

# ***Drift Detection e Machine Learning para Sistemas de Localização indoor RFID em Ambientes Dinâmicos***

**Eduardo L. Gomes<sup>1,2</sup>, Mauro Fonseca<sup>1</sup>, André Lazzaretti<sup>1</sup>,  
Carlos R. Guerber<sup>1,2</sup>, Anelise Munaretto<sup>1</sup>**

<sup>1</sup>Programa de Pós Graduação em Engenharia Elétrica e Informática Industrial (CPGEI)  
Universidade Tecnológica Federal do Paraná (UTFPR)  
Curitiba – PR – Brasil

<sup>2</sup>Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC)  
Canoinhas – SC – Brasil

eduardo.gomes@ifsc.edu.br, maurofonseca@utfpr.edu.br

**Resumo.** *A localização de objetos em ambientes internos e dinâmicos é uma tarefa desafiadora, pois além da presença de materiais reflexivos e excesso de obstáculos, a posição dos objetos são alteradas constantemente. Para contornar tais problemas, nós propomos o uso da tecnologia RFID e métodos de aprendizagem de máquina em conjunto com técnicas de Drift Detection para a construção de sistemas de localização indoor. A principal contribuição deste artigo é a proposta de um sistema de localização RFID de alta precisão (5 cm) para ambientes onde há mudanças incrementais na posição dos objetos. O resultado obtido com a utilização de técnicas de Drift Detection permitiu ao sistema manter acurácia acima de 96.90% ao longo das 110.000 instâncias.*

## **1. Introdução**

Um sistema de localização pode ser definido como um conjunto de dispositivos, técnicas, algoritmos e aplicações que trabalham em sincronia para estimar as coordenadas absolutas ou relativas de uma pessoa ou objeto em um determinado ambiente previamente mapeado [Goswami 2013, Wang et al. 2021].

Em termos de localização outdoor, a tecnologia GPS (*Global Positioning System*) é amplamente utilizada. No entanto, não tem desempenho apropriado para localização em ambientes internos, devido às limitações dos dispositivos atuais, como a falta de linha de visão entre os satélites e o receptor [Dinh-Van et al. 2017, Santos et al. 2021]. Desta forma, o foco de pesquisas associadas à localização em ambientes internos tem se baseado na busca por tecnologias mais apropriadas para a solução do problema [Torres-Sospedra et al. 2015].

Devido a sua versatilidade e custo-benefício, muitas pesquisas usaram a tecnologia RFID (*Radio Frequency Identification*) para propor sistemas de localização indoor. O RFID é um método de identificação automática que utiliza sinais de radio-frequência para identificar, rastrear e gerenciar objetos, animais ou pessoas sem qualquer contato físico ou campo de visão. Isso só é possível devido ao uso de etiquetas eletrônicas que refletem o sinal recebido de uma antena conectada a um leitor de RFID [R. Smith 2013, Rohei et al. 2021].

Algumas características frequentemente encontradas em ambientes internos aumentam a complexidade dos sistemas de localização que utilizam radiofrequência, como a alta densidade de itens, o grande número de obstáculos (paredes, móveis, equipamentos, plantas e outros) e a presença de materiais com superfície reflexiva (metais e vidros) [Calderoni et al. 2015, Yao and Hsia 2018]. Estas características são responsáveis por gerar interferências na propagação das ondas de radiofrequência. Os principais problemas gerados são: Reflexão [Rappaport 2001]; Difração [Westcott et al. 2011] e Dispersão [Rappaport 2001]. Estas interferências afetam a uniformização dos valores recebidos pelo leitor RFID, especialmente o RSSI (*Received Signal Strength Indicator*) que é a característica predominantemente utilizada para medir a distância entre os leitores RFID e as etiquetas eletrônicas [Ma et al. 2018].

Em um trabalho prévio [Gomes et al. 2017], nós apresentamos um sistema de localização indoor que extraía estatisticamente características dos dados brutos recebidos do leitor RFID e que contribuíram significativamente para que os algoritmos classificadores de aprendizagem de máquina identificassem a posição exata dos objetos. A arquitetura utilizava a etiqueta como uma classe alvo na construção dos classificadores e isso se tornou uma vulnerabilidade na implantação em ambientes de larga escala, pois a grande quantidade de objetos alvo (classes) aumentava a complexidade e tempo do aprendizado e predição do classificador. Esta vulnerabilidade foi estudada em outro trabalho prévio [Gomes et al. 2020], o qual utilizamos a técnica de aprendizagem de máquina Clusterização Hierárquica [Bonaccorso 2017] dividindo o ambiente em macro-regiões. Para localizar o objeto alvo, inicialmente realizamos uma predição que indicava qual macro região pertencia o objeto e, em seguida, havia uma segunda predição considerando somente as etiquetas da macro região. Desta maneira, reduzimos a quantidade de classes no aprendizado e predição dos classificadores. Na implantação dos dois trabalhos, percebemos que em ambientes onde há constante mudança de posição, adição e retirada de objetos, o qual denominamos neste trabalho como *ambientes dinâmicos*, o fluxo de dados recebidos não apresenta comportamento estático e os classificadores perdem sua acuracidade, frequentemente necessitando de novo treinamento. Essa mudança que ocorre no padrão dos dados é denominada por [Mitchell 1997] como *Concept Drift* e torna a utilização dos algoritmos clássicos de aprendizagem de máquina desafiadora para essa tarefa [Dongre and Malik 2014].

Com base nesses desafios, diversas pesquisas têm buscado desenvolver algoritmos que sejam capazes de identificar *Concept Drifts* de maneira rápida, já que isso previne que ocorra uma perda grande de acurácia que é motivada por erros de identificação de um novo padrão das instâncias de dados. Motivado por esses desafios, nós aprimoramos os trabalhos apresentados em [Gomes et al. 2017, Gomes et al. 2020], introduzindo o conceito de algoritmos para *Drift Detection* no treinamento dos classificadores. Para tanto, nós testamos 100 objetos alvo em um ambiente real realizando mudanças incrementais na posição de todos os objetos. Desta maneira, foi possível realizar uma avaliação de desempenho de cinco algoritmos *Drift Detection* (Seção 3.2.3) utilizados para aprendizagem de máquina em fluxo de dados online, os quais permitiram manter e até incrementar a acurácia do classificador *Random Forest* testado na implantação do sistema de localização *indoor*.

Este artigo está organizado da seguinte forma: A Seção 2 demonstra os trabalhos relacionados e as principais contribuições do presente trabalho. A Seção 3 apresenta em

detalhes como todos os módulos e processos do sistema de localização proposto foram construídos. A Seção 4 apresenta os resultados obtidos na aplicação do sistema usando dados de treinamento e em um ambiente real, e a Seção 5 apresenta as conclusões e sugestões para trabalhos futuros.

## 2. Trabalhos Relacionados e Principais Contribuições

A utilização da tecnologia RFID para criar sistemas de localização e posicionamento *indoor* é amplamente utilizada em vários estudos na literatura. Especialmente em razão dos benefícios que pode fornecer como: segurança contra furtos, agilidade de identificação de pessoas, localização de produtos, inventário de estoque, posicionamento e navegação de robôs, além de diversas outras aplicações [Torres-Sospedra et al. 2015, Rohei et al. 2021]. Conforme pudemos levantar, uma única solução multiuso que atenda a qualquer necessidade em aplicações de posicionamento interno ainda não existe. De fato, os sistemas propostos na literatura baseiam-se em diversas tecnologias e modelos adaptados, projetados para atender às exigências de cada contexto específico.

Com base nas tecnologias utilizadas é possível classificar os trabalhos existentes em três grupos [Lai et al. 2020, Al-Jarrah et al. 2019]: **i)** métodos baseados em RSSI [Calderoni et al. 2015, Torres-Sospedra et al. 2015, Dinh-Van et al. 2017, Hatem et al. 2018, Wang et al. 2019, Sabr and Belton 2019]; **ii)** métodos baseados em *Angle of Arrival* e *Time Difference of Arrival* [Steele et al. 2019]; e **iii)** métodos baseados em fases [Lai et al. 2020]. Neste artigo utilizamos métodos baseados em RSSI, pois priorizamos o uso de equipamentos disponíveis no mercado, não necessitando desenvolvimento de equipamentos especiais. O atributo RSSI é usualmente calculado em leitores RFID, desta maneira o sistema de localização proposto é baseado na independência de recursos e hardware. Como utilizamos métodos baseados em RSSI, focamos em trabalhos que utilizam estes métodos na revisão aqui apresentada.

Inicialmente, em [Wang et al. 2019], os autores apresentaram um sistema de localização *indoor* com o objetivo de reduzir ao máximo o processo de implantação, onde somente a posição dos leitores RFID devem ser conhecidas. Utilizaram o atributo RSSI de etiquetas ativas e os métodos de *Newton-Search* e *Kalman Filter* como técnicas de localização. Já em [Zhang et al. 2015], o uso de estatísticas de RSSI para identificar a posição de uma etiqueta RFID passiva em relação a um leitor é investigado. Além disso, concluíram que quanto maior a distância das antenas, maior a interferência da orientação da etiqueta em relação às antenas. No estudo de [Hatem et al. 2018], foram avaliados os erros na estimativa de distâncias para localização em ambientes internos com o objetivo de criar um modelo de perda de caminho que foi empregado para minimizar os erros de localização. Em [Ma et al. 2018], os autores criaram um modelo de detecção automática de leituras de falsos positivos em um leitor RFID. Para isso, realizaram a derivação do RSSI em conjunto com algoritmos de aprendizagem de máquina e concluíram que o RSSI é fortemente influenciado pelo ambiente, como os reflexos de múltiplos caminhos de sinais e obstrução que são muito comuns em ambientes internos. De forma similar, um sistema de localização *indoor* projetado para funcionar em cenários ruidosos é proposto em [Calderoni et al. 2015], onde as transmissões podem ser interferidas por outros dispositivos eletrônicos ou paredes protegidas, como é o caso de um hospital. A principal lacuna identificada nos trabalhos acima citados é que em nenhum deles houve preocupação na mudança do padrão no fluxo de dados recebidos pelos leitores RFID ou ambientes com

mudanças constantes na posição dos objetos (*Concept Drifts*).

Mesmo em sistemas de localização indoor que não utilizam RFID, mas têm abordagem utilizando radiofrequência como Wi-Fi, é notável que a principal preocupação é com a sensibilidade do ambiente para o RSSI e não com a detecção de *Concept Drifts*. Em [Torres-Sospedra et al. 2015], os autores visaram identificar qual a melhor função de distância a ser aplicada em algoritmos que usam a abordagem do algoritmo KNN (*K-Nearest Neighbors*) para minimizar a sensibilidade do RSSI. Eles concluíram que é muito difícil aplicar algoritmos de propagação de sinal ao posicionamento sem fio, devido à natureza severa da propagação de sinal. De forma similar em [Dinh-Van et al. 2017], um método de *Wi-Fi-Fingerprint* é apresentado para substituir o comportamento do GPS em ambientes internos. A principal contribuição foi usar uma técnica de suavização de dados brutos com o método de classificação (rede neural) para tratar a intensidade do sinal Wi-Fi ruidoso. Em [Sabr and Belton 2019], os autores propuseram uma solução de identificação e localização que utilizou uma combinação de duas tecnologias: reconhecimento facial e Wi-Fi utilizando trilateração do atributo RSSI. Fizeram testes em ambiente controlado e concluíram que a precisão do RSSI foi diretamente influenciado pela quantidade de equipamentos no ambiente.

A Tabela 1 apresenta uma comparação dos modelos de localização propostos nos trabalhos relacionados. A principal limitação encontrada é que em nenhum dos trabalhos houve preocupação com a detecção de *Concept Drifts* para ambientes dinâmicos. Diante desta lacuna, a principal contribuição deste artigo é um sistema de localização de alta precisão (5 cm) para ambientes de dinâmicos utilizando técnicas de *Drift Detection e Machine Learning*. A Seção 3 apresenta em detalhes como o sistema proposto neste trabalho foi construído.

**Tabela 1. Comparação de trabalhos relacionados.**

Pesquisa	Alvo	Cenário	Sensor	Precisão	<i>Concept Drift</i>
Este Artigo	Auto-peças	Prateleira	RFID (passivo)	0.05 m	✓
[Zhang et al. 2015]	Etiqueta	Controlado	RFID (passivo)	0.40 m	
[Calderoni et al. 2015]	Pessoa	Sala	RFID (ativo)	> 10.00 m	
[Torres-Sospedra et al. 2015]	<i>Fingerprint</i>	<i>UJI indoorLoc db</i>	<i>WiFi-Fingerprint</i>	1.70 m	
[Dinh-Van et al. 2017]	Veículo	Estacionamento	<i>WiFi-Fingerprint</i>	10.00 m	
[Gomes et al. 2017]	Livro	Prateleira	RFID (passivo)	0.05 m	
[Hernández et al. 2017]	Dispositivos Móveis	Universidade	Wi-Fi	1.22 m	
[Hatem et al. 2018]	Etiqueta	Controlado	RFID (ativo)	0.50 m	
[Yao and Hsia 2018]	Etiqueta	Controlado	RFID (passivo)	0.50 m	
[Wang et al. 2019]	Pessoa	Simulado	RFID (ativo)	1.00 m	
[Sabr and Belton 2019]	Pessoa	Controlado	Wi-Fi	1.00 m	
[Steele et al. 2019]	Pessoa	Centro de Saúde	RFID (ativo)	1.00 m	
[Zhang et al. 2020]	<i>Fingerprint</i>	Centro de Saúde	Wi-Fi	1.29 m	
[Gomes et al. 2020]	Livro	Prateleira	RFID (passivo)	0.05 m	
[Santos et al. 2021]	<i>Fingerprint</i>	Universidade	Wi-Fi	2.85 m	
[Rohei et al. 2021]	Pessoa	Simulado (Hospital)	RFID Sub-cutâneo	4.00 m	

### 3. Sistema de Localização indoor

Um sistema RFID é composto de componentes de infraestrutura e componentes de negócio. Os componentes de infraestrutura são as etiquetas, leitores, cabos, conectores e antenas. Os componentes de negócios são as aplicações, softwares de gerenciamento e o middleware que realiza a comunicação entre componentes de infraestrutura e componentes de negócio [R. Smith 2013]. A Figura 1 apresenta uma visão geral do sistema de

localização proposto neste trabalho, o mesmo é dividido em dois módulos. O Módulo de Leitura que faz parte dos componentes de infraestrutura e o Módulo de Aprendizagem de Máquina que faz parte dos componentes de negócio. As seções 3.1 e 3.2 apresentam em detalhes como os módulos do sistema foram construídos.

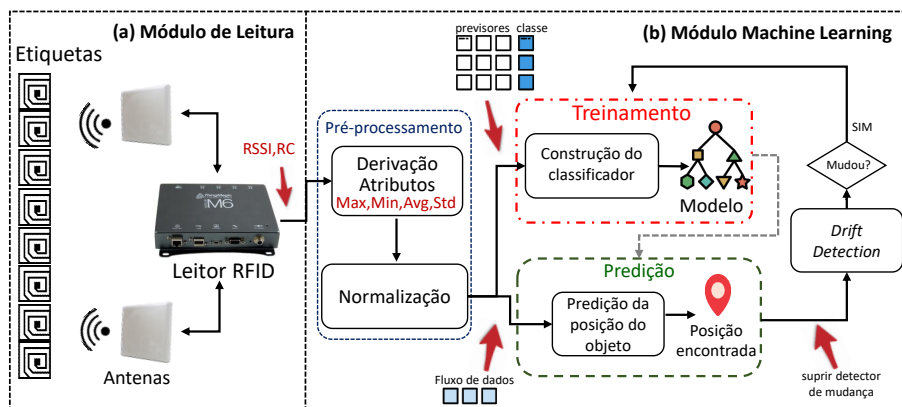


Figura 1. Visão geral do sistema de localização *indoor*.

### 3.1. Módulo de Leitura

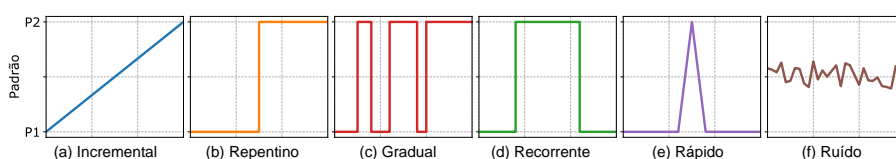
São os componentes básicos de infraestrutura de um sistema RFID, como mostra a Figura 1(a). O leitor geralmente está conectado a um computador com capacidade suficiente para processar os dados recebidos das etiquetas. Para a preparação do ambiente de implementação, nós utilizamos quatro antenas UHF mono-estáticas que operam na faixa de frequência de 902 a 928 MHz com ganho de 6 dBi. O equipamento que usamos para realizar as leituras foi o *ThingMagic Mercury 6*, que é um leitor RFID UHF de alto desempenho, suportando até quatro antenas monoestáticas, entradas e saídas digitais e conexão Wi-Fi. Ambos os equipamentos são comerciais e podem ser comprados por qualquer pessoa em empresas especializadas na comercialização de equipamentos RFID.

Nós fixamos 100 etiquetas em objetos posicionados lado a lado nas prateleiras de uma loja de peças automotivas. A distância entre as antenas e etiquetas foi de **115 cm** e a distância entre o grupo de antenas de **250 cm**. O tempo de interrogação definido foi de **5 segundos**. Este é o tempo que o leitor interroga as etiquetas e organiza as informações de entrada em uma coleção de dados formada pelos campos *timestamp*, *RSSI*, *TagID*, *RC* (*Read Count*) e *Antenna*. Para definir este tempo, nós identificamos o limite mínimo no qual todas as etiquetas foram identificadas pelo menos uma vez em uma antena. Para realizarmos a coleta de dados, os objetos foram colocados em posições conhecidas. Realizamos 1.100 leituras em cada etiqueta alvo, medindo o RSSI e o RC de cada uma das 4 antenas, totalizando 110.000 instâncias. A Figura 2 apresenta uma visão geral do ambiente de implantação e dos parâmetros definidos.

As constantes movimentações dos objetos é uma das características de um ambiente dinâmico. Para evidenciar esta característica no fluxo de dados, nós realizamos uma troca de 10% das posições dos objetos a cada 10.000 instâncias, o que permitiu alterar a posição de 100% dos objetos ao final das 110.000 instâncias. Este tipo de *Concept Drift* é denominado por [Dongre and Malik 2014] e por [Bifet et al. 2018] como *Incremental Drift* (Figura 3(a)), o qual é caracterizado pela mudança gradativa e contínua



**Figura 2. Ambiente de implementação.**



**Figura 3. Tipos de Concept Drift.**

do padrão atual do fluxo, migrando para outro padrão, e a partir de determinado momento começa a surgir apenas instâncias deste outro padrão. Além do tipo incremental, [Dongre and Malik 2014] também classificam a mudança de padrão nos tipos: repentino, gradual, recorrente, rápido e ruído. Nos tipos incremental, repentino, gradual ocorre a mudança completa do padrão do fluxo de dados. Nos tipos recorrente, rápido e ruído a mudança é temporária. A Figura 3 ilustra a troca de padrão no fluxo de dados para cada um dos tipos.

### 3.2. Módulo Aprendizagem de Máquina

Conforme apresentado na Figura 1(b), o Módulo Aprendizagem de Máquina é responsável por receber os dados brutos do Módulo de Leitura, organizar o fluxo de dados, derivar os atributos (RSSI e RC), normalizar os dados, treinar os modelos classificadores, realizar a predição das posições das etiquetas e verificar se houve mudança no padrão do fluxo de dados. Para a construção deste módulo nós utilizamos a linguagem de programação *Python* com as bibliotecas *Scikit-learn* [Pedregosa et al. 2011] e *Scikit-multiflow* [Batra et al. 2014], conforme detalhado a seguir.

#### 3.2.1. Pré-processamento

Tem como objetivo preparar os dados para treinamento e predição. Para isso, o pré-processamento recebe o fluxo de dados do Módulo de Leitura, organiza, extrai estatisticamente novos atributos e normaliza os valores RSSI e RC de cada uma das antenas.

Confirmando os problemas de propagação das ondas de radiofrequência apresentados na Seção 1, ao analisarmos os dados brutos nós identificamos uma variação significativa do RSSI. A Figura 4 apresenta um boxplot com amostra das leituras realizadas em seis posições, demonstrando a presença de muitos *outliers*. Além disso, o intervalo

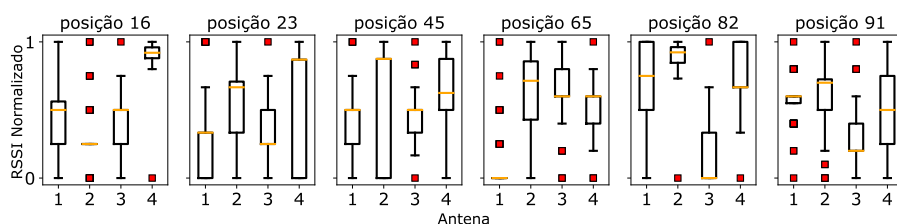


Figura 4. Análise de outliers do atributo RSSI.

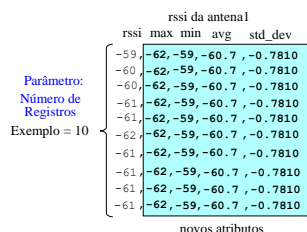


Figura 5. Exemplo do processo Derivação de Atributos.

médio entre os valores mínimo e máximo do RSSI de todas as etiquetas é de 81.56%, o que interfere no aprendizado dos classificadores.

Com o objetivo de contornar tais problemas nós calculamos e adicionamos no *dataset* de treinamento novas características para os atributos RSSI e RC (média, desvio padrão, valor mínimo e valor máximo). Baseado em um parâmetro (Número de Registros) definido pelo usuário, o sistema agrupa o fluxo de dados recebidos do Módulo de Leitura, calcula a média, desvio padrão e identifica o valor mínimo e máximo da sequência. A Figura 5 apresenta um exemplo de como o processo Derivação de Atributos trabalha e a Tabela 2 demonstra a estrutura do *dataset* após a finalização deste processo. O número de atributos passa a ser quarenta, dez para cada antena.

A base de dados de treinamento (Tabela 2) possui essencialmente atributos numéricos e de diferentes magnitudes. Para evitar que estas diferentes magnitudes interfiram no comportamento de alguns classificadores nós normalizamos os valores de cada atributo. Para normalizar os dados nós utilizamos a classe *StandardScaler*, que dimensiona cada valor de entrada separadamente subtraindo a média (chamada de centralização) e dividindo pelo desvio padrão para deslocar a distribuição a fim de ter uma média zero e um desvio padrão unitário [Pedregosa et al. 2011].

Tabela 2. Derivação de Atributos. Onde *n* representa o número da antena.

Atributo	Tipo de dado	Descrição
rssi_antena <i>n</i>	decimal	RSSI antena <i>n</i>
rc_antena <i>n</i>	inteiro	Quantidade de leituras da etiqueta na antena <i>n</i>
avg_rssi_antena <i>n</i>	decimal	Média do RSSI
avg_rc_antena <i>n</i>	decimal	Média do <i>Read Count</i>
min_rssi_antena <i>n</i>	decimal	Valor mínimo do RSSI
min_rc_antena <i>n</i>	decimal	Valor mínimo do <i>Read Count</i>
max_rssi_antena <i>n</i>	decimal	Valor máximo do RSSI
max_rc_antena <i>n</i>	decimal	Valor máximo do <i>Read Count</i>
stddev_rssi_antena <i>n</i>	decimal	Desvio padrão do RSSI
stddev_rc_antena <i>n</i>	decimal	Desvio padrão do <i>Read Count</i>

### 3.2.2. Treinamento e Predição

O processo de treinamento é responsável por criar os modelos de classificação para a futura predição da posição dos objetos em tempo real (Seção 3.2.3). Nesta fase, os algoritmos de aprendizado de máquina constroem os modelos baseados no conjunto de treinamento formado por tuplas de dados (atributos) e seus rótulos correspondentes (classes). A tupla é composta por um vetor de características  $n$ -dimensional representado por  $X = (x_1, x_2, \dots, x_n)$ . No modelo apresentado neste artigo, quarenta características são extraídas das leituras RFID (dez para cada antena (Tabela 2)), logo  $n = 40$  e a classe alvo são as posições das etiquetas no ambiente representadas pelo número da posição (1 a 100).

Para este experimento nós utilizamos a técnica de classificação *Random Forest* (RF), pois em estudo prévio apresentado em [Gomes et al. 2020] apresentou eficientes resultados e obteve melhor acurácia na implementação da pesquisa. A RF consiste em um grande número de classificadores *Decision Tree* (DT) individuais que funcionam como um conjunto. Cada DT individual na RF realiza uma previsão e a classe com mais votos se torna a previsão do modelo classificador. O conceito fundamental da RF é a sabedoria das multidões e a razão na qual o classificador se baseia é: um grande número de classificadores (DT) relativamente não correlacionados superará qualquer um dos classificadores individuais da floresta [Breiman 2001]. O processo de treinamento é acionado no *setup* inicial da implantação do sistema e também pelo processo de detecção de *Concept Drift* toda vez que o algoritmo de *Drift Detection* identifica uma mudança no padrão do fluxo de dados.

O processo de predição recebe o fluxo de dados em tempo real e utiliza os classificadores construídos na fase de treinamento para indicar a posição do objeto. Este processo também alimenta os algoritmos *Drift Detection*, os quais apresentamos detalhadamente na Seção a seguir.

### 3.2.3. Drift Detection

Em várias aplicações do mundo real, o padrão dos dados não é estável e muda ao longo do tempo. Essa mudança que ocorre no padrão dos dados é denominada de *Concept Drift* [Dongre and Malik 2014]. Com o *Concept Drift* em um fluxo de dados, se um modelo de aprendizado não passa por uma atualização periódica ele acaba se tornando obsoleto. Diante deste desafio várias pesquisas apresentam propostas de algoritmos para detecção de *Concept Drift* [Bifet et al. 2018]. No presente trabalho nós testamos cinco algoritmos comumente utilizados em aplicações de *Machine Learning* para *Datastreams* e que estão implementados na biblioteca *Scikit-multiflow* [Batra et al. 2014]: *Adaptive Windowing* (ADWIN), *Drift Detection Method* (DDM), *Early Drift Detection Method* (EDDM), *Drift Detection Method based on Hoeffding's* (HDDM\_A) e *Kolmogorov-Smirnov Windowing* (KSWIN), os quais descrevemos resumidamente a seguir:

1. ADWIN: é um algoritmo de janela deslizante adaptável para detectar mudanças e manter estatísticas atualizadas sobre um fluxo de dados em tempo real. O algoritmo mantém uma janela deslizante  $\Psi$  com os exemplos lidos mais recentemente. A ideia principal do ADWIN é sempre que duas subjanelas "grandes o sufici-



ente” de  $\Psi$  exibem médias ”distintas o suficiente”, pode-se concluir que os valores esperados correspondentes são diferentes, e a parte mais antiga da janela é descartada. Isso envolve responder a uma hipótese estatística: A média  $\mu_t$  permaneceu constante em  $\Psi$  com confiança  $\delta$ ? Se a hipótese for negativa, a mudança foi detectada. [Bifet and Gavaldà 2007];

2. DDM: é um método de detecção de *Concept Drift* baseado na premissa do modelo de aprendizagem PAC (*Probably approximately correct learning*) ([Mohri et al. 2018]), o qual considera que a taxa de erro do classificador diminuirá à medida que o número de amostras analisadas aumentar, desde que a distribuição de dados seja estacionária. Se o algoritmo detectar um aumento na taxa de erro, que ultrapassa um limite calculado, ou a mudança é detectada ou o algoritmo irá avisar o usuário que a mudança pode ocorrer em um futuro próximo, o que é chamado de zona de alerta [Gama et al. 2004];
3. EDDM: é um algoritmo apresentado por [Baena-García et al. 2006] baseado no algoritmo DDM, mas em vez de usar a taxa de erro do classificador, eles propõem o uso da taxa de erro de distância. Eles denotam  $p'_i$  como a distância média entre dois erros consecutivos e  $s'_i$  como seu desvio padrão. Usando esses valores, as condições de aviso e mudança são fornecidas pela equações 1 e 2 respectivamente.

$$\frac{p'_i + 2 \cdot s'_i}{p'_{max} + 2 \cdot s'_{max}} < \alpha \quad (1)$$

$$\frac{p'_i + 3 \cdot s'_i}{p'_{max} + 3 \cdot s'_{max}} < \beta \quad (2)$$

em que  $\alpha = 0,95$  e  $\beta = 0,90$  por padrão, mas pode ser parametrizado pelo usuário. Segundo os autores o EDDM funciona melhor do que DDM para desvio gradual lento, mas é mais sensível ao ruído. O que pode ser uma desvantagem dependendo do tipo de fluxo de dados;

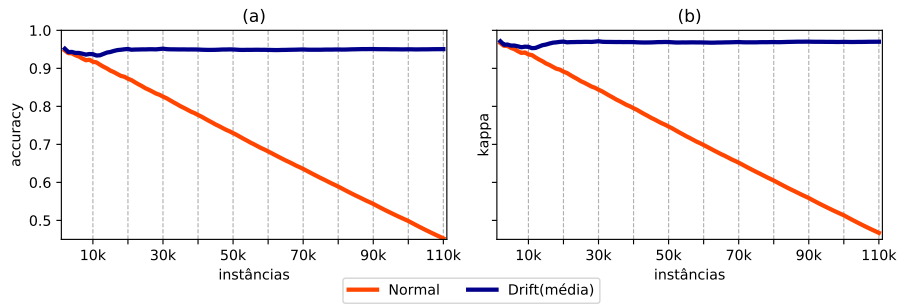
4. HDDM.A: é um método baseado na desigualdade do limite de *Hoeffding*. O limite de *Hoeffding* afirma que, com probabilidade  $1 - \delta$ , a média verdadeira de uma variável aleatória de intervalo  $R$  não será diferente da média estimada após  $n$  observações independentes por mais de:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}. \quad (3)$$

O HDDM.A usa a média como estimativa. Ele recebe como entrada um fluxo de valores reais e retorna a situação estimada do fluxo: estável, aviso ou mudança detectada [Frías-Blanco et al. 2015];

5. KSWIN: o método apresentado por [Raab et al. 2020] é baseado no teste estatístico de *Kolmogorov-Smirnov* (KS). O KSWIN mantém uma janela deslizante  $\Psi$  de tamanho fixo  $n$  (parâmetro *window\_size*). Os últimos  $r$  (parâmetro *stat\_size*) exemplos de  $\Psi$  são assumidos para representar o último padrão do fluxo de dados  $R$ . Das  $n - r$  amostras de  $\Psi$ ,  $r$  amostras são uniformemente extraídas representando o último padrão do fluxo de dados  $W$ . Então o teste KS é executado para as janelas  $R$  e  $W$  comparando a distância empírica cumulativa da distribuição de dados  $dist(R, W)$  e o *Concept Drift* é detectado se:

$$dist(R, W) > \sqrt{-\frac{\ln \alpha}{r}}. \quad (4)$$



**Figura 6. Com Drift Detection x Predição Normal.**

Independente do algoritmo escolhido para detecção de *Concept Drift*, toda vez que a mudança no fluxo de dados é detectada, o módulo de treinamento é acionado para realizar nova aprendizagem utilizando os previsores (atributos) do novo padrão. Na Seção 4 nós apresentamos os resultados obtidos na implementação do sistema de localização proposto neste trabalho.

#### 4. Avaliação Experimental

Após realizar a leitura e pré-processamento dos dados, nós realizamos análise de desempenho utilizando a métrica *Accuracy* e o coeficiente de concordância *Kappa* para comparar o sistema utilizando os algoritmos *Drift Detection* em relação à sua não utilização. A métrica *Accuracy* é a medida de desempenho mais intuitiva e é simplesmente uma razão de observação corretamente prevista para o total de observações e é calculada pela equação

$$accuracy = \frac{T_p + T_n}{T_p + F_p + F_n + T_n} \quad (5)$$

onde  $T_p$  são os verdadeiros positivos,  $T_n$  são os verdadeiros negativos,  $F_p$  são os falsos positivos e  $F_n$  são os falsos negativos. O coeficiente de concordância *Kappa* é utilizado para descrever a concordância entre dois ou mais classificadores quando realizam uma avaliação de uma mesma amostra e é calculado pela equação

$$kappa = \frac{\hat{P}_O - \hat{P}_e}{1 - \hat{P}_E} \quad (6)$$

em que  $\hat{P}_O$  é a proporção observada de concordâncias (soma das respostas concordantes dividida pelo total) e  $\hat{P}_E$  é proporção esperada de concordâncias (soma dos valores esperados das respostas concordantes dividida pelo total). A Figura 6 apresenta os resultados obtidos pelo classificador usando os algoritmos *Drift Detection* (média do resultado de todos eles) em comparação ao seu não uso (predição normal). Como esperado, a predição normal foi perdendo sua eficiência ao longo do tempo, pois conforme descrito na Seção 3.1, nós realizamos trocas incrementais a cada 10.000 instâncias.

Após identificar que a utilização dos algoritmos *Drift Detection* auxiliam na manutenção da acuracidade dos classificadores ao longo do tempo em ambientes dinâmicos, nossa pesquisa consistiu em identificar qual algoritmo *Drift Detection* obteve melhor desempenho na detecção dos *Concept Drifts*. A Figura 7 apresenta a linha do tempo de cada algoritmo e suas respectivas identificações de *Concept Drifts*. A linha

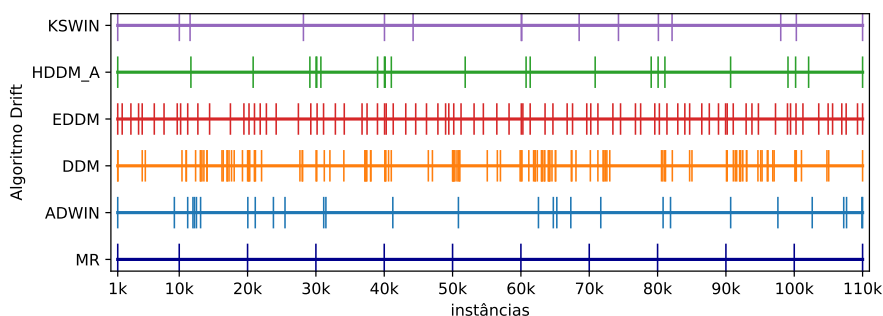


Figura 7. Detecção de *Concept Drift* ao longo do tempo.

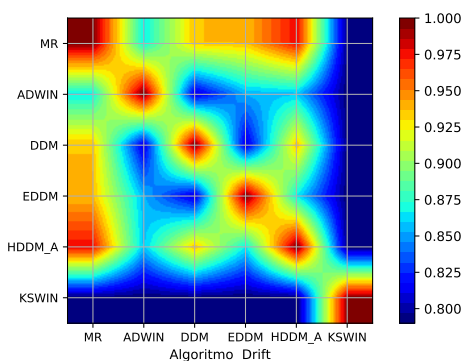


Figura 8. Correlação de *Pearson*.

MR (Mudança Real) indica a mudança exata que foi realizada no ambiente. Podemos perceber que os algoritmos DDM e EDDM foram mais sensíveis aos ruídos existentes no ambiente devido às interferências nas ondas de radiofrequência explicadas na Seção 1, os quais identificaram 122 e 81 *Concept Drifts* respectivamente. O algoritmo HDDM\_A obteve melhor desempenho, detectando *Concept Drifts* muito próximos das mudanças reais. O algoritmo ADWIN também detectou mudanças muito próximas, porém foi mais sensível aos ruídos. O algoritmo KSWIN detectou algumas mudanças próximas, mas não conseguiu detectar todos os *Concept Drifts*. A Figura 8 apresenta a matriz de correlação de *Pearson*, onde podemos confirmar que a melhor correlação com MR foi do algoritmo HDDM\_A no valor de 97,80%.

A Figura 9 apresenta os resultados da métrica *accuracy* e do coeficiente de concordância *kappa* para cada um dos algoritmos ao longo do tempo. Podemos perceber que os algoritmos DDM e EDDM obtiveram melhores resultados, isto aconteceu em razão de detectarem muitos *Concept Drifts*, pois sendo muito sensíveis aos ruídos exigiram treinamento constante dos classificadores. Este constante treinamento manteve os classificadores atualizados, e permitiu obterem melhores resultados, porém conforme demonstrado na Figura 10 o tempo de treinamento foi muito superior em relação aos algoritmos ADWIN, HDDM\_A e KSWIN. Na Figura 9 também é possível perceber que os algoritmos ADWIN e HDDM\_A conseguiram manter e melhorar suas performances ao longo do tempo, quanto que o algoritmo KSWIN perdeu rendimento conforme as mudanças foram acontecendo.

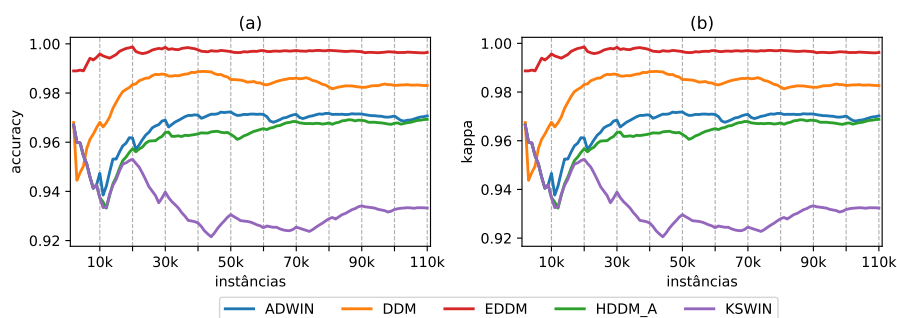


Figura 9. Performance dos algoritmos Drift Detection.

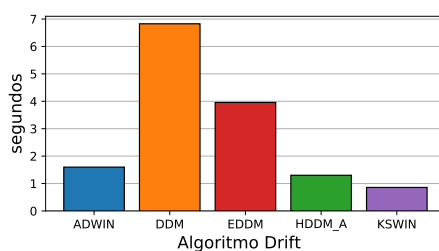


Figura 10. Tempos de Treinamento.

## 5. Conclusões

Sistemas de localização em ambientes internos são amplamente estudados devido aos diversos benefícios que podem proporcionar. A tecnologia RFID possui uma grande versatilidade e utilizar etiquetas RFID passivas para criar sistemas de localização *indoor* contribui para a redução dos custos de implantação. Entretanto, as interferências nas ondas de radiofrequência dificultam a obtenção de resultados satisfatórios em termos de acurácia, principalmente quando usada em ambientes com materiais reflexivos, excesso de obstáculos e principalmente em ambientes onde existe uma troca constante nas posições dos objetos.

Neste trabalho, nós apresentamos um sistema utilizando RFID, *Machine Learning* e *Drift Detection* para localização *indoor* com o objetivo de manter a acuracidade ao longo do tempo, mesmo depois de trocas incrementais nas posições dos objetos. Como resultado experimental em ambiente real onde havia prateleiras de metal, materiais reflexivos e uma troca incremental na posição dos objetos de 10% a cada 10.000 instâncias, a implementação do sistema apresentou manutenção constante e ganho de acurácia com os algoritmos HDDM\_A, ADWIN, DDM e EDDM. Sendo que o HDDM\_A foi o algoritmo que identificou *Concept Drifts* mais próximos da mudança real com uma correlação de *Pearson* de 97,80%.

Apesar do bom desempenho do sistema e de sua adequação em relação aos objetivos iniciais, alguns aspectos importantes merecem investigações adicionais e servem como motivação para trabalhos futuros. Teste em ambientes que possuam outros tipos de *Concept Drifts* como repentino, gradual e recorrente. Outros algoritmos *Drift Detection* podem ser testados como EDIST, Page-Hinkley e HDDM\_W. E uma proposta de algoritmo *Drift Detection* específico para sistemas de localização *indoor* e RFID que utilize características deste tipo de ambiente para identificar que ruídos na propagação das ondas

de radiofrequência não são necessariamente *Concept Drifts* minimizando erros e identificando apenas *Concept Drifts* reais.

## Referências

- Al-Jarrah, M. A., Al-Dweik, A., Alsusa, E., and Damiani, E. (2019). Rfid reader localization using hard decisions with error concealment. *IEEE Sensors Journal*, 19:7534–7542.
- Baena-Garca, M., Ávila, J. D. C., Fidalgo, R., Bifet, A., Gavaldà, R., and Bueno, R. M. (2006). Early drift detection method. In *Fourth International Workshop on Knowledge Discovery from Data Streams*.
- Batra, U., of Computer Science Engineering, I. U. D., Technology., I., of Electrical, I., and Engineers, E. (2014). *Souvenir of the 2014 IEEE International Advance Computing Conference (IACC) : February 21-22, 2014 : Gurgaon, India*.
- Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*, pages 443–448. SIAM.
- Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. (2018). *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press.
- Bonaccorso, G. (2017). *Machine Learning Algorithms: A Reference Guide to Popular Algorithms for Data Science and Machine Learning*. Packt Publishing.
- Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.
- Calderoni, L., Ferrara, M., Franco, A., and Maio, D. (2015). Indoor localization in a hospital environment using RF classifiers. *Expert Systems with Applications*.
- Dinh-Van, N., Nashashibi, F., Thanh-Huong, N., and Castelli, E. (2017). Indoor Intelligent Vehicle localization using WiFi received signal strength indicator. *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*.
- Dongre, P. B. and Malik, L. G. (2014). A review on real time data stream classification and adapting to various concept drift scenarios. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 533–537.
- Frías-Blanco, I., d. Campo-Ávila, J., Ramos-Jiménez, G., Morales-Bueno, R., Ortiz-Díaz, A., and Caballero-Mota, Y. (2015). Online and non-parametric drift detection methods based on hoeffding’s bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823.
- Gama, J., Medas, P., Castillo, G., and Rodrigues, P. P. (2004). Learning with drift detection. In *SBIA*, volume 3171, pages 286–295. Springer.
- Gomes, E. L., Fonseca, M., Munaretto, A., and Guerber, C. R. (2017). Arquitetura RF-Miner - Uma solução para localização indoor. *XXII Workshop de Gerência e Operação de Redes e Serviços(WGRS)*, pages 111–122.
- Gomes, E. L., Fonseca, M., Munaretto, A., and Guerber, C. R. (2020). Etiquetas rfid passivas e aprendizagem de máquina para sistema de localização indoor de alta precisão. *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 252–265.
- Goswami, S. (2013). *Indoor location technologies*. New York, Springer, 1st edition.
- Hatem, E., El-Hassan, B., Laheurte, J.-M., Abou-Chakra, S., Colin, E., and Marechal, C. (2018). Study the estimated distance error in indoor localization using uhf-rfid. *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*.
- Hernández, N., Alonso, J. M., and Ocaña, M. (2017). Fuzzy classifier ensembles for hierarchical wifi-based semantic indoor localization. *Expert Systems with Applications*, 90:394–404.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *14th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- Kvam, P. and Vidakovic, B. (2007). Nonparametric statistics with applications to science and engineering. 123.
- Lai, J., Luo, C., Wu, J., Li, J., Wang, J., Chen, J., Feng, G., and Song, H. (2020). Tagsort: Accurate relative localization exploring rfid phase spectrum matching for internet of things. *IEEE Internet of Things Journal*, 7:389–399.
- Ma, H., Wang, Y., and Wang, K. (2018). Automatic detection of false positive RFID readings using machine learning algorithms. *Expert Systems with Applications*.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. The MIT Press, 2nd edition.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- R. Smith, J. (2013). *Wirelessly Powered Sensor Networks and Computational RFID*.
- Raab, C., Heusinger, M., and Schleif, F.-M. (2020). Reactive soft prototype computing for concept drift streams. *Neurocomputing*, 416:340–351.
- Rappaport, T. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.
- Rohei, M. S., Salwana, E., Shah, N. B. A. K., and Kakar, A. S. (2021). Design and testing of an epidermal rfid mechanism in a smart indoor human tracking system. *IEEE Sensors Journal*, 21:5476–5486.
- Sabr, O. and Belton, J. (2019). Identifying and Tracking Individuals in a Smart Indoor Environment. *IEEE-Fifth International Engineering Conference on Developments in Civil & Computer Engineering Applications*.
- Santos, R., Leonardo, R., Barandas, M., Moreira, D., Rocha, T., Alves, P., Oliveira, J. P., and Gamboa, H. (2021). Crowdsourcing-based fingerprinting for indoor location in multi-storey buildings. *IEEE Access*, pages 1–1.
- Steele, A. M., Bopp, M. M., Rock, L., Boppvagog, M. M., Taylor, T. S., Sullivan, D. H., and Motivation, A. (2019). Patient Activity Monitoring Based on Real-Time Location Data. *2019 IEEE Int. Conf. on Bioinformatics and Biomedicine*.
- Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., and Huerta, J. (2015). Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23):9263–9278.
- Wang, H., Wang, C., and Xie, L. (2021). Intensity-slam: Intensity assisted localization and mapping for large scale environment. *IEEE Robotics and Automation Letters*, pages 1–1.
- Wang, J., Dhanapal, R. K., Ramakrishnan, P., Balasingam, B., Souza, T., and Maev, R. (2019). Active RFID Based Indoor Localization. *IEEE-22nd International Conference on Information Fusion*.
- Westcott, D. A., Coleman, D. D., Miller, B., and Mackenzie, P. (2011). *CWAP Certified Wireless Analysis Professional Official Study Guide*. SYBEX Inc., CA, USA.
- Yao, C. Y. and Hsia, W. C. (2018). An indoor positioning system based on the dual-channel passive rfid technology. *IEEE Sensors Journal*, 18:4654–4663.
- Zhang, C., Qin, N., Xue, Y., and Yang, L. (2020). Received signal strength-based indoor localization using hierarchical classification. *Sensors (Switzerland)*, 20.
- Zhang, X. b., Akre, J.-M. b., Baey, S. b., Fladenmuller, A. b., Kervella, B. b., Zancanaro, M. b. c., and Fonseca, M. (2015). Towards localization of RFID tags based on experimental analysis of RSSI. *IFIP Wireless Days*, (January).