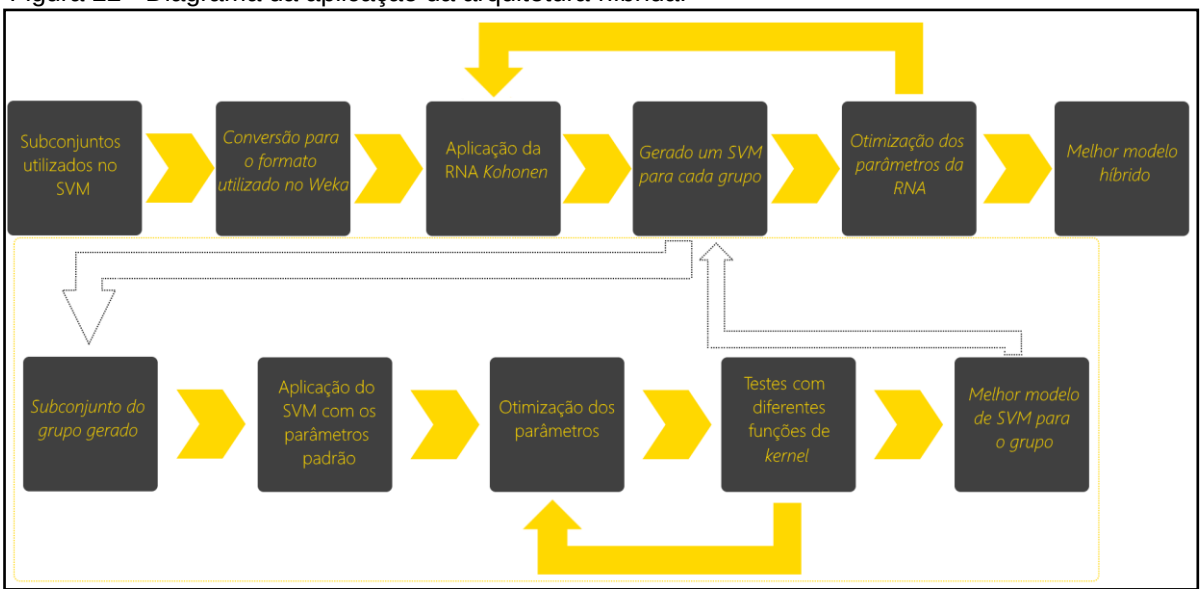
4.2.3 Aplicação da arquitetura híbrida

Nesta pesquisa optou-se pela realização do hibridismo com a utilização de uma RNA do tipo *Kohonen* para agrupar os dados do solo e após foi realizada a aplicação de um SVM para a tarefa de classificação em cada grupo de dados. A escolha deste tipo de hibridismo se deu após estudo dos trabalhos correlatos onde Liu, Xie e Wu (2008, tradução nossa) aplicaram um hibridismo similar, obtendo resultados melhores quando utilizada a arquitetura híbrida.

Nesta etapa foram utilizados os mesmos subconjuntos de treinamento e testes utilizados para a aplicação isolada do SVM, porém foi necessária a conversão destes para um novo formato. Na arquitetura não foi utilizado o conjunto que foi aplicada a técnica de SMOTE para balanceamento, pois conforme os resultados do SVM esta técnica não foi eficiente. Em seguida, os subconjuntos foram utilizados em

uma RNA do tipo *Kohonen* que gerou grupos para estes dados, conforme o parâmetro utilizado na RNA. Posteriormente para cada grupo foi gerado um SVM e otimizado da mesma forma quando se aplica o SVM. Assim, para cada grupo foi utilizado o *Gridsearch* para buscar os melhores parâmetros, testando-se com diferentes funções de *kernel* até que o melhor SVM para o grupo fosse encontrado, a fim de obter a melhor arquitetura os parâmetros da RNA também foram alterados em cada rodada e o processo do SVM de gerar um para cada grupo e otimizar se repetiu sempre que alterado os parâmetros da RNA (figura 22).

Figura 22 - Diagrama da aplicação da arquitetura híbrida.



Fonte: Do autor.

Para a aplicação da RNA do tipo *Kohonen* foi utilizada *Weka*, que é uma ferramenta de mineração de dados criada pela universidade de Waikato da Nova Zelândia a sua sigla é proveniente de ***W**aikato **E**nvironment for **K**nowledge **A**nalysis* (WEKA) (FRANK; HALL; WITTEN, 2016, tradução nossa). Esta ferramenta possibilita a aplicação de vários algoritmos de aprendizagem em conjuntos de dados, a *Weka* API disponibiliza as funcionalidades da ferramenta em forma de biblioteca para a linguagem Java.

A escolha da biblioteca *Weka* API se deu pelo fato da sua compatibilidade com o *LibSVM* que já estava sendo utilizado nesta pesquisa e também por possuir a

possibilidade de adicionar a implementação da RNA do tipo *Kohonen* por meio do seu gerenciador de pacotes ou realizando o *download* diretamente pelo site[12].

A biblioteca utiliza arquivos no formato *arff,* para realizar a conversão dos arquivos que estavam no formato do *LibSVM* para o formato *arff* foi utilizado o *script* disponível no *GitHub*[13] na linguagem *Python*. No entanto, foram necessárias adaptações na forma de leitura do arquivo no formato original.

O arquivo *arff* foi desenvolvido para ser utilizado junto com o *Weka*, este formato trata-se de um arquivo de texto que conta com uma estrutura de cabeçalho e dados. No cabeçalho é identificado primeiramente de qual arquivo ele se originou por meio da notação `@RELATION`, após são identificados os atributos com seus identificadores e tipos, utiliza-se a notação `@ATTRIBUTE NOME TIPO`, como último atributo é informado a classe também se utiliza a notação para atributos, porém nesta em vez de colocar o tipo é identificado os possíveis valores que este atributo pode assumir, como por exemplo, `@ATTRIBUTE class {1,2,3}`. Por fim os dados são informados após a notação `@DATA`, estes devem estar em uma única linha para cada registro e devem informar os valores dos atributos na ordem em que foram declarados no cabeçalho e separados por vírgula. Na figura 22 encontra-se o arquivo em formato *arff* utilizado nesta pesquisa, para demonstrar toda a estrutura os atributos foram resumidos.

Figura 23 - Exemplo de arquivo no formato *arff*.

```
1  @RELATION .\dataset.treinamento.scale
2  @ATTRIBUTE a28 REAL
3  @ATTRIBUTE a33 REAL
4  @ATTRIBUTE a4 REAL
5  @ATTRIBUTE a9 REAL
6  ...
7  @ATTRIBUTE a22 REAL
8  @ATTRIBUTE a35 REAL
9  @ATTRIBUTE a52 REAL
10 @attribute class {1,2,3,4,5,6,7,8}
11 @DATA
12 -1,-1,-0.777778,-0.151515,0.538462,-1,-1,-1,0.393586,0.758007,-0.6,0.5,-1,-1
13 1,-1,-0.777778,-0.454545,0.538462,-1,-1,-1,0.381924,0.608541,-0.866667,1,-1,
14 -1,-1,-0.777778,-0.954545,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-0.81818
15 -1,-1,0.333333,-0.515152,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-0.818182
16 0.0,-1,0.333333,-0.590909,0.230769,-1,-1,-1,0.54519,0.701068,-0.733333,0.75,
17 0.0,-1,0.333333,-0.839394,0.384615,-1,-1,-1,-0.0204082,0.11032,0.0666667,0.5
```

Fonte: Do autor.

[12] Disponível em: http://weka.sourceforge.net/packageMetaData/SelfOrganizingMap/index.html
[13] Disponível em: https://github.com/wammar/wammar-utils/blob/master/convert-libsvm-format-to-arff-format.py

Após a conversão do arquivo pode então ser aplicada a RNA do tipo *Kohonen*, nesta pesquisa foram utilizados diferentes parâmetros na RNA para avaliar quais apresentavam melhores resultados. Conforme o funcionamento da RNA os parâmetros de largura (*w*) e altura (*h*) definem a quantidade de grupos que são gerados, foram utilizadas variações nos parâmetros, para cada grupo gerado foram seguidas as etapas conforme a aplicação do SVM, que foram utilizar inicialmente no SVM a função de *kernel* RBF com os parâmetros padrão, em seguida foram realizadas as otimizações com a técnica de *Gridsearch* e teste com diferentes funções de *kernel*.

Para a implementação do hibridismo foi primeiramente executada a RNA que separa o conjunto de dados em grupos de acordo com os parâmetros informados. Nesta etapa é executada a RNA no conjunto de treinamento, posteriormente no conjunto de testes, separando ambos os conjuntos em grupos. Por meio de um laço de repetição é criado um SVM para cada grupo, passando como argumento os conjuntos de treinamento e de teste do respectivo grupo, em seguida executa-se o SVM, conforme código fonte apresentado na figura 24.

Figura 24 - Código fonte minimizado do hibridismo.

```
public void executaHibridismo() {
    Thread t1 = new Thread(new Runnable() {
        @Override
        public void run() {
            Lê parâmetros
            try {
                Seta variáveis
                dto = hibrido.executaCluster();
                Seta os valores dos tempos do cluster na tela
                numCluster = dto.getNumCluster();
                Inicializa arrays
                for (int i = 0; i < numCluster; i++) {
                    //Cria um SVM pra cada arquivo de treinamento e teste criados a partir da RNA
                    svm = new SVM("tmp/treinamento." + i + ".arff", "tmp/teste." + i + ".arff");
                    //Executa o SVM
                    dto = svm.executaSVM(true, i, numCluster);
                    //Armazena os resultados gerais
                    tempoTotalTreinamento += dto.getTempoTreinamento();
                    tempoTotalTeste += dto.getTempoTeste();
                    totalAcertos += dto.getAcertos();
                    totalErros += dto.getErros();
                    precisaoGeral += dto.getPrecisao();
                    arrayDTO[i] = dto;
                    Seta dados na tela
                }
                precisaoGeral = (double) totalAcertos / (totalErros + totalAcertos);
                Seta os dados finais na tela
            } catch (Exception ex) {
                mostrarErro("Erro executando o SVM.\n" + ex.getMessage());
            }
```

Fonte: Do autor.

Foram realizadas rodadas testando a variação de parâmetros da RNA do tipo *Kohonen* para que fossem geradas as quantidades de grupos de dois, quatro, cinco, oito e quinze. Para isto foram utilizados os parâmetros *w* e *h* de um e dois, dois e dois, um e cinco, dois e quatro e um e quinze (tabela 8).

Tabela 8 - Parâmetros utilizados em cada rodada da arquitetura híbrida.

Rodada	Parâmetros	Quantidade de grupos
1	w 1 e *h* 2	2
2	w 2 e *h* 2	4
3	w 1 e *h* 5	5
4	w 2 e *h* 4	8
5	w 1 e *h* 15	15

Fonte: Do autor.

Em cada rodada foram realizadas as otimizações de cada SVM para cada grupo, em algumas rodadas as variações da função de *kernel* mostrou-se eficiente, como foi o caso das rodadas 1, 2, 3 e 4. Nestas rodadas alguns grupos obtiveram melhores performances com as funções de *kernel* polinomial e sigmoidal, que foi o caso do grupo 1 da primeira rodada em que utilizou a função polinomial, na segunda rodada o grupo 3 também utilizou a mesma função, já na terceira rodada os grupos 3 e 4 mostraram-se eficientes com o *kernel* polinomial e na quarta rodada os grupos 3 e 5 utilizaram o *kernel* polinomial e o grupo 8 teve melhor performance com o *kernel* sigmoidal (tabela 9).

Em relação as quantidades de erros de cada rodada foi de 33 na primeira, mantendo assim a mesma performance em termos de precisão da aplicação do SVM isoladamente, isto é, a precisão manteve-se em aproximadamente 77,5%, para a segunda rodada a quantidade de erros teve um aumento para 37, a sua precisão ficou em aproximadamente 74,8%. Na terceira rodada houve um aumento ainda maior na quantidade de erros que foi para 43 e a precisão da arquitetura foi de aproximadamente 70,7%, já na quarta rodada a quantidade de erros foi menor que a anterior, porém manteve-se maior que quando aplicado somente o SVM, que foi de 36 e sua precisão foi de aproximadamente 75,5%. Na última rodada a quantidade de erros teve um pequeno aumento em comparação a rodada anterior e foi para 37 e sua

precisão foi de aproximadamente 74,8%, apresentando assim o mesmo resultado que a segunda rodada (tabela 9).

Tabela 9 - Resultados dos experimentos com diferentes números de grupos

Grupos	Grupo	Quantidade	Parâmetro do SVM	Erros
2	1	117	*Kernel* polinomial, d 4, r 0.8, γ 0.1 e C 26	24
	2	30	*Kernel* RBF, γ 0.114 e C 10	9
	Total	147		33
4	1	30	*Kernel* RBF, γ 0.09672 e C 5.5	9
	2	29	*Kernel* RBF, γ 0.04 e C 1.5	4
	3	29	*Kernel* polinomial, d 4 , γ 0.077 e C 0.6	8
	4	59	*Kernel* RBF, γ 0.09672 e C 1.1	16
	Total	147		37
5	1	30	*Kernel* RBF, γ 0.116 e C 16.15	9
	2	21	*Kernel* RBF, γ 0.11 e C 1.6	3
	3	29	*Kernel* polinomial, d 2, γ 0.04705 e C 1.5	7
	4	40	*Kernel* polinomial, d 3, γ 0.0223 e C 5.5	11
	5	27	*Kernel* RBF, γ 0.017 e C 62.9	13
	Total	147		43
8	1	25	*Kernel* RBF, γ 0.025 e C 111	4
	2	18	*Kernel* RBF, γ 0.04161499999999636 e C 1	4
	3	26	*Kernel* polinomial, d 17, γ 0.5	5
	4	15	*Kernel* RBF, γ 0.014 e C 6	3
	5	16	*Kernel* polinomial, d 11, γ 0.014 e com pesos de 5.75	6
	6	6	*Kernel* RBF, γ 0.01 e C 0.1	1
	7	11	*Kernel* RBF, γ 0.0132 e C 51	5
	8	30	*Kernel* sigmoidal, γ 0.0063 e C 111	8
	Total	147		36
15	1	16	*Kernel* RBF, γ 0.0008 e C 100	4
	2	8	*Kernel* RBF, γ 0.3 e C 1	3
	3	1	*Kernel* RBF, γ 0.01 e C 10	0
	4	9	*Kernel* RBF, γ 0.01 e C 10	0
	5	15	*Kernel* RBF, γ 0.015 e C 10	6
	6	16	*Kernel* RBF, γ 0.1 e C 1.6	2
	7	20	*Kernel* RBF, γ 0.1 e C 1	4
	8	1	*Kernel* RBF, γ 0.0001 e C 0.01	0
	9	6	*Kernel* RBF, γ 0.01 e C 10	1
	10	11	*Kernel* RBF, γ 0.01 e C 120	5
	11	10	*Kernel* RBF, γ 0.1 e C 10	5
	12	4	*Kernel* RBF, γ 0.0001 e C 200	0
	13	12	*Kernel* RBF, γ 0.05 e C 10	5
	14	7	*Kernel* RBF, γ 0.01 e C 50	1
	15	11	*Kernel* RBF, γ 0.1 e C 10	1
	Total	147		37

Fonte: Do autor.

Com relação aos tempos de treinamento e execução, como a RNA do tipo *Kohonen* tem um tempo maior de treinamento, em torno de dois minutos para o caso com mais grupos, esta medida não será utilizada nas análises, já o tempo de execução, a RNA possui um tempo de execução baixo em média de 0,2 milissegundos para esta arquitetura foram realizadas análises entre os tempos de execução.

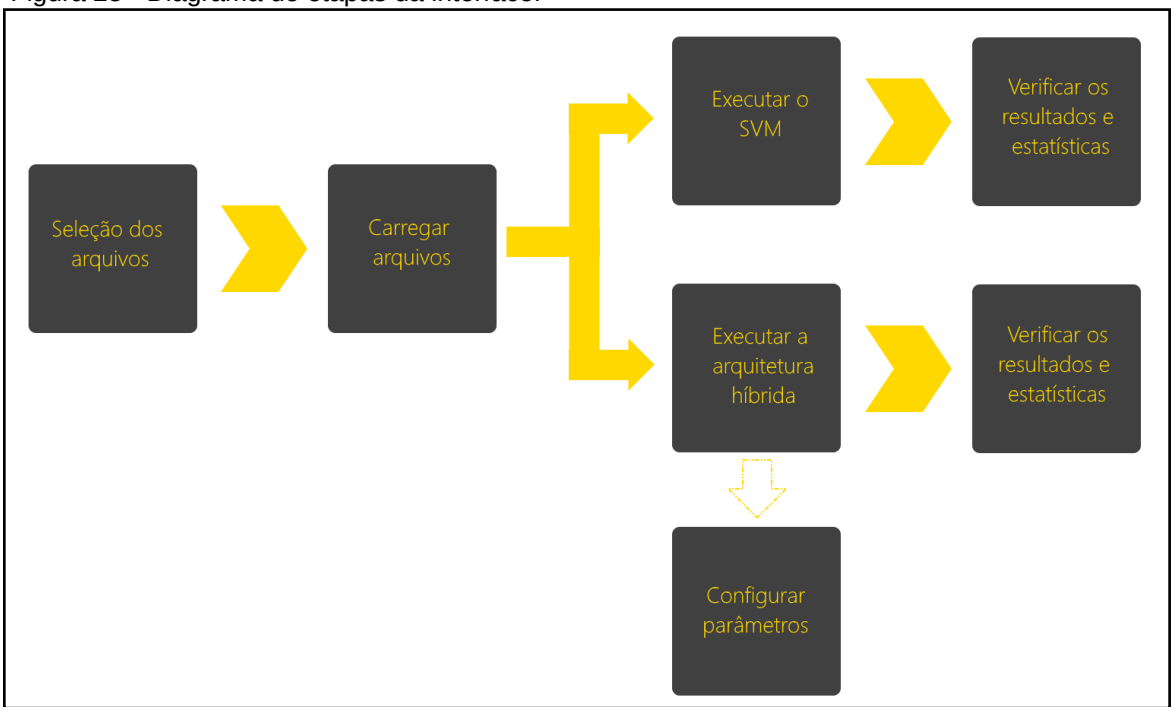
Na primeira rodada o tempo de execução foi em média de 10,6ms, nas rodadas seguintes foram de 8,1ms, 5,4ms, 4,5ms e 3,9ms para a segunda, terceira, quarta e quinta rodada respectivamente.

A melhor arquitetura em relação a precisão foi a utilizada na primeira rodada que foram utilizados os valores de um e dois para os parâmetros *w* e *h* da RNA do tipo *Kohonen*. Para o SVM do grupo um foi utilizado o kernel *polinomial* e os parâmetros de 26 para *C*, 0,1 para *γ*, 4 para *d* e 0,8 para *r*, no SVM do grupo dois foi utilizado o *kernel* RBF com os parâmetros de 10 para *C* e 0,114 para *γ*.

4.2.4 Interface gráfica

Nesta etapa foi desenvolvida uma interface gráfica com o objetivo de facilitar a interação com o usuário e visualização dos dados, na interface é necessário selecionar os arquivos de treinamento e testes, ambos no formato *arff*, em seguida os mesmos são carregados e pode-se executar a técnica do SVM isoladamente ou então executar a arquitetura híbrida da RNA do tipo *Kohonen* e SVM, finalmente pode-se visualizar os resultados na tela e também visualizar os dados estatísticos (figura 25).

Figura 25 - Diagrama de etapas da interface.



Fonte: Do autor.

Para que o usuário realize a seleção dos arquivos de treinamento e teste, foram criados na tela principal, no quadro Arquivos dois campos, os arquivos devem estar no formato *arff*. Após a seleção é necessário carregá-los para o sistema utilizando o botão carregar, o sistema então irá converter os arquivos em instâncias utilizadas pelo *Weka*, também será demonstrado a quantidade de registros em cada arquivo (figura 26).

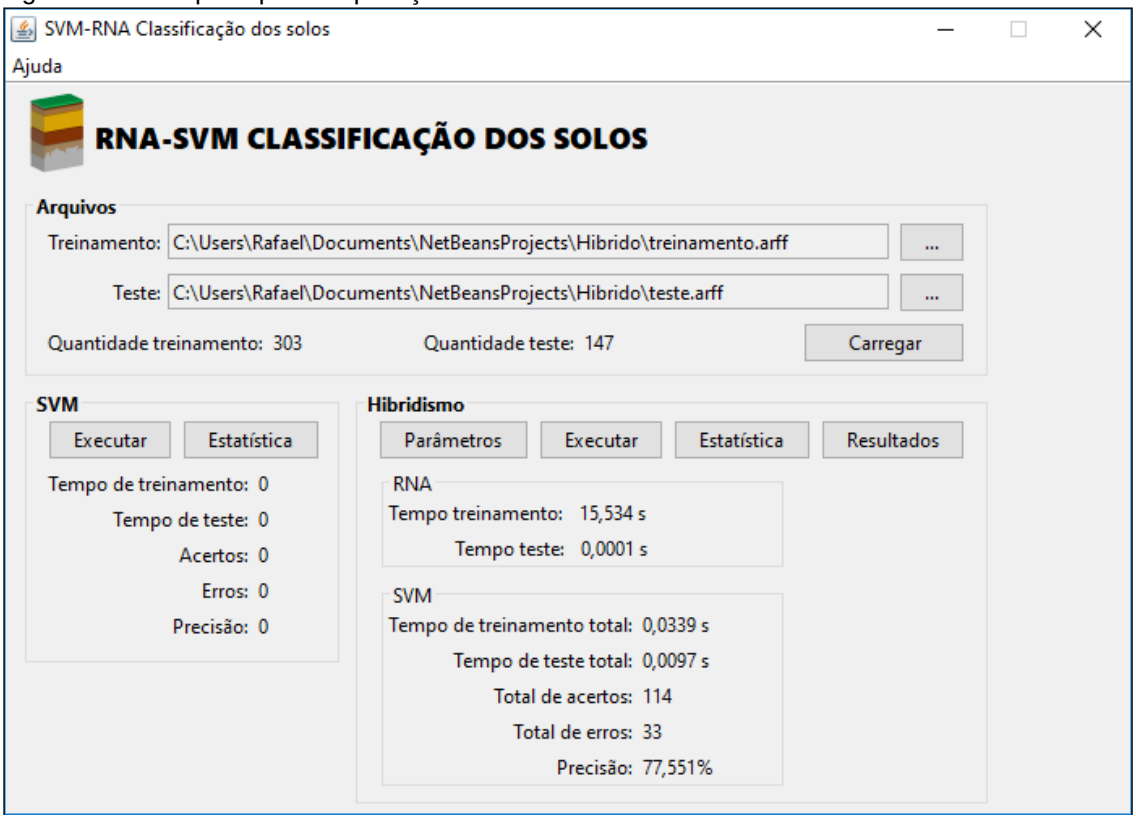
No quadro SVM será possível executar a técnica de máquinas de vetores de suporte isoladamente, esta execução utilizará os melhores parâmetros que foram encontrados. Após a execução são preenchidas as informações de tempo de treinamento, tempo de teste, quantidade de acertos, quantidade de erros e precisão.

No quadro Hibridismo é possível configurar os parâmetros da RNA *Kohonen*, a tela de configuração que é aberta pelo botão *Parâmetros*, permite alterar a largura (*w*) e a altura (*h*) da RNA. Os parâmetros padrão são os melhores obtidos em relação a precisão, ou seja, um e dois respectivamente. Caso seja informado um parâmetro diferente destes ou diferentes dos parâmetros que foram otimizados nesta pesquisa, a arquitetura utilizará o método *Gridsearch* para selecionar os melhores parâmetros de cada SVM. As utilizações de parâmetros não otimizados podem ocasionar erros ou uma baixa precisão, pois podem gerar grupos sem registros ou a

busca pelos melhores parâmetros não ser ideal já que o intervalo de busca é limitado. No quadro RNA é exibido os tempos despendidos nas fases de treinamento e de teste da rede neural artificial do tipo *Kohonen.*

No quadro Resumo do SVM são demonstradas as informações do tempo total de treinamento, tempo total de teste, estes tempos levam em conta a execução do SVM inteiro, ou seja, somatório dos tempos de cada SVM de cada grupo, também são demonstradas as quantidades de acertos e erros e a precisão da arquitetura híbrida.

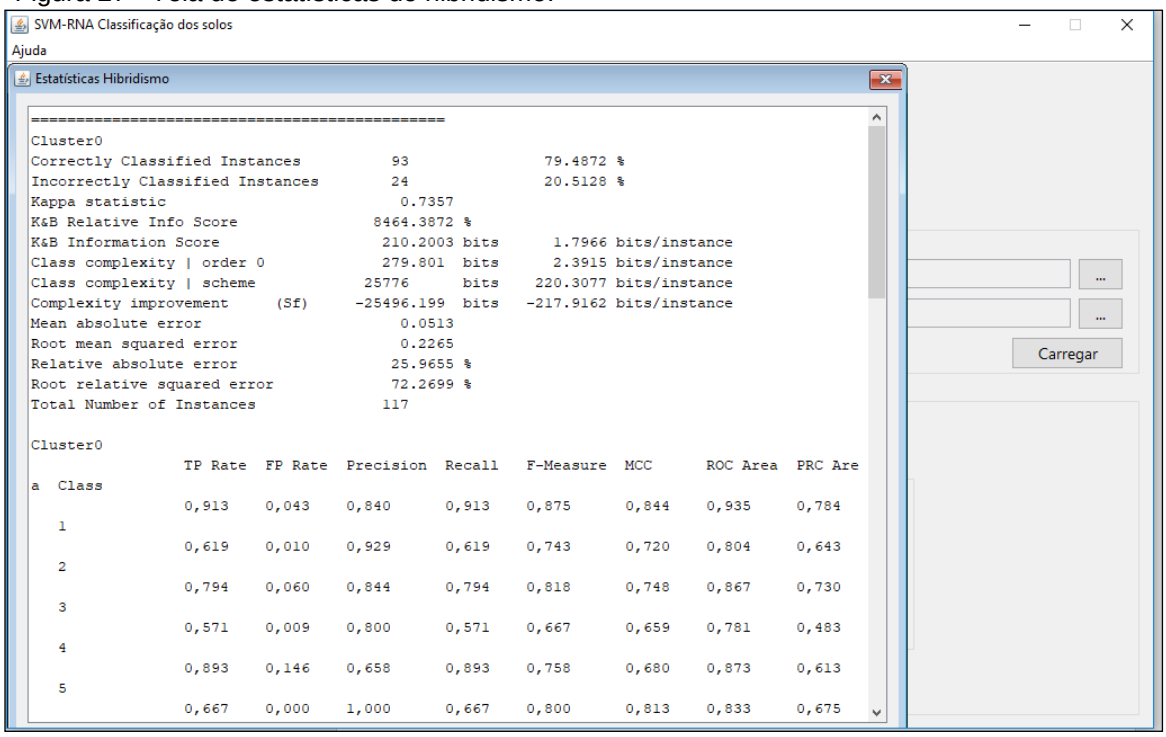
Figura 26 - Tela principal da aplicação.



Fonte: Do autor.

Ao clicar no botão *Resultados* é aberta uma tela na qual pode-se visualizar os dados individuais de cada grupo com as informações de quantidade de acertos, quantidade de erros e precisão do SVM. Finalmente é habilitado o botão *Estatísticas*, ao clicar é aberta uma janela na qual pode ser visualizado os dados estatísticos, quando clicado no quadro SVM são exibidas suas informações e quando clicado no quadro Hibridismo são exibidas as informações de cada SVM de cada grupo (figura 27).

Figura 27 - Tela de estatísticas do hibridismo.




```
===============================================
Cluster0
Correctly Classified Instances          93               79.4872 %
Incorrectly Classified Instances        24               20.5128 %
Kappa statistic                          0.7357
K&B Relative Info Score               8464.3872 %
K&B Information Score                  210.2003 bits      1.7966 bits/instance
Class complexity | order 0             279.801  bits      2.3915 bits/instance
Class complexity | scheme            25776      bits    220.3077 bits/instance
Complexity improvement     (Sf)     -25496.199  bits   -217.9162 bits/instance
Mean absolute error                      0.0513
Root mean squared error                  0.2265
Relative absolute error                 25.9655 %
Root relative squared error             72.2699 %
Total Number of Instances              117

Cluster0
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Are
a  Class
                 0,913    0,043    0,840      0,913    0,875      0,844    0,935     0,784
   1
                 0,619    0,010    0,929      0,619    0,743      0,720    0,804     0,643
   2
                 0,794    0,060    0,844      0,794    0,818      0,748    0,867     0,730
   3
                 0,571    0,009    0,800      0,571    0,667      0,659    0,781     0,483
   4
                 0,893    0,146    0,658      0,893    0,758      0,680    0,873     0,613
   5
                 0,667    0,000    1,000      0,667    0,800      0,813    0,833     0,675
```

Fonte: Do autor.

4.3 RESULTADOS E DISCUSSÃO

Nesta pesquisa foram utilizados métodos estatísticos para avaliação tanto da aplicação do SVM isolado quanto para a aplicação da arquitetura híbrida. Como o hibridismo escolhido consistiu na implementação de um SVM para cada grupo gerado pela RNA foi necessária a utilização de uma média ponderada para que os valores pudessem ser comparados entre si, já que as estatísticas são geradas para cada SVM e não para o modelo híbrido em geral.

Desta forma, para calcular a taxa geral de verdadeiros positivos foi realizado um somatório do valor individual de VP (VP_n) e multiplicado pela quantidade de registros que o mesmo classificou (W_n), em seguida foi dividido o somatório pela quantidade total de registros no conjunto de testes (equação 10). A mesma equação foi aplicada para todas as medidas estatísticas a fim de permitir a comparação entre cada modelo gerado.

$$Taxa\ geral\ de\ VP = \frac{VP_1.W_1 + VP_2.W_2 + \cdots + VP_n.W_n}{\text{Total registros de teste}} \qquad (10)$$

Primeiramente foram avaliados os dados estatísticos dos classificadores gerados, para isso empregaram-se as medidas de taxa de verdadeiros positivos, taxa de falsos positivos, taxa de verdadeiros negativos, taxa de falsos negativos, sensibilidade (*recall*), especificidade e área abaixo da curva ROC. Um classificador é considerado bom quando possui uma maior sensibilidade, especificidade e acurácia, e um menor número de erros, taxa de falsos negativos e taxa de falsos positivos.

Para análise deve-se considerar que a taxa de VP, taxa de VN, *Recall*, especificidade e área abaixo da curva ROC quanto mais próximo a 1 seus valores, melhor é o classificador. Para a taxa de FP e taxa de FN é o contrário, ou seja, quanto mais próximos a zero melhor (LAROSE; LAROSE, 2014, tradução nossa).

Na aplicação do SVM o melhor modelo obteve a quantidade de erros de 33 e sua precisão foi de aproximadamente 77,5% (tabela 10), o SVM ainda contou com uma taxa de VP de 0,7755 esta medida é o mesmo valor que a precisão na qual a taxa de VP considera as classificações corretas do total de classificações, taxa de FP de 0,0822, taxa de VN de 0,9178, taxa de FN 0,2245, *recall* de 0,7755, especificidade de 0,9178 e capacidade de classificar corretamente de acordo com a área abaixo da curva ROC de 0,8467 (tabela 11).

A aplicação da arquitetura híbrida com dois grupos apresentou a mesma precisão da aplicação de somente o SVM, isto é, manteve-se em aproximadamente 77,5% (tabela 10). No entanto, na arquitetura ocorreram algumas diferenças nas medidas de qualidade, a taxa de FP foi menor em comparação ao SVM, na qual a arquitetura apresentou 0,0731, enquanto que somente o SVM foi de 0,0822, houve uma diferença de 0,0091 para a arquitetura híbrida com dois grupos, a taxa de VN foi maior no hibridismo (0,9269) do que no SVM (0,9178), o que apresenta uma pequena melhora de 0,0091 para o hibridismo. Esta mesma melhora pode ser observada na especificidade, a área abaixo da curva ROC também foi um pouco melhor na arquitetura com o valor de 0,8512 (tabela 11). Embora a arquitetura apresente algumas pequenas melhoras nas medidas de avaliação não se pode concluir que estas melhoras são estatisticamente significativas.

Ao utilizar quatro grupos no hibridismo a precisão teve uma queda em relação as aplicações anteriores ficando em 74,8% (tabela 10). Suas medidas de qualidade também tiveram uma queda quando comparada ao hibridismo com dois grupos, apresentou uma taxa de FP de 0,0805, taxa de VN de 0,9195 esta medida

apresentou uma discreta melhora com relação a aplicação do SVM, taxa de FN de 0,2517, *recall* 0,7483, especificidade e área abaixo da curva ROC de 0,9195 e 0,8339 respectivamente (tabela 11).

No hibridismo com cinco grupos houve uma queda na precisão de 6,8% quando comparado ao SVM ou ao hibridismo com dois grupos, a precisão foi de 70,7% (tabela 10). A mesma queda de desempenho pode ser vista nas medidas de qualidade, na qual apresentou uma taxa de FP de 0,0731, taxa de VN de 0,9061, taxa de FN de 0,2925, *recall* de 0,7075, a menor diferença foi encontrada na especificidade em que foi observado o valor de 0,9061, sua capacidade de realizar classificações corretas também foi menor, com o valor de 0,8068 (tabela 11).

Na arquitetura híbrida com oito grupos a precisão alcançada foi próxima ao SVM, o seu valor ficou aproximadamente em 75,5% uma diferença de 2% para o SVM, este foi o segundo melhor resultado em relação a precisão (tabela 10). No entanto, suas medidas de qualidade indicam o contrário, pois foram inferiores inclusive a aplicação do hibridismo com cinco grupos, ou seja, a precisão não refletiu na qualidade do classificador. Sua taxa de FP foi a maior entre todas as aplicações com o valor de 0,1666, a taxa de VN também foi a menor de todas aplicações com 0,8334, a taxa de FN e *recall* apresentaram os valores de 0,2449 e 0,7551 respectivamente, a especificidade e capacidade de classificação também foram as menores com os valores de 0,8334 e 0,7942 (tabela 11).

Finalmente a aplicação do hibridismo com quinze grupos, apresentou uma precisão menor quando comparada com a de oito grupos, com aproximadamente 74,8% (tabela 10). As análises das medidas de qualidade do classificador indicam os mesmos casos do hibridismo com oito grupos, porém apresentam pequenas melhoras quando comparado ao mesmo. Nesta aplicação a taxa de FP apresentou um valor de 0,1293, sendo superior apenas ao hibridismo anterior com oito grupos, isto também ocorre na taxa de VN e taxa de FN com 0,8571 e 0,2517 respectivamente. O valor do recall e especificidade foram de 0,7483 e 0,8707, sua capacidade classificação correta foi de 0,7959 (tabela 11).

Tabela 10 - Quantidade de erros e precisão das aplicações.

Aplicação	Quantidade total	Quantidade de erros	Precisão
SVM	147	33	77,5%
Arquitetura híbrida com 2 grupos	147	33	77,5%
Arquitetura híbrida com 4 grupos	147	37	74,8%
Arquitetura híbrida com 5 grupos	147	43	70,7%
Arquitetura híbrida com 8 grupos	147	36	75,5%
Arquitetura híbrida com 15 grupos	147	37	74,8%

Fonte: Do autor.

Tabela 11 - Medidas estatísticas dos experimentos.

Aplicação	Taxa VP	Taxa FP	Taxa VN	Taxa FN	*Recall*	Especificidade	Área ROC
SVM	**0,7755**	0,0822	0,9178	**0,2245**	**0,7755**	0,9178	0,8467
2 grupos	**0,7755**	**0,0731**	**0,9269**	**0,2245**	**0,7755**	**0,9269**	**0,8512**
4 grupos	0,7483	0,0805	0,9195	0,2517	0,7483	0,9195	0,8339
5 grupos	0,7075	0,0939	0,9061	0,2925	0,7075	0,9061	0,8068
8 grupos	0,7551	0,1666	0,8334	0,2449	0,7551	0,8334	0,7942
15 grupos	0,7483	0,1293	0,8571	0,2517	0,7483	0,8707	0,7959

Taxa VP, Taxa VN, *Recall,* Especificidade e Área ROC - valores próximos a 1
Taxa FP e Taxa FN - valores próximos a 0
Fonte: Do autor.

Dentre as avaliações dos modelos também foram considerados seus tempos de execução, para a avaliação foram utilizados médias e desvio padrão, ainda por meio do teste U de *Mann-Whitney* foi identificado a existência de significância estatística, para isto o valor-p deve ser menor que 0,05.

A aplicação do SVM obteve uma média de tempo de execução de 19,5ms com ±6,2ms de desvio padrão, no hibridismo com dois grupos o tempo foi de 10,6ms ±5,5ms, apresentou uma diferença em relação ao tempo de execução do SVM, conforme foram aumentando as quantidades de grupos esta diferença foi aumentando. Na arquitetura híbrida com quatro grupos o tempo foi de 8,1ms ±5,7ms, para as aplicações seguintes os tempos foram de 5,4ms ±3,9ms, 4,5ms ±3.3ms e 3,9ms ±3,1ms para o hibridismo com cinco, oito e quinze grupos respectivamente. Todos os tempos apresentaram significância estatística, pois apresentaram o valor-p $< 0,001$ (tabela 12).

Tabela 12 - Média e desvio padrão dos tempos de execução em ms das aplicações.

	Média ± Desvio padrão	
	Execução n = 30	Valor-p†
SVM	19,5±6,2	<0,001
Arquitetura híbrida com 2 grupos	10,6±5,5	<0,001
Arquitetura híbrida com 4 grupos	8,1±5,7	<0,001
Arquitetura híbrida com 5 grupos	5,4±3,9	<0,001
Arquitetura híbrida com 8 grupos	4,5±3,3	<0,001
Arquitetura híbrida com 15 grupos	3,9±3,1	<0,001

†Valores obtidos por meio da aplicação do teste U de Mann-Whitney.
Fonte: Do autor.

Para analisar se as diferenças entre os tempos de execução foram significativas estatisticamente foi utilizado o teste H de Kruskal-Wallis e *post hoc* de Dunn, que realiza a comparação em pares. As representações das letras ao lado dos valores de tempos indicam se possuem ou não diferenças significativas, os que possuem a mesma letra não apresentam, caso as letras não sejam as mesmas, pode-se interpretar que os pares apresentam diferenças significativas (tabela 13).

Os tempos obtidos com a aplicação do SVM e do hibridismo com dois grupos não apresentaram diferenças, já o SVM quando comparado com as demais aplicações do hibridismo apresentou diferença significativa. Quando comparada a aplicação do hibridismo entre si, pode-se avaliar que o hibridismo com dois grupos, quatro e cinco não apresentaram diferenças entre si, porém apresentaram diferença quando comparado com os de oito e quinze grupos (tabela 13).

Tabela 13 - Média e desvio padrão dos tempos de execução em ms das aplicações com aplicação do teste H de Kruskal-Wallis e *post hoc* de Dunn.

	Média ± Desvio padrão	
	Execução n = 30	Valor-p†
SVM[a]	19,5±6,2	< 0,001
Arquitetura híbrida com 2 grupos[a,b]	10,6±5,5	
Arquitetura híbrida com 4 grupos[b]	8,1±5,7	
Arquitetura híbrida com 5 grupos[b]	5,4±3,9	
Arquitetura híbrida com 8 grupos[c]	4,5±3,3	
Arquitetura híbrida com 15 grupos[c]	3,9±3,1	

†Valores obtidos por meio da aplicação do teste H de Kruskal-Wallis.
[a,b,c]Letras diferentes representam diferença estatisticamente significativa obtida por meio da aplicação do teste *post hoc* de Dunn.
Fonte: Do autor.

Com os resultados obtidos pode-se avaliar que o melhor modelo híbrido para a base de dados utilizada nesta pesquisa foi o de dois grupos que obteve a melhor precisão e as melhores medidas de avaliação, no entanto o mesmo não apresentou uma precisão maior do que a aplicação isolada do SVM e também não apresentou otimizações significativas estatisticamente no tempo de execução. Os demais modelos híbridos apresentaram melhoras significativas nos tempos de execução quando comparada com a aplicação isolada do SVM, porém estes modelos perderam precisão e obtiveram resultados inferiores nas medidas de qualidade empregadas para a avaliação dos modelos.

Pode-se realizar uma análise para considerar até que ponto a perda da eficácia do classificador compensa no ganho de tempo de execução (tabela 14).

Tabela 14 - Principais medidas de avaliação do classificador com os tempos de execução.

Aplicação	Precisão	Taxa FP	Taxa VN	*Recall*	Especificidade	Área ROC	Tempo execução (ms)
SVM	77,5%	0,0822	0,9178	0,7755	0,9178	0,8467	19,5±6,2
2 grupos	77,5%	0,0731	0,9269	0,7755	0,9269	0,8512	10,6±5,5
4 grupos*	74,8%	0,0805	0,9195	0,7483	0,9195	0,8339	8,1±5,7
5 grupos*	70,7%	0,0939	0,9061	0,7075	0,9061	0,8068	5,4±3,9
8 grupos*	75,5%	0,1666	0,8334	0,7551	0,8334	0,7942	4,5±3,3
15 grupos*	74,8%	0,1293	0,8571	0,7483	0,8707	0,7959	3,9±3,1

* apresentaram diferenças estatisticamente significantes quando comparado os tempos de execução com o tempo de execução do SVM
Taxa VP, Taxa VN, *Recall,* Especificidade e Área ROC - valores próximos a 1
Taxa FP e Taxa FN - valores próximos a 0
Fonte: Do autor.

Os resultados obtidos demonstraram que para esta base de dados o modelo híbrido não melhorou a acurácia, conforme as aplicações de hibridismo vistas nos trabalhos correlatos, em que na pesquisa de Li e Huang (2009, tradução nossa) aplicando o hibridismo entre uma RNA do tipo *Kohonen* e um SVM para cada grupo obteve uma melhora na precisão de aproximadamente 85% para precisões acima de 90%.

Na pesquisa de Lee, Roh e Choi (2009, tradução nossa) na utilização do hibridismo de SVM com RNA, onde a arquitetura difere-se da maneira em que foi implementada neste trabalho, obtiveram melhor precisão e tempos de treinamento e execução quando aplicado o modelo híbrido. Com os resultados obtidos foi possível analisar que o ganho no tempo de execução custou uma menor precisão.

5 CONCLUSÃO

As técnicas de inteligência artificial vêm crescendo suas aplicações atualmente, devido ao fato de que o avanço da tecnologia aumentou o poder de processamento, sendo assim tornando possível a aplicação de técnicas de aprendizado de máquina cada vez mais complexas para resolução de problemas.

A tarefa de classificação vem sendo aplicada com sucesso em diversos estudos e áreas do conhecimento, esta pesquisa que foi desenvolvida teve como base o desenvolvimento de uma arquitetura híbrida para a classificação dos solos e avaliação do modelo por meio de métodos estatísticos.

Foram encontradas dificuldades para escolha do modelo híbrido que seria implementado, pois nas revisões bibliográficas encontradas o foco do hibridismo era o domínio de problema e não o modelo computacional. Também foram encontradas dificuldades para buscar os melhores modelos de SVM, dificuldade esta que foi superada com a revisão bibliográfica e então a utilização da técnica de *Gridsearch*. A otimização dos classificadores também se demonstrou um processo que demanda uma quantidade de tempo elevada.

Além das dificuldades encontradas, a pesquisa demonstrou algumas limitações, como a pouca quantidade de registros na base de dados, o que pode ocasionar uma precisão baixa do SVM, outra limitação está relacionada a parte estatística da pesquisa em que devido ao pouco tempo hábil foi pouco explorada.

Apesar das dificuldades encontradas, os objetivos propostos nesta pesquisa foram alcançados, dado que foi aplicado o SVM, desenvolvida a arquitetura híbrida entre redes neurais artificias e SVM e foi possível avaliar a aplicação, tanto do SVM isoladamente, quanto do modelo híbrido na problemática da classificação dos solos.

Com os resultados obtidos foi possível avaliar que o melhor modelo híbrido encontrado foi utilizando a RNA com dois grupos, o qual manteve a mesma acurácia que o SVM isolado, que foi de 77,5%, enquanto as demais variações na quantidade de grupos obtiveram uma precisão inferior ao SVM. Em relação ao tempo de treinamento não houve melhora devido ao fato da RNA do tipo *Kohonen* apresentar um tempo de treinamento alto. Em seu tempo de execução houve uma pequena melhora no modelo de dois grupos que ficou em 10,6ms ± 5,5ms enquanto o SVM

ficou em 19,5ms ± 6,2ms, porém esta melhora não foi confirmada, visto que não existe significância estatística na diferença apresentada. Para os demais hibridismos realizados houve uma melhora no tempo de execução, no entanto custou a perda de precisão do modelo.

Desta forma para esta base de dados da classificação dos solos no estado de Santa Catarina a utilização de uma arquitetura híbrida não se mostrou eficiente, sendo que a melhora obtida no tempo de execução não foi estatisticamente significativa. Em relação a precisão não houve uma melhora mantendo-se assim a precisão 77,5%, o que não confirmou para esta base de dados os resultados apresentados nas pesquisas de Li e Huang (2009, tradução nossa), Lee, Roh e Choi (2009, tradução nossa) e Liu, Xie e Wu (2007, tradução nossa) em que a precisão foi otimizada quando empregada a arquitetura híbrida.

Dando continuidade ao desenvolvimento desta pesquisa, sugerem-se algumas possibilidades de trabalhos futuros:

a) aplicar a arquitetura híbrida de redes neurais artificiais e máquinas de vetores de suporte, desenvolvidas nesta pesquisa, para outros domínios de aplicação;
b) utilizar outras técnicas de hibridismo na problemática da classificação dos solos;
c) aplicar a arquitetura híbrida utilizando técnicas diferentes para outros domínios de aplicação;
d) aplicar no mesmo domínio de conhecimento uma arquitetura híbrida utilizando a biblioteca do SVM na linguagem R, substituindo assim a *LibSVM* e avaliando se os resultados são superiores;
e) avaliar estatisticamente as características dos atributos da base de dados para verificar se ocorre melhora nos resultados da classificação;
f) avaliar técnicas de pré-processamento para superar a limitação de poucos registros na base de dados.

REFERÊNCIAS

ANBAR, Mohammed et al. A Machine Learning Approach to Detect Router Advertisement Flooding Attacks in Next-Generation IPv6 Networks. **Cognitive Computation**, [s.l.], v. 10, n. 2, p.201-214, 23 out. 2017. Springer Nature.

BALLINGS, Michel et al. Evaluating multiple classifiers for stock price direction prediction. **Expert Systems With Applications**, v. 42, n. 20, p.7046-7056, nov. 2015.

BI, Xia-an et al. Classification of Autism Spectrum Disorder Using Random Support Vector Machine Cluster. **Frontiers In Genetics**, [s.l.], v. 9, p.0-0, 6 fev. 2018. Frontiers Media SA.

BISOGNIN, Gustavo. **Utilização de Máquinas de Suporte Vetorial Para Predição de Estruturas Terciárias de Proteínas.** 2007. 102 f. Dissertação (Mestrado) - Curso de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2007.

BRAGA, Antonio De Pádua; CARVALHO, André Ponce De Leon F. De; LUDERMIR, Teresa Bernarda. **REDES NEURAIS ARTIFICIAIS - TEORIA E APLICAÇÕES**. Ltc, 2000.

BRUNGARD, Colby W. et al. Machine learning for predicting soil classes in three semi-arid landscapes. **Geoderma**, [s.l.], v. 239-240, p.68-83, fev. 2015. Elsevier BV.

BUOL, Stanley W. et al. **Soil Genesis and Classification.** 6. ed. [s.i.]: John Wiley & Sons, Inc, 2011. 543 p.

CHAWLA, N. V. et al. SMOTE: Synthetic Minority Over-sampling Technique. **Journal Of Artificial Intelligence Research**, [s.l.], v. 16, p.321-357, 1 jun. 2002. AI Access Foundation.

COPPIN, Ben. **Inteligência artificial.** Rio de Janeiro: LTC, 2010. xxv, 636p.

CRISOSTOMO, Edson; LOPES, Rieuse. FUNÇÕES E SUAS DERIVADAS: UMA PESQUISA REALIZADA NA PERSPECTIVA DO PENSAMENTO MATEMÁTICO AVANÇADO. **REVISTA EDUCAÇÃO MATEMÁTICA EM FOCO**, v. 6, n. 2, 2018.

CRISTIANINI, Nello; SHAWE-TAYLOR, John. **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods**. Cambridge: Cambridge University Press, 2000. 204 p.

DEBAENE, Guillaume; PIKULA, Dorota; NIEDZWIECKI, Jacek. Use of vis-nirs for land management classification with a support vector machine and prediction of soil organic carbon and other soil properties. **Ciencia e Investigación Agraria**, v. 41, n. 1, p.5-6, abr. 2014.

DESAI, Srinivas et al. Spectral Mapping Using Artificial Neural Networks for Voice Conversion. **Ieee Transactions On Audio, Speech, And Language Processing**, [s.l.], v. 18, n. 5, p.954-964, jul. 2010. Institute of Electrical and Electronics Engineers (IEEE).

FABRIS, Fabio; MAGALHÃES, João Pedro de; FREITAS, Alex A.. A review of supervised machine learning applied to ageing research. **Biogerontology**, [s.l.], v. 18, n. 2, p.171-188, 6 mar. 2017. Springer Nature.

FACELI, Katti (Et al.). **Inteligência artificial:** uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2011. xxvi, 378 p.

FARQUAD, M.a.h.; RAVI, Vadlamani; RAJU, S. Bapi. Churn prediction using comprehensible support vector machine: An analytical CRM application. **Applied Soft Computing**, [s.l.], v. 19, p.31-40, jun. 2014. Elsevier BV.

FLACH, Peter A. **Machine learning:** the art and science of algorithms that make sense of data. Cambridge: Cambridge University Press, 2012. xvii, 396 p.

FRANCO, Neide Bertoldi. **Cálculo numérico**. Pearson, 2006.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel Lopes. **Data mining:** uma guia prático : conceitos, técnicas, ferramentas, orientações e aplicações. Rio de Janeiro: Elsevier, 2005. 261 p.

HAN, H. et al. Study on a hybrid SVM model for chiller FDD applications. **Applied Thermal Engineering**, [s.l.], v. 31, n. 4, p.582-592, mar. 2011. Elsevier BV.

HAYKIN, Simon. **Redes neurais:** princípios e prática. 2.ed. Porto Alegre: Bookman, 2001. 900 ISBN 857307182.

HEUNG, Brandon et al. An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping. **Geoderma**, [s.l.], v. 265, p.62-77, mar. 2016. Elsevier BV.

HOMAYOUN, Mohammad et al. Determining relationships between soil properties and plant distribution in a protected area in central Iran. **Maejo International Journal Of Science And Technology**, [s. L.], v. 9, n. 2, p.136-146, abr. 2015.

JORDAN, M. I.; MITCHELL, T. M.. Machine learning: Trends, perspectives, and prospects. **Science**, [s.l.], v. 349, n. 6245, p.255-260, 16 jul. 2015. American Association for the Advancement of Science (AAAS).

KABIR, Enamul et al. A novel statistical technique for intrusion detection systems. **Future Generation Computer Systems**, [s.l.], v. 79, p.303-318, fev. 2018. Elsevier BV.

KANTARDZIC, Mehmed. **Data Mining**: Concepts, Models, Methods, and Algorithms. 2. ed. New Jersey: John Wiley & Sons, 2011.

KOVAČEVIĆ, Miloš; BAJAT, Branislav; GAJIĆ, Boško. Soil type classification and estimation of soil properties using support vector machines. **Geoderma**, v. 154, n. 3-4, p.340-347, jan. 2010.

LAROSE, Daniel T.; LAROSE, Chantal D. **Discovering knowledge in data**: an introduction to data mining. New Jersey: John Wiley & Sons, 2014.

LEE, Sang-myeong; ROH, Tae-seong; CHOI, Dong-whan. Defect diagnostics of SUAV gas turbine engine using hybrid SVM-artificial neural network method. **Journal of Mechanical Science and Technology**, [s.l.], v. 23, n. 2, p.559-568, fev. 2009. Springer Nature

LI, Te-sheng; HUANG, Cheng-lung. Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing**. Expert Systems with Applications**, [s.l.], v. 36, n. 1, p.374-385, jan. 2009. Elsevier BV.

LI, X. Z.; KONG, J. M.. Application of GA–SVM method with parameter optimization for landslide development prediction. **Natural Hazards And Earth System Science**, [s.l.], v. 14, n. 3, p.525-533, 4 mar. 2014. Copernicus GmbH.

LIU, Hongbin; XIE, Deti; WU, Wei. Soil water content forecasting by ANN and SVM hybrid architecture. **Environmental Monitoring And Assessment**, [s.l.], v. 143, n. 1-3, p.187-193, 16 set. 2007. Springer Nature.

LLANOS, Jacqueline et al. Load estimation for microgrid planning based on a self-organizing map methodology. **Applied Soft Computing**, [s.l.], v. 53, p.323-335, abr. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.asoc.2016.12.054.

LORENA, Ana Carolina; DE CARVALHO, André CPLF. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43-67, 2007.

LOURIDAS, Panos; EBERT, Christof. Machine Learning. **Ieee Software,** [s.l.], v. 33, n. 5, p.110-115, set. 2016. Institute of Electrical and Electronics Engineers (IEEE).

LUGER, George F. **Inteligência artificial.** 6. ed. São Paulo: Pearson Education do Brasil, 2013. xvii, 614 p. ISBN 9788581435503 (broch.).

MITCHELL, Tom M.. **Machine Learning.** United Kingdom: Mcgraw-hill Science/engineering/math, 1997. 432 p.

OMIDVAR, Mohammadnabi; DEZFOULI, Mashaallah Abbasi; RAHMANI, Amirmasoud. A face recognition method using artificial neural networks. **Second International Conference On Digital Image Processing**, [s.l.], p.1-5, 26 fev. 2010. SPIE.

OWUSU, Ebenezer; ZHAN, Yonzhao; MAO, Qi Rong. An SVM-AdaBoost facial expression recognition system. **Applied Intelligence**, [s.l.], v. 40, n. 3, p.536-545, 29 set. 2013. Springer Nature.

PANDEY, Abhishek; PRASAD, R.; JHA, Sunil Kr.. Classification of two different rough soil surfaces by using microwave X-band data through support vector machine (SVM). **Russian Agricultural Sciences**, v. 36, n. 2, p.141-145, abr. 2010.

RICH, Elaine; KNIGHT, Kevin; NAIR, Shivashankar B. **Artificial intelligence.** 3. ed. New Delhi: Tata McGraw-Hill, 2009. xviii, 586 p.

RODRÍGUEZ, Fermín et al. Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control. **Renewable Energy**, [s.l.], v. 126, p.855-864, out. 2018. Elsevier BV.

ROUHI, Rahimeh; JAFARI, Mehdi. Classification of benign and malignant breast tumors based on hybrid level set segmentation. **Expert Systems With Applications**, [s.l.], v. 46, p.45-59, mar. 2016. Elsevier BV.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial.** Rio de Janeiro: Elsevier, 2013. xxi, 988 p.

SACHDEVA, Jainy et al. A package-SFERCB-“Segmentation, feature extraction, reduction and classification analysis by both SVM and ANN for brain tumors”. **Applied Soft Computing**, [s.l.], v. 47, p.151-167, out. 2016. Elsevier BV.

SANTOS, H.G. dos; JACOMINE, P.K.T.; ANJOS, L.H.C. dos; OLIVEIRA, V.A. de; OLIVEIRA, J.B. de; COELHO, M.R.; LUMBRERAS, J.F.; CUNHA, T.J.F. (Ed.). Sistema brasileiro de classificação de solos 2.ed. Rio de Janeiro: Embrapa Solos, 2006. 306p.

SEO, Dong-hyuck; ROH, Tae-seong; CHOI, Dong-whan. Defect diagnostics of gas turbine engine using hybrid SVM-ANN with module system in off-design condition. **Journal Of Mechanical Science And Technology**, [s.l.], v. 23, n. 3, p.677-685, mar. 2009. Springer Nature.

SERGEEV, Alexander V. et al. Cardiovascular Disease Treatment Outcomes in Patients with Diabetes: Prediction Models Using Artificial Neural Networks and Logistic Regression. **Annals Of Epidemiology**, [s.l.], v. 25, n. 9, p.705-705, set. 2015. Elsevier BV.

SETTLES, Burr. Active Learning. **Synthesis Lectures On Artificial Intelligence And Machine Learning**, [s.l.], v. 6, n. 1, p.1-114, 30 jun. 2012. Morgan & Claypool

SILVA, Augusto Felix Tavares et al. Classificação de sinais de voz através da aplicação da transformada Wavelet Packet e redes neurais artificiais. **Revista Principia - Divulgação Científica e Tecnológica do Ifpb**, [s.l.], v. 1, n. 37, p.34-41, 21 dez. 2017. Instituto Federal de Educacao, Ciencia e Tecnologia da Paraiba.

SILVA, Thays Aparecida de Abreu. **Previsão de Cargas Elétricas através de um Modelo Hibrído de Regressão com Redes Neurais.** 2012. 64 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Unesp, Ilha Solteira, 2012.

SOARES, Fátima Cibele et al. Redes neurais artificiais na estimativa da retenção de água do solo. **Ciência Rural**, Santa Maria, v. 44, n. 2, p.293-300, fev. 2014. Disponível em: <https://www.redalyc.org/html/331/33129833016/>. Acesso em: 06 dez. 2018.

SUTTON, Richard S.; BARTO, Andrew G.. **Reinforcement Learning:** An Introduction. 2. ed. United States Of America: A Bradford Book, 1998. 322 p.

THEODORIDIS, Sergios; KOUTROUMBAS, Konstantinos. **Pattern Recognition**. 4. ed. Londres: Academic Press, 2009. 967 p.

TONG, Simon; KOLLER, Daphne. Support vector machine active learning with applications to text classification. **The Journal Of Machine Learning Research**, [s.l.], v. 2, n. 1, p.45-66, mar. 2002.

TSAI, Wen-ping et al. A data-mining framework for exploring the multi-relation between fish species and water quality through self-organizing map. **Science Of The Total Environment**, [s.l.], v. 579, p.474-483, fev. 2017. Elsevier BV.

VAPNIK, Vladimir N.. **Statistical Learning Theory**. United States Of America: John Wiley & Sons, Inc, 1998. 740 p.

WANG, Haifeng et al. A support vector machine-based ensemble algorithm for breast cancer diagnosis. **European Journal Of Operational Research**, [s.l.], v. 267, n. 2, p.687-699, jun. 2018. Elsevier BV.

WANG, Wei et al. Attribute Normalization in Network Intrusion Detection. **2009 10th International Symposium On Pervasive Systems, Algorithms, And Networks**, [s.l.], p.448-453, 2009. IEEE. http://dx.doi.org/10.1109/i-span.2009.49.

WESTON, Jason et al. Feature selection for SVMs. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 13. 2000, Canadá. Atas... p. 668-674.

WU, Ji; ZHANG, Xiao-lei. An efficient voice activity detection algorithm by combining statistical model and energy detection. **Eurasip Journal On Advances In Signal Processing**, [s.l.], v. 2011, n. 1, p.0-0, 12 jul. 2011. Springer Nature.

YONG, D. de; BHOWMIK, S.; MAGNAGO, F.. An effective Power Quality classifier using Wavelet Transform and Support Vector Machines. **Expert Systems With Applications**, v. 42, n. 15-16, p.6075-6081, set. 2015.

ZHU, Xiaojin; GOLDBERG, Andrew B.. Introduction to Semi-Supervised Learning. **Synthesis Lectures On Artificial Intelligence And Machine Learning**, [s.l.], v. 3, n. 1, p.1-130, jan. 2009. Morgan & Claypool Publishers LLC.

APÊNDICE(S)

APÊNDICE A - Tabela de atributos selecionados para classificação dos solos

Atributos do solo Continua...

Atributo	Valores
Saturação por bases ou por alumínio	Álico, Distrófico, Eutrófico
Atividade da argila	Ta, Tb, Muito duro
Horizonte diagnóstico superficial	A Chernozêmico, A Húmico, A Moderado, A Proeminente, Firme e friável
Grupamento de classe textural superficial	Arenosa, Argilosa, Argilosa cascalhenta, Argilosa com cascalhos, Média, Média cascalhenta, Média com cascalho, Muito argilosa, Muito pegajosa
Grupamento de classe textural subsuperficial	Argilosa, Argilosa com cascalhos, Muito argilosa
Classe de drenagem	Acentuadamente drenado, Bem a acentuadamente drenado, Bem drenado, Excessivamente drenado, Imperfeitamente a mal drenado, Imperfeitamente drenado, Mal drenado, Mal drenado a muito mal drenado, Moderadamente a bem drenado, Moderadamente drenado, Muito mal drenado
Símbolo Horizonte	2BC, 2Bt1, 2Bt2, A, A/C, A1, A2, AB, AB1, AB2, AC, Ap, Ap1, Ap2, B/C, BA, BA1, BA2, BC, BC1, BC2, Bg, Bg1, Bg2, Bh, Bi, Bi1, Bi2, Bi3, Big, Bt, Bt1, Bt2, Bt3, Bt4, Bw, Bw1, Bw1, Bw2, Bw3, C, C1, C2, C3, Cg, E, E/B, E1, E2, HD, Hdo, Hod, Hod1, Hod2, IIC1, IIIC2, R
Profundidade Superior	Numérico
Profundidade Inferior	Numérico
Classe de Textura	Muito argilosa (argila pesada),Argila, Franco-argilosa, Franca, Franco-argilo-arenosa, Franco-arenosa, Franco-argilo-siltosa, Argila cascalhenta, Franco-argilosa cascalhenta, Areia, Argilo-siltosa, Muito argilosa cascalhenta, Pouca, Areia-franca, Franco-arenosa cascalhenta
Cor da Amostra Úmida - Valor	Numérico
Cor da Amostra Úmida - Croma	Numérico
Grau de Desenvolvimento - 1ª Ocorrência	Forte, Fraca, Fraca a moderada, Moderada, Moderada a forte
Grau de Desenvolvimento - 2ª Ocorrência	Forte, Fraca, Fraca a moderada, Moderada, Moderada a forte
Grau de Desenvolvimento - 3ª Ocorrência	Forte, Moderada
Tamanho - 1ª Ocorrência	Grande, Média, Média a grande, Muito pequeno, Muito pequena a pequena, Pequena, Pequena a média, Ultrapequena

Atributos do solo (Continua...)

Atributos	Valores
Tamanho - 2ª Ocorrência	Grande, Média, Média a grande, Muito pequena, Muito pequena a pequena, Pequena, Pequena a média, Ultrapequena
Tamanho - 3ª Ocorrência	Muito pequena, Pequena a média
Forma - 1ª Ocorrência	Granular, Prismática, Blocos angulares, Blocos subangulares, Prismática que se desfaz, Maciça que se desfaz,Grãos simples
Forma - 2ª Ocorrência	Maciça porosa, Granular, Blocos angulares, Blocos subangulares, Grãos simples, Blocos subangulares que se desfaz, Prismática, Maciça que se desfaz
Forma - 3ª Ocorrência	Blocos angulares, Blocos subangulares, Granular, Maciça porosa
Grau de Consistência - Seca	Dura, Dura a muito dura, Ligeiramento dura, Ligeiramento dura a dura, Macia, Macia a ligeiramente dura, Muito dura, Solta
Grau de Consistência - Úmida	Firme, Firme a friável, Firme a muito firme, Friável, Friável a muito friável, Muito friável, Solta
Molhada - Plasticidade	Ligeiramente plástica, Ligeiramente plástica a plástica, Muito plástica, Não-plástica, Plástica, Plástica a muito plástica
Molhada - Pegajosidade	Ligeiramente pegajosa, Ligeiramente pegajoso a pegajoso, Muito pegajosa, Não pegajosa, Pegajosa, Pegajosa a muito pegajosa
Cerosidade - Grau de desenvolvimento	Forte, Fraca, Moderada
Cerosidade - Quantidade	Abundante, Comum, Pouca
Transição de horizonte subjacente - Topografia	Ondulada, Plana
Transição de horizonte subjacente - Nitidez	Abrupta, Clara, Difusa, Gradual
Mosqueado - Valor	Numérico
Mosqueado - Croma	Numérico
Frações da Amostra Total - Calhaus (g/Kg)	Numérico
Frações da Amostra Total - Cascalho (g/Kg)	Numérico
Frações da Amostra Total - Terra Fina (g/Kg)	Numérico
Composição Granulométrica da terra fina - Areia Grossa (g/Kg)	Numérico
Composição Granulométrica da terra fina - Areia Fina (g/Kg)	Numérico
Composição Granulométrica da terra fina - Areia Total (g/Kg)	Numérico
Composição Granulométrica da terra fina - Silte (g/Kg)	Numérico
Composição Granulométrica da terra fina - Argila (g/Kg)	Numérico
Composição Granulométrica da terra fina - Relação Silte Argila (g/Kg)	Numérico

Atributos do solo Conclusão

Atributos	Valores
Composição Granulométrica da terra fina - Argila Dispersa em Água (g/Kg)	Numérico
Densidade - Partículas (real)	Numérico
Densidade - Solo (aparente)	Numérico
Grau de Floculação (%)	Numérico
pH - H2O	Numérico
pH - KCl	Numérico
Complexo Sortivo - Cálcio (cmolc/kg)	Numérico
Complexo Sortivo - Magnésio	Numérico
Complexo Sortivo - Ca + Mg	Numérico
Complexo Sortivo - Potássio	Numérico
Complexo Sortivo - Sódio	Numérico
Complexo Sortivo - Valor S (Ca2++Mg2++K++Na+)	Numérico
Complexo Sortivo - Hidrogênio (H+)	Numérico
Complexo Sortivo - Alumínio Trocável (Al3+)	Numérico
Complexo Sortivo - Valor T (S+H++Al3+)	Numérico
Complexo Sortivo - Valor V Δ Saturação por Bases (100. S/T)%	Numérico
Complexo Sortivo - Saturação por Alumínio (100. Al3+/S+Al3+)%	Numérico
Ataque sulfúrico - SiO2	Numérico
Ataque sulfúrico - Al2O3	Numérico
Ataque sulfúrico - Fe2O3	Numérico
Ataque sulfúrico - TiO2	Numérico
Ataque sulfúrico - P2O5	Numérico
Ataque sulfúrico - SiO2 / Al2O3 (Ki)	Numérico
Ataque sulfúrico - SiO2 / R2O3 (Kr)	Numérico
Ataque sulfúrico - Al2O3 / Fe2O3	Numérico
Fósforo assimilável (mg/kg)	Numérico
Carbono orgânico	Numérico
Nitrogênio total	Numérico
Relação C/N (%)	Numérico
Saturação por Sódio (100.Na+/T)	Numérico
Condutividade Elétrica	Numérico

Fonte: Do autor.

APÊNDICE B - Artigo

Arquitetura híbrida de máquinas de vetores de suporte e redes neurais artificiais aplicada a classificação dos solos

Rafael de Bona[1], Merisandra Cortês de Matos Garcia[2]

[1] Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma– SC

[2] Professora do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma– SC

rafael.bferro@hotmail.com, mem@unesc.net

Abstract. *The advances made by the technology has been allowing the application of complex artificial intelligence techniques, e.g. support vector machines. However, this technique has the disadvantage of a high amount of time spent in training and execution phases. In this paper, it is proposed the development of a hybrid architecture between support vector machines and artificial neural network. The hybrid approach has the objective to overcome those disadvantages and this approach has been a focus in various researches.*

Resumo. *O avanço da tecnologia permite a utilização de técnicas mais complexas de inteligência artificial, como as máquinas de vetores de suporte. Esta técnica possui a desvantagem de possuir um alto tempo de treinamento e execução. Neste artigo propõe-se o desenvolvimento de uma arquitetura híbrida com redes neurais artificiais para superar essa limitação. As arquiteturas híbridas visam otimizar as técnicas envolvidas e atualmente são alvos de diversas pesquisas.*

1. Introdução

O aprendizado de máquina atualmente é uma das áreas que mais cresce. O emprego do aprendizado de máquina pode ser visto em diversas áreas, como por exemplo, ciência, tecnologia e comércio. Dentre as técnicas de aprendizado de máquina destacam-se, as máquinas de vetores de suporte e as redes neurais artificiais.

A máquina de vetor de suporte é uma técnica muito utilizada para a classificação, a mesma tem como objetivo traçar retas que separem dois conjuntos com a maior distância possível (HAYKIN, 2001) e vem sendo aplicadas com sucesso em várias áreas, como por exemplo, Engenharia Ambiental e na área da Geologia.

Na área da geologia tem-se a classificação dos solos que é um processo laboratorial trabalhoso e de alto custo, que por vezes devido à falta de amostras suficientes é necessária a utilização de técnicas de predição. A predição das propriedades do solo

quando realizada por meio da inteligência computacional possui resultados melhores (SOARES et al., 2014).

As máquinas de vetores de suporte apesar de serem eficientes nas resoluções de problemas de classificação, possui uma desvantagem que é o alto tempo despendido nas fases de aprendizagem e testes. Com o intuito de otimizar a aplicação desta técnica algumas abordagens híbridas com outras técnicas de inteligência computacional podem ser aplicadas.

Na inteligência computacional, uma arquitetura hibrida é a mescla de técnicas de aprendizado de máquina. Desta forma, a arquitetura híbrida pode gerar uma solução mais eficiente e robusta.

Nesta pesquisa foi desenvolvida a arquitetura híbrida empregando-se o modelo de rede neural artificial *Kohonen* para o agrupamento dos dados e aplicando-se uma máquina de vetores de suporte para cada um dos grupos gerados a fim de se classificar os solos. Posteriormente, por meio de métodos estatísticos avaliou-se o desempenho da aplicação das máquinas de vetores de suporte e da arquitetura híbrida, considerando-se os parâmetros de taxa de erro, tempo de treinamento, tempo de execução, acurácia, entre outras medidas de avaliação de um classificador.

2. Materiais e métodos

Esta pesquisa metodologicamente classifica-se como aplicada e de base tecnológica, tendo-se desenvolvido as seguintes etapas principais: levantamento bibliográfico, obtenção e preparação da base de dados, aplicação da técnica de máquinas de vetores de suporte, desenvolvimento e aplicação da arquitetura híbrida entre RNA e SVM, avaliação dos modelos por meio de métodos estatísticos e análise dos resultados.

Os materiais empregados no desenvolvimento desta pesquisa em termos de recursos de dados foram os disponíveis no site da Embrapa sobre a classificação do solo no estado de Santa Catarina; em termos de software empregou-se o ambiente de desenvolvimento *NetBeans* na versão 8.2, biblioteca em Java do *LibSVM* na versão 3.23 e a biblioteca *Weka* API, sendo que nesta última está a implementação da rede neural artificial do tipo *Kohonen*.

2.1. Base de dados

A base de dados utilizada neste trabalho foi obtida através do site da Empresa Brasileira de Pesquisa em Agropecuária (EMBRAPA), nesta base constam os dados do solo de pesquisas realizadas no país, nesta pesquisa a base de dados é referente a apenas dados do estado de Santa Catarina.

Esta base de dados contém um total de 450 registros que englobam oito das treze classes de solo do Sistema Brasileiro de Classificação dos Solos, sendo elas, argissolo, cambissolo, gleissolo, latossolo, neossolo, nitossolo, organossolo e planossolo.

Como a base de dados é do estado de Santa Catarina alguns solos são menos predominantes no estado, como por exemplo, as classes gleissolo, neossolo, organossolo e planossolo (tabela 1).

1. Tabela 1. Quantidade de registros por classe de solo

Classe do solo	Quantidade
Argissolo	74
Cambissolo	95
Gleissolo	9
Latossolo	102
Neossolo	26
Nitossolo	135
Organossolo	5
Planossolo	4
Total	450

2.2. Pré-processamento da base de dados

No pré-processamento foram realizadas a transformação dos atributos para o tipo numérico, normalização dos atributos e balanceamento de classes

A base de dados possui uma grande quantidade de atributos categóricos, para que as técnicas de aprendizado de máquina pudessem ser aplicadas foi realizada a transformação de dados categóricos para numéricos

Após a transformação dos atributos para o tipo numérico, foi também realizada a normalização, a mesma consiste em colocar os atributos dentro de uma escala, para que cada atributo não domine os outros, ou seja, para que não seja considerado apenas os atributos de maior valor como relevante

Ainda foi avaliado o desbalanceamento de classes devido ao fato da base de dados ser do estado de Santa Catarina em que predominam algumas classes do solo. Dessa forma, algumas classes possuem menor frequência em relação as demais, o que pode ocasionar classificações incorretas para as classes de menor frequência. Algumas técnicas podem ser usadas como tentativa de minimizar este problema.

Uma dessas técnicas é a SMOTE, nesta o algoritmo por meio de cálculos matemáticos cria registros sintéticos a partir dos dados originais. Outra técnica que alguns classificadores possuem é a utilização de pesos no seu treinamento, assim pode-se informar pesos diferentes para cada classe, neste caso os pesos serão considerados para a penalização por classificação incorreta.

Nesta pesquisa como a técnica do SVM permite utilizar pesos foram geradas duas bases de dados para a aplicação, uma em que foi aplicada a técnica SMOTE e uma sem a aplicação de nenhuma técnica de balanceamento.

2.3. Aplicação das máquinas de vetores de suporte

Antes da aplicação das máquinas de vetores de suporte separou-se a base de dados em subconjuntos de treinamento e teste, para que fosse possível validar e avaliar a aplicação por meio de métodos estatísticos. Para a separação foi utilizada a técnica de *holdout*, no qual dois terços dos dados foram utilizados no treinamento e um terço para teste. A distribuição de registros nos subconjuntos foi de 303 para treinamento e 147 para teste.

Para a implementação do SVM foi utilizada a biblioteca do Java *LibSVM*, a mesma possui as principais funções de *kernel* para serem utilizadas, as quais são linear, RBF, polinomial e sigmoidal. Cada função de *kernel* utiliza diferentes quantidades de parâmetros, a RBF utiliza o parâmetro *gamma* (γ), a sigmoidal os parâmetros γ e coef0 (r) e por último o polinomial, que utiliza a maior quantidade de parâmetros, γ, *r e degree* (d) (tabela 2).

Tabela 2. Quantidade de registros por classe de solo

Tipo de *kernel*	Função	Parâmetros
RBF	$\exp(-\gamma \lVert x_i - x_j \rVert^2)$	γ
Sigmoidal	$\tanh(\gamma(x_i . x_j) + r)$	γ e r
Polinomial	$(\gamma(x_i \, . \, x_j) + r)^d$	γ, r e d

Alguns parâmetros são gerais para o SVM que é o caso do parâmetro de custo (C), este parâmetro é utilizado para indicar a penalização por uma classificação incorreta

Para encontrar os melhores valores para estes parâmetros foi empregada a técnica *Gridsearch,* que consiste em utilizar vetores para cada parâmetro, combinar os valores dos vetores e testar o SVM com a base de dados. Nesta técnica são selecionados os melhores parâmetros da rodada e alteram-se os valores dos vetores para valores mais próximos aos encontrados na rodada anterior e o processo se repete até que se obtenha os parâmetros ótimos.

Na primeira execução do SVM foi utilizada a base de dados sem o emprego de técnicas de balanceamento e com os parâmetros padrão da biblioteca, assim como foi utilizada a função de *kernel* RBF. Foi então obtida uma quantidade de erros de 59 do total de registros de 147, a precisão do modelo ficou em aproximadamente 59,8%. O tempo de treinamento e execução foi em média de 71,4 e 19,5 milissegundos respectivamente.

A segunda rodada de execução utilizou a técnica *Gridsearch*, a qual encontrou os melhores parâmetros para C e para γ. A quantidade de erro do modelo foi de 33 de um total de 147, sua precisão foi de aproximadamente 77,5% (tabela 3).

Tabela 3. Quantidade de erros na classificação por classe do solo

Classe do solo	Total no treinamento	Erros
Argissolo	24	8
Cambissolo	31	3
Gleissolo	3	1
Latossolo	34	9
Neossolo	8	3
Nitossolo	45	8
Organossolo	1	1
Planossolo	1	0
Total	147	33

Para este conjunto de dados os testes com diferentes funções de *kernel* não se mostraram eficientes. Ainda foram realizadas rodadas com o conjunto que foi aplicada a técnica SMOTE e também foram utilizados os pesos no treinamento do SVM.

Utilizando o SMOTE a quantidade de erros e a precisão se mantiveram as mesmas do que quando não utilizado. Com a utilização dos pesos a precisão piorou, com uma quantidade de erros de 42 de um total de 147, a precisão ficou em aproximadamente 71,4%. Além de não melhorar a precisão ela também não otimizou a quantidade de acertos das classes que estão desbalanceadas.

2.3. Aplicação da arquitetura híbrida

Nesta pesquisa optou-se pela realização do hibridismo com a utilização de uma RNA do tipo *Kohonen* para agrupar os dados do solo e após foi realizada a aplicação de um SVM para a tarefa de classificação em cada grupo de dados. A escolha deste tipo de hibridismo se deu após estudo dos trabalhos correlatos onde Liu, Xie e Wu (2007, tradução nossa) aplicaram um hibridismo similar, obtendo resultados melhores quando utilizada a arquitetura híbrida.

Para a aplicação da RNA do tipo *Kohonen* foi utilizada *Weka* API, que é uma ferramenta de mineração de dados, a escolha da biblioteca *Weka* API se deu pelo fato da sua compatibilidade com o *LibSVM* que já estava sendo utilizado nesta pesquisa e também por possuir a possibilidade de adicionar a implementação da RNA do tipo *Kohonen* .

Conforme o funcionamento da RNA os parâmetros de largura (*w*) e altura (*h*) definem a quantidade de grupos que são gerados, foram utilizadas variações nos parâmetros para gerar diferentes quantidades de grupos. Para cada grupo gerado foram seguidas as etapas conforme a aplicação do SVM, que foram utilizar inicialmente no SVM a função de *kernel* RBF com os parâmetros padrão, em seguida foram realizadas as otimizações com a técnica de *Gridsearch* e teste com diferentes funções de *kernel*.

Foram realizadas rodadas variando os parâmetros da RNA do tipo *Kohonen* para que fossem geradas as quantidades de grupos de dois, quatro, cinco, oito e quinze.

Em cada rodada foram realizadas as otimizações de cada SVM para cada grupo, em algumas rodadas as variações da função de *kernel* mostrou-se eficiente onde alguns grupos obtiveram melhores performances com as funções de *kernel* polinomial e sigmoidal.

Em relação as quantidades de erros de cada rodada foi de 33 na primeira, mantendo assim a mesma performance em termos de precisão da aplicação do SVM isoladamente. A segunda rodada a quantidade de erros teve um aumento para 37, a sua precisão ficou em aproximadamente 74,8%. Na rodada seguinte houve um aumento ainda maior na quantidade de erros que foi para 43, já na quarta rodada a quantidade de erros foi menor, porém manteve-se maior que quando aplicado somente o SVM, que foi de 36. Finalmente na última rodada a quantidade de erros teve um pequeno aumento em comparação a rodada anterior e foi para 37 e sua precisão foi de aproximadamente 74,8%.

Em relação aos tempos, na primeira rodada o tempo de execução foi em média de 10,6ms, nas rodadas seguintes foram de 8,1ms, 5,4ms, 4,5ms e 3,9ms para a segunda, terceira, quarta e quinta rodada respectivamente.

A melhor arquitetura em relação a precisão foi a utilizada na primeira rodada que foram utilizados dois grupos na RNA do tipo *Kohonen*. Para o SVM do grupo um foi utilizado o kernel *polinomial* e no SVM do grupo dois foi utilizado o *kernel* RBF.

Para facilitar a visualização dos resultados tambfoi desenvolvida uma interface gráfica.

3. Resultados e discussão

esta pesquisa foram utilizados métodos estatísticos para avaliação tanto da aplicação do SVM isolado quanto para a aplicação da arquitetura híbrida. Como o hibridismo escolhido consistiu na implementação de um SVM para cada grupo gerado pela RNA foi necessária a utilização de uma média ponderada para que os valores pudessem ser comparados entre si, já que as estatísticas são geradas para cada SVM e não para o modelo híbrido em geral.

Primeiramente foram avaliados os dados estatísticos dos classificadores gerados, para isso empregaram-se as medidas de taxa de verdadeiros positivos, taxa de falsos positivos, taxa de verdadeiros negativos, taxa de falsos negativos, sensibilidade (*recall*), especificidade e área abaixo da curva ROC. Um classificador é considerado bom quando possui uma maior sensibilidade, especificidade e acurácia, e um menor número de erros, taxa de falsos negativos e taxa de falsos positivos

Para análise deve-se considerar que a taxa de VP, taxa de VN, *Recall*, especificidade e área abaixo da curva ROC quanto mais próximo a 1 seus valores, melhor é o classificador. Para a taxa de FP e taxa de FN é o contrário, ou seja, quanto mais próximos a zero melhor.

A aplicação da arquitetura híbrida com dois grupos apresentou a mesma precisão da aplicação de somente o SVM, isto é, manteve-se em aproximadamente 77,5% (tabela 4). No entanto, na arquitetura ocorreram algumas diferenças nas medidas de qualidade, a taxa de FP foi menor em comparação ao SVM, na qual a arquitetura apresentou 0,0731, enquanto que somente o SVM foi de 0,0822, houve uma diferença de 0,0091 para a arquitetura híbrida com dois grupos, a taxa de VN foi maior no hibridismo (0,9269) do que no SVM (0,9178), o que apresenta uma pequena melhora de 0,0091 para o hibridismo. Esta mesma melhora pode ser observada na especificidade, a área abaixo da curva ROC também foi um pouco melhor na arquitetura com o valor de 0,8512 (tabela 5).

Ao utilizar quatro grupos no hibridismo a precisão teve uma queda em relação as aplicações anteriores ficando em 74,8% (tabela 4). Suas medidas de qualidade também

tiveram uma queda quando comparada ao hibridismo com dois grupos (tabela 5). Suas medidas de qualidade também tiveram uma queda quando comparada ao hibridismo com dois grupos.

No hibridismo com cinco grupos houve uma queda na precisão de 6,8% quando comparado ao SVM ou ao hibridismo com dois grupos, a precisão foi de 70,7% (tabela 4). A mesma queda de desempenho pode ser vista nas medidas de qualidade.

Na arquitetura híbrida com oito grupos a precisão alcançada foi próxima ao SVM, o seu valor ficou aproximadamente em 75,5% uma diferença de 2% para o SVM, este foi o segundo melhor resultado em relação a precisão (tabela 4). No entanto, suas medidas de qualidade indicam o contrário, pois foram inferiores inclusive a aplicação do hibridismo com cinco grupos (tabela 5).

Finalmente a aplicação do hibridismo com quinze grupos, apresentou uma precisão menor quando comparada com a de oito grupos, com aproximadamente 74,8% (tabela 10). As análises das medidas de qualidade do classificador indicam os mesmos casos do hibridismo com oito grupos, porém apresentam pequenas melhoras quando comparado ao mesmo.

Tabela 4. Precisão das aplicações

Aplicação	Quantidade total	Quantidade de erros	Precisão
SVM	147	33	77,5%
Arquitetura híbrida com 2 grupos	147	33	77,5%
Arquitetura híbrida com 4 grupos	147	37	74,8%
Arquitetura híbrida com 5 grupos	147	43	70,7%
Arquitetura híbrida com 8 grupos	147	36	75,5%
Arquitetura híbrida com 15 grupos	147	37	74,8%

Tabela 5. Medidas de qualidade do classificador

Aplicação	Taxa VP	Taxa FP	Taxa VN	Taxa FN	*Recall*	Especificidade	Área ROC
SVM	0,7755	0,0822	0,9178	0,2245	0,7755	0,9178	0,8467
2 grupos	0,7755	0,0731	0,9269	0,2245	0,7755	0,9269	0,8512
4 grupos	0,7483	0,0805	0,9195	0,2517	0,7483	0,9195	0,8339
5 grupos	0,7075	0,0939	0,9061	0,2925	0,7075	0,9061	0,8068
8 grupos	0,7551	0,1666	0,8334	0,2449	0,7551	0,8334	0,7942
15 grupos	0,7483	0,1293	0,8571	0,2517	0,7483	0,8707	0,7959

Dentre as avaliações dos modelos também foram considerados seus tempos de execução, para a avaliação foram utilizados médias e desvio padrão, por meio do teste U de *Mann-Whitney* foi possível identificar se houve a existência de significância estatística, para isto o valor-p deveria ser menor que 0,05.

A aplicação do SVM obteve uma média de tempo de execução de 19,5ms com ±6,2ms de desvio padrão, no hibridismo com dois grupos o tempo foi de 10,6ms ±5,5ms, apresentou uma diferença em relação ao tempo de execução do SVM, conforme foram aumentando as quantidades de grupos esta diferença foi aumentando.

Para analisar se as diferenças entre os tempos de execução foram significativas estatisticamente foi utilizado o teste H de *Kruskal-Wallis* para avaliar a significância e *post hoc* de *Dunn*, que realiza a comparação em pares

Os tempos obtidos com a aplicação do SVM e do hibridismo com dois grupos não apresentaram diferenças, já o SVM quando comparado com as demais aplicações do hibridismo apresentou diferença significativa estatisticamente (tabela 6).

Tabela 6. Média e desvio padrão dos tempos de execução em ms das aplicações com aplicação do teste H de *Kruskal-Wallis* e *post hoc* de *Dunn*

	Média ± Desvio padrão	
	Execução n = 30	Valor-p[†]
SVM[a]	19,5±6,2	< 0,001
Arquitetura híbrida com 2 grupos[a,b]	10,6±5,5	
Arquitetura híbrida com 4 grupos[b]	8,1±5,7	
Arquitetura híbrida com 5 grupos[b]	5,4±3,9	
Arquitetura híbrida com 8 grupos[c]	4,5±3,3	
Arquitetura híbrida com 15 grupos[c]	3,9±3,1	

[†]Valores obtidos por meio da aplicação do teste H de *Kruskal-Wallis*.
[a,b,c]Letras diferentes representam diferença estatisticamente significativa obtida por meio da aplicação do teste *post hoc* de *Dunn*.

Com os resultados obtidos pode-se avaliar que o melhor modelo híbrido para a base de dados utilizada nesta pesquisa foi o de dois grupos que obteve a melhor precisão e as melhores medidas de avaliação, no entanto o mesmo não apresentou uma precisão maior do que a aplicação isolada do SVM e também não apresentou otimizações significativas estatisticamente no tempo de execução

Os demais modelos híbridos apresentaram melhoras significativas nos tempos de execução quando comparada com a aplicação isolada do SVM, porém estes modelos perderam precisão e obtiveram resultados inferiores nas medidas de qualidade empregadas para a avaliação dos modelos.

Os resultados obtidos demonstraram que para esta base de dados o modelo híbrido não melhorou a acurácia, conforme as aplicações de hibridismo vistas nos trabalhos correlatos, em que na pesquisa de Li e Huang (2009, tradução nossa) aplicando o hibridismo entre uma RNA do tipo *Kohonen* e um SVM para cada grupo obteve uma melhora na precisão de aproximadamente 85% para precisões acima de 90%.

Na pesquisa de Lee, Roh e Choi (2009, tradução nossa) na utilização do hibridismo de SVM com RNA, onde a arquitetura difere-se da maneira em que foi implementada neste trabalho, obtiveram melhor precisão e tempos de treinamento e execução quando aplicado o modelo híbrido.

Foi possível observar que o ganho no tempo de execução da arquitetura com uma maior quantidade de grupos na RNA custou uma menor precisão.

4. Considerações finais

O avanço da tecnologia aumentou o poder de processamento tornando possível a aplicação de técnicas de aprendizado de máquina cada vez mais complexas para resolução

de problemas. O aprendizado de máquina na tarefa de classificação vem sendo aplicada com sucesso em diversos estudos e áreas do conhecimento.

Esta pesquisa que foi desenvolvida teve como base o desenvolvimento de uma arquitetura híbrida para a classificação dos solos e avaliação do modelo por meio de métodos estatísticos.

Com os resultados obtidos foi possível avaliar que o melhor modelo híbrido encontrado foi utilizando a RNA com dois grupos, o qual manteve a mesma acurácia que o SVM isolado, que foi de 77,5%, enquanto as demais variações na quantidade de grupos obtiveram uma precisão inferior ao SVM.

Em relação ao tempo de execução houve uma pequena melhora no modelo de dois grupos que ficou em 10,6ms ± 5,5ms enquanto o SVM ficou em 19,5ms ± 6,2ms, porém esta melhora não foi confirmada, visto que não existe significância estatística na diferença apresentada. Para os demais hibridismos realizados houve uma melhora no tempo de execução, no entanto custou a perda de precisão do modelo.

Sendo assim para esta base de dados da classificação dos solos no estado de Santa Catarina a utilização de uma arquitetura híbrida não se mostrou eficiente, sendo que a melhora obtida no tempo de execução não foi estatisticamente significativa. Em relação a precisão não houve uma melhora mantendo-se assim a precisão 77,5%, o que não confirmou para esta base de dados os resultados apresentados nas pesquisas de Li e Huang (2009, tradução nossa), Lee, Roh e Choi (2009, tradução nossa) e Liu, Xie e Wu (2007, tradução nossa) em que a precisão foi otimizada quando empregada a arquitetura híbrida.

Referências

HAYKIN, Simon. **Redes neurais:** princípios e prática. 2.ed. Porto Alegre: Bookman, 2001.

LIU, Hongbin; XIE, Deti; WU, Wei. Soil water content forecasting by ANN and SVM hybrid architecture. **Environmental Monitoring and Assessment**, v. 143, n. 1-3, p.187-193, 16 set. 2007. Springer Nature.

SOARES, Fátima Cibele et al. Redes neurais artificiais na estimativa da retenção de água do solo. **Ciência Rural**, Santa Maria, v. 44, n. 2, p.293-300, fev. 2014. Disponível em: <https://www.redalyc.org/html/331/33129833016/>. Acesso em: 06 dez. 2018.

LEE, Sang-myeong; ROH, Tae-seong; CHOI, Dong-whan. Defect diagnostics of SUAV gas turbine engine using hybrid SVM-artificial neural network method. **Journal of Mechanical Science and Technology**, v. 23, n. 2, p.559-568, fev. 2009. Springer Nature.

LI, Te-sheng; HUANG, Cheng-lung. Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing. **Expert Systems with Applications**, v. 36, n. 1, p.374-385, jan. 2009. Elsevier BV.