

# PREVISÃO DE SÉRIES TEMPORAIS COM CLUSTERIZAÇÃO, ANÁLISE DE PREVISORES E COMITÊ DE MÁQUINAS

Francisco Wagner Costa Aquino, Gustavo Augusto de Lima Campos, Jerffeson Teixeira de Souza

Universidade Estadual de Ceará (UECE)  
wcaquino@gmail.com, {gustavo, jerffeson}@larces.uece.br

**Resumo** – Este trabalho busca apresentar um método para previsão de séries temporais dotado de vários previsores especializados em classes de dados (clusters) distintas, todas com um erro médio atrelado à mesma. A rede preditora de cada cluster é gerada automaticamente (considerando sua estrutura e outros parâmetros, como janela de tempo e os atributos) sendo treinada com um conjunto de exemplos representativos e os mais semelhantes possíveis. Várias estratégias são abordadas seguindo esta ideia até a definição de uma que, dependendo dos dados de entrada, seja decidido se ela será prevista por uma rede neural simples ou um complexo comitê de máquinas.

**Abstract** – This paper presents a method for time series forecasting with several forecasters specialized in distinct data classes (clusters), each one with an average error associated itself. The network predictor of each cluster is generated automatically (according to their structure and other parameters such as the time window and attributes) being trained with a set of representative examples as similar as possible. Several strategies are discussed following this idea to the definition of a that, depending on the input data, it is decided whether it will be provided by a simple neural network or a complex committee machines.

**Keywords** – Previsão, séries temporais, clusterização, *backpropagation*, comitê de máquinas.

## 1 Introdução

A ideia de conseguir acertar (prever) eventos é de interesse e, em alguns casos, necessidade das pessoas. Para isto, muitas pesquisas foram realizadas com sucesso nesta área, em diversos domínios de aplicação, que vai da predição de séries temporais financeiras (Dablemont, 2003), previsão de vazão de rios (Atiya, 1999), modelagem de séries temporais biomédicas (Coyle, 2005) e predição de tráfego de rede (Doulamis 2003) [1].

Série temporal é uma coleção de observações realizadas em ordem cronológica. As formas de se trabalhar na previsão de séries temporais se tornaram mais complexas com o tempo, sempre na busca de reduzir a distância entre o valor previsto e o valor desejado na previsão, como disse (Kalyvas, 2001), o melhor a se fazer com tanta incerteza é tentar reduzi-la [2]. Outro problema é saber quais dados (atributos) podem ajudar na previsão de uma série temporal. A calibragem de um sistema preditor, na sua forma mais simples, consiste em várias simulações com ajustes de parâmetros e atributos utilizados em um grande experimento com tentativas e erros.

Fundamentalmente a abordagem é um comitê de máquinas cujo valor estimado final combina: uma rede de Kohonen; e uma combinação linear envolvendo as informações sobre a distância euclidiana entre o padrão de entrada e os "neurônios" na grade, e as saídas de várias redes predictoras associadas aos neurônios da grade, geradas automaticamente, inclusive o espaço de *features* (seleção de atributos), empregando-se algoritmo genético e os subconjuntos de exemplos associados aos seus respectivos neurônios na grade (seleção de exemplos).

Mais um ponto importante neste trabalho é a utilização do erro médio de cada cluster na previsão dos dados. Sabendo do comportamento médio de cada classe de dados, uma previsão pode ser feita com maior ou menor segurança. O que não impede que novas estratégias sejam utilizadas nas classes "menos seguras" em busca de melhorias.

Este trabalho está organizado da seguinte forma: A seção 2 apresenta alguns trabalhos relacionados à área de previsão de séries temporais; a seção 3 fala sobre as tecnologias utilizadas ao longo do trabalho; a seção 4 apresenta a abordagem modularizada em subseções; A seção 5 exhibe alguns resultados da avaliação da proposta; e a seção 6 faz algumas considerações finais e apresenta algumas sugestões de melhorias.

## 2 Previsão

O problema da previsão de séries temporais tem sido estudado há bastante tempo. Em 1970, George Box e Gwilym Jenkins aplicaram modelos auto regressivos de médias móveis (ARMA ou ARIMA) na tentativa de encontrar semelhanças em trechos passados da série temporal de forma a ajudá-los na tarefa da previsão de dados futuros [3]. Em 1986, foi criado um modelo de processos de heterocedasticidade condicional auto regressiva generalizada (GARCH) para modelar a volatilidade de séries temporais e estimar a variância condicional [4, 5].

Com o tempo, Comrie (1997) mostrou que redes neurais artificiais (RNA) apresentam resultados um pouco superiores quando comparadas a métodos de regressão tradicionais, na previsão do nível de ozônio em diferentes cidades. Cho (2003) mostrou que as RNAs se saíram melhores que o modelo ARIMA na previsão de turistas em Hong Kong, principalmente em casos onde não existe um padrão definido, como no caso da oscilação do número de turistas oriundos do Japão devido à crise

asiática [6]. Cybenko (1989) provou que, se os pesos corretos entre as conexões forem encontrados, uma rede neural com uma quantidade de neurônios suficiente na camada intermediária pode ser usado para aproximar qualquer função multidimensional para qualquer grau de precisão [7].

O fato de muitas séries temporais serem ruidosas, não lineares e, em alguns casos, caóticas [8], faz com que os modelos de RNA, geralmente, apresentem desempenho melhores que técnicas lineares tradicionais [1]. Chegando-se a conclusão que as técnicas lineares convencionais têm cedido cada vez mais espaço para técnicas não lineares que conseguem capturar, com mais eficiência, a dinâmica dos sistemas complexos [1].

Os algoritmos genéticos começaram a ser utilizados junto com redes neurais na tentativa de automatizar processos como: projeto automático da rede, seleção de atributos, seleção de exemplos dentre outros [6, 2, 9, 10]. A utilização em conjunto de algoritmos genéticos e RNAs vem crescendo, já que os sistemas resultantes identificam e mapeiam melhor as relações não lineares entre variáveis, trabalham com dados incompletos e ruidosos e não requisitam qualquer suposição a priori sobre os dados [11].

Apesar de Antunes falar que a classificação não tem obtido muita atenção nas séries temporais, alguns trabalhos na área são encontrados [12]. Siqueira (2006) diz que mapas auto-organizáveis foram utilizados para encontrar padrões selecionando os neurônios que eram ativados mais vezes [13]. Chappelier (1996) classifica assinaturas levando em consideração a sequência da caneta para “desenhá-la” [14]. Lendasse (1998) clusteriza os dados e calcula a saída usando de estatística, escolhendo a saída que possui maior frequência [15]. Além de mapas auto-organizáveis, o *K-means* também é utilizado para a clusterização de dados [16, 17]. Outra técnica que tem apresentado bons resultados na previsão de séries temporais é o comitê de máquinas. Em [18], uma melhora significativa foi obtida apenas com a média aritmética das saídas de 30 redes *multi-layer perceptron* (MLP).

### 3 Fundamentação teórica

Para o desenvolvimento deste trabalho várias tecnologias amplamente conhecidas na área de inteligência computacional serão utilizadas. A primeira delas é a Redes de Kohonen (mapas auto-organizáveis) que é um algoritmo neural não supervisionado usado para extrair regularidades estatísticas presentes em conjuntos de dados de entrada [19]. As redes de Kohonen utilizam o conceito de vizinhança, tanto para atualizar os pesos durante o treinamento, como para definir o padrão que mais se assemelha ao dado de entrada apresentado. Barreto (2001) destaca esta rede pela sua poderosa habilidade de clusterização e classificação [20]. O principal objetivo da clusterização é identificar uma estrutura dentro de um conjunto de dados não rotulados onde os dados mais semelhantes sejam organizados em grupos homogêneos [21].

A rede de Kohonen é utilizada neste trabalho para realizar uma seleção de exemplos para as redes de previsão através de clusterização. Segundo Franco (2000), uma das propriedades fundamentais que tornam as redes neurais tão importantes é capacidade de aprender a partir dos exemplos apresentados [22].

Além de utilizar as redes de Kohonen para clusterização, outra arquitetura de RNA é utilizada: *multi-layer perceptron* (MLP). A arquitetura MLP é a mais amplamente aplicada na previsão de aplicações [23], redes *feedforward* são consideradas aproximadores universais de funções [7, 24, 25].

Uma forma para buscar uma configuração ótima para o problema em questão sem a necessidade de ser um especialista no assunto para fazer análises detalhadas nos parâmetros de calibração da rede é a utilização de metas-heurísticas. Algoritmos genéticos têm sido utilizados no intuito de maximizar o projeto da topologia de uma RNA com bons resultados [26].

Para realizar o treinamento de um sistema, devem-se definir quais dados serão utilizados como entrada do mesmo. Cybenko (1989) diz que cada dado de entrada deve representar alguma variável independente que tem alguma relação com a saída da rede neural [7]. Zhang (2003) diz que muitas das variáveis podem ser irrelevantes ou redundantes com a previsão. Introduzindo mais atributos na camada de entrada do modelo não necessariamente estará introduzindo novas informações, mas sempre estará introduzindo novos ruídos [2].

Máquina de comitê é um tipo de arquitetura que combina a saída de vários especialistas (redes neurais) em uma saída apenas. Segundo Haykin (2001), Utiliza a estratégia “Dividir para Conquistar”, onde uma tarefa computacional complexa é resolvida dividindo-a em um número de tarefas computacionais simples e então combinando as soluções destas tarefas. [27]

### 4 Abordagem

A abordagem proposta neste trabalho busca minimizar o erro na previsão de séries temporais. Para isto, uma série de passos é apresentada onde cada um irá melhorar a acurácia do sistema finalizando em uma máquina de comitê. Ao todo, teremos uma heurística que faz clusterização dos dados (seleção de exemplos), projeto automático, seleção de atributos, análise de resultados/desempenho de cada previsor e um comitê de máquinas.

O algoritmo da abordagem se divide em duas etapas, uma de pré-processamento (Algoritmo 1) bem mais demorada onde é realizada a clusterização dos dados de treinamento e configuração das redes MLP associadas aos neurônios da grade de Kohonen e outra fase (Algoritmo 2) onde a previsão, de fato, é realizada:

### Algoritmo 1: Pré-processamento

1. Treinar uma rede de Kohonen com os dados de treinamento
2. Para cada neurônio da rede de Kohonen:
  1. Selecionar todos os dados (exemplos) que ativam o neurônio em questão (melhor neurônio)
  2. Usando apenas os dados obtidos no passo acima, projetar automaticamente, ou seja, buscar os atributos (inclusive suas janelas de tempo) e configuração da rede MLP que realize a previsão dos dados (do subconjunto de dados definido no passo acima) com o menor erro médio.
  3. Armazenar os atributos e estrutura da rede definidos no passo acima.

### Algoritmo 2: Previsão

1. Usando a divisão dos dados e a rede de Kohonen treinada no algoritmo 1, para cada dado do conjunto de teste:
  1. Buscar o melhor neurônio na rede Kohonen para o dado em questão
  2. Obter a rede MLP e os atributos a serem usados como entrada que foram definidos no pré-processamento para o neurônio em questão
  3. Se o erro médio da MLP for baixo, apresentar a entrada para a MLP em questão. A saída da mesma será a saída final.
  4. Senão, apresentar a entrada para o comitê de máquinas. A saída final será calculada pelo comitê.

## 4.1 Seleção de exemplos via Rede de Kohonen

Um ponto a se pensar na criação de uma estrutura de previsão de séries temporais é a qualidade dos exemplos utilizados no treinamento. Os problemas já surgem na hora de quantificar a qualidade de um determinado exemplo, este pode ser bom na aprendizagem de um padrão de entrada e péssimo para outro. Uma solução seria realizar o treinamento apenas com os exemplos bons para o dado de entrada em questão, a questão é como saber quais os melhores exemplos para uma entrada nunca apresentada ao sistema.

Esta abordagem busca resolver este problema fazendo a clusterização dos dados através de uma rede de Kohonen. Após a rede estar treinada (com todos os dados do conjunto de treinamento), para cada entrada, calcula-se qual neurônio na rede será ativado (melhor neurônio) e apenas os exemplos que ativam o mesmo neurônio serão utilizados no treinamento da rede de previsão específica para aquela entrada. Desta forma, estaremos utilizando apenas os exemplos mais “semelhantes” (segundo o critério da rede) e descartando exemplos considerados dispare.

Em séries temporais, principalmente as caóticas, dificilmente os dados são uniformes, ou seja, enquanto alguns clusters possuem vários exemplos associados, existem outros com pouquíssimos exemplos, o que pode ocasionar previsões piores nas redes criadas para trabalhar com os dados destes clusters. A proposta para aumentar a quantidade de exemplos N nestes casos é relaxar a restrição de proximidade, ampliando o raio de aceitação, a partir do melhor neurônio (neurônio ativado pelo dado de entrada).

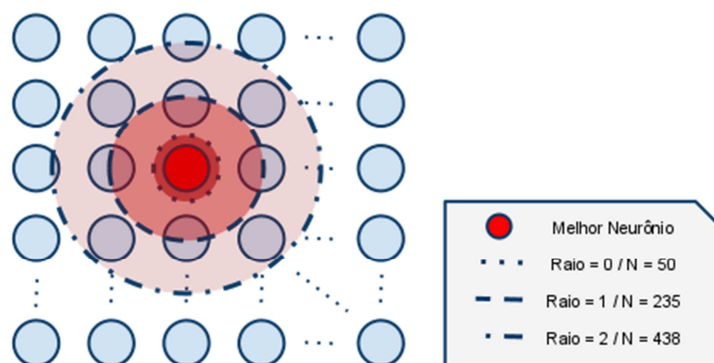
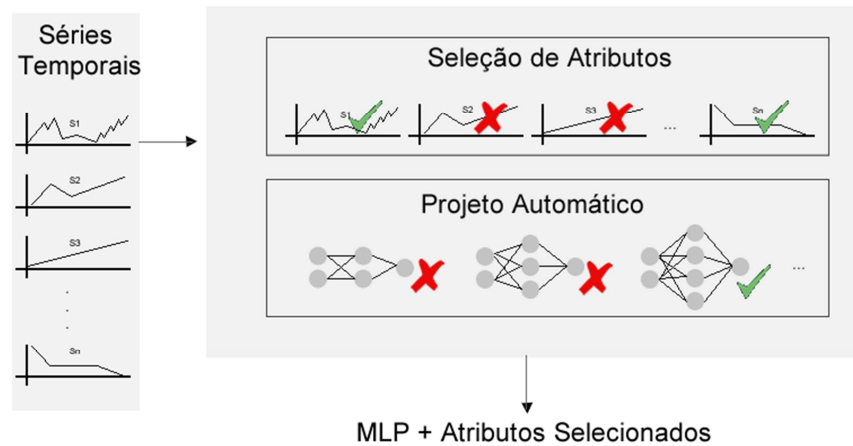


Figura 1 – Relaxamento na restrição de proximidade.

A figura 1 ilustra o processo de relaxamento na restrição de proximidade (semelhança). A estratégia é aumentar o raio gradativamente até atingir uma quantidade mínima de exemplos selecionados. O número mínimo ideal de exemplos a serem coletados não pode ser definido por uma constante, este deve ser analisado caso a caso, deve-se buscar um meio termo para não ter problemas por utilizar poucos exemplos no conjunto e treinamento e evitar apresentar dados fora do padrão que a rede está sendo treinada, o que pode causar ruídos. Desta forma, é possível aumentar a quantidade de exemplos sempre buscando selecionar os dados mais semelhantes dentro do universo do conjunto de treinamento.

## 4.2 Projeto automático de MLP e Seleção de atributos

A rede de Kohonen permite que, para cada entrada, sejam selecionados apenas os exemplos mais similares a ela, pode-se dizer que cada neurônio representa um cluster, uma classe de dados. Estes exemplos são utilizados para o treinamento de uma rede específica para realizar a previsão daquela entrada, generalizando um pouco mais, da classe de dados da qual aquela entrada pertence.

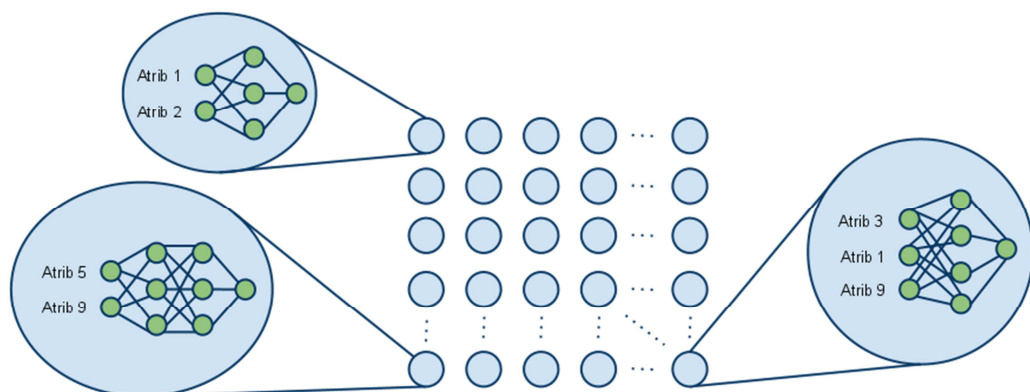


**Figura 2** – Processo de treinamento da rede de previsão.

Cada classe de dados é independente das demais e podem ser manipuladas de formas distintas. Estando de posse dos exemplos de cada classe, é possível buscar a melhor configuração para uma rede de previsão, MLP com *backpropagation*. A figura 2 exibe como o processo é realizado, para cada classe, a seleção de atributos e projeto automático de uma rede MLP onde apenas os dados daquela classe serão utilizados.

A seleção de atributos e o projeto automático da rede neural são realizados simultaneamente por um algoritmo genético onde cada cromossomo define quais atributos são utilizados como entrada e qual a estrutura da rede MLP. Junto com os atributos que podem ser utilizados, o sistema também permite que sejam determinadas janelas de tempo independentes para cada um dos atributos, ou seja, pode-se utilizar como entrada apenas a última observação do atributo A (D+0), as três últimas observações do atributo B (D+0, D-1 e D-2) e até mesmo nenhuma observação do atributo C. O tamanho máximo da janela é parametrizável, sendo que, quanto maior seja este parâmetro, maior será o tamanho do cromossomo.

Após definir a estrutura de previsão de cada classe, tem-se um mecanismo de previsão, onde é possível mapear, através da rede de Kohonen, qual a rede MLP (treinada apenas com os exemplos daquela classe) e atributos de uma determinada entrada que devem ser utilizados para minimizar o erro na previsão. A figura 3 ilustra como cada neurônio da rede de Kohonen está associado a uma rede MLP e a um subconjunto de atributos para realizar a previsão de entradas pertencentes àquela classe.



**Figura 3** – Rede de Kohonen, cada neurônio com uma rede de previsão distinta associada.

Utilizando o conceito de poder trabalhar com cada classe de dados de forma independente, é possível estender um pouco mais este raciocínio fazendo avaliações independentes de cada rede previsora, de cada neurônio da grade. O fato de realizar previsão em separado permite avaliar o desempenho das redes MLP que preveem cada classe. Esta informação é importante, pois permite ao sistema extrair informações importantes sobre o comportamento de cada classe de dados.

**Tabela 1** – Exemplo dos erros médios das classes de dados.

Neurônio de Kohonen	Erro Médio da Classe
A	0,56
B	0,000042
C	5,79

A Tabela 1 exhibe um exemplo de avaliação dos previsores das classes de dados. Todos os neurônios devem possuir uma taxa média de erro associada a eles, de posse desta informação é possível inferir que as entradas que ativam o neurônio B foram previstas com uma boa precisão (baixa taxa de erro), já os dados da classe C estão com uma taxa de erro muito alta. Não é possível afirmar que as previsões da rede para a classe B tenha um alto grau de confiança, mas é possível afirmar que, em média, a classe B tem realizado previsões próximas dos valores desejados, ou até mesmo podemos suspeitar que os dados da classe C podem ser muito ruidosos.

Esta medida é importante para ajudar a definir como o sistema deve trabalhar com cada entrada a ser prevista. Por exemplo, caso alguma entrada esteja associada à classe B, o sistema realiza a previsão com uma segurança maior devido ao fato de o erro médio para previsões naquela classe ser baixo. Caso a entrada esteja associada ao neurônio da classe C, o sistema pode simplesmente se abster de realizar uma previsão para aquela entrada, pois os erros de previsão para entradas similares, em média, estão altos. Como cada classe é independente, pode-se tentar melhorar a rede de previsão das classes com taxas de erros altas ou, até mesmo, buscar alguma outra estratégia que melhore as previsões dos dados daquela classe sem interferir ou impactar nas redes de classes com baixa taxa de erro.

### 4.3 Comitê de Máquinas

Conforme visto na seção anterior, podem-se criar novas estratégias para melhorar a previsão de alguma classe em específico sem interferir nos resultados das demais. A estratégia apresentada neste trabalho também define um comitê de máquinas onde todo o conceito de classe de dados e proximidade (semelhança) é aproveitado.

O valor previsto é calculado através de uma composição de todas as saídas das redes predictoras de cada neurônio da grade de Kohonen. O valor de cada saída será ponderado ( $\alpha$ ) pela distância que o neurônio em questão está em relação ao melhor neurônio (neurônio que foi ativado pela entrada), conforme definido na equação 1.

$$S = \frac{\sum_{i=0}^N \alpha_i * S_i}{\sum_{i=0}^N \alpha_i} \quad (1)$$

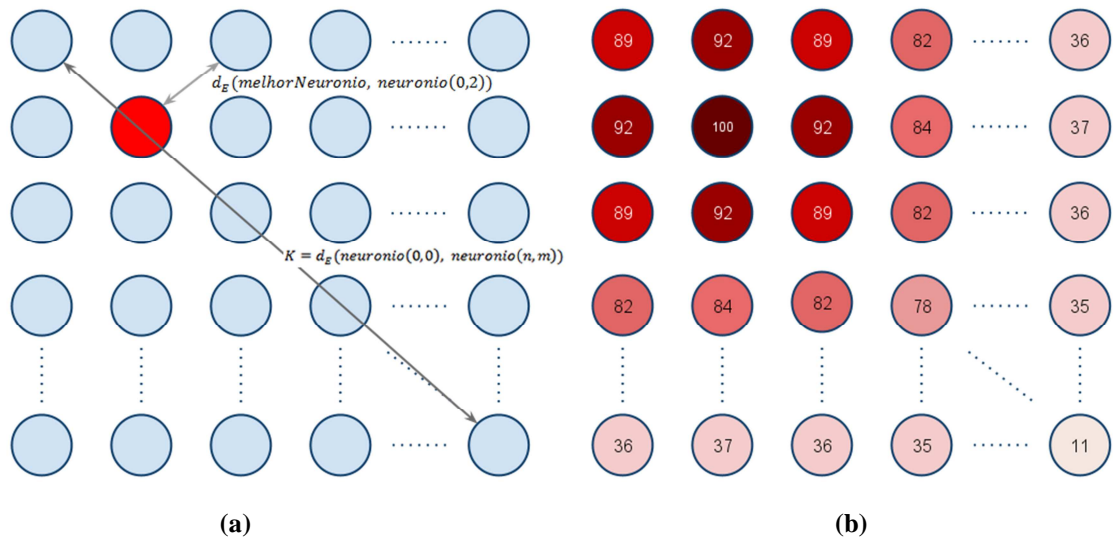
onde  $S_i$  representa a saída do neurônio  $i$  e  $\alpha_i$  é calculado a partir da equação 2.

$$\alpha_i = \frac{d_E(\text{melhorNeuronio}, \text{neuronio}_i)}{K} \quad (2)$$

onde  $K$  representa a distância euclidiana entre os pontos de uma das diagonais da rede de Kohonen, conforme equação 3 e ilustrado na figura 4(a).

$$K = d_E(\text{neuronio}(0,0), \text{neuronio}(n,m)) \quad (3)$$

onde  $n$  representa a quantidade de linhas da grade de Kohonen e  $m$  representa a quantidade de colunas da mesma. O  $K$  foi criado para normalizar o  $\alpha_i$  que tem função de definir a participação de cada saída pela distância entre o neurônio em questão e o melhor neurônio. A figura 4(b) exhibe a participação percentual (x100) de cada neurônio ( $\alpha_i$ ) no cálculo da saída final do sistema, para este exemplo, quando o neurônio (1, 1) for o neurônio ativado, ou seja, quando este for o melhor neurônio.



**Figura 4** – Exemplo de um cálculo da distância para o melhor neurônio e cálculo do  $K$  (a). Participações percentuais ( $\alpha_i * 100$ ) das saídas de cada neurônio da grade na composição da saída resultante. (b)

Na figura 4(b) é possível notar um gradiente onde os neurônios mais próximos do neurônio escolhido, ou seja, os mais similares ao padrão da entrada, possuem uma participação maior na saída resultante do comitê aqui definido. Desta forma, a tarefa de previsão foi dividida entre vários previsores (Combinação de conhecimento) com o diferencial que as saídas das redes mais aptas a realizar a previsão terão um peso maior no resultado final. Cada neural pode ter uma área de conhecimento específica, a junção destes valores aumenta a acurácia das previsões.

## 5 Avaliação

Para avaliar a proposta, foram utilizadas séries temporais econômicas. O sistema possui nove séries temporais da empresa Vale (VALE5) no período compreendido entre os anos 2000 e 2009, também é possível flexibilizar o tamanho da janela de cada atributo entre 1 (D+0) e 3 (D+0, D-1, D-2) dias, onde cada dia é considerado um atributo distinto, totalizando 27 (3\*9) atributos a disposição do sistema. Os dados foram divididos em 50% para o conjunto de treinamento, 25% para o conjunto de validação e 25% para o conjunto de teste. Os conjuntos foram formados através de sorteios sem reposição dentro do universo de dados e os mesmos conjuntos sorteados foram utilizados em todos os testes. A medida de avaliação considerada é o *Root Mean Square Error* (RMSE) do conjunto de testes.

Como a abordagem definida neste trabalho é modular, foi realizada uma sequência de testes com algumas diferenças entre si (alterando a ordem que as tecnologias são aplicadas ou, até mesmo, não utilizando algumas delas). Esta sequência tem o objetivo de provar a modularidade da abordagem, onde cada tecnologia adicionada ao sistema implica em melhoria e mostrar que a sequência em que as mesmas são empregadas também influencia o resultado final. Os seguintes cenários de testes foram levantados:

1. **Teste de controle:** Teste mais simples que utiliza apenas uma MLP, a melhoria aplicada foi a Seleção de Atributos e o projeto automático (4.2) realizado em cima de todo o conjunto de treinamento. Chamado de “Controle” pois será a base para comparação de todas as outras estratégias.
2. **Projeto + Atributos antes da clusterização:** Utilizando a mesma estrutura da rede e atributos definidos no teste 1, várias redes foram treinadas após a seleção de exemplos obtida pela clusterização (4.1). Desta forma, cada neurônio da grade possuía uma rede de mesma configuração das demais com a diferença que os pesos entre as conexões (definidos nos treinamentos) estavam “otimizados” para aquela classe de dados, devido ao fato de a rede ter sido treinada apenas com dados daquela classe. A saída resultante é a calculada pela rede MLP associada à classe do dado de entrada.
3. **Projeto + Atributos antes da clusterização com erro baixo:** Utiliza a mesma estrutura definida no teste 2, com a diferença de o sistema se abster de realizar previsões para as classes de dados cujo erro associado a ela estava muito alto (RMSE > 5, vale ressaltar que este valor foi obtido empiricamente e não deve ser generalizado para todos os domínios de aplicação).
4. **Projeto + Atributos depois da clusterização:** Aplica a clusterização (4.1), depois faz o projeto automático e seleção de atributos para cada cluster utilizando apenas os exemplos daquela classe (4.2). A diferença deste para o teste 2 é que cada classe de dados tem uma rede distinta e otimizada para aquele cluster. Inclusive os atributos de entrada (e sua janela de tempo) podem variar entre as classes (neurônios da grade). A saída resultante é a calculada pela rede MLP associada à classe do dado de entrada.

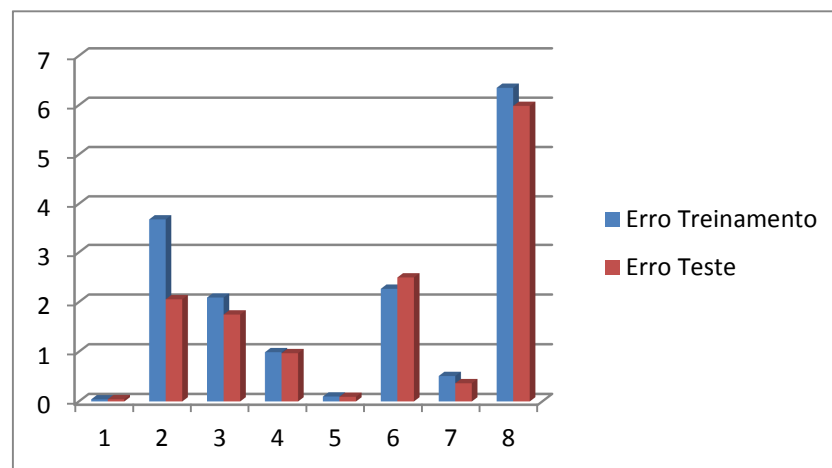
5. **Projeto + Atributos depois da clusterização com erro baixo:** Utiliza a mesma estrutura definida no teste 4, com a diferença que, assim como no teste 3, o sistema se abstém de realizar previsões para classes com erro alto.
6. **Comitê:** Aplica a clusterização (4.1), faz o projeto automático com seleção de atributos (4.2) e, para cada dado do conjunto de teste, calcula a saída resultante através do comitê de máquinas definido na seção 4.3.
7. **Erro baixo + Comitê:** Estratégia semelhante à do teste 5, com a diferença que, ao invés de se abster de previsões para classes com erro alto, será aplicado o comitê (4.3) nestas classes “problemáticas” (similar ao utilizado no teste 6). A saída resultante é a calculada pela rede MLP associada à classe do dado de entrada quando está possuir um erro médio associado baixo, caso contrário, a saída é calculada através do comitê de máquinas.

**Tabela 2** – Avaliação das estratégias.

#	Teste	RMSE	Abstenções	Melhoria
1	Teste de Controle	1,67299	0	-
2	Projeto + Atributos antes da clusterização	1,23811	0	26%
3	Projeto + Atributos antes da clusterização com erro baixo	0,83802	36	49,9%
4	Projeto + Atributos depois da clusterização	0,96635	0	42,24%
5	Projeto + Atributos depois da clusterização com erro baixo	0,66071	46	60,51%
6	Comitê	0,97295	0	41,84%
7	Erro baixo + Comitê	0,82637	0	50,6%

A tabela 2 expõe os resultados dos cenários de testes já definidos nesta seção. O teste 2 já é suficiente para validar que a seleção de exemplos via clusterização trás melhorias pois foi a única alteração aplicada em relação ao teste 1. Os testes 3 e 5 mostram que a utilização da taxa de erro no auxílio da tomada de decisão trazem ótimos resultados em relação aos testes semelhantes que não foi utilizado (testes 2 e 4 respectivamente), com a ressalva das abstenções (relativamente pequenas se considerar que o conjunto de teste possui 559 registros). O teste 4 prova que configurar o sistema predictor de um neurônio da grade apenas com os dados daquela classe é melhor que se fosse utilizado todos os dados juntos (teste 2). O teste 6 mostra que o comitê puro conseguiu algumas melhorias mas não foi a melhor estratégia. O teste 7 prova que cada classe de dados realmente pode se trabalhada de forma distinta na busca de reduzir o erro médio do sistema como um todo.

O gráfico na Figura 5 exhibe o erro médio do conjunto de treinamento e do conjunto de testes calculado em alguns neurônios da grade de Kohonen. Neste gráfico é possível notar que a diferença entre os erros em treinamento e testes é pequena, ou seja, a rede preditora de cada classe está com desempenho semelhante para os dados daquela classe. Sabendo disto pode-se concluir que utilizar o erro médio do conjunto de treinamento como medida de segurança para avaliar se uma previsão no conjunto de testes terá um erro alto ou baixo é válida. Outra prova que valida esta abordagem, já mencionada no parágrafo acima, é o fato de o erro ter caído consideravelmente, sempre que este raciocínio foi levado em consideração (testes 3, 5 e 7).



**Figura 5** – Erro médio do conjunto de treinamento e do conjunto de testes calculado em alguns neurônios da grade de Kohonen.

Com base em todos os testes, a estratégia com menor erro médio foi a do teste 5, porém as abstenções para alguns dados, dependendo do caso/domínio de aplicação, pode ser inaceitável. A estratégia sem abstenções que obteve melhor resultado foi a 7, onde foi usado o poder das máquinas de comitês para melhorar o desempenho nas previsões das classes mais “difíceis”.

## 6 Conclusão

De uma forma geral, a clusterização dos dados é válida pois todas as abordagens resultaram em melhorias consideráveis em relação ao teste de controle. Realizar o projeto automático e a seleção de atributos apenas com dados específicos da classe de dados obtiveram resultados melhores do que fazer estes passos antes da clusterização. A utilização do erro médio de cada preditor como auxiliar na tomada de decisão reduziu consideravelmente o erro, porém uma pequena quantidade (menos de 10%) de dados ficou sem previsão (abstenções). A máquina de comitê gerou bons resultados, mas se sobressaiu apenas como estratégia secundária auxiliando nas classes onde o erro médio associado da primeira estratégia estava muito alto.

Apesar de a validação ter sido realizada em séries temporais econômicas, não existem restrições para que esta abordagem seja utilizada em outros domínios de aplicação. Basta reunir as séries temporais consideradas relevantes para o problema e definir qual atributo que se deseja realizar as previsões, o que é mais um ponto positivo para abordagem, não é necessário ser um especialista na área de domínio, a seleção de atributos se encarregará de desconsiderar os atributos irrelevantes.

Um dos pontos fracos da abordagem é o tempo necessário para o pré-processamento definido no Algoritmo 1, o tamanho da grade de Kohonen no tempo de treinamento. Após esta fase inicial, as previsões (Algoritmo 2) são realizadas em questão de segundos.

Atualmente as configurações da rede de Kohonen, os atributos considerados para a clusterização e o limite mínimo de dados para treinamento das redes MLP de cada neurônio foram definidos empiricamente. Após vários testes e observações foi concluído que, assim como em [20], a quantidade de neurônios e épocas influenciam na qualidade do sistema, uma possível melhoria seria buscar alguma heurística capaz de calibrar estes dados.

## 7 Referências Bibliográficas

- [1] Menezes, J. M. P., Redes neurais dinâmicas para predição e modelagem não-linear de séries temporais
- [2] Kalyvas, E., *Using neural networks and genetic algorithms to predict stock market returns*, 2001.
- [3] Box, George and Jenkins, Gwilym, *Time series analysis: Forecasting and control*, San Francisco: **Holden-Day**, 1970.
- [4] Oliveira, M. A., et al., Previsão de retornos de ações de empresas dos setores financeiro e de alimentos por meio de redes neurais e modelos ARIMA-GARCH
- [5] Bollerslev, T., *Generalized Autoregressive Conditional Heteroskedasticity*, **Journal of Econometrics**, 1986.
- [6] Moura, F. A., O uso de redes neurais artificiais na previsão de tendências no mercado de ações.
- [7] Cybenko, G., *Approximation by Superposition of a Sigmoidal Function*, *Mathematics of Control, Signals, and Systems* (**Springer-Verlag**, New York Inc., 1989).
- [8] Elliot Wave Theory - <http://www.investopedia.com/articles/technical/111401.asp>, acessado em 09/01/2010
- [9] Sikora, G., Projeto automático de redes neurais artificiais para o problema de previsão em séries temporais.
- [10] Boozarjomehry, R. B., Svrcek, W. Y., *Automatic design of neural network structures*, *Computer and Chemical Engineering*, 2001.
- [11] Vellido, A., Lisboa, P. J. G., Vaughan, J. (1999), *Neural networks in bussiness: a survey of applications (1992-1998)*. **Expert Systems with Applications**.
- [12] Antunes, C. M., Oliveira A. L., *Temporal Data Mining: An overview*, **Lecture notes in computer science**, INESC
- [13] Siqueira, E. S., Almeida, I. L., Cechin, A. L., Utilização de Mapas Auto-organizáveis no pré-tratamento de Dados de Microarranjos. **CRICTE 2006**
- [14] Chappelier, J. C., Grumbanch, A., *A Kohonen Map for Temporal Sequences*, **In Proceedings of NEURAP'95**, 1996
- [15] Lendasse, A., Verleysen, M., Bodt, E., *Forecasting Time-Series by Kohonen Classification*. European Symposium on Artificial Neural Networks (**ESANN'1998**)
- [16] Singhal, A., Seborg, D. E., *Clustering multivariate time-series data*, **Journal of Chemometrics** **19**, 2006
- [17] Kakizawa, Y., Shumway, R. H., Taniguchi M., *Discrimination and Clustering for Multivariate Time Series*, **Journal of the American Statistical Association**, Vol. 93, No. 441, 1998
- [18] Adeodato, P. J. L., Arnaud, A. L., et al., *The Role of Temporal Feature Extraction and Bagging of MLP Neural Networks for Solving the WCCI 2008 Ford Classification Challenge*.
- [19] Kohonen, *Self-organized maps*, 2a. edição, Springer-Verlag, 1997.
- [20] Barreto, G. A., Araújo, A. F. R., Aplicação de rede auto-organizável de Kohonen na Predição de Séries Temporais Caóticas, Departamento de Engenharia elétrica (USP)
- [21] Liao, Warren T., *Clustering of time series data - A survey*, **Pattern Recognition** **38**, Elsevier, 2005.
- [22] Franco, L., *Improving Network Generalization through Selection of Examples*.
- [23] Zhang, G. P., *Neural Networks in Bussiness Forecasting*.
- [24] Hornik, K., *Approximation capabilities of multilayer feedforward networks*. **Neural Networks**. Vol **4**. 1991.
- [25] Hornik, K., Stinchcombe, M., White H., *Multilayer feedforward networks are universal approximators*. **Neural Networks**. Vol **2**. 1989.
- [26] Santos, R. T., Nievola, J. C., Freitas, A. A., Lopes, H. S., Extração de regras de redes neurais via algoritmos genéticos. Anais do IV congresso brasileiro de redes neurais. ITA. 1999.
- [27] Symon Haykin, *Redes Neurais - Princípio e prática*, 2ª edição, Editora Bookmam, 2001