Redes e Sistemas Autónomos
Apontamentos
Parte II

Universidade de Aveiro

Sebastian D. González

Redes e Sistemas Autónomos
Apontamentos
Parte II

Dept. de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

sebastian.duque@ua.pt(103690)

25 de junho de 2024

Warning!!

Isto são apenas uns apontamentos realizados por uma pobre alma de MIECT, feitas a partir dos slides da disciplina e outras fontes 😈. Por favor, não usem apenas estes apontamentos como material de estudo.

Dito isto, boa sorte a todos e ámen CT 🙏.

Agradecimentos à Prof. Susana Sargento susana@ua.pt por todo o material fornecido nas aulas.

Conteúdo

Lista de Figuras

Glossário

AODV Ad Hoc On-Demand Distance Vector Routing.

BATMAN A Better Approach to Mobile Ad hoc Networking.

LAR Location-Aided Routing.

LoRa Long Range.

ML Machine Learning.

MPR Multipoint relays.

OLSR Optimized Link State Routing Protocol.

SGD Federated stochastic gradient descent.

Ad-Hoc Networks

Uma Rede Ad Hoc Móvel é um tipo de rede local que é formada espontaneamente para permitir a conexão de dois ou mais dispositivos sem fio. Essas redes são dinâmicas, permitindo que terminais apareçam e desapareçam a qualquer momento, com nós que podem atuar como routers ou terminais. As redes são independentes, podendo se fundir ou dividir a qualquer momento e podem coexistir com diferentes meios de acesso. [1]

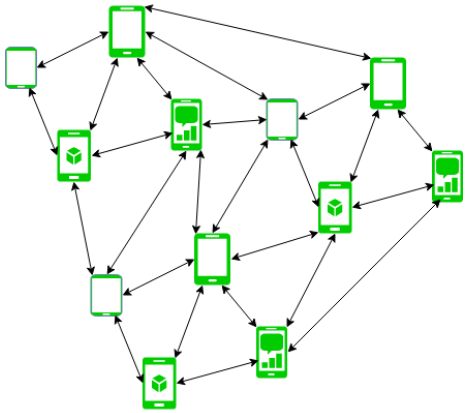


Figure - Mobile Ad Hoc Network

Além disso, são inteligentes e auto-organizadas, porém possuem as suas limirações:

⇒ **Limitações da rede sem fios:**

- Falta de uma entidade central para organização disponível
- Alcance limitado da comunicação sem fios
- Perda de pacotes devido a erros de transmissão
- Ligações de capacidade variável
- Desconexões/partições frequentes
- Banda de comunicação limitada

⇒ **Limitações impostas pela mobilidade:**

- Topologias/rotas em constante mudança de forma dinâmica
- Falta de consciência de mobilidade por parte do sistema/aplicações

⇒ **Limitações do computador móvel:**

- Vida útil curta da bateria
- Capacidades limitadas

1.1 Cenários de aplicação

Devido às redes ad-hoc não necessitarem de uma infraestrutura backbone de suporte e serem fáceis de implementar, são ideais para implementar quando o objetivo é ter redes auto-adaptáveis e auto-suficientes, especialmente em situações que requerem mobilidade ou onde exista a ausência de qualquer configuração externa e processo de gestão.

⇒ **Rede de Área Pessoal:**

- Telemóvel, computador portátil, auscultadores, relógio de pulso

⇒ **Ambientes Militares:**

- Soldados, tanques, aviões

⇒ **Ambientes Civis:**

- Rede de táxis, salas de reuniões, estádios desportivos, barcos, aviões pequenos...

⇒ **Operações de Emergência:**

- Busca e resgate, Policiamento, combate, incêndios...

1.2 Routing

1.2.1 Routing: Challenges and Requirements

Como já abordado noutras cadeiras, o routing é o processo de determinar o melhor caminho ou rota numa rede de comunicação para encaminhar dados de um ponto de origem para um destino. Em redes sem fio ad hoc, onde os dispositivos se comunicam diretamente uns com os outros sem uma infraestrutura de rede fixa, o routing apresenta desafios únicos e requesitos específicos para garantir uma comunicação eficiente e confiável.

Principais desafios:

- **Mobilidade**: Nós em movimento podem causar quebras de caminho, colisões de pacotes e laços transitórios.
- **Restrição de largura de banda**: O canal de comunicação é partilhado por todos os nós na região, resultando em limitações de largura de banda e fica mais propenso a erros
- **Concorrência dependente da localização**: Aumento da concorrência pelo acesso ao canal com o aumento do número de nós.

Principais requisitos:

- **Mínimo atraso na aquisição de rotas**.
- **Reconfiguração rápida de rotas:** para lidar com quebras de caminho.
- **Routing Loop-free:** para evitar desperdício de recursos.
- **Distributed routing approach:** para reduzir a largura de banda consumida e a sobrecarga de controlo.
- **Sobrecarga mínima de controlo:** para evitar problemas de largura de banda e colisões.
- **Escalabilidade:** para lidar com grandes redes, minimizando a sobrecarga de controlo.
- **Provisão de Qualidade de Serviço (QoS):** para suportar tráfego sensível ao tempo.
- **Segurança e privacidade:** garantindo que a rede seja resiliente a ameaças e vulnerabilidades.

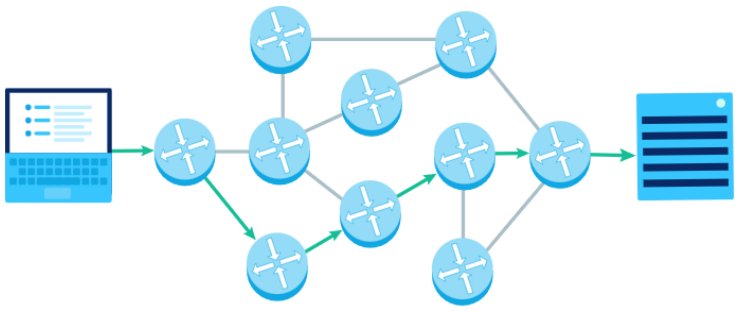


Figura 1.1: Routing [2]

1.2.2 Reactive protocols

- Menor sobrecarga, uma vez que as rotas são determinadas quando necessárias
- Atraso significativo na determinação de rotas
- Utilização de flooding
- O tráfego de controlo pode ser bursty

1.2.2.1 AODV

Ad Hoc On-Demand Distance Vector Routing é um protocolo de routing para redes ad hoc que estabelece rotas apenas quando necessário, reduzindo a disseminação de tráfego de controlo e eliminando a sobrecarga de tráfego de dados.

⇒ Cada nó mantém sua própria routing table e utiliza pacotes RREQ e RREP para estabelecer rotas.

⇒ Um nó intermediário também pode enviar um RREP desde que saiba de um caminho mais recente do que o anteriormente conhecido pelo remetente.

⇒ Os nós intermediários que encaminham o RREP também registam o next hop para o destino

RREQ - Route Request

- Quando um nó retransmite um RREQ, ele estabelece um caminho reverso apontando para a origem.
- O AODV pressupõe ligações simétricas (bidirecionais).
- Inclui o último número de sequência conhecido para o destino.

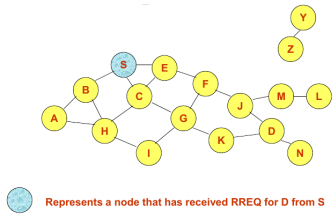


Figura 1.2: Route Requests in AODV(1)

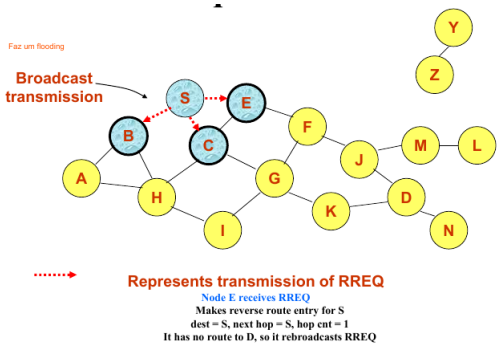


Figura 1.3: Route Requests in AODV(2)

RREP - Route Reply

- Quando o destino recebe um RREQ, ele responde enviando um RREP.
- A Route Reply segue ao longo do caminho reverso estabelecido quando o RREQ foi retransmitido.

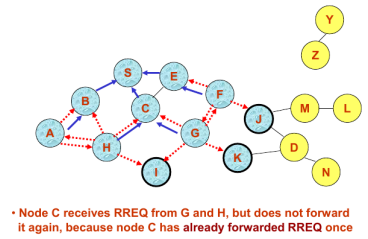


Figura 1.4: Reverse Path Setup in AODV(1)

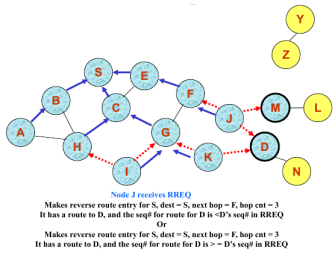


Figura 1.5: Reverse Path Setup in AODV(2)

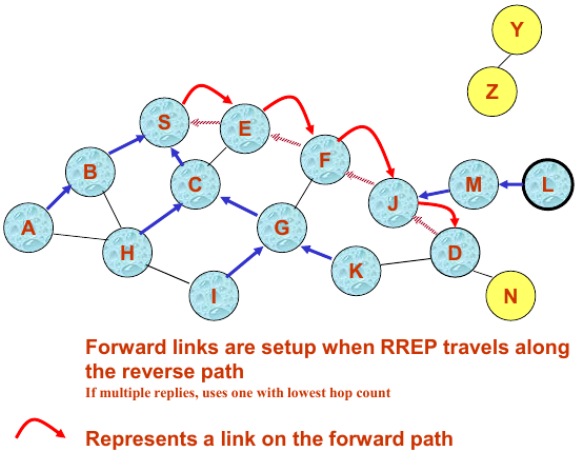


Figura 1.6: Forward Path Setup in AODV

Link Failure

No AODV, a deteção de falhas de ligação é realizada por meio de mensagens de `Route Error` (RERR), que atualizam os números de sequência de destino. Quando uma ligação de next hop numa entrada da routing table se rompe, todos os vizinhos ativos são informados.

⇒ Existe a troca periódica de mensagens hello entre os nós vizinhos, para manter e atualizar as informações das tabelas de routing

⇒ Um vizinho é considerado ativo numa entrada da routing table se o vizinho enviou um pacote dentro do intervalo de `active_route_timeout`.

Route Error

Quando um nó não consegue encaminhar um pacote para o destino, ele gera uma mensagem RERR, incrementa o número de sequência de destino e envia a RERR de volta para o nó de origem. O nó de origem inicia então um novo processo de descoberta de rota usando um número de sequência pelo menos tão grande quanto o número de sequência no RERR. O nó de destino atualiza o seu número de sequência para combinar o número de sequência no RERR, a menos que ele já seja maior. Isso deteta e lida com falhas de ligação numa rede.

1.2.3 Proactive protocols

- Mantém sempre as rotas
- Pouco ou nenhum atraso na determinação de rotas
- Consume largura de banda para manter rotas atualizadas
- Mantém as rotas que podem nunca ser utilizadas

1.2.3.1 OLSR

Optimized Link State Routing Protocol é um protocolo proativo que utiliza um mecanismo de encaminhamento de pacotes eficiente chamado Multipoint relays (MPR). O OLSR é projetado para funcionar bem em redes sem fio grandes e dinâmicas, comunicando informações de routing de forma eficiente sem usar `flooding`. No entanto, pode haver algum atraso na reação a alterações na topologia e na obtenção de parâmetros ótimos.

⇒ Cada dispositivo emite um **"Hello"** periódico, que se anuncia aos seus vizinhos, e determina quem mais está presente na rede e seleciona alguns sistemas para atuarem como MPR.

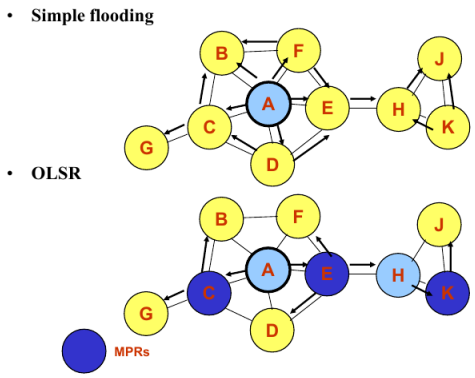


Figura 1.7: Example of MPR in OLSR

1.2.3.2 Link state fowarding

- Cada nó na rede tem um papel específico.
- Os nós C e E são MPR do nó A, ajudando na comunicação.
- Os MPR de A são selecionados de modo que cada 2-hop neighbors de A seja alcançável através de um único salto de um desses relay.
- Os nós trocam listas de vizinhos para identificar os seus 2-hop neighbors e selecionar os MPR de forma eficiente.

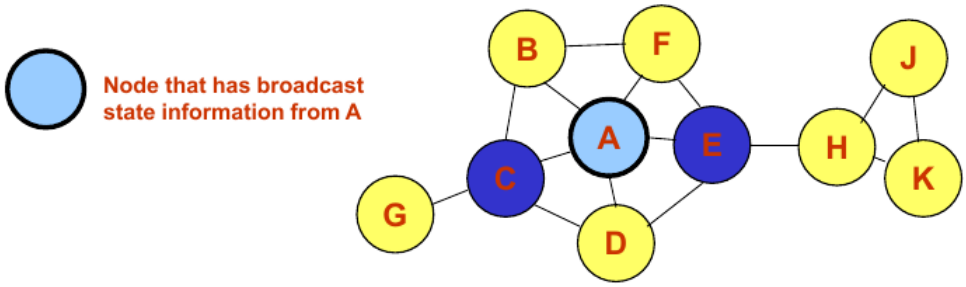


Figura 1.8: Link state fowarding

- Os nós C e E encaminham informações recebidas de A.
- Os nós E e K são MPR para o nó H.
- O nó K encaminha informações recebidas de H.

1.2.3.3 MPR sets and MPR selectors

- **MPR sets:**
  - Conjunto de nós que são MPR na rede.
  - Cada nó seleciona um conjunto MPR para processar e encaminhar cada link state packet que ele próprio origina.
  - Outros nós na rede processam os link state packets, mas não os encaminham.
- **MPR selectors:**
  - Conjunto de vizinhos que selecionaram o nó como MPR.
  - O nó encaminha pacotes recebidos dos nós MPR selectors.

Os MPR sets e MPR selectors não são fixos, eles podem mudar conforme a dinâmica da rede. Existem mecanismos eficientes de seleção que permitem que os nós se adaptem às mudanças na topologia da rede e nas condições de comunicação.

1.2.3.4 Processo de Seleção de um MPR:

- **Cálculo de Rotas:** Cada nó na rede calcula rotas para todos os destinos através dos membros do conjunto de MPR.

- **Decisão de Membro do Conjunto MPR:** Os nós decidem quais outros nós devem fazer parte do conjunto MPR. Isso geralmente é feito com base em critérios específicos, como os seguintes:
  - Selecionar como MPR cada nó na vizinhança de 2-hops que tenha um link bidirecional com o nó.
  - Selecionar como MPR os nós que cobrem nós "isolados", ou seja, nós que têm apenas um único nó vizinho que age como pai.
  - Selecionar como MPR o nó que cobre o maior número possível de outros nós na rede.

A troca de pacotes **"Hello"** também influencia na seleção de um MPR:

- Ao receber mensagens Hello, os nós atualizam suas Routing Tables de dois saltos.
- Com base nessas informações e nos critérios de seleção de MPR, cada nó seleciona seus próprios MPRs.

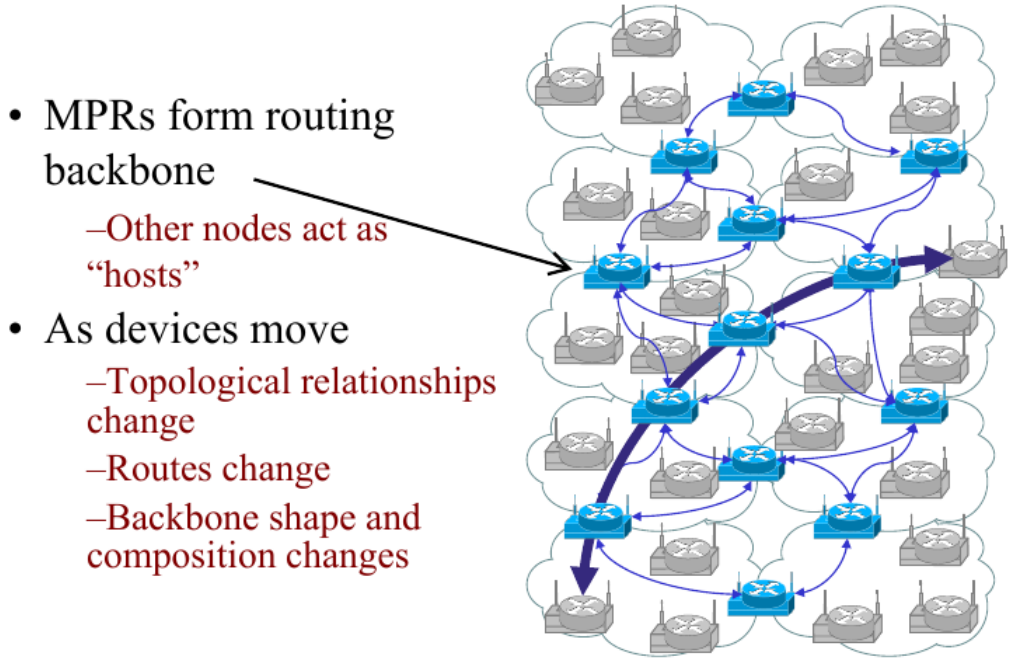


Figura 1.9: Structure of an OLSR Network

1.2.4 Location-based routing protocols

Os principais problemas dos mecanismos anteriores eram principalmente o facto da localização dos nós mudar rapidamente e a falta de informação sobre a localização atual, velocidade ou direção.

⇒ Sabendo a localização dos nossos nós na rede podemos Minimizar a zona de busca e evitar o uso de floodings.

⇒ Sabendo a velocidade e/ou direção podemos minimizar a zona de busca e aumentar a probabilidade de encontrar o nó necessário.

Foi daí que surgiram protocolos que nos dão alguma informação da localização.

1.2.4.1 LAR

Location-Aided Routing (LAR) é um método de routing em redes de comunicação sem fio onde cada nó tem conhecimento da sua própria localização geográfica e usa essa informação para auxiliar na descoberta e manutenção de rotas para enviar dados para outros nós na rede.

Esta abordagem ajuda a otimizar o processo de routing, tornando-o mais eficiente e adaptável às mudanças na topologia da rede.

O routing é feito usando a última localização conhecida + uma suposição:

- A descoberta de rota é iniciada quando:
  - S não conhece uma rota para D.
  - A rota anterior de S para D está quebrada.

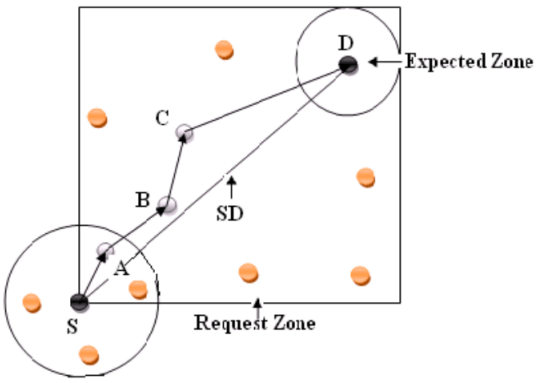


Figura 1.10: Directional-location aided routing scheme [3]

1.2.4.1.1 Expected Zone e Request Zone

Expected Zone: Quando um nó, digamos S, deseja enviar dados para outro nó, D, numa rede de comunicação sem fio, ele precisa conhecer a localização de D. A `EZ` é o local onde S espera que D esteja num determinado momento (t1). Isto é baseado na localização conhecida de D num momento anterior (t0) e na velocidade máxima ou média (v) em que D pode se mover. Portanto, a EZ é uma área onde S acredita que D provavelmente estará no momento em que os dados forem enviados.

Request Zone (RZ): É uma área na qual o nó S solicita uma rota para alcançar o nó D. Em vez de apenas enviar uma solicitação de rota para todos os nós na rede, S limita a difusão dessa solicitação para uma área específica, que é a RZ. Determinar o tamanho e a forma da RZ é crucial. Algumas considerações incluem:

- Se a EZ de D não incluir S, significa que S está fora do alcance direto de D. Portanto, S precisa expandir a RZ para incluir outras regiões onde D pode estar, além da EZ.
- Nem sempre é garantido que uma rota será encontrada usando uma determinada RZ. Isso pode depender da topologia da rede, da disponibilidade de nós intermediários e de outros fatores.

1.2.4.2 BATMAN

Tradicionalmente, os nós trocam pacotes de controlo que contêm informações sobre o estado da conexão (utilização atual da conexão, largura de banda, etc.). Os nós determinam os melhores caminhos com base nos pacotes de controlo e cada nó deve ter informações quase exaustivas sobre toda a rede.

O BATMAN adota uma abordagem muito diferente:

⇒ A presença ou ausência de pacotes de controlo é usada para indicar a qualidade da conexão (e do caminho).

1.2.4.2.1 Batman Operation

⇒ Cada nó tem um conjunto de vizinhos de ligação direta. Por exemplo, na figura 1.11, o nó A tem os vizinhos B e C. Esses são os nós pelos quais A envia e recebe todos os seus pacotes.

⇒ Cada nó na rede envia periodicamente uma Originator Message (OGM) para informar a todos os outros nós de sua presença.

- As OGMs incluem um número de sequência.

⇒ Se todas as ligações mostradas forem perfeitas, o nó A receberá a OGM do nó D através de ambos os seus vizinhos B e C.

- Se todas as OGMs de D chegarem através de B e C, então quando A precisar enviar algo para D, pode usar tanto B quanto C como next hop em direção ao nó de destino D.

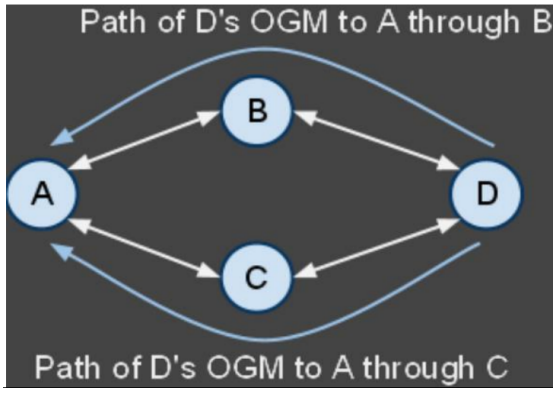


Figura 1.11: Batman OGM Operation

⇒ Se a ligação entre os nós A e C cair:

- A OGM do nó D só chegará ao nó A através do nó B.
- Portanto, o nó A considera o nó B como o melhor next hop para todos os pacotes destinados ao nó D.
- Além disso, as OGMs do nó C também só alcançarão o nó A através do nó B. Portanto, o nó B é o melhor next hop para dados destinados ao nó C.

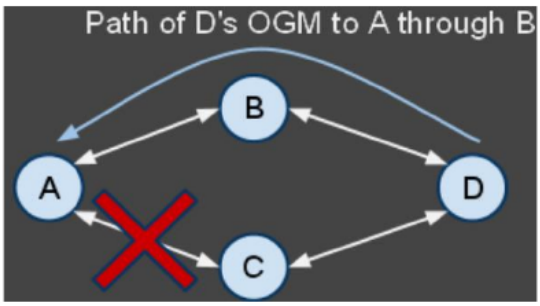


Figura 1.12: a ligação entre os nós A e C caiu

1.2.4.2.2 Sliding Window

Indica quais dos últimos `WINDOW_SIZE` (na figura 1.13 são 8) números de sequência foram recebidos através das OGMs.

	Out of Range				In Window Range								Out of Range			
Seq. Numbers:	...	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Arrived:	...	-	-	-	1	1	1	0	1	0	1	1	-	-	-	...

Figura 1.13: Sliding Window de 8 a 15

Quando um número de sequência fora do intervalo é recebido, neste caso, seq# 17, a janela desliza até 17.

	Out of Range				In Window Range								Out of Range			
Seq. Numbers:	...	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...
Arrived:	...	-	-	-	1	0	1	0	1	1	0	1	-	-	-	...

Figura 1.14: sliding window shifting

1.2.4.2.3 Routing table

Todos os nós têm uma sliding window para cada originador (outro nó) na rede para cada vizinho.

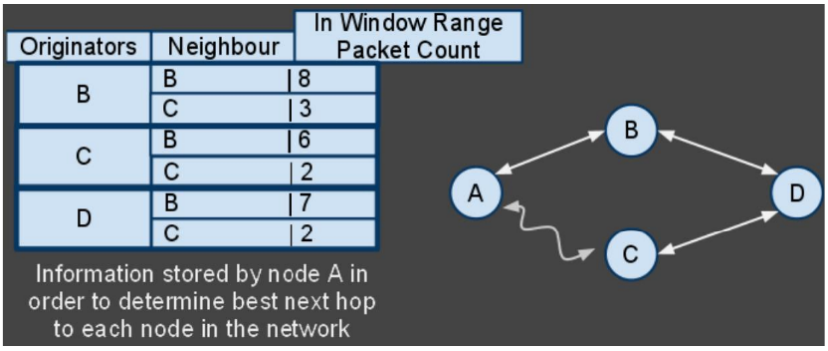


Figura 1.15: BATMAN Routing table

1.3 IP Address Assignment

1.3.1 MANETconf

O MANETconf é um algoritmo de exclusão mútua distribuída usado para verificar a singularidade de um endereço IP. A atribuição de um endereço a um novo nó requer um acordo de todos os nós conhecidos na rede.

- Cada nó tem uma tabela global de alocação de endereço em que mantém os endereços atualmente em uso
- O nó (solicitante) difunde uma mensagem NEIGH_REQ aos vizinhos de um salto
- Cada vizinho responde ao solicitante com uma mensagem NEIGH_REP
- O nó solicitante seleciona um dos vizinhos como seu agente, que realiza a alocação de endereço em nome do solicitante
- Em seguida, envia um REQUESTER_REQ ao agente para solicitar um endereço
- O agente escolhe um endereço atualmente não utilizado de sua tabela e difunde um ADDR_REQ para obter um acordo de todos os outros nós configurados na rede
- Cada nó na rede envia uma resposta ADDR_REP de volta ao agente
- Se o agente receber uma resposta de todos os outros nós, ele atribui o endereço ao solicitante enviando ADDR_ALLOC

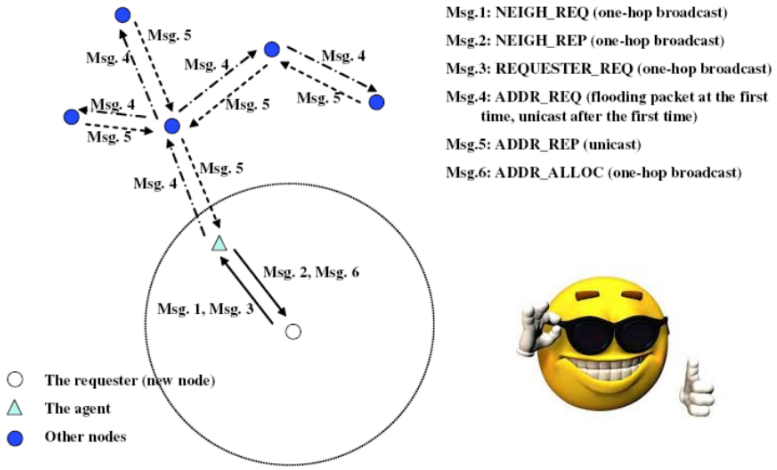


Figura 1.16: MANETconf

1.3.2 PyMesh Addressing

É um sistema de comunicação ad-hoc que utiliza LoRa. Ele usa múltiplos gateways para conectar a rede local à internet. Os nós na rede ligam dados internos à nuvem e usam um protocolo `Listen Before Talk` para evitar colisões. A segurança é implementada em vários níveis. Dispositivos LoRa podem assumir diferentes papéis na rede, como líder, router, filho ou Border Router.

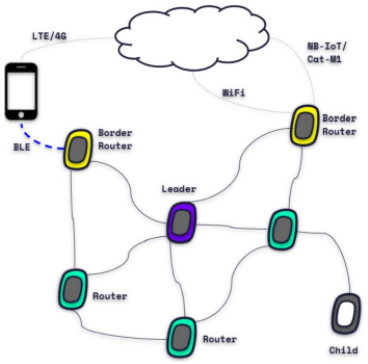


Figura 1.17: PyMesh

1.3.2.1 PyMesh Addressing and Communication

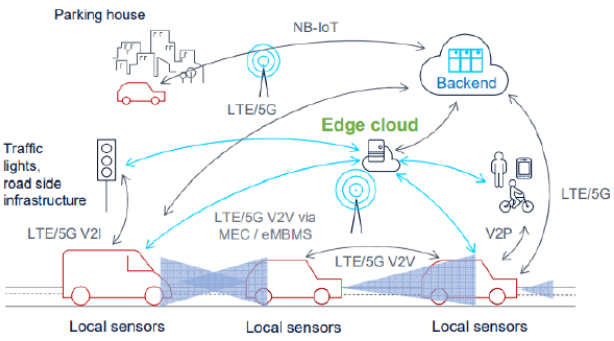
- Um prefixo de endereço de rede é declarado para o border router, por exemplo, ”2001:dead:beef:cafe::/64”.
- Esse prefixo de endereço de rede é enviado para o líder da rede.
- Todos os nós recebem um endereço unicast IPv6 aleatório com esse prefixo de rede. Por exemplo, ”2001:dead:beef:cafe:1234:5678:9ABC:DEF0”.
- Se um nó envia dados para um endereço IPv6 externo (um prefixo de rede inexistente no PyMesh), o datagrama UDP será roteado para o Border router.
- Este pacote UDP terá como origem o endereço IPv6 aleatório do router de fronteira.
- O router de fronteira receberá os datagramas UDP externos com um cabeçalho anexado, contendo:
  - Um byte ”MAGIC”(0xBB).
  - O destino IPv6 como um array de bytes (16 bytes).
  - O destino da porta como 2 bytes (valores de 1 a 65535).

O endereço IPv6 de destino é importante porque permite que o router de fronteira encaminhe o conteúdo do datagrama UDP para a interface correta (Wi-Fi, BLE, celular).

Self-organized systems: Data, learning and decisions

No contexto do processamento de dados em redes de comunicação, tomadas de decisão são fundamentais, abrangendo desde otimizações de rede até a personalização de experiências para os utilizadores. Este processo de análise de dados envolve decisões tais como:

- **Rede e Serviços:** Decidir se aumentar a largura de banda ou priorizar filas para reduzir atrasos.
- **Utilizador ou Sistema:** Determinar se há obstáculos ou controlar ações, como direção de um robô.
- **Rede com Dados dos utilizadores:** Antecipar movimentos e necessidades dos utilizadores, ajustando conteúdo da CDN e priorizando emergências.
- **Utilizador com Dados da Rede:** Escolher caminhos com melhor conexão para atividades específicas, como jogos ou streaming, e selecionar locais adequados para experiências de realidade virtual ou aumentada.



2.1 Machine Learning

Defenição Pragmática:

⇒ Uma coleção de algoritmos e modelos estatísticos para que as máquinas realizem tarefas automatizadas com base na observação de entradas e/ou saídas de um processo. O objetivo do Machine Learning é produzir uma estimativa ou classificação dada um conjunto de valores de entrada.

2.1.1 Types of Learning

2.1.1.1 Supervised

Supervised learning é um tipo de Machine Learning que utiliza dados rotulados para treinar algoritmos para prever resultados ou reconhecer padrões. Ele envolve o uso de um conjunto de dados de treino que inclui entradas e valores de saída corretos, permitindo que o modelo aprenda ao longo do tempo através de um processo de ajuste de parâmetros para minimizar erros

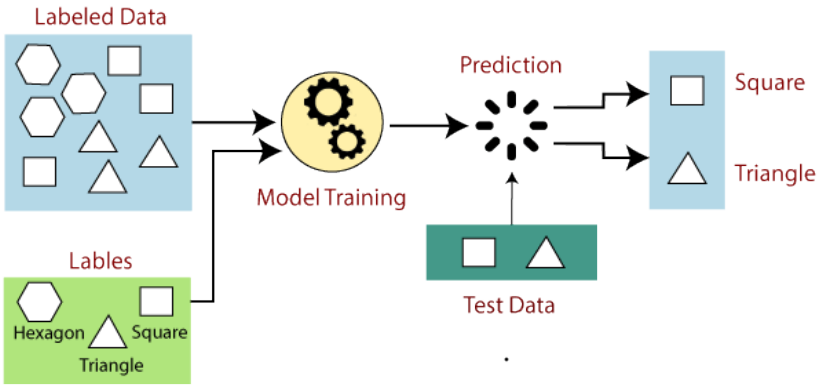


Figura 2.1: Supervised Machine Learning [4]

A **ground truth** é usada como uma referência para avaliar o desempenho do modelo durante o treino e também para avaliar sua precisão ao fazer previsões sobre novos dados. A qualidade e a precisão da ground truth são cruciais para garantir que o modelo seja treinado corretamente e possa generalizar bem para novos dados. Possui duas fases:

- **Training stage:** O modelo aprende com dados rotulados(padrões e relações entre as entradas e as saídas esperadas) durante esta fase.

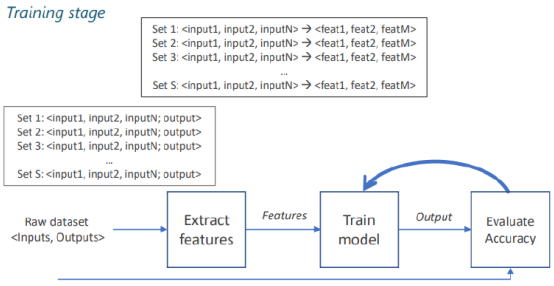


Figura 2.2: Training stage

- **Inference stage:** O modelo faz previsões ou inferências com base no que aprendeu durante o treino, quando confrontado com novos dados, sem acesso à ground truth

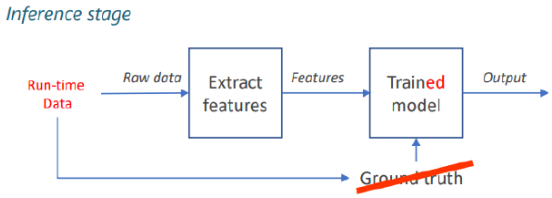


Figura 2.3: Inference stage

2.1.1.2 Unsupervised

A aprendizagem não supervisionada é um método de aprendizagem de máquina em que algoritmos aprendem padrões exclusivamente a partir de dados não rotulados, sem saídas correspondentes.

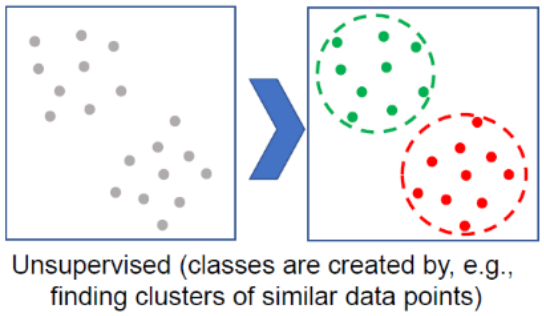
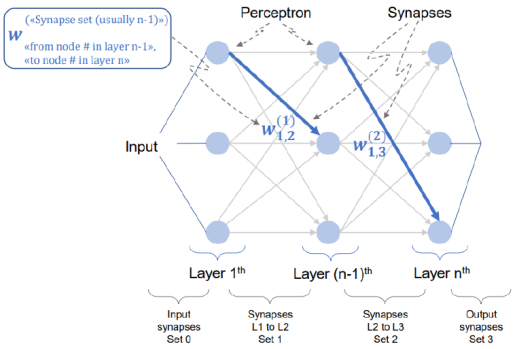


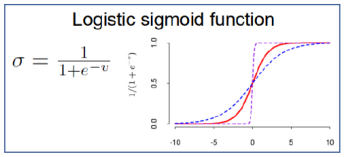
Figura 2.4: Unsupervised Learning

2.1.2 Redes Neuronais

Um dos métodos mais bem-sucedidos de ML que utiliza blocos de construção:

- **Perceptron:** tipicamente uma função que mapeia todo o intervalo natural num intervalo limitado ([0,1] ou [-1,1])
  - Exemplo: função logística sigmoide, função ReLU, tanh, softmax, etc.
- **Sinapses:** Conexões dos perceptrons da camada (n-1) para perceptrons da camada n, cada uma aplicando um peso ao valor transmitido.
  - Treinar Redes Neuronais é principalmente sobre encontrar os pesos dessas sinapses





2.1.3 Reinforcement Learning

É tipo de técnica de ML que permite a um agente aprender num ambiente interativo por tentativa e erro, usando feedback de suas próprias ações e experiências.

1. **Ambiente** - Mundo físico no qual o agente opera
2. **Estado** - Situação atual do agente
3. **Recompensa** - Feedback do ambiente
4. **Política** - Método para mapear o estado do agente para ações
5. **Valor** - Recompensa futura que um agente receberia ao tomar uma ação num estado específico

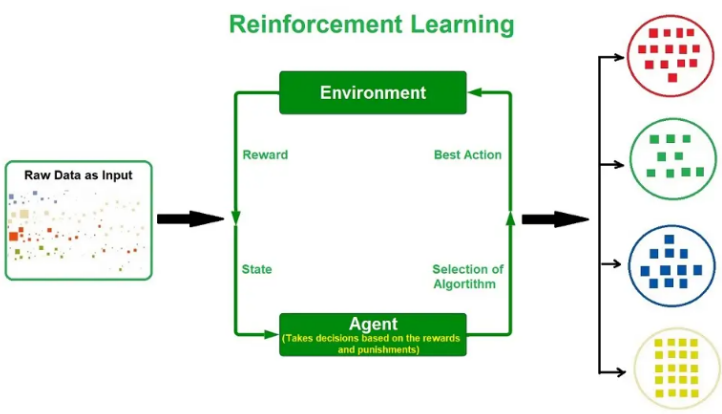


Figura 2.5: Reinforcement Learning [5]

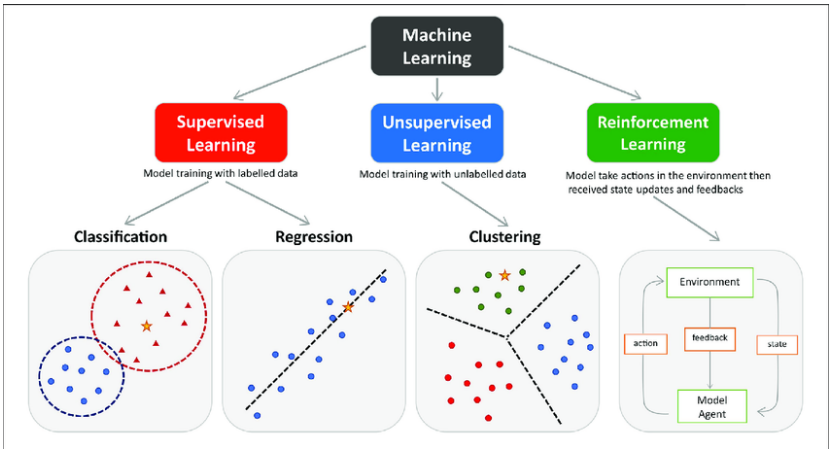


Figura 2.6: The main types of machine learning [6]

2.1.4 K-means

Centroide: pontos non-data que indicam o centro do cluster conforme identificado pelo algoritmo K-means.

1. Implantar N centroides aleatoriamente (N proporcional ao número de classes esperadas).

2. Atribuir aleatoriamente pontos de dados às classes.

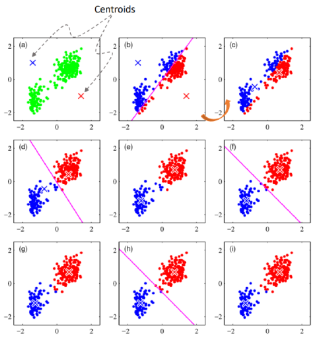
3. Repetir iterativamente:

(a) Calcular o centro de gravidade de cada classe.

(b) O centroide é reposicionado nesse centro de gravidade.

(c) Atualizar o limite.

4. Parar quando as atualizações se tornarem negligenciáveis.



2.2 Learning

2.2.1 Centralized

É um paradigma de machine learning no qual os modelos de aprendizagem são treinados num ambiente centralizado usando dados coletados de um grupo de dispositivos distribuídos. Nesse cenário, os dados dos dispositivos são agregados num local central, onde o modelo é treinado.

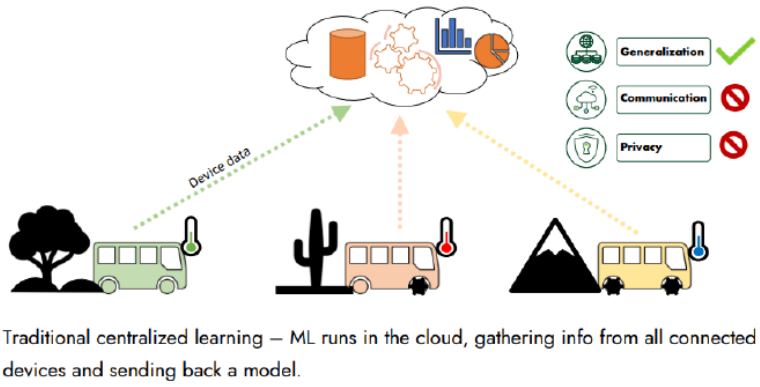


Figura 2.7: Machine Learning Centralized

2.2.2 De-Centralized

É um paradigma onde o processo de aprendizagem de máquina ocorre localmente, em cada dispositivo conectado, em oposição à centralização dos dados e do treino do modelo num servidor remoto.

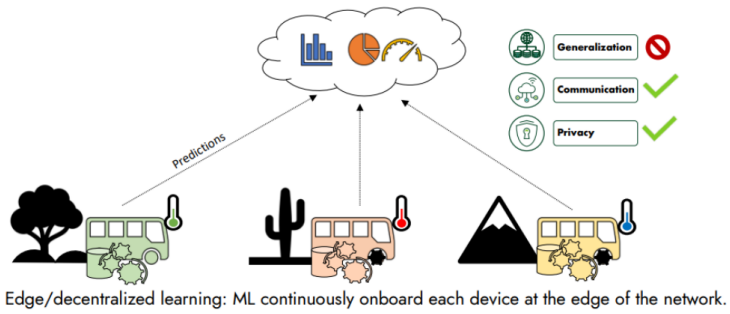


Figura 2.8: Machine Learning De-Centralized

2.2.3 Federated

É um paradigma que visa treinar algoritmos em dispositivos periféricos distribuídos, mantendo as amostras de dados localmente. O Google iniciou como o principal interveniente com o objetivo de treinar modelos de Aprendizagem Automática em bilhões de telemóveis, respeitando a privacidade dos utilizadores. A abordagem consiste em enviar apenas frações dos resultados de treino, ou seja, derivadas do treino, para a nuvem, nunca armazenando qualquer informação no dispositivo.

⇒ Quando recolhidos na nuvem, os resultados parciais do treino podem ser montados para formar um novo supermodelo que, na próxima etapa, pode ser enviado de volta para os dispositivo

⇒ Novos dispositivos podem começar com um modelo de um dispositivo semelhante, em vez de um Super Model.

2.2.3.1 Federated learning: iterative

O Federated Learning emprega um método iterativo contendo múltiplas trocas cliente-servidor, conhecidas como `federated learning round`:

- Difundir o estado atual/atualizado do modelo global para os nós contribuintes (participantes).
- Treinar os modelos locais nesses nós para obter certas atualizações potenciais do modelo a partir dos nós.

- Processar e agregar as atualizações dos nós locais numa atualização global agregada, de modo que o modelo central possa ser atualizado em conformidade.

Figura 2.9: Federated learning: iterative

O servidor FL é usado para esse processamento e agregação das atualizações locais para atualizações globais. O treino local é realizado pelos nós locais. Além disso, a Federated learning distribui a aprendizagem profunda ao eliminar a necessidade de reunir os dados num único local. Na aprendizagem federada, o modelo é treinado em diferentes locais em várias iterações.

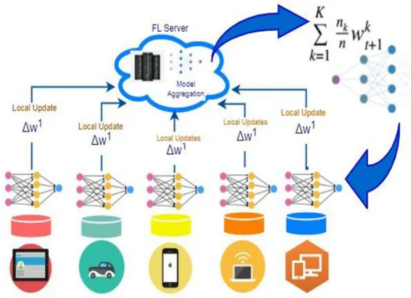


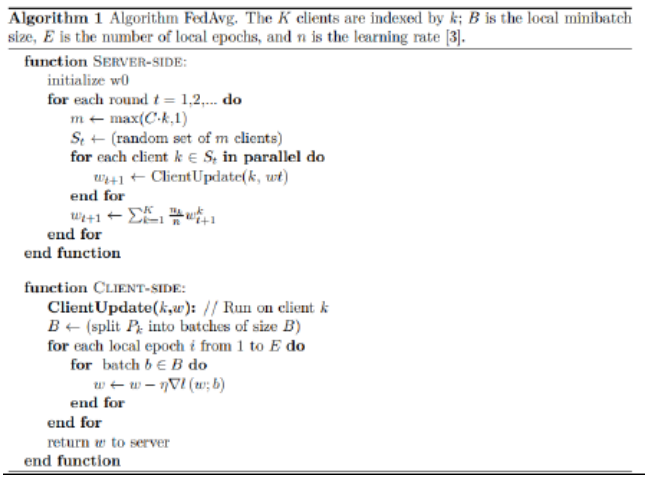
Figura 2.10: Federated learning

2.2.3.2 Learning: model aggregation

- A agregação eficaz de modelos distribuídos é essencial para criar um modelo global generalizado afetando a precisão, o tempo de convergência, o número de rondas e a sobrecarga de rede.
- O Federated stochastic gradient descent (SGD) utiliza uma única instância do conjunto de dados para o treino local em cada cliente por ronda de comunicação, sendo o algoritmo base da aprendizagem federada.
- O FedAvg parte do SGD, com cada cliente realizando um treino local utilizando seus próprios dados e o modelo atual, aplicando múltiplos passos

de SGD, e os modelos atualizados são enviados de volta para o servidor para agregação.

- O FedAvg reduz a sobrecarga de comunicação, mas exige mais computação total por parte dos clientes durante o treino.
- Um "Local epoch"refere-se a uma passagem completa do conjunto de dados de treino pelo algoritmo.



Algorithm 1 Algorithm FedAvg. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and n is the learning rate [3].

function SERVER-SIDE:
  initialize w0
  for each round $t = 1,2,...$ **do**
    $m \leftarrow \max(C \cdot k, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{k+1} \leftarrow \text{ClientUpdate}(k, wt)$
    end for
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
  end for
end function

function CLIENT-SIDE:
  ClientUpdate(k,w): // Run on client k
  $B \leftarrow$ (split P_k into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in B$ **do**
      $w \leftarrow w - \eta \nabla l\,(w; b)$
    end for
  end for
  return w to server
end function

Figura 2.11: Model Aggregation

⇒ **Fault Tolerant Federated Average:** permite que um sistema continue a funcionar mesmo em caso de falha, tolerando alguns nós offline durante a agregação segura.

⇒ **Q-Federated Average:** ajusta o objetivo para alcançar equidade no modelo global, atribuindo pesos mais elevados aos dispositivos com baixo desempenho, tornando a distribuição de precisão na rede mais uniforme.

⇒ **Federated Optimization:** utiliza um otimizador de cliente durante várias épocas de treino e um otimizador de servidor durante a agregação do modelo, com exemplos de otimizadores incluindo ADAGRAD, ADAM e Yogi.

Bibliografia

[1] geeksforgeeks, *Introduction of Mobile Ad hoc Network (MANET)*, https://www.geeksforgeeks.org/introduction-of-mobile-ad-hoc-network-manet/.

[2] M. Kubilius, *A Comprehensive Guide to Network Routing*, https://www.ipxo.com/blog/network-routing/, 2022.

[3] S. Das, *Feasibility Evaluation of VANET using Directional-Location Aided Routing (D-LAR) Protocol*, https://www.researchgate.net/publication/233898931_Feasibility_Evaluation_of_VANET_using_Directional-Location_Aided_RoutingD-LAR_Protocol, 2021.

[4] JavatPoint, *Supervised Machine Learning*, https://www.javatpoint.com/supervised-machine-learning.

[5] N. Arya, *Three Types of Machine Learning*, https://www.ejable.com/tech-corner/ai-machine-learning-and-deep-learning/types-of-machine-learning/.

[6] J. Peng, E. C, J. P. Dönnes e C. Ciurtin, *Machine Learning Techniques for Personalised Medicine Approaches in Immune-Mediated Chronic Inflammatory Diseases: Applications and Challenges*, https://www.researchgate.net/publication/354960266_Machine_Learning_Techniques_for_Personalised_Medicine_Approaches_in_Immune-Mediated_Chronic_Inflammatory_Diseases_Applications_and_Challenges, 2021.