PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Um Método de Detecção de Mudanças em Ambientes Dinâmicos Baseado na Combinação de Confiança na Decisão e Distribuição dos Dados.

Ana Paula Vieira Mota

Manaus - AM
05 de julho de 2013

Ana Paula Vieira Mota

Um Método de Detecção de Mudanças em Ambientes Dinâmicos Baseado na Combinação de Confiança na Decisão e Distribuição dos Dados.

Dissertação apresentada ao Programa de Pós-graduação em Informática da Universidade Federal do Amazonas como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Visão Computacional e Robótica

Orientadora:

Prof[a]. Dra. Eulanda Miranda dos Santos

Manaus - AM

05 de julho de 2013

Ficha Catalográfica
(Catalogação realizada pela Biblioteca Central da UFAM)

Ana Paula Vieira Mota

Um Método de Detecção de Mudanças em Ambientes Dinâmicos Baseado na Combinação de Confiança na Decisão e Distribuição dos Dados.

Dissertação apresentada ao Programa de Pós-graduação em Informática da Universidade Federal do Amazonas como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Visão Computacional e Robótica

Aprovado em __ de xxxx de 2013.

Banca Examinadora

_____
Profª. Dra. Eulanda Miranda dos Santos (Orientadora)
Universidade Federal do Amazonas

_____
Prof. Dr. José Luiz de Souza Pio
Universidade Federal do Amazonas

_____
Prof. Dr. José Reginaldo Hughes Carvalho
Universidade Federal do Amazonas

Lista de Figuras

Lista de Tabelas

1. Introdução

Sistemas de classificação são algoritmos projetados para aprender a distinguir padrões a partir de informações extraídas do ambiente. Normalmente, o processo que envolve o projeto e a implementação de um classificador é dividido em duas etapas [9]: (1) treinamento, o sistema aprende a classificar as diferentes classes existentes; e (2) operação, o sistema identifica informações desconhecidas.

Nesse processo, portanto, assume-se que a distribuição das informações no ambiente em que o sistema irá operar permanecerá igual à distribuição representada no conjunto de treinamento, isto é, assume-se que o ambiente é estático. Entretanto, a literatura mostra que os problemas do mundo real evoluem, assim como suas características, pois ocorrem mudanças com o passar do tempo no cenário de aplicação do sistema [9].

Existem diversos exemplos de problemas com ambiente dinâmico, tais como: monitoramento de fraude em cartão de crédito [9], detecção de *spam* [21] e outros problemas de detecção de intrusos em redes, monitoramento ambiental, dentre outros. Para Baena et. al. [1], considerar que os dados de treinamento são gerados a partir de uma fonte estacionária é uma hipótese falsa nesses problemas dinâmicos, especialmente quando os dados de treinamento são coletados durante um longo período de tempo. Nesse tipo de problema, os conceitos (*concepts*, em Inglês) não permanecem estáveis à medida que o tempo evolui.

Áreas de pesquisa como aprendizagem de máquina, reconhecimento de padrões, estatística e mineração de dados, têm concentrado esforços na proposta de métodos de detecção e reação a mudanças. Em função da área de pesquisa, problemas com ambientes dinâmicos são chamados de ambientes não estacionários, *concept drift*, etc. enquanto que a tarefa de detecção de mudanças em ambientes dinâmicos é conhecida como detecção de novidades, detecção de *drift*, etc. Em mineração de dados, por exemplo, há problemas em que os dados são organizados na forma de fluxos, ao invés de bancos estáticos, e é bastante incomum que os conceitos e distribuições de dados permaneçam estáveis com o passar do tempo [20].

Dentre as estratégias de reação a mudanças em ambientes dinâmicos, destaca-se o uso de múltiplos classificadores, pois é difícil para um único classificador responder a vários tipos de *concept drift* [15]. Entretanto, antes que o sistema possa reagir a mudanças, é necessário que estas sejam detectadas. Esta pesquisa tem como objetivo propor um método de detecção de mudanças em ambientes dinâmicos.

1.1. Definição do Problema

Os ambientes podem apresentar diferentes tipos de mudanças. Kuncheva [9, 10]

classifica-as em quatro tipos: ruídos, *blip* (evento raro), gradual e abrupta. A detecção de ruídos pode ser realizada através de mecanismos de filtragem para serem removidos antes que o dado seja submetido à classificação. Ou ainda, podem ser incluídos durante o treinamento dos classificadores para tornar o sistema mais efetivo.

Um evento raro é considerado um *outlier,* ou seja, quando um dado está fora do padrão dos demais dados conhecidos da classe em questão. Detecção de intrusão e de fraudes [11] são exemplos de aplicações em que ocorrem eventos raros. Nesse contexto, estruturas conhecidas de detecção de *outlier* podem ser usadas para que um sistema de classificação detecte esse tipo de alteração.

As mudanças do tipo gradual e abrupta são consideradas mudanças constantes. Segundo Kuncheva [9], se as mudanças são graduais, podemos utilizar uma janela de tempo móvel sobre os dados para calcular algum tipo de métrica comparativa entre as janelas, como por exemplo, a taxa de acerto do classificador, para que as mudanças sejam detectadas. Se as mudanças são abruptas, podemos optar por usar uma estratégia de classificação estática. Quando for detectada a mudança, através do monitoramento de taxa de acerto, por exemplo, a reação será re-treinar o classificador.

1.2. Justificativa

Um classificador quando se destina a uma aplicação real deve ser equipado com um mecanismo para se adaptar às alterações no ambiente [9]. Conforme mencionado anteriormente, áreas de pesquisa como reconhecimento de padrões e aprendizagem de máquina buscam por soluções para problemas com ambientes dinâmicos. Esta busca é inspirada pelo próprio comportamento do cérebro humano, pois Gavrilov e Lee [5] relatam que a capacidade de reconhecimento invariante de objetos em ambientes dinâmicos é uma das mais importantes capacidades do cérebro natural.

A detecção de *concept drift* ajuda o sistema a responder rapidamente a mudanças e a permanecer produzindo elevadas taxas de acerto através de métodos de reação. A detecção de mudanças é relevante para problemas reais, como, monitoramento na área da biomedicina e processos industriais, detecção de falhas e diagnósticos, e segurança de sistemas complexos [15]. Além disso, em muitos domínios do mundo real o conceito de interesse pode depender de algum contexto escondido, que não é fornecido explicitamente nos dados de treinamento [20].

Para Ahiskali et. al. [7], um algoritmo típico de problemas com ambientes dinâmicos precisa implementar um ou mais dos seguintes procedimentos: detecção de mudança; detecção da magnitude da mudança; aprender o conceito novo; e esquecer o que não é mais relevante. Portanto, a detecção das mudanças é o primeiro passo envolvido nesse processo e é fundamental

para o sucesso das demais etapas.

Os métodos de detecção de mudança podem ser implícitos e explícitos. Nos métodos implícitos, o sistema não se preocupa em perceber as mudanças, os mecanismos de detecção estão implícitos no método de reação. O sistema é constantemente atualizado, ocorrendo ou não mudança. Já nos métodos explícitos, o sistema se preocupa em primeiramente perceber a mudança, para após adaptar-se ao novo conceito. Embora os métodos implícitos sejam mais simples, em termos de modelagem, eles apresentam um custo computacional elevado porque necessitam que o sistema seja atualizado mesmo sem ocorrência de mudanças. Além de serem pouco úteis em aplicações do mundo real, como problemas com fluxo contínuo de dados que surgem e devem ser descartados rapidamente.

Apesar dessas desvantagens, a maioria dos trabalhos encontrados na literatura utiliza como solução a abordagem de detecção implícita de mudanças. As soluções variam entre métodos de seleção de instância, conjuntos de classificadores e métodos dinâmicos de combinação. Dentre os métodos de detecção explícita reportados na literatura, a maioria utiliza como métrica para monitorar as mudanças no ambiente a taxa de acerto do sistema [1] [2] [3]. Entretanto, monitorar a taxa de acerto de um sistema de classificação em problemas do mundo real nem sempre é viável, pois para isso é necessário que o sistema receba um *feedback* a respeito de suas decisões de classificação. Dessa forma, a dependência de um *feedback* que pode existir ou não, ou ainda, ser muito demorado, pode inviabilizar o uso do sistema.

Portanto, para que sistemas de classificação automática sejam aplicados em problemas reais com ambiente dinâmico, é necessária a propostas de novas soluções com detecção explícita de mudanças e que não sejam baseadas na taxa de acerto do classificador. Uma alternativa pode ser o monitoramento dos dados. Nesse caso, quando os novos dados apresentam diferença significativa em relação aos dados aprendidos, isso pode indicar mudança no ambiente. Pechenizkiy et. al. [18] apresentam uma proposta que verifica se existem diferenças estatisticamente significativas entre as médias de cada possível janela de dados.

Outra informação que pode ser relevante para indicar a ocorrência de mudanças no ambiente é a confiança do classificador na decisão a ser tomada. A confiança de classificação é como estamos chamando a probabilidade de um classificador k rotular uma amostra de teste x como pertencendo a uma classe c do conjunto de classes de um problema de classificação. Na prática, trata-se da probabilidade *a posteriori* do classificador atribuída à classe c, dada a amostra x. Nossa hipótese é que a queda significativa da confiança do classificador pode indicar uma mudança no ambiente em que este opera.

1.3. Objetivos da Pesquisa

O objetivo geral desta proposta é o desenvolvimento de um método de detecção de mudanças em ambientes dinâmicos baseado na combinação do monitoramento da confiança de classificação e da variação dos dados.

Os objetivos específicos são os seguintes:

a) Demonstrar que o método proposto é eficiente na detecção de mudanças tanto em problemas sintéticos quanto em problemas reais.

b) Demonstrar experimentalmente que o método proposto de detecção de mudanças em ambientes dinâmicos detecta mais rapidamente as mudanças, quando comparado aos métodos baseados no monitoramento da taxa de erro.

1.4. Organização da Dissertação

Esta proposta está organizada em cinco capítulos. Neste primeiro capítulo foi contextualizado o problema de mudanças em ambientes dinâmicos. O segundo capítulo apresenta a fundamentação teórica com os conceitos importantes para a definição de detecção de mudanças e de comportamento na distribuição das informações que podem ocorrer em aplicações do mundo real. O terceiro capítulo descreve os principais trabalhos relacionados. No quarto capítulo está descrita a metodologia proposta. O quinto capítulo apresenta informações sobre experimentos realizados. Por fim, no sexto capítulo é apresentado o cronograma das atividades desenvolvidas.

.

2. Conceitos Básicos

Este capítulo descreve os principais conceitos relacionados com esta proposta de trabalho. São apresentados conceitos de ambientes dinâmicos, algoritmos de classificação e *concept drift*.

2.1. Ambientes Dinâmicos

Ambientes Dinâmicos são ambientes que estão em constantes mudanças. O problema de detecção de intrusão, por exemplo, é considerado dinâmico porque novos tipos de métodos de invasão podem alterar o ambiente regularmente. Há ainda a possibilidade de perda de características (*missing features*) devido a ruídos, falhas mecânicas, entre outras causas, que podem gerar mudanças no cenário de aplicação do sistema. A detecção de mudanças é uma maneira de identificar a mudança de conceito de aprendizagem, pois existem vários exemplos de problemas reais onde a detecção de mudanças é relevante tais como modelagem de usuários, monitoramento em biomedicina e processos industriais, detecção de falhas e diagnósticos, dentre outros.

2.2. Algoritmos de Classificação

Apesar de diferentes classes de algoritmos de classificação terem sido desenvolvidas e aplicadas com sucesso em uma ampla gama de domínios do mundo real, os problemas investigados são massivamente estáticos. Para Oza e Tumer [16] garantir que o algoritmo de classificação particular corresponda às propriedades dos dados é crucial no fornecimento de resultados que atendam as necessidades do domínio de aplicação particular. Portanto, é necessário que algoritmos de classificação sejam dotados de capacidade de detecção e de reação a mudanças ocorridas no ambiente de aplicação.

2.3. Concept Drift

Na literatura de Aprendizagem de Máquina as alterações no ambiente são denominadas de *concept drift*. Para Narasimhamurthy e Kuncheva [14], o termo *concept* é usado para definir coisas ligeiramente diferentes, é por vezes utilizado para se referir à classe de interesse. Elwell e Polikar [2] definem *concept drift* como uma mudança nas classes definidas ao longo do tempo. Para Kurlej e Wozniak [13], de um modo geral *concept drift* pode ser causado por alterações nas probabilidades *a priori* das classes ou nas distribuições de probabilidade condicional das classes, sendo que pode-se distinguir as seguintes fontes deste fenômeno:

- Mudança prévia de probabilidade para a classe (probabilidade *a priori*);

- Mudança de distribuição de probabilidade de classe-condicional;
- Mudança de probabilidade *a posteriori*.

Para Tsymbal [20] três abordagens para lidar com *concept drift* podem ser destacadas, são elas:

1. Seleção de instância – o objetivo é selecionar instâncias relevantes para o *concept* atual, ou seja, consiste em generalizar a partir de uma janela que se move sobre instâncias recém-chegadas e utiliza os *concepts* aprendidos para a previsão no futuro imediato;
2. Peso da instância – usa a capacidade de alguns algoritmos de aprendizagem para processar os pesos das instâncias, as quais podem ser ponderadas de acordo com a sua idade e sua competência no que diz respeito ao *concept* atual;
3. Conjuntos de classificadores – mantém as previsões que são combinadas através de um formulário de votação, usa alguns critérios para excluir dinamicamente, reativar ou criar novos membros do conjunto, que normalmente são baseados em coerência.

No próximo capítulo são discutidos detalhes de métodos propostos na literatura para detectar mudanças em ambientes dinâmicos. Serão abordados métodos de detecção implícita e explícita.

3. Trabalhos Relacionados

Na literatura encontram-se propostas de soluções para o problema de classificação em ambientes dinâmicos que podem ser agrupadas nas categorias apresentadas neste capítulo.

3.1. Classificador Individual versus Conjuntos de Classificadores

As abordagens baseadas em classificadores individuais usam normalmente métodos de detecção independentes do algoritmo de classificação. Um exemplo é o método de Klinkenberg e Joachims [8] que utiliza SVM (*Support Vector Machines*). Por outro lado, as estratégias baseadas em conjuntos de classificadores são muito utilizadas na literatura.

A técnica de conjuntos de classificadores tornou-se uma estratégia dominante em muitas áreas de aplicação como aprendizagem de máquina, reconhecimento de padrões e *data mining*. A razão desse sucesso são os estudos experimentais e teóricos que mostram que conjuntos de classificadores podem ter taxas de acerto superiores às taxas obtidas por classificadores individuais [19].

Uma vez criado um conjunto de classificadores, a estratégia mais comum é a combinação da saída de todos os classificadores. Essa estratégia assume que todos os membros são igualmente importantes e que apresentam decisões independentes. Por outro lado, a seleção de classificadores parte do princípio de que os membros do conjunto inicial são redundantes [12], ou seja, esta estratégia busca encontrar o mais relevante membro ou subconjunto de classificadores.

A seleção de classificadores pode ser estática ou dinâmica. A seleção estática define o melhor classificador ou subconjunto de classificadores na fase de treinamento [20]. Já a seleção dinâmica de classificadores baseia a escolha do melhor classificador durante a fase de uso do sistema.

3.2. Classificador Específico versus Classificador Livre

Os mecanismos de detecção e reação a mudanças só podem ser aplicados a um classificador modelo que deve ser exclusivo, ou seja, específico.

Qualquer classificador pode ser usado, pois a detecção de mudanças e a atualização do sistema não dependem do classificador. Normalmente, os critérios usados são baseados na taxa de acerto da classificação, e não do modelo. O algoritmo é independente de um modelo de classificador específico, e pode ser usado com qualquer classificador que se encaixe nas características do problema subjacente [7].

3.3. Detecção de Mudança Explícita (Passiva) versus Detecção de Mudança Implícita (Ativa)

Para tornar mais clara a classificação das abordagens estão destacadas a seguir as principais características dos dois grupos:

3.3.1. Abordagens Passivas de Detecção de Mudanças.

A detecção de mudanças implícitas equivale ao uso de uma estratégia de atualização do sistema independentemente de haver ou não ocorrido mudanças. Por exemplo, ao usar um conjunto de classificadores on-line, os pesos dos membros do conjunto são modificados após cada nova instância ser classificada pelo conjunto, com base nos registros da taxa de acerto recente dos membros do conjunto. Nessa situação, se não ocorrer mudanças no ambiente, a taxa de acerto da classificação será estável e os pesos irão convergir, por outro lado, se a mudança ocorrer, os pesos irão mudar sem a necessidade de detecção explícita. Portanto, as estratégias de detecção implícitas não se preocupam em detectar a mudança, mas sim, em manter o sistema constantemente atualizado, ocorrendo ou não mudança [6].

O trabalho de Widmer e Kubat [21] é baseado no uso de conjuntos de dados para representar o conhecimento do ambiente. Os conjuntos são denominados janelas. As janelas são utilizadas para monitorar os dados de entrada no intuito de detectar os dados mais relevantes ao contexto corrente. Quando o contexto é conhecido por variar em função do tempo, o sistema armazena apenas os exemplos mais recentes. Outros exemplos são adicionados à janela à medida que chegam, enquanto os exemplos mais antigos são eliminados. Ambas as ações (adição e eliminação) provocam modificações no conceito a ser aprendido, para mantê-lo consistente com os exemplos na janela. Podem ser usadas janelas de tempo de tamanho variável ou fixo. No caso mais simples, a janela é de tamanho fixo, e o exemplo mais antigo é descartado sempre que um novo dado aparecer.

A ideia de que quando novos exemplos chegam são inseridos no início da janela e um número correspondente de exemplos é removido do final da janela também é defendida por outros autores. Seguindo esta ideia, Hulten et. al. [6] relata que se a janela é muito pequena, isso pode resultar em exemplos suficientes para aprender o conceito de forma satisfatória, porém o custo computacional pode ser elevado.

Segundo Klinkenberg e Joachims [8], para janelas de tamanho fixo, a escolha de um tamanho de janela "ideal" é um compromisso entre adaptabilidade rápida (janela pequena) e boa generalização em fases sem mudança no conceito (janela grande). Uma das principais dificuldades deste tipo de solução é que os conceitos relevantes podem depender de algum

contexto escondido. Por exemplo, o termo clima ameno tem significados diferentes na Sibéria e na África Central, ou seja, mudanças no contexto escondido podem induzir mudanças mais ou menos relevantes para os conceitos alvos.

Outras abordagens de detecção passiva aplicam conjuntos de classificadores que usam regras de combinação dinâmicas e heurísticas de descarte de aprendizagem para conservar o sistema invariavelmente atualizado. O comportamento do conjunto para cada treinamento padrão é registrado como um vetor cujos elementos são as decisões dos classificadores do conjunto [12].

Um exemplo de método baseado em conjuntos de classificadores é o Learn++ proposto por Elwell e Polikar [3]. Learn++ originalmente gera um conjunto de classificadores que procura especificamente a informação mais relevante de cada conjunto de dados. Uma versão mais recente, Learn++.NSE [17] é um abordagem passiva para ambientes dinâmicos. Segundo os autores, o método pode acompanhar de perto e acomodar as mudanças, mesmo nos ambientes mais adversos, independentemente da sua taxa e tipo, ou adição ou remoção de classes do conceito aprendido.

Baena et. al. [1] aproveitaram as ideias de seleção de instância e atribuição de pesos e as utilizaram com conjuntos de classificadores. O método treina um novo classificador e adiciona-o ao conjunto. O conhecimento irrelevante é descartado removendo-se ou desativando-se do conjunto os membros que os detêm. A vantagem desse método é que o sistema não precisa ser atualizado o tempo todo.

A literatura mostra que conjuntos de classificadores apresentam elevado potencial para detectar alterações pois podem possuir modos de detecção diferentes e fontes de informações de diferentes tipos e magnitudes de mudanças [10]. Por outro lado, os conjuntos de classificadores podem elevar o custo computacional dos sistemas.

3.3.2. Abordagens Ativas de Detecção de Mudanças.

Após a detecção da mudança explícita uma ação é tomada, por exemplo, a criação de uma janela de dados mais recente para re-treinar o classificador. Esse processo é denominado detecção de mudanças explícitas [22].

Nesta categoria de métodos, há abordagens que observam o número de erros produzidos pelo modelo de aprendizado durante a classificação. Baena et. al. [1] relatam em seu trabalho propostas de outros autores, sendo um deles o método de detecção de mudança que utiliza uma distribuição binomial. Essa distribuição fornece a forma geral da probabilidade para a variável aleatória que representa o número de erros em uma base de n exemplos. A limitação dessas abordagens está na velocidade com que as mudanças ocorrem, se for uma mudança gradual muito lenta o sistema não consegue percebê-la.

Pechenizkiy et. al. [18] considera quatro abordagens diferentes para detectar os pontos de mudança usando testes estatísticos, e assim facilitar a modelagem das transições de um estado do sistema para outro. As duas primeiras abordagens são baseadas em controle estatístico de desempenho. Uma abordagem é não paramétrica, baseada no teste Mann-Whitney U, enquanto o outro método é baseado em um teste paramétrico sobre o desempenho dos modelos locais. As outras duas abordagens usam os dados brutos para detecção de mudanças. A primeira abordagem é ADWIN, que verifica se existem diferenças estatisticamente significativas entre as médias de cada possível divisão da seqüência. A segunda abordagem é baseada em heurística adaptada às peculiaridades do problema investigado.

Folino et. al. [4] relatam que mudanças podem causar uma grave deterioração do desempenho, e em tal caso, o método adotado deve ser capaz de se ajustar rapidamente às novas condições. Os dados que chegam na forma de fluxos contínuos, por exemplo, normalmente não são armazenados, pelo contrário, são processados assim que chegam e descartados logo em seguida.

Os trabalhos destacados nesta seção mostram que a maioria dos métodos encontrados na literatura pesquisada usa o monitoramento da taxa de acerto/erro para detectar as mudanças.

4. Metodologia

O projeto será desenvolvido em 10 (dez) etapas:

1ª. Etapa: Será realizado um levantamento bibliográfico, leitura dos principais métodos relacionados na literatura;

2ª. Etapa: Estudo e análise de ferramentas disponíveis para tratar o problema;

3ª. Etapa: Implementação de um método de detecção de mudança baseado no monitoramento da taxa de acerto para ser usado como *baseline*;

4ª. Etapa: Implementação de um método de detecção de mudança baseado na distribuição dos dados;

5ª. Etapa: Implementação de um método de detecção de mudança baseado na confiança de decisão;

6ª. Etapa: Combinação do método baseado na confiança de decisão com o método baseado na distribuição dos dados;

7ª. Etapa: Comparação entre os quatro métodos de detecção implementados;

8ª. Etapa: Análise dos resultados obtidos gerados pelo método proposto;

9ª. Etapa: Elaboração de artigos científicos com resultados parciais e finais, na conclusão do projeto;

10ª.Etapas: Elaboração da dissertação com a documentação completa incluindo detalhes da pesquisa e análise dos resultados.

5. Experimentos e Análise dos Resultados

Este capítulo descreve os experimentos realizados neste trabalho, bem como os resultados obtidos. Nosso objetivo com esses experimentos é compararmos o comportamento da taxa de acerto do classificador ao longo do tempo, em bases estáticas e bases dinâmicas. Para esse fim, utilizamos três bases de dados: (1) uma base estática; (2) uma base com mudanças graduais; e (3) uma base com mudanças abruptas. Todas as bases estão disponíveis publicamente para a realização de pesquisas. O classificador escolhido foi SVM. Na próxima seção são descritos detalhes das bases investigadas em nossos experimentos.

5.1. Definição da Base de Dados

As 03 (três) bases de dados investigadas são: DNA, CIRCLE e SINE. A base DNA, disponível publicamente em (http://www.sgi.com/tech/mlc/db/), é composta por 180 atributos e 3.186 amostras distribuídas em 03 (três) classes: (ie) "intron → exon boundary "; (ei) "exon → intron boundary"; e (n) nenhum dos dois. As amostras estão distribuídas entre as classes da seguinte forma:

- Classe 1: 767 amostras.
- Classe 2: 765 amostras.
- Classe 3: 1.654 amostras.

As bases CIRCLE e SINE encontram-se disponíveis em (http://www.cs.bham.ac.uk/~minkull/opensource/). A base SINE apresenta mudança de conceito do tipo abrupta, enquanto que a base CIRCLE, apresenta mudança de conceito do tipo gradual. As duas bases são compostas por 2.500 amostras, as quais estão distribuídas em 02 (duas) classes, sendo 1.000 amostras por classe nas primeiras 2.000 amostras, e 250 amostras por classe nas 500 amostras restantes. A mudança ocorre inicialmente a cada 1.000 amostras e posteriormente a cada 250 amostras.

5.2. Ferramentas Utilizadas

Conforme mencionado anteriormente, SVM foi utilizado como classificador nesses primeiros experimentos.

5.2.1. Biblioteca Libsvm

Considerando o fato de haver diversas bibliotecas com implementações de SVM, nós utilizamos a biblioteca Libsvm, que está disponível publicamente em (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). Essa é a biblioteca mais utilizada na literatura. Trata-se de uma ferramenta integrada que inclui diversas versões de SVM.

Com Libsvm, o parâmetro tipo de kernel de SVM foi ajustado para cada base de dados. Portanto, foram realizados testes com SVM com kernel do tipo linear, polinomial e função de base radial (RBF), e o parâmetro interno de cada kernel, exceto kernel linear, também foi ajustado.

Para observarmos o comportamento da taxa de acerto em função do tempo, nós dividimos as bases utilizando o método de validação cruzada (k-*fold cross-validation*). Esse método consiste na divisão do conjunto total de amostras em k janelas (*k-folds*) com tamanhos aproximadamente iguais. Em nossos experimentos, o valor escolhido para k foi 10.

Primeiramente nós dividimos cada base em 10 janelas. Em seguida, a primeira partição foi utilizada para treinar SVM. Por fim, as 09 janelas restantes foram utilizadas como bases de teste, isto é, a taxa de acerto do classificador foi calculada para cada janela. Essa estratégia foi utilizada para simularmos a evolução do tempo, ou seja, cada janela representa um bloco de dados que é submetido ao classificador, à medida que o tempo passa.

5.2.2. Toolbox PRTools do Ambiente Matlab

Utilizamos o ambiente Matlab especificamente uma Toolbox chamada PRTools (ferramentas para reconhecimento de padrões), nessa Toolbox tem uma série de classificadores como o SVM, conhecida no ambiente como SVC (*support vector classifier*).

Com o SVC, o parâmetro tipo de kernel foi ajustado para cada base de dados, CIRCLE e SINE, foram realizados testes com kernel do tipo linear, polinomial e função de base radial (RBF), e o parâmetro interno de cada kernel, exceto kernel linear, também foi ajustado. Para a base CIRCLE o kernel função de base radial foi o que demonstrou melhor comportamento na fase de treinamento com parâmetro 0.1 e regularização do parâmetro 100. Para a base SINE o kernel polinomial foi o que melhor se comportou na fase de treinamento com parâmetro 3 e regularização do parâmetro 100. Para a escolha do parâmetro e da regularização do parâmetro "C", variamos os mesmos em 0,1 de 1 a 1000, achando os mais adequados para os testes.

Para analisarmos o desempenho da taxa de acerto ao longo do tempo, nós dividimos as bases utilizando o método de validação cruzada (k-*fold cross-validation*), com tamanhos iguais. Em nossos experimentos, o valor escolhido para k foi 50.

Primeiro dividimos a base CIRCLE e SINE em 50 janelas cada. Posteriormente, a primeira janela foi utilizada para treinar SVC e as 49 janelas restantes para testes, cada uma com um cálculo da taxa de acerto do classificador. Cada janela representa um bloco de dados que é submetido ao classificador ao longo do tempo.

Os resultados obtidos nos experimentos são apresentados na próxima seção.

5.3. Resultados Obtidos

5.3.1. Comportamento Baseado no Monitoramento da Taxa de Acerto

Os parâmetros de SVM foram ajustados para cada base, e os resultados obtidos em cada base são mostrados nas figuras 1, 2, e 3. A Figura 1 mostra o comportamento da taxa de acerto em função do tempo na base DNA. O melhor parâmetro de kernel para essa base foi o kernel linear. É possível observar nessa figura que a taxa de acerto para uma base estática mantém-se estável, ou seja, não ocorre uma alteração significativa em função da mudança de amostras (nesse caso representando evolução do tempo). A taxa de acerto varia entre 90% e 82% de acerto. É importante destacar que esse comportamento já era esperado.

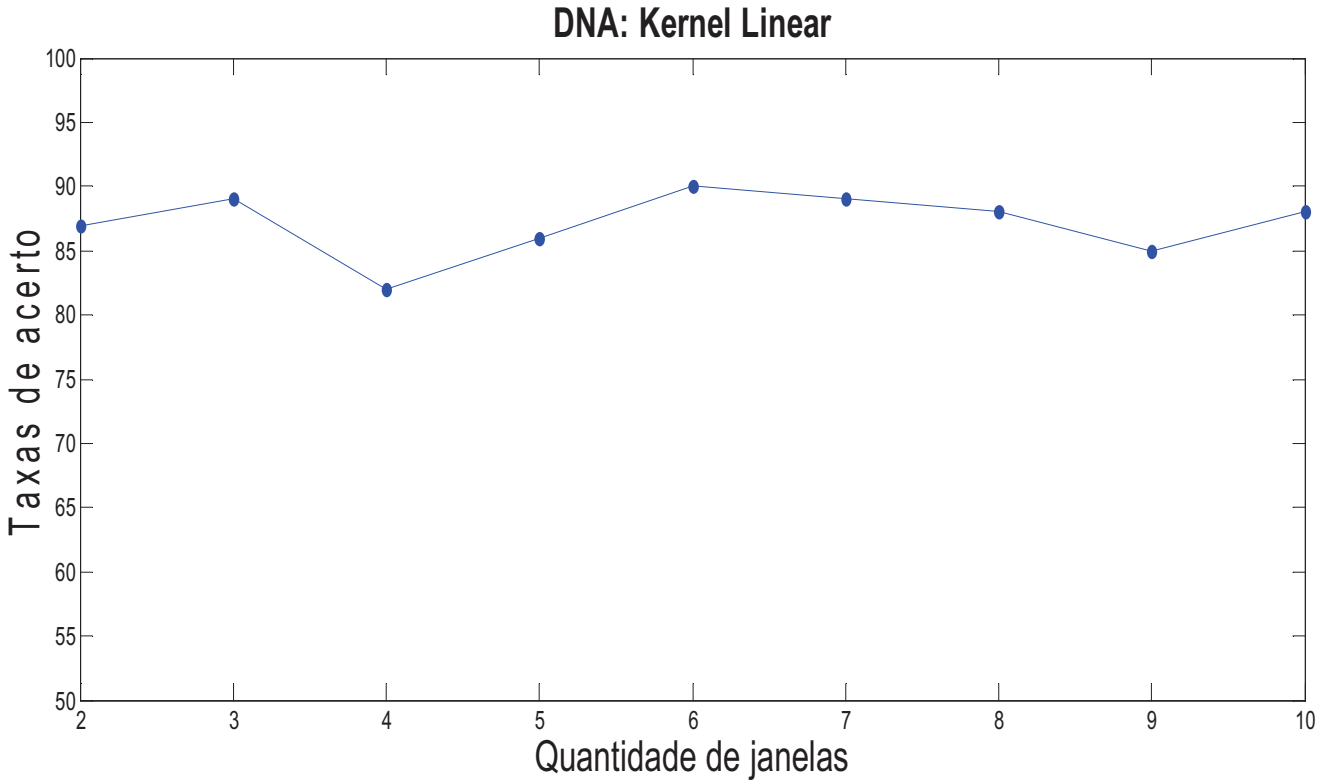


Figura 1: Comportamento da taxa de acerto de SVM na base estática DNA em função da mudança das janelas dos dados. SVM foi treinado com kernel linear.

O mesmo comportamento não é observado nas bases dinâmicas.

Na base CIRCLE (Figura 3) é visível a mudança de contexto ocorrida a partir da amostra 1.000. Porém, a queda na taxa de acerto é muito mais acentuada na base SINE por esta ser uma base com mudanças abruptas. A diferença entre a taxa de acerto obtida antes da mudança e a taxa de acerto obtida com as últimas amostras da base é de mais 45% na base SINE. Na base CIRCLE, essa diferença é menor que 25%.

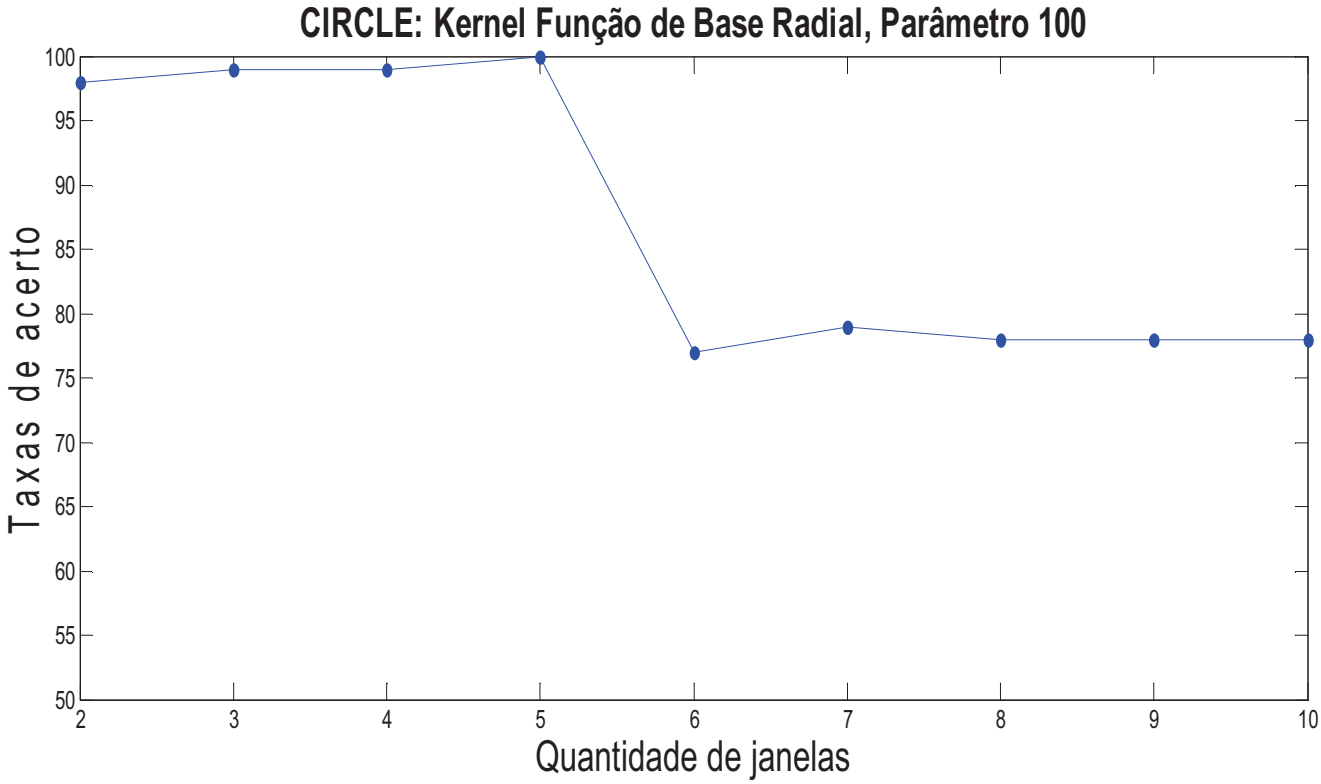


Figura 3: Comportamento da taxa de acerto de SVM na base com mudanças graduais CIRCLE, em função da mudança de partição dos dados. SVM foi treinado com kernel RBF.

Na base SINE também é visível a mudança de contexto ocorrida a partir da amostra 1.000. Na Figura 3 podemos observar a queda da taxa de acerto na base SINE, que é uma base com mudança abrupta. Ao compararmos esta curva com o comportamento verificado na base estática DNA, mostrado na Figura 1, é perceptível que a mudança ocorrida a partir da amostra 1.000, é bastante significativa.

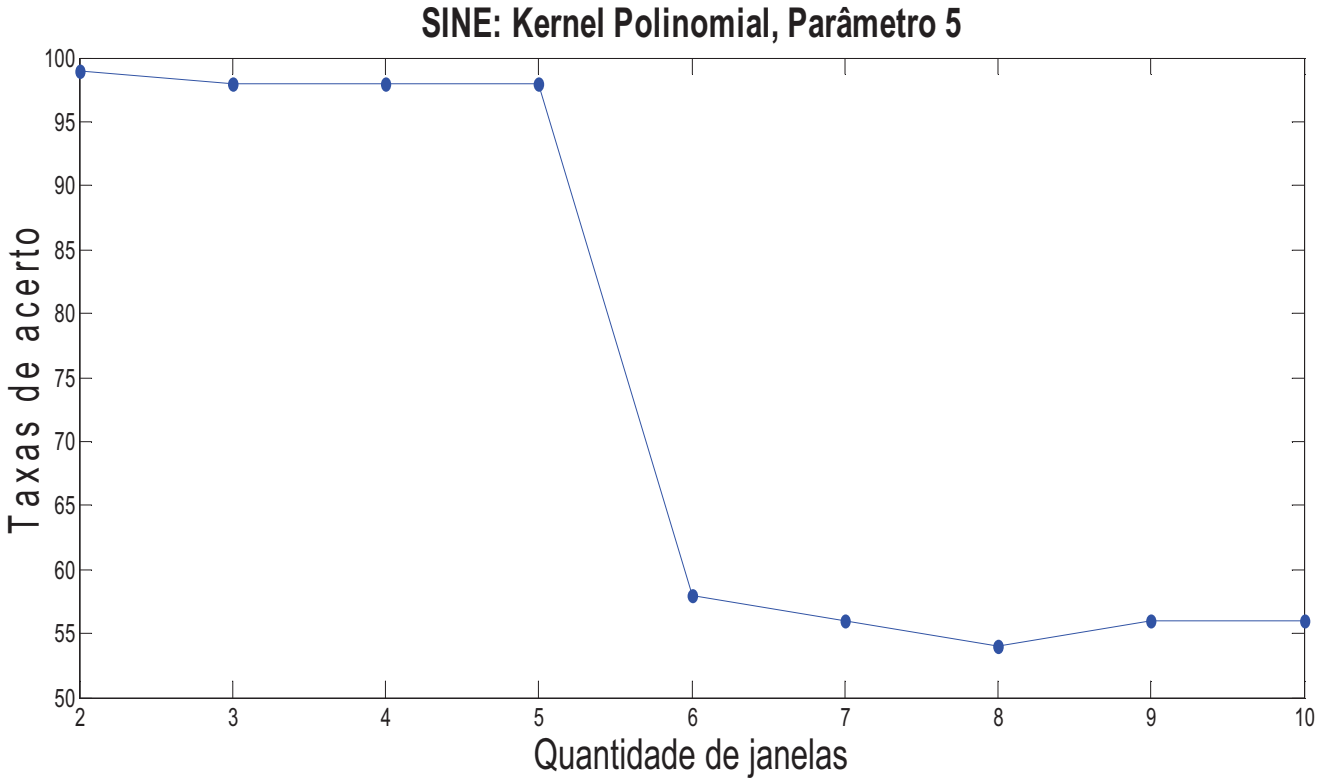


Figura 3: Comportamento da taxa de acerto de SVM na base com mudanças abruptas SINE, em função da mudança de partição dos dados. SVM foi treinado com kernel Polinomial 5.

Além desses resultados já esperados, nossos experimentos mostraram que a detecção de mudanças baseada no monitoramento da taxa de acerto é dependente da definição correta dos parâmetros do classificador. Se a taxa da acerto inicial for elevada, é maior a probabilidade de detecção de mudanças. Esse fato é confirmado nas figuras 4 e 5.

Por exemplo: se observamos os comportamentos da taxa de acerto de SVM nas figuras 4a e 4b, é perceptível que um método de detecção de mudanças seria capaz de perceber a mudança ocorrida a partir da amostra 1.000 mostrada na Figura 4a, porém, esse mesmo método provavelmente não conseguiria detectar essa mudança no contexto mostrado na Figura 4b. Os dois gráficos da Figura 4 mostram o comportamento da taxa de acerto, em função da evolução do tempo, na base SINE. Entretanto, a Figura 4a mostra as taxas de acerto obtidas por SVM com os melhores parâmetros definidos para essa base, enquanto que na Figura 4b, SVM foi treinado com os piores parâmetros detectados em nossos experimentos.

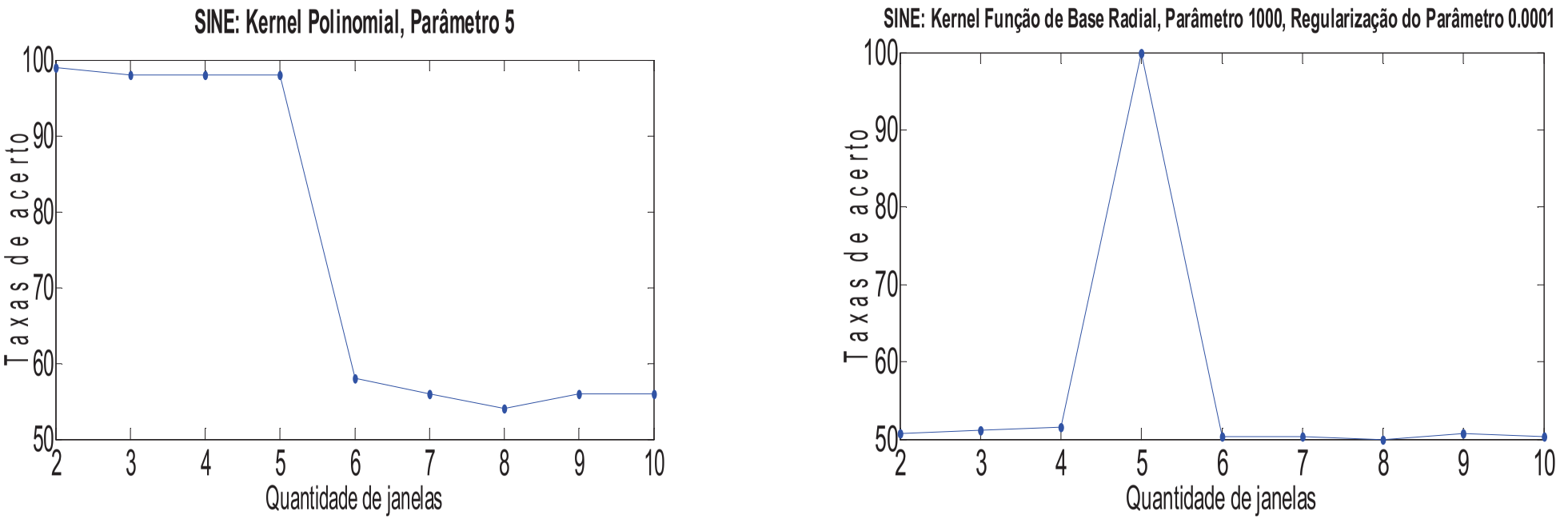


Figura 4: Comportamento da taxa de acerto de SVM na base SINE. Em (a) SVM foi treinado com os melhores parâmetros definidos para a base, kernel Polinomial grau 5. Em (b), SVM foi treinado com o pior parâmetro, kernel RBF.

O mesmo comportamento pode ser observado na base CIRCLE. Na Figura 5a observamos que o comportamento da taxa de acerto apresenta mudanças visivelmente mais significativas do que na curva mostrada na Figura 5b. Novamente, no primeiro caso os parâmetros do classificador foram ajustados, enquanto que no segundo caso, o classificador não teve os parâmetros bem definidos.

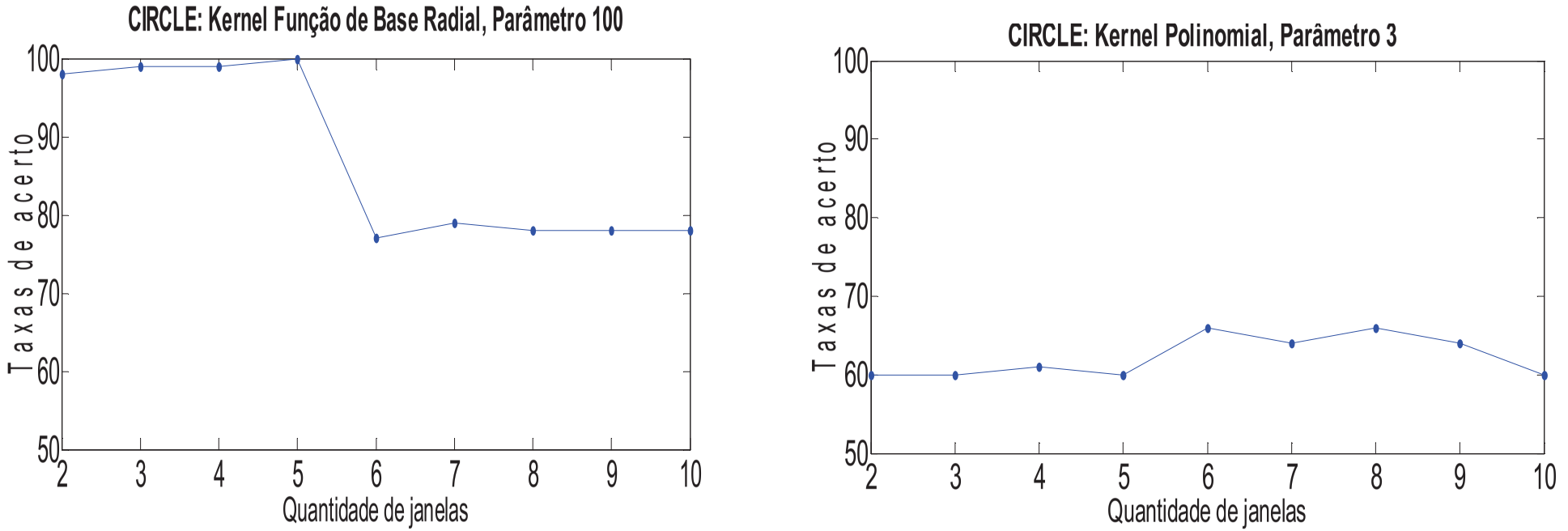


Figura 5: Comportamento da taxa de acerto de SVM na base CIRCLE. Em (a) SVM foi treinado com os melhores parâmetros definidos para a base, kernel RBF. Em (b), SVM foi treinado com o pior parâmetro, kernel polinomial grau 3.

Portanto, métodos de detecção de mudança baseados no monitoramento da taxa de acerto, além de serem difíceis de implementar em aplicações práticas, também são dependentes de um treinamento minucioso do classificador. Caso contrário, nenhuma mudança significativa será observada.

Os nossos próximos passos envolvem a implementação de um método baseado no monitoramento da taxa de acerto, para ser usado como *baseline*.

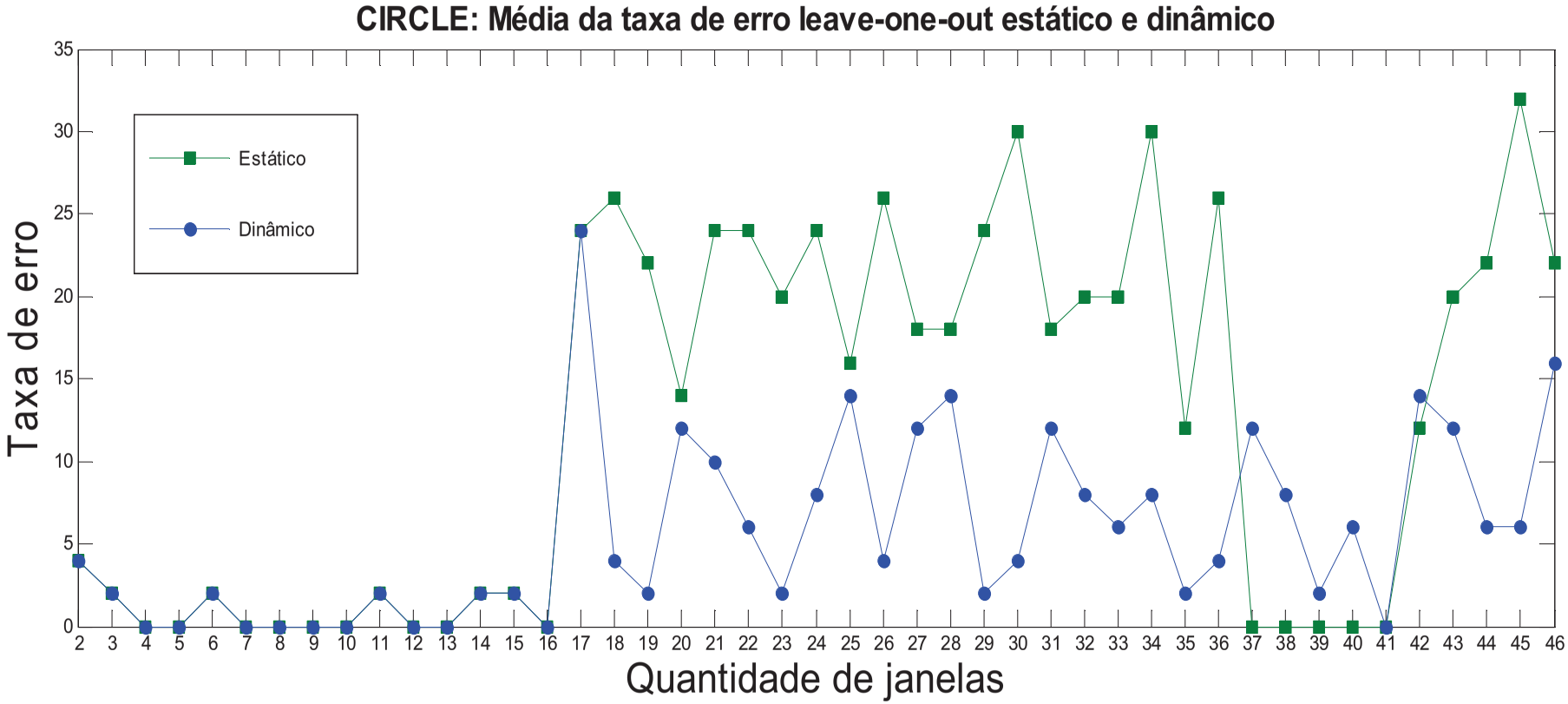
5.3.2. Comportamento ao Longo do Tempo considerando a Média de Classificação usando *leave-one-out*

Os parâmetros de SVC foram ajustados para as bases CIRCLE e SINE, e os resultados obtidos em cada base são mostrados nas figuras 6 e 7.

Para o experimento com *leave-one-out* dividimos cada base em 46 janelas, a primeira janela com 250 amostras reservadas para treinamento de SVM e as 45 janelas restantes possuíam 50 amostras cada, estas foram utilizadas como bases de teste, ou seja, foram calculadas para casa janela a média da taxa de erro e do desvio padrão do classificador utilizando o *leave-one-out*. Essa estratégia foi utilizada para simularmos a evolução ao longo do tempo.

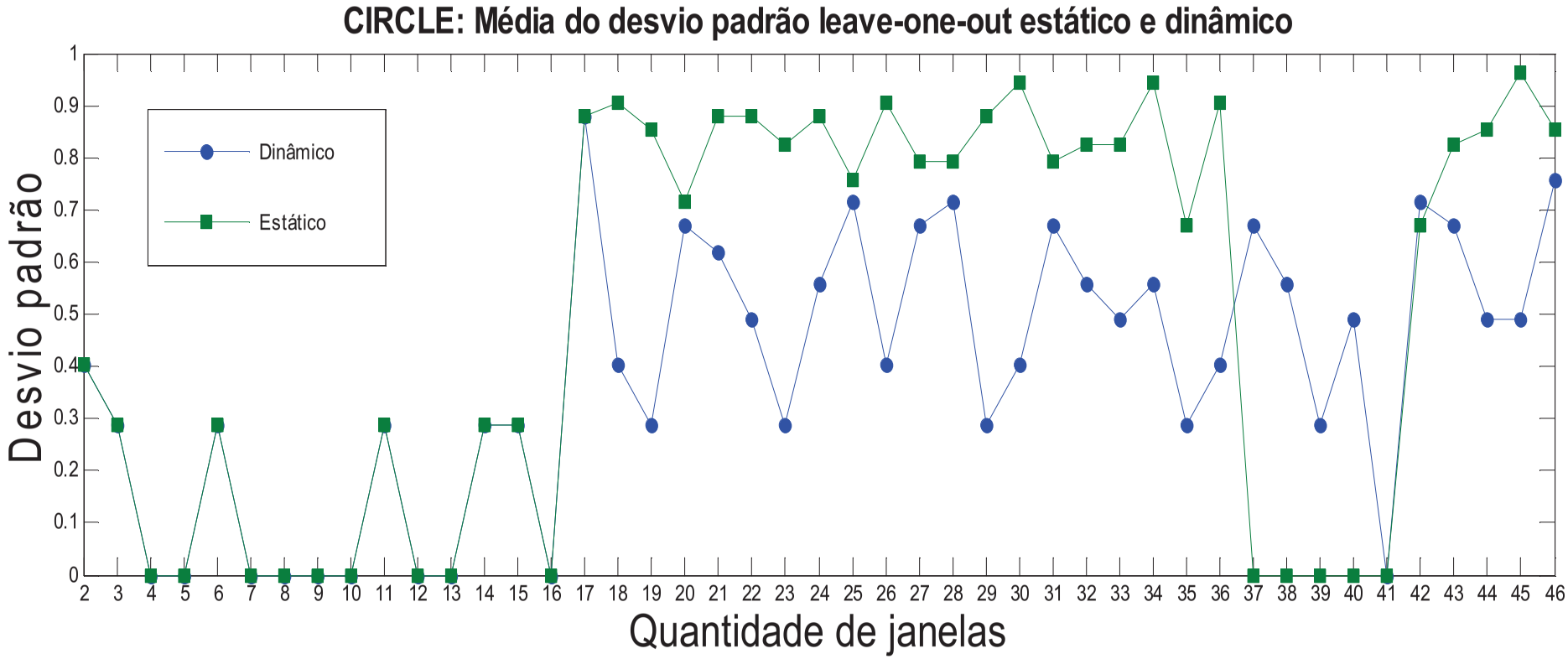
Os gráficos a seguir são resultados de experimentos estáticos e dinâmicos. O estático é a media da taxa de erro e desvio padrão de cada janela usando *leave-one-out* sem aplicação de método de detecção de mudança. O dinâmico é o resultado da media da taxa de erro e desvio padrão da cada janela usando *leave-one-out,* mas com aplicação do método de detecção de mudança.

O método de detecção implementado é um método de detecção à mudança em ambientes dinâmicos existente na literatura baseado no monitoramento da taxa de erro a ser utilizado como *baseline* neste trabalho.



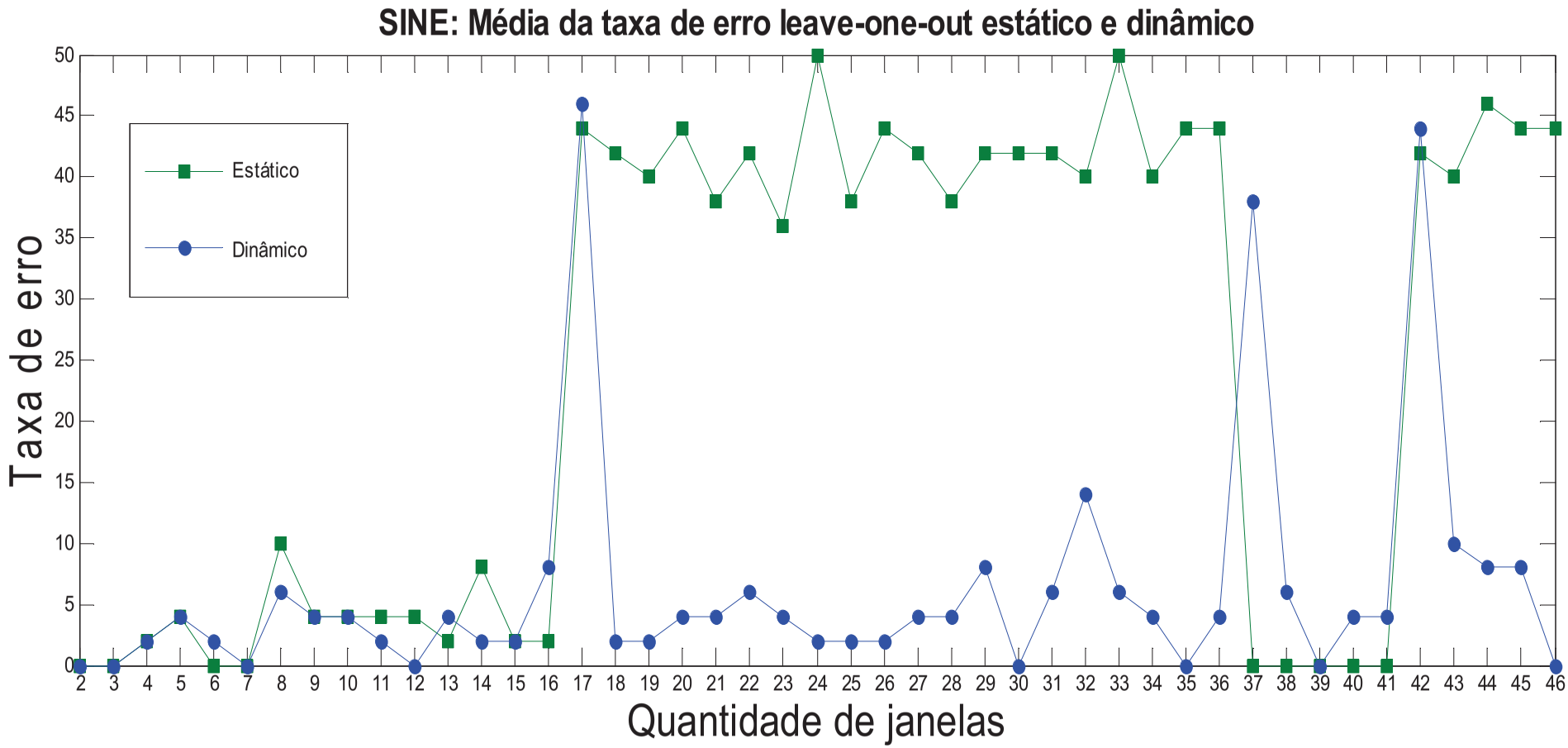
A Figura 6 mostra o comportamento da média da taxa de erro ao longo do tempo na base CIRCLE. O melhor parâmetro de kernel para essa base foi o kernel função de base radial, parâmetro 0.1 e da regularização do parâmetro "C" 100.

Observamos na figura 6, para o experimento estático, que até a janela 16 a media da taxa de erro permanece estável, porém houve uma mudança da janela 16 para a 17 mudando de 0% para 23% de erro permanecendo nessa variação até a janela 36, pois outra variação ocorre da janela 36 para 37 mudando de 23% para 0% de erro, e consequentemente o mesmo ocorre com a janela 41 para 42, ou seja, essas variações ocorrem justamente no momento em que ocorre a mudança de contexto (classe).



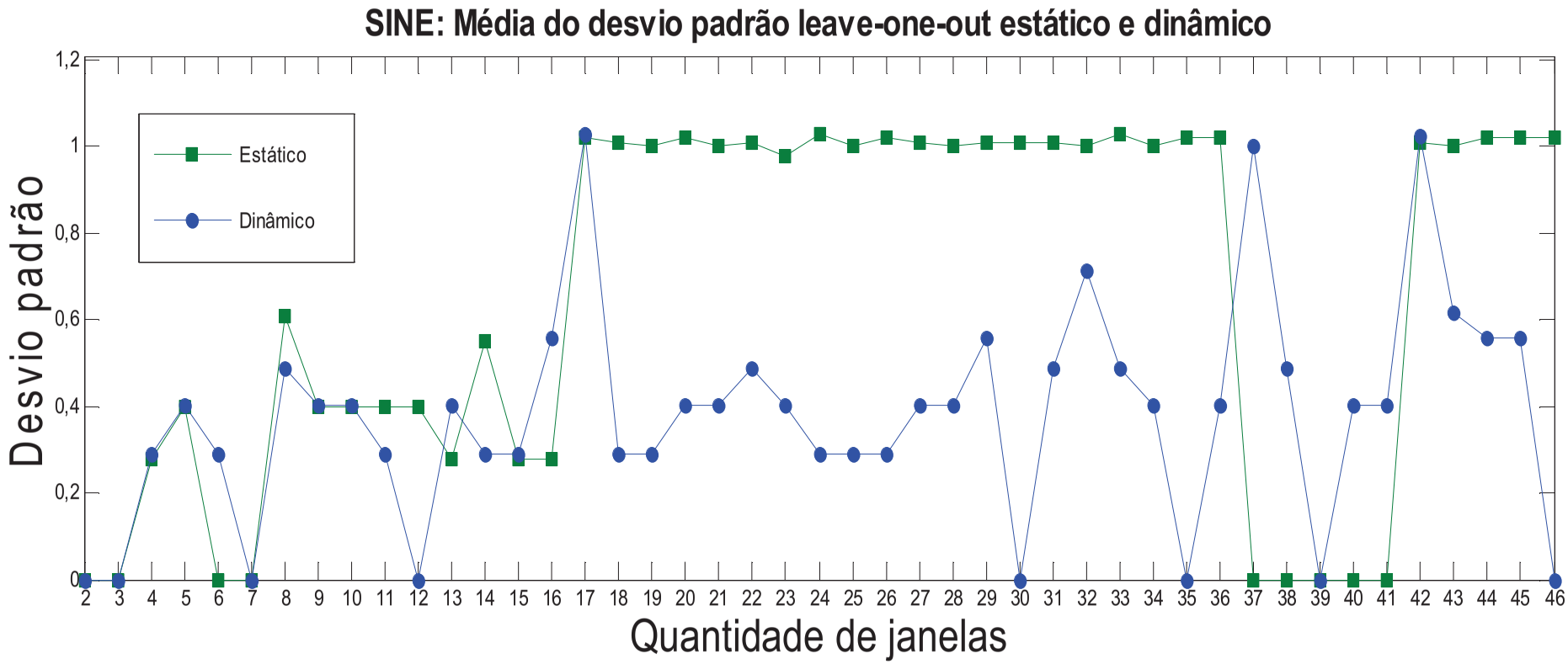
As mesmas observações realizadas na figura 6 podem ser observadas na figura 7, assim como o experimento dinâmico, pois nas janelas em que houve a mudanca ao ser detectada pelo método houve um novo treinamento, ocorrendo a queda da taxa.

A Figura 7 mostra o comportamento da média do desvio padrão ao longo do tempo na base CIRCLE. Com o seguinte parâmetro função de base radial, parâmetro 0.1 e da regularização do parâmetro “C” 100.



A Figura 8 mostra o comportamento da taxa de erro ao longo do tempo na base SINE. Com o seguinte parâmetro função polinomial, parâmetro 3 e da regularização do parâmetro “C” 100.

O comportamento observado na figura 8 é bastante perceptível referente ao experimento estático em relação ao dinâmico, é o que se espera de uma base com mudança abrupta. A queda na taxa de erro após o detecção de mudança e um novo treinamento é bastante significativa.



A Figura 9 mostra o comportamento da média da taxa do desvio padrão ao longo do tempo na base SINE. Parâmetro - função polinomial, parâmetro 3 e da regularização do parâmetro “C” 100.

6. Cronograma

Na Tabela 1 é apresentado o cronograma de atividades a serem realizadas.

Tabela 1- Cronograma de atividades a serem concluídas neste Mestrado.

Atividades	*julho*	*agosto*	*setembro*	*outubro*
4	*x*			
5	*x*	*x*		
6		*x*		
7		*x*	*x*	
8			*x*	*x*
9			*x*	*x*
10				*x*

6.1. Descrição das Atividades

O cronograma acima apresenta as atividades que serão realizadas no período de produção e estudo dessa proposta.

1. Cursar disciplinas

Cursar disciplinas obrigatórias e eletivas do Programa de Pós-Graduação visando a obtenção de conhecimento específico e qualificado para realização do projeto;

2. Revisão bibliográfica

Nesta etapa, será realizada uma revisão bibliográfica através da leitura de artigos relacionados ao tema, e posterior revisão sistemática.

3. Avaliação das soluções existentes

Aqui serão analisadas as soluções existentes na literatura para detecção de mudanças em ambientes dinâmicos baseadas no monitoramento da taxa de acerto e na variação dos dados.

4. Implementação do método de detecção

Durante esse período, será desenvolvido um método de detecção à mudança em ambientes dinâmicos existente na literatura.

5. Criação do método de detecção baseado na confiança de decisão

Nessa etapa será desenvolvido um método de detecção baseado na confiança de decisão.

6. Combinação dos métodos de detecção baseados na variação dos dados e na

confiança de decisão.

Nessa fase, será implementado o método que combina as duas estratégias de detecção.

7. Elaboração de artigos

Com base nos resultados obtidos por meio da metodologia desenvolvida, serão elaborados artigos científicos para publicação à comunidade científica.

8. Escrita da dissertação

Nesse período, através dos resultados obtidos, inicia-se o processo de escrita da dissertação.

9. Defesa da dissertação

Por fim, busca-se apresentar os resultados da pesquisa à comunidade científica.

Referências

[1] BAENA-GARCIA, M., DEL CAMPO-ÁVILA, J., FIDALGO, R., AND BIFET, A. Early drift detection method. In *ECML PKDD 2006 Workshop on Knowledge Discovery from Data Streams* (2006), pp. 77-86.

[2] ELWELL, R., AND POLIKAR, R. Incremental Learning of Concept Drift in Nonstationary Enviroments. In *IEEE Transactions on Neural Networks*. Vol, 22. Nº 10, October 2011.

[3] ELWELL, R., AND POLIKAR, R. Incremental learning of variable rate concept drift. In *Multiple Classifier Systems*, J. Benediktsson, J. Kittler, and F. Roli, Eds., vol. 5519 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 142-151.

[4] FOLINO, G., PIZZUTI, C., AND SPEZZANO, G. An adaptive distributed ensemble approach to mine concept-drifting data streams. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Volume 02* (Washington, DC, USA, 2007), ICTAI '07, IEEE Computer Society, pp. 183-188.

[5] GAVRILOV, A., AND LEE, S. An Approach for Invariant Clustering and Recognition in Dynamic Environment. In *Proceedings of IEEE International Conference* - CISSE-2006, December, 2006.

[6] HULTEN, G., SPENCER, L., AND DOMINGOS, P. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2001), KDD '01, ACM, pp. 97-106.

[7] KARNICK, M., AHISKALI, M., MUHLBAIER, M., AND POLIKAR, R. Learning concept drift in nonstationary environments using an ensemble of classifiers based approach. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on* (2008), pp. 3455 -3462.

[8] KLINKENBERG, R., AND JOACHIMS, T. Detecting concept drift with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 2000, pp. 487-494.

[9] KUNCHEVA, L. I. Classifier ensembles for Changing Environments. In *Multiple Classifier Systems*, vol. 3077 of *Lecture Notes in Computer Science*. 2004, pp. 1-157.

[10] KUNCHEVA, L. I. Classifier ensembles for Detecting Concept Change in Streaming Data: Overview and Perspectives. In *2nd Workshop SUEMA 2008 (ECAI 2008)*, pp. 5-10.

[11] KUNCHEVA, L. I. Change Detection in Streaming Multivariate Data Using Likelihood Detectors. In *IEEE computer Society Digital Library. IEEE Transactions on*

Knowledge and Data Engineering, 19 Oct. 2011.

[12] KUNCHEVA, L. I. Clustering-and-selection model for classifier combination. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on* (2000), vol. 1, pp. 185-188 vol.1.

[13] KURLEJ, B. AND WOZNIAK, M. Active Learning Approach to Concept Drift Problem. In *Logic Journal of IGPL Advance Access published*, February 24, 2011.

[14] NARASIMHAMURTHY, A. AND KUNCHEVA, L. I. A framework for generating data to simulate changing environments. In *Proceedings of the 25th IASTED International Multi-Conference*, 12-14 February 2007, pp. 384-389.

[15] NISHIDA, K. Learning and Detecting Concept Drift. Ph.D. Dissertation, Hokkaido University, Japan, 2008. [Online] Available: http://lis2.huie.hokudai.ac.jp/_knishida/paper/nishida2008-dissertation.pdf

[16] OZA, N., and TUMER, K. Classifier ensembles: Select real-world applications. *Information Fusion* 9, 1 (Jan. 2008), 4-20.

[17] PARIKH, D., P. R. An ensemble-based incremental learning approach to data fusion. *Systems, Man, and Cybernetics-PartB, IEEE Transactions on 37*, 2 (2007), 500-508.

[18] PECHENIZKIY, M., BAKKER, J., ZLIOBAITE, I., IVANNIKOV, A., KARKKAINEN, T. Online Mass Flow Prediction in CFB Boilers with Explicit Detection of Sudden Concept Drift. In *ACM SIGKDD Explorations Newsletter*. Volume 11 Issue 2, December 2009, 109-116.

[19] RUTA, D., AND GABRYS, B. Classifier selection for majority voting. *Information Fusion 6*, 1 (January 2005), 63-81.

[20] TSYMBAL, A., PECHENIZKIY, M. AND PUURONEN, S. Dynamic Integration of Classifiers for Handling Concept Drift. *Inf. Fus*. Vol. 9, nº 1, pp. 56-68, Jan. 2008.

[21] WIDMER, G., AND KUBAT, M. Learning in the presence of concept drift and hidden contexts. Vol. 23. Springer Netherlands, 1996, pp. 69-101.

[22] ZHU, Q. Pattern Classification in Dynamic Environments: Tagged Feature-Class Representation and the Classifiers. In *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, nº 5, September/October 1989.