

# NEVE: Um Comitê de Classificadores Neuro-Evolucionário para Aprendizagem Adaptativa

Tatiana Escovedo, André Vargas Abs da Cruz, Marley Vellasco, Adriano S. Koshiyama

Departamento de Engenharia Elétrica  
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)  
Rio de Janeiro – Brasil  
E-mail: {tatiana, andrev, marley, adriano}@ele.puc-rio.br

*Abstract*—This work describes the use of a quantum-inspired evolutionary algorithm (QIEA-R) to construct a weighted ensemble of neural network classifiers for adaptive learning in concept drift problems. The proposed algorithm, named NEVE (meaning Neuro-EVolutionary Ensemble), uses the QIEA-R to train the neural networks and also to determine the best weights for each classifier belonging to the C when a new block of data arrives. After running eight simulations using two different datasets and performing two different analysis of the results, we show that NEVE is able to learn the data set and to quickly respond to any drifts on the underlying data, indicating that our model can be a good alternative to address concept drift problems.

*Keywords*— adaptive learning, concept drift, neuro-evolutionary ensemble, quantum-inspired evolution

## I. INTRODUÇÃO

A capacidade de um classificador aprender a partir de dados incrementais e atualizados extraídos de um ambiente não estacionário representa um desafio para o campo da inteligência computacional. Adicionalmente, o uso de redes neurais como classificadores torna o problema ainda mais complexo, pois redes neurais geralmente são vistas como ferramentas que devem ser treinadas novamente usando todo o conjunto de instâncias aprendidas até então quando um novo bloco de dados se torna disponível.

A fim de lidar com esse tipo de problema, um classificador deve, idealmente, ser capaz de [1]:

- Monitorar e detectar qualquer tipo de mudança na distribuição da base de dados;
- Aprender com novos dados sem a necessidade de apresentar novamente todo o conjunto de dados para o classificador;
- Ajustar os seus próprios parâmetros, a fim de tratar as alterações detectadas nos dados;
- Esquecer o que foi aprendido quando esse conhecimento não for mais útil para a classificação de novas instâncias.

Todas essas habilidades buscam, de uma forma ou de outra, tratar um fenômeno chamado de concept drift [2], [3]. Este fenômeno define conjuntos de dados que sofrem alterações ao

longo do tempo como, por exemplo, quando há mudança na relevância das variáveis, ou quando a média e a variância de séries temporais sofre alterações. Muitas abordagens foram concebidas a fim de contemplar algumas ou todas as capacidades acima mencionadas. A mais simples delas consiste em utilizar uma janela deslizante sobre os dados de entrada e treinar o classificador com os delimitados dados por esta janela [4]. Outra abordagem consiste em detectar desvios e, em seguida, ajustar o classificador de acordo com esta variação.

Uma abordagem mais bem sucedida consiste em utilizar um conjunto de classificadores (comitê de classificadores). Este tipo de abordagem utiliza um grupo de diferentes classificadores, a fim de ser capaz de controlar as alterações no ambiente. Vários modelos diferentes de comitê de classificadores foram propostos na literatura [5, 6, 7]:

- Comitês de classificadores que criam novos classificadores para cada novo bloco de dados e classificadores com peso de acordo com a precisão em dados recentes;
- Comitês de classificadores não ponderados que podem lidar com novos dados que pertencem a um conceito diferente dos dados de treinamento mais recentes;
- Comitês de classificadores que são capazes de descartar classificadores à medida que se tornem imprecisos, ou quando uma mudança de conceito é detectada.

A maioria dos modelos que utilizam comitês de classificadores ponderados determinam os pesos para cada classificador usando algum conjunto de heurísticas relacionadas a quantidade de erros do classificador quando trabalha com os dados mais recentes [3]. Embora a princípio qualquer classificador possa ser usado para construir os comitês de classificadores, os mais frequentemente utilizados são árvores de decisão, redes neurais e Naïve Bayes [8].

Neste trabalho, apresentamos uma abordagem baseada em redes neurais treinadas por algoritmos evolucionários com inspiração quântica. Algoritmos evolucionários com inspiração quântica [9-13] são uma classe de algoritmos de estimação de distribuição que apresentam - segundo várias referências - um melhor desempenho para otimização combinatória e numérica em comparação com seus algoritmos genéticos canônicos homólogos. O algoritmo evolucionário com inspiração quântica para otimização numérica (AEIQ-R), tem

demonstrado bom desempenho quando usado para treinar uma rede neural para séries temporais de previsão e problemas de aprendizagem por reforço. O treinamento uma rede neural utilizando um algoritmo evolutivo pode ser benéfico especialmente em problemas de aprendizagem por reforço, nos quais a geração de instâncias de dados de entrada-saída não é simples ou até mesmo impossível. Além disso, usando um algoritmo evolutivo para a formação de uma rede neural, pode-se eventualmente treinar arquiteturas complexas como redes com funções de ativação não contínuas e redes neurais recorrentes de uma forma direta, ou até mesmo definir a topologia da rede neural durante o treinamento [10, 14].

O AEIQ-R também foi usado para determinar os pesos de votação para cada classificador que faz parte do comitê de classificadores. Cada vez que um novo bloco de dados chega, um novo classificador é treinado para este bloco e todos os pesos são otimizados para que o comitê de classificadores melhore o seu desempenho na classificação deste novo conjunto de dados.

Desta forma, apresentamos uma nova abordagem para aprendizagem adaptativa, que consiste em um comitê de classificadores de redes neurais treinado por um algoritmo evolucionário com inspiração quântica, a fim de aprender conjuntos de dados (possivelmente com concept drifts) de forma incremental. Este modelo foi denominado NEVE, do acrônimo em inglês (Neuro-Evolutionary Ensemble). Para avaliar a performance e a acurácia do modelo, serão utilizados 2 diferentes conjuntos de dados reais para a realização de diversos experimentos, variando as configurações do ensemble a fim de analisar como elas influenciam o resultado final.

Este artigo está organizado em quatro seções adicionais. A seção 2 apresentará alguns conceitos teóricos relacionados a concept drift. A seção 3, por sua vez, detalhará o modelo proposto e a seção 4 apresentará e discutirá os resultados dos experimentos realizados. Finalmente, a seção 5 concluirá este trabalho e apresentará alguns possíveis trabalhos futuros.

## II. O MODELO PROPOSTO

### A. Modelo Neuro-Evolutivo com Inspiração Quântica

Neuro-evolução é uma forma de aprendizado de máquina que usa algoritmos evolucionários para treinar redes neurais artificiais. Este tipo de modelo é particularmente interessante para problemas de aprendizagem por reforço, nos quais a disponibilidade de pares de entrada-saída é geralmente difícil ou impossível de se obter e, além disso, o desempenho da rede é avaliado medindo-se diretamente o seu desempenho em uma tarefa pré-definida. Como o treinamento dos pesos em uma rede neural é um problema de otimização global não linear, é possível minimizar a função de erro usando uma estratégia de algoritmo evolucionário.

O algoritmo evolucionário com inspiração quântica, apresentado na subseção anterior, é uma classe de algoritmos de estimação de distribuição que apresenta rápida convergência e geralmente provê uma solução melhor, com menos avaliações, do que os algoritmos genéticos tradicionais [6, 8]. Neste modelo, os genes com inspiração quântica são representados pelas funções densidade de probabilidade que

são usadas para gerar indivíduos clássicos através de um operador de observação. Após serem observados, os indivíduos clássicos são avaliados como nos algoritmos genéticos tradicionais e, usando a informação de fitness, um conjunto de operadores com inspiração quântica é aplicado aos indivíduos quânticos, a fim de atualizar a informação guardada por eles de forma que nas próximas gerações, melhores indivíduos terão mais chance de serem selecionados. Mais detalhes de como este método funciona podem ser encontrados em [9-13].

Com base neste algoritmo, o modelo neuro-evolucionário com inspiração quântica proposto consiste em uma rede neural (um multi-layer perceptron - MLP) e uma população de indivíduos, cada um deles codificando uma configuração diferente de pesos e biases para a rede neural. Se a rede neural tem  $n_i$  entradas,  $n_h$  processadores escondidos e  $n_o$  saídas, então o número total de pesos e biases que devem ser codificados pelos genes dos indivíduos é dado por:

$$t_p = n_i * n_h + n_h * n_h + n_h * n_o + n_o \quad (2)$$

que considera as conexões entre as entradas e os processadores escondidos, as conexões entre os processadores escondidos e os processadores de saída e os biases para os processadores escondidos e de saída.

O processo de treinamento ocorre através da construção de um MLP para cada indivíduo clássico usando os genes deste indivíduo como pesos e biases. Em seguida, o conjunto completo de dados de treinamento (ou o conjunto de tarefas a serem executadas) é apresentado para o MLP e o erro médio para este conjunto de dados é calculado para cada MLP. Este erro médio é usado como avaliação para cada indivíduo associado a este MLP, o que permite ao algoritmo evolucionário se ajustar e passar para a próxima geração, quando o processo completo será repetido até que uma determinada condição de parada seja satisfeita.

Esta subseção apresentou o modelo neuro-evolutivo com inspiração quântica que servirá de base para o algoritmo proposto neste trabalho. A subseção a seguir apresentará o algoritmo NEVE, que consiste em um comitê de classificadores neuro-evolucionário.

### B. NEVE: Comitê de Classificadores Neuro-Evolucionário

Para algumas aplicações, tais como as que utilizam data streams, a estratégia de modelos mais simples é a mais adequada porque pode não haver tempo para executar e atualizar um comitê de classificadores. Porém, quanto o tempo não é a maior preocupação, mas se requer alta acurácia, um comitê de classificadores é a solução natural. O maior potencial desta estratégia para detecção de drifts é a habilidade de utilizar diferentes formas de detecção e fontes de informação para lidar com os diversos tipos e magnitudes de mudanças [7].

Um dos maiores problemas na utilização de um classificador simples (uma rede neural, por exemplo) para tratar problemas de Concept drift é que quando o classificador aprende um conjunto de dados e precisamos que ela aprenda um novo conjunto, ela precisará ser treinada novamente com

todos os dados, ou então, ela vai “esquecer” tudo que já tinha aprendido. Já com o comitê de classificadores, não há a necessidade de treiná-lo novamente, uma vez que ele consegue “reter” o conhecimento prévio e ainda aprender novos dados.

Desta forma, para ser capaz de aprender à medida que novos blocos de dados chegam, implementamos um comitê de classificadores de redes neurais que é treinado pelo algoritmo evolucionário. Esta abordagem torna o comitê de classificadores útil para aprendizado por reforço online, por exemplo. O algoritmo funciona como demonstrado na figura 1 e cada passo é descrito de forma detalhada nos próximos parágrafos.

1. Criar um comitê de classificadores vazio P
2. Definir o tamanho do comitê s
3. Para cada bloco de dados  $D_i$ ,  $i = 1, 2, 3, \dots, m$  faça
  - 3.1. Treine o classificador usando o AEIQ-R e MLP e calcule seu erro  $E'$  referente ao bloco de dados
  - 3.2. Se o comitê está cheio (número de classificadores = s) então
    - i) Calcule o erro de classificação  $E_j$  para cada classificador  $c_j$  do comitê
    - ii) Se ( $E' > \max(E_j)$ )
      - a. Substitua o classificador com  $\max(E_j)$  pelo novo classificador
  - 3.3. Senão
    - i) Adicione o novo classificador ao comitê
  - 3.4. Evolua os pesos de votação  $w_j$  para cada classificador usando o último bloco de dados  $D_i$

Fig. 1. Algoritmo neuro-evolutivo de treinamento do comitê de classificadores

No passo 1, é criado um comitê de classificadores vazio com tamanho pré-definido igual a s. Quando o primeiro bloco de dados é recebido, uma rede neural é treinada usando o AEIQ-R até que uma condição de parada seja atingida (por exemplo, o número de gerações evolucionárias ou um limiar de erro). Se o número de classificadores no comitê de classificadores é menor que “s”, então nós simplesmente adicionamos este novo classificador no comitê de classificadores. Isto dá ao comitê de classificadores a habilidade de aprender o novo conjunto de dados sem precisar analisar os dados antigos. Se o comitê de classificadores já estiver cheio, avaliamos cada classificador com o novo conjunto de dados e removemos aquele que tiver a maior taxa de erro (incluindo o novo, ou seja, o novo classificador só se tornará parte do comitê se a sua taxa de erro for menor que a taxa de erro de um dos classificadores existentes no comitê). Isto dá ao comitê de classificadores a habilidade de esquecer os dados que não são mais necessários.

Finalmente, nós usamos o AEIQ-R novamente, desta vez para gerar um peso de votação para cada classificador. A otimização dos pesos permite ao comitê de classificadores se adaptar facilmente a mudanças bruscas nos dados, através da atribuição de pesos maiores aos classificadores mais bem adaptados aos conceitos correntes que governam os dados. O cromossomo que codifica os pesos tem apenas um gene para cada peso de votação, e a população é avaliada usando o erro do classificador como função de avaliação. É importante notar que quando os primeiros s-1 blocos de dados são recebidos, o tamanho do comitê de classificadores é menor que o seu tamanho final e consequentemente, o tamanho do cromossomo também é menor. A partir do s-ésimo bloco de dados, o tamanho do cromossomo será constante e igual a s.

Neste trabalho, usamos apenas classificadores binários, mas não há perda de generalização e o algoritmo pode ser usado com qualquer número de classes. Para o classificador binário, nós discretizamos as saídas das redes neurais como “1” ou “-1” e o processo de votação para cada amostra de dados é feita através da soma das NN saídas multiplicada pelo seu peso de votação. Em outras palavras, a saída do comitê de classificadores para uma amostra k do i-ésimo bloco de dados é dada por:

$$P(D_{ik}) = \sum_{j=0}^s w_j c_j(D_{ik}) \quad (3)$$

onde  $P(D_{ik})$  é a saída do comitê de classificadores para a amostra  $D_{ik}$ ,  $w_j$  é o peso do j-ésimo classificador e  $c_j(D_{ik})$  é a saída do j-ésimo classificador para esta amostra. Se  $P(D_{ik}) < 0$ , assumimos que a saída do comitê de classificadores é “-1”. Se  $P(D_{ik}) > 0$ , assumimos que a saída do comitê de classificadores é “1”. Se  $P(D_{ik}) = 0$ , escolhemos uma classe randomicamente.

### III. RESULTADOS EXPERIMENTAIS

Esta seção descreve e avalia os resultados obtidos nos diversos experimentos realizados usando NEVE, objetivando em determinar a melhor configuração dos parâmetros: tamanho do comitê e número de neurônios, a partir de análises estatísticas.

Além disso, são detalhadas as bases de dados usadas para aprendizado não estacionário (SEA Concepts e Nebraska) e os detalhes de execução para cada base de dados. Após isso, as análises estatísticas conduzidas para identificar qual a melhor configuração do NEVE para aquela específica base de dados. Por fim, é apresentado e discutido os resultados obtidos.

#### A. Descrição das Bases de Dados

De forma a checar a habilidade do classificador em aprender em ambientes não estacionários, foram usados duas diferentes bases de dados (SEA Concepts e Nebraska, também usadas em [6]) com as quais foram efetuadas diferentes simulações e cenários.

O SEA Concepts foi criado por Street [15] e tem sido usado por diferentes algoritmos como um benchmark para estudar modelos que identificam concept drift. A base de dados, disponível em [16], é caracterizada por extensos períodos sem grandes alterações no ambiente, mas com ocasionais e bruscas alterações nas regiões de discriminação das classes (concept drift). A base de dados consiste em 50000 padrões criados aleatoriamente em um plano tri-dimensional (3 atributos). Os padrões para cada atributo estão no intervalo [0, 10], mas somente dois dos três atributos são relevantes para a determinação da classe de saída. Esses padrões são divididos em quarto blocos, com diferentes conceitos. As classes são atribuídas a cada padrão baseado na soma dos atributos relevantes (no caso dois atributos); um determinado padrão pertence a classe 1 se a soma dos atributos relevantes for maior que um limiar; e pertence a classe 2 caso contrário. Esse limiar é alterado ao longo do tempo, em intervalos regulares, cujos

valores são (8→9→7.5→9.5), criando portanto abruptas alterações nas regiões de discriminação das classes.

A base de dados Nebraska, também disponível em [16], apresenta uma compilação de medidas de clima a partir de 9000 estações climáticas espalhadas mundialmente pela U.S. National Oceanic and Atmospheric Administration desde 1930s, providenciando uma grande quantidade e períodos climáticos. Medições diárias incluem uma variedade de atributos (temperatura, pressão, velocidade do vento, etc.) e indicadores para precipitação e outras informações e eventos relativos ao clima. Como essa base de dados possui um tamanho significativo, foram escolhidos os dados provenientes da subestação Offutt Air Force Base em Bellevue, Nebraska, para esse experimento, devido à larga quantidade de padrões (50 anos de dados diários (1949–1999)) e diversos atributos climáticos, perfazendo essa base de dados como um problema real de classificação/previsão com concept drift. Os rótulos das classes são baseadas em uma indicação binária, onde para cada dia (padrão) há a possibilidade de chover, ou não chover. Cada microbase de treinamento consistem em 30 dias, com os subsequentes 30 dias como base de teste. Esta base de dados inclui 583 microbases de treinamento e teste consecutivamente.

### B. Detalhes da Execução

Todas as execuções começam em  $t=0$  e terminam quando  $T$  consecutivos blocos de padrões são apresentados para treinamento e teste, sendo que cada bloco pode tomar diferentes cenários de concept drift, com taxas e naturezas desconhecidas. O valor de  $T$  determina o número de passos (ou amostragens) tomado a partir dos dados durante o período de transição dos blocos. Um  $T$  grande corresponde a um nível de concept drift baixo, enquanto que um  $T$  pequeno está ligado à alta taxa de concept drift, devido ao algoritmo ter menos passos para reefetuar a aprendizagem.

Para cada configuração foi usada uma topologia fixa para a rede neural, consistindo em três entradas para o SEA Concepts e oito entradas para o Nebraska, representando os atributos de entrada de cada base de dados. Para ambas as bases de dados foram usadas uma única saída, e para a camada intermediária o número de neurônios foi variado. Cada neurônio usa uma função de ativação do tipo tangente hiperbólica, tal que se a saída for negativa esta é discretizada para “-1” e quando positiva para “1”. O algoritmo evolucionário treina cada rede neural por 100 gerações. A população quântica e clássica possui 10 e 20 indivíduos, respectivamente. A taxa de cruzamento é 0:9 (ver [10, 11] para detalhes sobre os parâmetros). Os mesmos parâmetros são usados para evoluir os pesos para cada classificador.

Os pesos e interceptos da rede neural, e os pesos do comitê são permitidos variar entre -1 e 1, devido a estes valores possibilitarem melhores resultados em algumas execuções prévias.

Na base de dados SEA Concepts, quatro configurações foram executadas 10 vezes cada uma, usando 50 blocos de dados de tamanho 250 (resultando em 12500 padrões para treinamento e 12500 para a fase de teste). Para a base de dados Nebraska, quatro configurações foram estabelecidas e

executadas 10 vezes cada uma, nas quais o classificador devia prever se chovia ou não 30 dias à frente, baseado nas 30 amostras passadas. Estas 30 subsequentes do teste, tornavam-se próximo passo a base de treinamento, até que os 583 blocos de dados serem usados.

Para ambas as bases de dados, de forma a sistematicamente avaliar a influencia de cada parâmetro no resultado para cada base de dados, foram variados o número de neurônios na camada escondida (5 ou 10) e o tamanho do comitê (5 ou 10).

### C. Delineamento das Análises

Baseado nas subseções passadas, após executar 10 vezes cada uma das quatro configurações (totalizando em 40 simulações) usando a base de dados SEA e em seguida, executando 10 vezes cada uma das quatro configurações (totalizando em 40 simulações) com o Nebraska, foram realizados duas diferentes análises de modo a encontrar um conjunto de parâmetros, que viabilizassem uma taxa de erro baixa na fase de teste para ambas as bases de dados. Para tanto, foi realizada um desenho factorial [17] de modo a verificar não somente o efeito da variação de um parâmetro no erro produzido pelo classificador, mas também a interação simultânea das configurações desses parâmetros. A Tabela I exhibe os fatores (número de neurônios na camada escondida e tamanho do comitê) com diferentes valores (5 e 10) e a saída (média do erro para as 10 execuções na fase de teste) para cada configuração.

TABELA I. RESULTADOS PARA AS BASES DE DADOS SEA E NEBRASKA.

SEA				
Número de neurônios	Tamanho do comitê	Config	Erro na fase de teste	
			Média	Desv. Pad.
10	10	A	24,99%	0,17%
5	5	B	24,88%	0,19%
10	5	C	25,06%	0,21%
5	10	D	24,75%	0,17%
Nebraska dataset				
Número de neurônios	Tamanho do comitê	Config	Erro na fase de teste	
			Média	Desv. Pad.
10	10	E	32,30%	0,48%
5	5	F	32,85%	0,43%
10	5	G	33,04%	0,37%
5	10	H	32,10%	0,46%

De forma a avaliar qual configuração proporcionou um menor erro significativo, torna-se necessário múltiplas comparações entre cada configuração. Se para cada configuração distinta (para cada base de dados) fosse decidido aplicar o teste-t [18], por exemplo, seria necessário realizar seis comparações ao todo, e, portanto, a probabilidade de que todas as estivessem corretas é substancialmente afetada. Dado isso, para realizar a comparação entre todas as configurações simultaneamente foi ajustado uma Análise de Variância (ANOVA) com um fator para cada base de dados [18], descrito por:

$$Y_{ij} = \mu + CF_j + \varepsilon_{ij}; \varepsilon_{ij} \sim N(0, \sigma^2) \quad (3)$$

onde  $Y_{ij}$  é o  $i$ -ésimo erro da  $j$ -ésima configuração,  $\mu$  é a média global,  $CF_j$  é a configuração executada com  $j$ -níveis ( $j = 1, 2, 3, 4$ ) para cada base de dados (A, B, C e D, e E, F, G, H, para a SEA e Nebraska, respectivamente) e  $\varepsilon_{ij}$  é o resíduo do modelo, com distribuição Normal com média zero e variância constante e finita. Se  $CF_j$  é estatisticamente significativo, então existe alguma configuração (ou algumas) que apresentou um erro médio menor do que as demais. Para verificar qual foi esta configuração foi aplicado o teste de Tukey [19].

Um segundo experimento foi realizado, visando a investigação da relação entre o número de neurônios e tamanho do comitê nos resultados auferidos pelo classificador, unindo os resultados para ambas as bases de dados. Para tanto, foi ajustada uma ANOVA para dois fatores [18], dado por:

$$Z_{ijk} = \mu + NN_j + ES_k + \varepsilon_{ijk}; \varepsilon_{ijk} \sim N(0, \sigma^2) \quad (4)$$

onde  $Z_{ijk}$  é o  $i$ -ésimo erro normalizado, na fase de teste, para a  $j$ -ésima e  $k$ -ésima configuração,  $NN_j$  é o número de neurônios com  $j$  níveis ( $j = 1, 2$ ),  $ES_k$  é o tamanho do comitê com  $k$  níveis ( $k = 1, 2$ ) e  $\varepsilon_{ijk}$  é o resíduo do modelo. Deve se notar que o  $Y_{ijk}$  (erro na fase de teste desnormalizado) foi normalizado para  $Z_{ijk}$ , com o objetivo de tornar os erros comensuráveis para base de dados distintas.

Para ambos os experimentos, foi aplicado o teste de Normalidade de Shapiro-Wilk nos resíduos [20] e o teste de homogeneidade das variâncias de Bartlett [18], como avaliação dos pressupostos da ANOVA. As análises estatísticas foram conduzidas no pacote estatístico R [21], admitindo um nível de significância de 5%.

#### D. Resultados do Primeiro Experimento

Os principais resultados do primeiro experimento são descritos na tabela II.

TABELA II. RESULTADOS DA ANOVA PARA SEA E NEBRASKA.

ANOVA – SEA		
Método	Estatística de Teste	p-valor
Teste de Bartlett	0,4443	0,9309
Fator: Config	5,4360	0,0035
Shapiro-Wilk	0,9260	0,2137
ANOVA – Nebraska		
Método	Estatística de Teste	p-valor
Bartlett's test	0,6537	0,8840
Fator: Config	11,5900	< 0,0001
Shapiro-Wilk	0,9708	0,3803

Analisando os resultados dispostos na tabela III, em ambas as bases de dados às variâncias dos erros na fase de teste são homogêneas (teste de Bartlett's,  $p$ -valor > 0,05). Após verificar

esse pressuposto inicial, foi ajustado o modelo ANOVA para um fator. Para ambas as bases de dados, alguma configuração (A, B, C e D para SEA, e E, F, G e H for Nebraska) demonstrou um erro médio diferente ( $p$ -value < 0,05). Em adicional, para ambos os modelos, os resíduos seguem uma distribuição Normal (Shapiro-Wilk,  $p$ -valor > 0,05).

De modo a identificar qual configuração obteve, em media, um menor erro que as demais, foi realizado o teste de Tukey, cujos resultados são exibidos na tabela III.

TABELA III. TESTE DE TUKEY PARA AS CONFIGURAÇÕES DE SEA E NEBRASKA.

SEA			Nebraska		
Config	Diferença de média	p-valor	config	Diferença de média	p-valor
A-B	0,11%	0,5665	E-F	-0,55%	<b>0,0145</b>
A-C	-0,07%	0,8018	E-G	-0,74%	<b>0,0013</b>
A-D	0,24%	<b>0,0317</b>	E-H	0,20%	0,8849
B-C	-0,18%	0,1399	F-G	-0,19%	0,7434
B-D	0,13%	0,4010	F-H	0,75%	<b>0,0014</b>
C-D	0,31%	<b>0,0029</b>	G-H	0,94%	<b>0,0001</b>

Verifica-se que na base de dados SEA a configuração D auferiu um erro significativamente menor quando comparado às configurações A e C, e estatisticamente não diferente da configuração B. Na base de dados Nebraska a configuração E e H obtiveram métricas de erro substancialmente menores que a F e G. De fato, foi escolhido as configurações D e H, para a SEA e Nebraska. Essa escolha deveu-se a dois critérios: menor erro médio e custo computacional menor para treinar esses modelos.

Baseado nesses resultados, a próxima seção realiza uma análise de sensibilidade do algoritmo a alterações nos valores dos parâmetros. Essas informações podem ajudar a compreender o motivo das configurações D e H terem sido mais competitivas quando comparadas as demais analisadas.

#### E. Resultados do Segundo Experimento

Na tabela IV são apresentados os principais resultados da ANOVA e do teste de Tukey, de forma a analisar a influência de dois fatores: número de neurônios e tamanho do comitê. Os resultados indicam tanto o número de neurônios ( $p$ -valor = 0,0006) e o tamanho do comitê ( $p$ -valor < 0,0001) afetam significativamente na variabilidade das métricas de erro. Ainda, observa-se que o tamanho do comitê tende a afetar em dobro (20,50%) os resultados do classificador, quando comparado a variações no número de neurônios (11,18%).

O teste de Tukey (tabela IV) demonstra que a diferença entre as médias de erro entre os níveis no fator tamanho do comitê (-0,42%) é maior do que a de número de neurônios (0,20%). Ainda, os resultados apontam uma tendência de redução do número de neurônios na camada escondida, e aumento do tamanho do comitê podem acarretar resultados ainda melhores ao encontrado.

Essas configurações encontradas podem ser usadas nas próximas análises, como parametrizações iniciais do sistema. Porém, deve se atentar que grande parte da variabilidade destes

resultados (68,32%) não é explicada somente pelos dois fatores estudados: número de neurônios e tamanho do comitê. Portanto, novas análises semelhantes a essa devem ser conduzidas, mas com o foco nos parâmetros do AEIQ-R.

TABELA IV. RESULTADOS DO SEGUNDO EXPERIMENTO.

ANOVA			
Método	Varição Explicada	Estatística de Teste	p-valor
Teste de Bartlett	-	0,0027	0,9589
Shapiro-Wilk	-	0,9869	0,5916
Fator: Número de neurônios	11,18%	12,5900	0,0006
Fator: Tamanho do comitê	20,50%	23,1100	< 0,0001
Resíduo	68,32%	-	-
Teste de Tukey			
Fator	Nível	Diferença de média	p-valor
Número de neurônios	10-5	0,20%	0,0006
Tamanho do comitê	10-5	-0,42%	< 0,0001

#### IV. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou um modelo que usa um comitê de classificadores de redes neurais treinadas por um algoritmo evolucionário com inspiração quântica para aprender de forma incremental usando blocos de dados (possivelmente com concept drifts). Verificamos a habilidade do modelo através de 4 diferentes simulações para cada um dos 2 diferentes conjuntos de dados e analisando os resultados.

Como verificado nos resultados, encontramos uma boa configuração para ambos os conjuntos de dados e demonstramos como o número de neurônios e o tamanho do ensemble afetam o erro médio produzido pelo modelo. Como demonstrado nos resultados, o tamanho do ensemble tem um impacto quase duas vezes maior nos resultados do NEVE do que o número de neurônios. Há uma indicação que configurações com um número menor de neurônios com um tamanho de ensemble maior parecem produzir resultados melhores em média. Assim, um grande ensemble composto por pequenas redes neurais aumenta a importância da estratégia de voto executada pelo AEIQ-R.

Analisar a performance e a acurácia do NEVE utilizando outros conjuntos de dados reais é uma possibilidade de trabalho futuro, embora não seja fácil determinar se um conjunto de dados real tem algum tipo de mudança significativa nos dados. De qualquer forma, é possível introduzir estas mudanças em qualquer conjunto de dados real de forma artificial.

Apesar de o algoritmo NEVE ter demonstrado performance satisfatória para os conjuntos de dados utilizados nos experimentos deste trabalho, recomenda-se fortemente que sejam realizados novos testes, com diferentes configurações, a

fim de que se possa confirmar ou não os resultados aqui apresentados. Também pretendemos no futuro dar continuidade neste trabalho, analisando outras abordagens existentes, como [5], [22] e [23], a fim de realizar novos experimentos de forma comparativa a estes algoritmos.

#### REFERÊNCIAS

- [1] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data", *Machine Learning*, vol. 1, no. 3, pp. 317–354, 1986.
- [2] A. Tsymbal, "The problem of concept drift: Definitions and related work", *Tech. Rep.*, 2004.
- [3] M. T. Karnick, M. Ahiskali, M. Muhlbaier, and R. Polikar, "Learning concept drift in nonstationary environments using an ensemble of classifiers based approach," in *IJCNN*, pp. 3455–3462, 2008.
- [4] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," In *Proc. Of The 2001 Acm Sigkdd Intl. Conf. On Knowledge Discovery And Data Mining*, pp. 97–106, 2001.
- [5] Ryan Elwell, Robi Polikar: Incremental Learning of Concept drift in Nonstationary Environments. *IEEE Transactions on Neural Networks* 22(10): 1517-1531, 2011.
- [6] L. I. Kuncheva, "Classifier ensemble for changing environments," in *Multiple Classifier Systems*, vol. 3077. New York: Springer-Verlag, 2004.
- [7] L. I. Kuncheva, "Classifier ensemble for detecting concept change in streaming data: Overview and perspectives," in *Proc. Eur. Conf. Artif. Intell.*, pp. 5–10, 2008.
- [8] N. C. Oza, "Online Ensemble Learning," *Dissertation*, University of California, Berkeley, 2001.
- [9] A. V. Abs da Cruz, M. M. B. R. Vellasco, and M. A. C. Pacheco, "Quantum-inspired evolutionary algorithms for numerical optimization problems," in *Proceedings of the IEEE World Conference in Computational Intelligence*, 2006.
- [10] A. V. Abs da Cruz, "Algoritmos evolutivos com inspiração quântica para otimização de problemas com representação numérica," Ph.D. dissertation, Pontifical Catholic University – Rio de Janeiro, 2007.
- [11] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, 2002.
- [12] K.-H. Han and J.-H. Kim, "On setting the parameters of qea for practical applications: Some guidelines based on empirical evidence," in *GECCO*, pp. 427–428, 2003.
- [13] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, H<sub>g</sub> gate, and two-phase scheme," *IEEE Trans. Evolutionary Computation*, vol. 8, no. 2, pp. 156–169, 2004.
- [14] W. Liefijm, "Heterogeneous neuro-evolutionary specialization of collective rover behaviors," 2008.
- [15] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, pp. 377–382, 2001.
- [16] R. Polikar and R. Elwell. Benchmark Datasets for Evaluating Concept drift/NSE Algorithms. At: <http://users.rowan.edu/~polikar/research/NSE>. Last access at December 2012.
- [17] Bailey, R. A. *Design of Comparative Experiments*. Cambridge University Press, 2008.
- [18] Montgomery, Douglas C. *Design and analysis of experiments*. Wiley, 2008.
- [19] Yandell, B. S. *Practical Data Analysis for Designed Experiments*. Chapman & Hall, 1997.
- [20] Patrick Royston. An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31, 115–124, 1982.
- [21] R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. Download at: [www.r-project.org](http://www.r-project.org), 2012.
- [22] J. Kolter and M. Maloof. "Dynamic weighted majority: An ensemble method for drifting concepts". *Journal of Machine Learning Research*, 8, pp. 2755-2790, 2007.
- [23] K. Jackowski. "Fixed-size ensemble classifier system evolutionarily adapted to a recurring context with an unlimited pool of classifiers." *Pattern Analysis and Applications*, 2013.