# Metodologia

Nessa dissertação, propõe-se uma metodologia para a análise de risco em projetos de software, a partir de dados históricos de registros de riscos, por meio da utilização de redes neurais artificiais. Para desenvolver essa metodologia, primeiramente, necessitamos de uma base de dados extensa e real de riscos em projetos de software. No entanto, há uma dificuldade em encontrar publicamente bases de dados representativas e confiáveis nessa área. Uma base de dados de risco chamada PERIL \cite{kendrick2003identifying} mostrou atender as necessidades básicas para essa pesquisa, além de já estar totalmente classificada (mais detalhes na Seção \ref{sec:perildataset}). Nesse estudo, precisa-se de uma base de dados extensa de registros de riscos oriundos de diversos projetos espalhados pelo Mundo e em diversos períodos de tempo, com o objetivo de mostrar evidências do que ocorre no gerenciamento desses projetos.

Segundo, é necessário realizar o pré-processamento dos dados para que as variáveis de entradas sejam transformadas, normalizadas e selecionadas para o estudo. Terceiro, avaliar o desempenho das ferramentas de estado da arte quanto ao erro de previsão: Simulação de Monte Carlo e Análise PERT.

Quarto, implementar cada modelo de previsão para que pudéssemos selecionar o melhor modelo em redes neurais artificiais: Perceptron de Múltiplas Camadas, Máquinas de Vetor de Suporte, Redes de Função de Base Radial e o Sistema de Inferência Adaptativo Neuro-Fuzzy. Alguns desses modelos selecionados apresentam alguns parâmetros e devido a diversidade de possíveis valores para cada parâmetro, necessita-se otimizar os parâmetros dos modelos. Nesse estudo, implementa-se uma variação da meta-heurística Otimização por Enxame de Partículas (PSO - \textit{Particle Swarm Optimization}) com coeficiente de constrição de Clerk \cite{engelbrecht2007computational} para realizar a tarefa de otimização dos parâmetros das RNAs. O espaço de busca do problema de otimização desses parâmetros é multi-dimensional, complexo e apresenta diversos mínimos locais. Na Figura \ref{fig:method2}, observa-se um esquema ilustrando que um algoritmo de otimização executa os quatro modelos em suas diversas configurações paramétricas para que seja possível eleger o modelo mais coerente com a base de dados adotada.

\begin{figure}[h]

\centering

\includegraphics[width=.45\textwidth]{image/MetodologiaDissertacao2.png}

\caption{Esquema de Seleção da Melhor Rede Neural Artificial}

\label{fig:method2}

\end{figure}

Quinto, um teste de validação dos resultados é realizado para verificar se os modelos baseados em redes neurais são mais eficazes que os modelos de estado da arte em análise de riscos: Simulação de Monte Carlo, Análise PERT, Modelo de Regressão Linear Múltipla, Modelo de Regressão em Árvore. Os dois últimos modelos de regressão linear foram considerados como linha de base, por serem modelos mais simples, já que realizam a regressão linear. Os dois últimos modelos foram implementados e também são avaliados quanto ao erro de previsão.

A seleção do melhor modelo é feita em termos da precisão na previsão. É difícil obter uma métrica representativa da precisão de um modelo. No entanto, Engelbrecht \cite{engelbrecht2007computational} e Saxena \cite{saxena2012software} sugerem que a Raiz do Erro Médio Quadrático(REMQ), do inglês \textit{Root Mean Square Error} é uma medida conveniente e aplicável para a maioria dos problemas. A precisão, nesse caso, significa o grau de proximidade de uma saída calculada para a esperada. A REMQ é representada pela Equação \ref{eq:RMSE}:

\begin{equation}\label{eq:RMSE}

REMQ= \sqrt[2]{\frac{1}{n}\sum\_{i=1}^{n} (e\_i)^2},

\end{equation}

onde $e\_i=f\_i - y\_i$, $f\_i$ é o resultado calculado, $y\_i$ é o resultado esperado e $n$ é o número de pares de dados.

%where $e\_i=f\_i - y\_i$, $f\_i$ is the calculated outcome, $y\_i$ is the expected outcome and $n$ is the number of data pairs.

Todas as técnicas estudadas nesse trabalho estimam a saída para o impacto de riscos. A REMQ é calculada trinta vezes para cada abordagem. Um Teste Estatístico de Wilcoxon Não-pareado \cite{siegel1956nonparametric} pode ser necessário para determinar qual é uma abordagem mais precisa para a base de dados (nesse estudo, o PERIL). O Teste de Wilcoxon Não-pareado é utilizado porque não há evidências que as amostras sejam oriundas de uma população normalmente distribuída, como também não há relação de ordem entre os valores de diferentes amostras.

A validação cruzada \cite{amari1996statistical} é utilizada para evitar a ocorrência de \textit{overfitting} ou \textit{underfitting} durante o treinamento das redes neurais. Nesse caso, o treinamento com parada prematura é utilizado para identificar o início do \textit{overfitting} porque esse método tem provado ser capaz de melhorar a capacidade de generalização da Rede Neural Artificial em comparação com o treinamento exaustivo \cite{haykin-1994} \cite{engelbrecht2007computational} \cite{amari1996new}. Portanto, o método de validação cruzada é utilizado para cada abordagem, excluindo a Simulação de Monte Carlo e Análise PERT, para promover maior capacidade de generalização. Quando adotamos o uso da validação cruzada, significa que necessitamos particionar a base de dados em três partes: conjunto de treinamento, conjunto de validação cruzada e conjunto de testes. O conjunto de treinamento é utilizado na fase de treinamento das redes neurais, momento em que o aprendizado ocorre. O conjunto de validação cruzada é processado ao mesmo tempo, com o conjunto de treinamento e determina a parada no treinamento. Já o conjunto de testes, é utilizado para produzir a métrica de precisão do modelo (REMQ).

Por fim, a previsão do impacto do risco e a definição de um intervalo de confiança para uma amostra do conjunto de treinamento são obtidas utilizando o modelo de previsão mais preciso após os testes de validação. É necessário produzir um intervalo de confiança da previsão para que os gerentes de projetos e analistas de risco possam estabelecer o nível de confiança de acordo com a sua necessidade. Afinal de contas, esse é o resultado que eles esperam.

A Figura \ref{fig:method} apresenta um fluxograma com a metodologia estabelecida para esse estudo. O procedimento inicia com a seleção da base de dados que serve como conjunto de dados de entrada, nesse caso o PERIL, e finaliza com a atividade "Definição do Intervalo de Confiança". Todas as atividades são executadas sequencialmente, exceto as atividades "Avaliação dos Modelos de Estado da Arte" e "Seleção da Melhor Rede Neural Artificial" que são executados paralelamente.

\begin{figure}[h]

\centering

\includegraphics[width=.45\textwidth]{image/MetodologiaDissertacao.png}

\caption{Fluxograma do estudo realizado}

\label{fig:method}

\end{figure}

**PERIL Database**

Um melhor gerenciamento de riscos inicia com a identificação de problemas potenciais, atribuído como fatores de risco. A adoção dos métodos disponíveis como: revisar lições aprendidas, \textit{brainstorming}, entrevistas e opinião especializada são alternativas relativamente eficientes, no entanto, na maioria das situações elas envolvem alto custo. Uma proposta de baixo custo, extensiva e acessível é utilizar a base de dados \textit{Project Experience Risk Information Library}(PERIL) \cite{kendrick2003identifying}. A base de dados PERIL provê informações e experiências de outras gestão de risco de projetos.

Por mais de uma década, em \textit{Workshops} de Gerenciamento de Riscos, Kendrick coletou dados anônimos de centenas de líderes de projetos lidando com seus problemas em projetos passados. Ele compilou essas informações na base de dados PERIL, que sumariza tanto uma descrição do que houve de errado quanto o impacto que ele teve em cada projeto. A base provê uma perspectiva preocupante do que projetos futuros podem encarar e é valiosa na ajuda para a identificação do que pelo menos poderiam ter sido riscos invisíveis ou \textit{black swans} \cite{kendrick2003identifying}.

Em projetos, os riscos identificados podem ser classificados como "conhecidos", aqueles antecipados durante o planejamento, ou "desconhecidos", identificados futuramente durante a execução do projeto. O propósito dessa base de dados é prover um \textit{framework} para identificar riscos, de modo a aumentar o número de "conhecidos" e diminuir o número de "desconhecidos".

Algumas características da PERIL são:

\begin{itemize}

\item Os dados não estão relacionados, eles representam somente uma pequena fração de dezenas de milhares de projetos realizados pelos líderes de projetos, cujos riscos foram coletados;

\item Apresentam viés, a informação não foi coletada aleatoriamente;

\item Representam somente os riscos mais significativos;

\item São de todo o mundo, com a maioria nas Américas;

\item Não identificam oportunidades;

\item Contém seis centos e quarenta e nove registros, cujo impacto relativo é baseado no número de semanas de atraso no cronograma;

\item Projetos comuns tiveram um cronograma inicialmente planejado entre seis meses e um ano;

\item Tamanho da equipe raramente foi maior que vinte pessoas.

\end{itemize}

Os registros de risco são categorizados em escopo, cronograma e recurso. O escopo é decomposto nas subcategorias mudança e defeito. O cronograma é decomposto em dependência, estimativa e atraso. Os recursos são decompostos em dinheiro, \textit{outsourcing} e pessoas. Um benefício da PERIL é que o autor contempla "\textit{black swans}": riscos com grande impacto, difíceis de prever e com ocorrência rara \cite{taleb2001fooled}.

Kendrick decidiu padronizar o impacto usando como métrica o tempo, medido em semanas de atraso no projeto. Essa tática faz sentido à luz da obsessão de hoje por cumprimento de prazos e foi uma escolha fácil, pois, de longe, o impacto mais sério relatado nesses dados foi o atraso no prazo. Focar-se em tempo também é apropriado porque entre as restrições triplas de projeto - escopo, tempo e custo -, tempo é o único é completamente fora de nosso controle. Afinal, quando se vai, se foi \cite{KEND2003BOOK}.

\begin{table}[h]

\caption{Raw project numbers in the PERIL database}\label{tab:peril\_numbers} \centering

\begin{tabular}{|l|c|c|c|c|}

\hline

\multicolumn{1}{|c|}{} &Americas &Asia &Europe/Middle East &Total \\

\hline

IT/Solution Project & 256 & 57 & 18 & 331 \\

Product Development Project & 224 & 66 & 28 & 318 \\

\hline

\textbf{Total} & \textbf{480} & \textbf{123} & \textbf{46} & \textbf{649} \\

\hline

\end{tabular}

\end{table}

Na Tabela \ref{tab:peril\_numbers}, apresenta-se o número de projetos em TI e de desenvolvimento de produtos nas Américas, Ásia e Europa.

\begin{table}[h]

\caption{Total project impact by root-cause categories and subcategories \cite{KEND2003BOOK}}\label{tab:peril\_pareto} \centering

\begin{tabular}{|l|c|c|c|c|}

\hline

Root-Cause & & &Cumulative &Average \\

Subcategories &Definition &Cases &Impact(weeks) &Impact(weeks) \\

\hline

Scope: & Revision made to scope & & & \\

Changes & during the project & 177 & 1,460 & 8.2 \\

\hline

Resource: & & & & \\

People & Issues arising from internal staffing & 123 & 706 & 5.7 \\

\hline

Scope: & Failure to meet deliverable & & & \\

Defects & requirements & 93 & 654 & 7.0 \\

\hline

Schedule: & Project slippage due to factors & & & \\

Delays & under the control of the project & 102 & 509 & 5.0 \\

\hline

Schedule: & Inadequate durations allocated & & & \\

Estimates & to project activities & 49 & 370 & 7.6 \\

\hline

Resource: & & & & \\

Outsourcing & Issues arising from external staffing & 47 & 316 & 6.7 \\

\hline

Schedule: & Project slippage due to factors & & & \\

Dependencies & outside the project & 41 & 262 & 6.4 \\

\hline

Scope: & & & & \\

Changes & Insufficient project funding & 17 & 228 & 13.4 \\

\hline

\end{tabular}

\end{table}

A Tabela \ref{tab:peril\_pareto} apresenta o número de casos, o impacto cumulativo e médio em semanas para cada categoria e sub-categoria de causa-raiz; além do significado de cada subcategoria.

Uma desvantagem dessa base de dados, é que ela somente contabiliza riscos que tiveram impacto negativo no projeto. As oportunidades não foram identificadas e maximizadas com esse estudo. No entanto, um dos grandes benefícios é que o autor apresenta alguns riscos como \textit{black swans} \cite{KEND2003BOOK}: representando a idéia de riscos com amplo impacto, difíceis de prever e raros de ocorrer. Se o risco tiver impacto negativo, é conhecido como catástrofe, ao passo que, se tiver impacto positivo, é conhecido como recompensa.

**Black Swans**

Denominar alguns riscos como "\textit{black swans}" têm sido popularizado desde os textos de Nassim Nicholas Taleb \cite{taleb2001fooled}. A noção de um "\textit{black swan}" originou-se na Europa antes de ser popularizada pelo Mundo. Já que todos os cisnes observados na Europa eram brancos, um cisne negro era considerado impossível de existir. Porém, foi como um choque quando uma espécie de cisne negro foi descoberta na Austrália. Esse fato deu origem ao uso metafórico do termo "\textit{black swan}" para descrever algo erroneamente acreditado ser impossível.

O conceito de Taleb acerca de "\textit{black swan}" é um evento raro, difícil de prever e de grande impacto. No entanto, é aplicável à gestão de risco do projeto. Nos projetos, é comum que os líderes de projeto para descontar principais riscos do projeto, porque eles são estimados para ter probabilidades extremamente baixas. No entanto, esses riscos ocorrem - a PERIL é cheia deles - e a severidade dos problemas que eles causam significa que ignorá-los pode ser imprudente. Quando esses riscos ocorrem, o mesmo gerente de projetos que inicialmente os negaram começam a percebê-los como muito mais previsíveis - às vezes até mesmo inevitável \cite{KEND2003BOOK}.

Na PERIL, há 127 casos representando os maiores atrasos no cronograma. Como a base de dados mostra, estes riscos mais danosos não são tão raros quanto devem ser pensados, e não devem ser tão difíceis de ser previstos por gerentes se eles dedicarem a atenção apropriada para o processo de gerenciamento de riscos \cite{KEND2003BOOK}.

Em muitas situações, a tarefa mais difícil é identificar e estimar "\textit{black swans}" devido a sua característica: emergente, inesperado, imprevisível e com alto impacto. Portanto, \textit{"black swans"} também são considerados nesse estudo.

**Data Preprocessing**

PERIL contém valores nominais e numéricos. Logo, variáveis nominais foram expressadas através de variáveis binárias. Nesse estudo, utilizam-se doze variáveis binárias para representar as variáveis nominais. Segundo, o impacto que representa a saída real, são números inteiros. Foi observado que a função de distribuição de probabilidade do impacto ajusta-se às funções log-normal e gamma. Portanto, foi realizada uma normalização gamma \cite{han2006data}. Terceiro, a seleção das variáveis mais significativas para o estudo foi realizado após o resultado da análise promovida pelo algoritmo \textit{Random Forest} proposto no livro de Luís Torgo \cite{torgo2003data}.

A Figura \ref{fig:input16} e \ref{fig:input712} apresentam os histogramas das variáveis de entrada. Todos os dados encontram-se binarizados como pode ser observado nos gráfico em barras, que contém o número de ocorrências para cada intervalo de valores.

\begin{figure}[h]

\vspace{-0.2cm}

\centering

\includegraphics[width=\columnwidth]{image/input1\_6.pdf}

\caption{First six input variables}

\label{fig:input16}

\end{figure}

\begin{figure}[h]

\vspace{-0.2cm}

\centering

\includegraphics[width=\columnwidth]{image/input7\_12.pdf}

\caption{Last six input variables}

\label{fig:input712}

\end{figure}

A Figura \ref{fig:impacthistogram} apresenta a saída normalizada pela função gamma para a PERIL num histograma. A forma da função de distribuição de probabilidade é exibida como uma curva sobre o histograma.

\begin{figure}[h]

\vspace{-0.2cm}

\centering

\includegraphics[width=0.8\columnwidth]{image/impact\_histogram.pdf}

\caption{Histogram of impact and shape of the distribution fitting function}

\label{fig:impacthistogram}

\end{figure}

Para o nosso objetivo, PERIL foi dividida em três subconjuntos disjuntos - treinamento, validação cruzada e teste, correspondendo a cinquenta, vinte e cinco porcento e o restante da base de dados, respectivamente. O método de validação-cruzada com divisão de amostras foi utilizado tanto para MRLM e MRA quanto para MLP, SVM, RBF e ANFIS; sendo que para esses a parada prematura também foi utilizada no treinamento \cite{priddy2005artificial}.

**Modelos de Estado da Arte**

Nesta seção, são descritos as configurações adotadas para a experimentação dos modelos de estado da arte.

**Monte Carlo Simulation**

A Simulação de Monte Carlo utilizou a base de dados inteira com o objetivo de aumentar o desempenho do modelo durante a previsão. Além disso, foram filtradas somente as saídas possíveis para uma dada entrada para que uma nova saída pudesse ser calculada, através disso, o desempenho da previsão também aumentou.

**Análise PERT**

A Análise PERT também utilizou a base de dados inteira e somente as saídas possíveis para uma determinada entrada foi utilizada no cálculo da nova saída, tal qual a Simulação de Monte Carlo.

A partir dessa configuração pudemos otimizar esses modelos e observar que há uma falta de habilidade na capacidade de previsão do impacto de riscos a partir de uma base de dados através de modelos estatísticos.

**Modelos de Regressão Linear**

Nesta seção, são descritos as configurações adotadas para a experimentação com os modelos de regressão linear.

O código-fonte dos modelos de regressão linear foram adaptados de Torgo \cite{torgo2003data} para realizar o treinamento, validação cruzada, teste e avaliação da Raiz do Erro Médio Quadrático. Os modelos de RLM e MAR foram comparados estatisticamente para que se pudesse definir um modelo de regressão linear padrão para estudos futuros com a base de dados correspondente.

**Modelo de Regressão Linear Múltipla**

Após a otimização do Modelo de Regressão Linear através da seleção das melhores variáveis de entrada utilizando o critério \textit{Akaike Information Criterion}(AIC) sete das onze variáveis foram selecionadas, além do termo independente.

**Modelo de Árvore de Regressão**

O modelo de árvore de regressão constrói uma árvore de classificação das variáveis de entrada, nesse caso pode não ser necessário utilizar todas as variáveis de entrada para a construção do modelo.

Na Seção \ref{cap:experiments}, três modelos de árvore de regressão foram analisados com o modelo de regressão linear múltipla.

**Otimização por Enxame de Partículas**

Para o PSO, os parâmetros descritos na Tabela \ref{tab:pso\_configuration} foram utilizados. A variação do PSO utilizada é aquela que implementa o coeficiente de constrição de Clerk, como explicado anteriormente.

\begin{table}[h]

\caption{Parâmetros do PSO.}\label{tab:pso\_configuration} \centering

\begin{tabular}{|c|c|}

\hline

Parameter & Value \\

\hline

Coeficiente Cognitivo & 2.05 \\

\hline

Coeficiente Social & 2.05 \\

\hline

Fator de Inércia & 0.8 \\

\hline

Número de partículas & 30 \\

\hline

Número de ciclos & 600 \\

\hline

\end{tabular}

\end{table}

Cada partícula no PSO representa uma configuração candidata. A função de avaliação das partículas é a REMQ de trinta avaliações da rede neural artificial analisada cujos parâmetros são cada partícula. Os parâmetros das redes MLP, SVM e RBF são determinados após a execução desse algoritmo.

**Redes Neurais Artificiais**

Nesta seção, descrevemos as configurações das redes neurais artificiais utilizadas nesse estudo.

**MLPs**

Algumas variações do modelo da MLP foram utilizadas nesse estudo. Diversos parâmetros podem ser alterados como: a quantidade de camadas escondidas, quantidade de neurônios escondidos em cada camada escondida, taxa de aprendizado, momento, número máximo de ciclos de treinamento e regra de aprendizado. Um modelo MLP utilizado nesse estudo é apresentado na Figura \ref{fig:mlp\_example}. Nesse modelo tem-se dez neurônios escondidos em uma única camada escondida.

\begin{figure}[!h]

\vspace{-0.2cm}

\centering

\def \svgwidth{0.55\columnwidth}

\input{image/mlp.pdf\_tex}

\caption{Um modelo MLP utilizado no estudo.}

\label{fig:mlp\_example}

\end{figure}

A quantidade de camadas escondidas estudadas foram uma e duas camadas. O número de neurônios na(s) camada(s) escondida(s), a taxa de aprendizado e o momento foram determinados por um algoritmo de otimização como o PSO para aumentar a precisão na estimativa de erros. Para as análises, o número máximo de ciclos de treinamento foi configurado para seiscentos.

A taxa de aprendizado, momento e a quantidade de neurônios escondidos variam de acordo com os valores apresentados na Tabela \ref{tab:mlp\_configuration\_investigation}.

\begin{table}[h]

\caption{Parameters intervals to MLP model.}\label{tab:mlp\_configuration\_investigation} \centering

\begin{tabular}{|c|c|c|}

\hline

Parameter & Min. Value & Max. Value \\

\hline

Momentum & 0.1 & 0.9 \\

\hline

Learning rate & 0.1 & 0.9 \\

\hline

Hidden Neurons & 1 & 100 \\

\hline

\end{tabular}

\end{table}

Por fim, as regras de aprendizado utilizadas nesse estudo são \textit{Backpropagation}, Levenberg-Marquardt, BFGS Quasi-Newton, \textit{Resilient Backpropagation}, \textit{Polak-Ribiére Conjugate Gradient}, Gradiente Conjugado Escalonado e \textit{One Step Secant}.

Em particular, uma MLP com uma camada escondida e cuja regra de aprendizado tem o objetivo de minimizar o erro médio quadrático mais uma penalidade quadrática através do método BFGS Quasi-Newton teve um melhor desempenho que as demais variações.

**SVM**

O algoritmo SVM para regressão utilizado é o SMOReg. Nesse algoritmo RegSMOImproved é o algoritmo de otimização e PolyKernel é a função de kernel como descrito em \cite{Shevade1999}. O pseudo-código para esse algoritmo é apresentado no Algoritmo \ref{code:svm}.

\begin{small}

\label{code:svm}

\begin{verbatim}

Begin

peril <- read\_file();

peril\_train <- partition(peril, 0, 50);

peril\_crossvalidation <- partition(peril, 50, 75);

peril\_test <- partition(peril, 75, 100);

smo <- SMOReg();

options <- [peril\_train, peril\_crossvalidation,

RegSMOImproved, PolyKernel]

SMOReg.runClassifier(smo, options);

for instance in peril\_test:

calculated <- smo.classifyInstance(instance);

wished <- instance.classValue();

REMQ <- REMQ + (wished - calculated)^2

end

n <- peril\_test.size();

REMQ <- REMQ/n;

REMQ <- sqrt(REMQ);

End

\end{verbatim}

\end{small}

No Algoritmo \ref{code:svm}, os dados são lidos a partir de um arquivo, dividido nos subconjuntos treinamento, validação cruzada e teste. Instanciar o modelo de regressão SMOReg, executar o treinamento do modelo, gerar a saída calculada e calcula o REMQ a partir da saída real e calculada. Esse algoritmo é utilizado como a função de otimização para o algoritmo de otimização por enxames.

**RBF**

A rede neural RBF utilizada é a RBFRegressor, ela minimiza o erro quadrático através do método BFGS. Os centros iniciais das gaussianas são encontrados utilizando SimpleKMeans, um algoritmo que implementa K-Médias. O sigma inicial é configurado para a maior distância entre qualquer centro e o vizinho mais próximo no conjunto de centros. O parâmetro de cume é usado para penalizar o tamanho dos pesos na camada de saída, o qual implementa uma combinação linear simples. O número de funções de base pode também ser especificado. Para esse estudo somente um sigma global é utilizado para todas as funções de base. O pseudo-código para esse algoritmo é apresentado no Algoritmo \ref{code:rbf}.

\begin{small}

\label{code:rbf}

\begin{verbatim}

Begin

peril <- read\_file();

peril\_train <- partition(peril, 0, 50);

peril\_crossvalidation <- partition(peril, 50, 75);

peril\_test <- partition(peril, 75, 100);

rbf <- RBFRegressor();

options <- [peril\_train, peril\_crossvalidation, peril\_test]

RBFRegressor.runClassifier(rbf, options);

for instance in peril\_test:

calculated <- rbf.classifyInstance(instance);

wished <- instance.classValue();

REMQ <- REMQ + (wished - calculated)^2

end

n <- peril\_test.size();

REMQ <- REMQ/n;

REMQ <- sqrt(REMQ);

End

\end{verbatim}

\end{small}

No Algoritmo \ref{code:rbf}, os dados são lidos a partir de um arquivo, dividido nos subconjuntos treinamento, validação cruzada e teste. O modelo de regressão SMOReg é instanciado, o treinamento do modelo é executado, a saída calculada é gerada e a REMQ é obtida a partir da saída real e calculada. Esse algoritmo é utilizado como a função de otimização para o algoritmo de otimização por enxames.

**ANFIS**

O ANFIS é um sistema neuro-fuzzy implementado no Matlab por Sugeno \cite{jang1997neuro}. ANFIS usa um algoritmo de aprendizado híbrido para identificar parâmetros do sistema de inferência fuzzy Sugeno. Ele aplica uma combinação do método dos mínimos quadrados e o método do gradiente descendente \textit{backpropagation} para o treinamento dos parâmetros da função de pertinência do sistema de inferência fuzzy. O sistema de inferência fuzzy utilizado foi o genfis2, já que há um número grande de variáveis de entrada. O pseudo-código para esse algoritmo é apresentado no Algoritmo \ref{code:anfis}.

\begin{small}

\label{code:anfis}

\begin{verbatim}

Begin]

inputs = csvread(peril,0,0,[0,0,648,10])

targets = csvread(peril,0,11)

tData = [inputs targets];

in\_fis = genfis2(inputs,targets, 0.7);

trainOpts = [100,0.1,0.01,0.9,1.1]

displayOpts = [1,1,1,1];

chkData = []

[fis,error,stepsize,chkFis,chkErr] =

anfis(tData,in\_fis,trainOpts,displayOpts,

chkData,1);

for err in error:

REMQ <- REMQ + (err)^2

end

n <- peril.size();

REMQ <- REMQ/n;

REMQ <- sqrt(REMQ);

End

\end{verbatim}

\end{small}

No Algoritmo \ref{code:anfis}, os dados de entrada e saída são lidos a partir de um arquivo. O sistema de inferência fuzzy é instanciado, o treinamento do modelo é executado, o erro é gerado e a REMQ é obtida a partir da saída real e calculada. Esse algoritmo é utilizado como a função de otimização para o algoritmo de otimização por enxames.