HyperDetect: A Real-Time Hyperdimensional Solution for Intrusion Detection in IoT Networks

APRESENTAÇÃO ARTIGO 2

APRENDIZADO DE MÁQUINA NA SAÚDE

AUGUSTO CESAR DA F. DOS SANTOS

Sumário

- **Contextualização**
- **Motivação**
- **Objetivo**
- **Classificação Hiperdimensional**
- **Operações Básicas em HDC (Hyperdimensional Computing)**
- **Metodologia**
- **Resultados Experimentais**
- **Conclusão**
- **Referências**

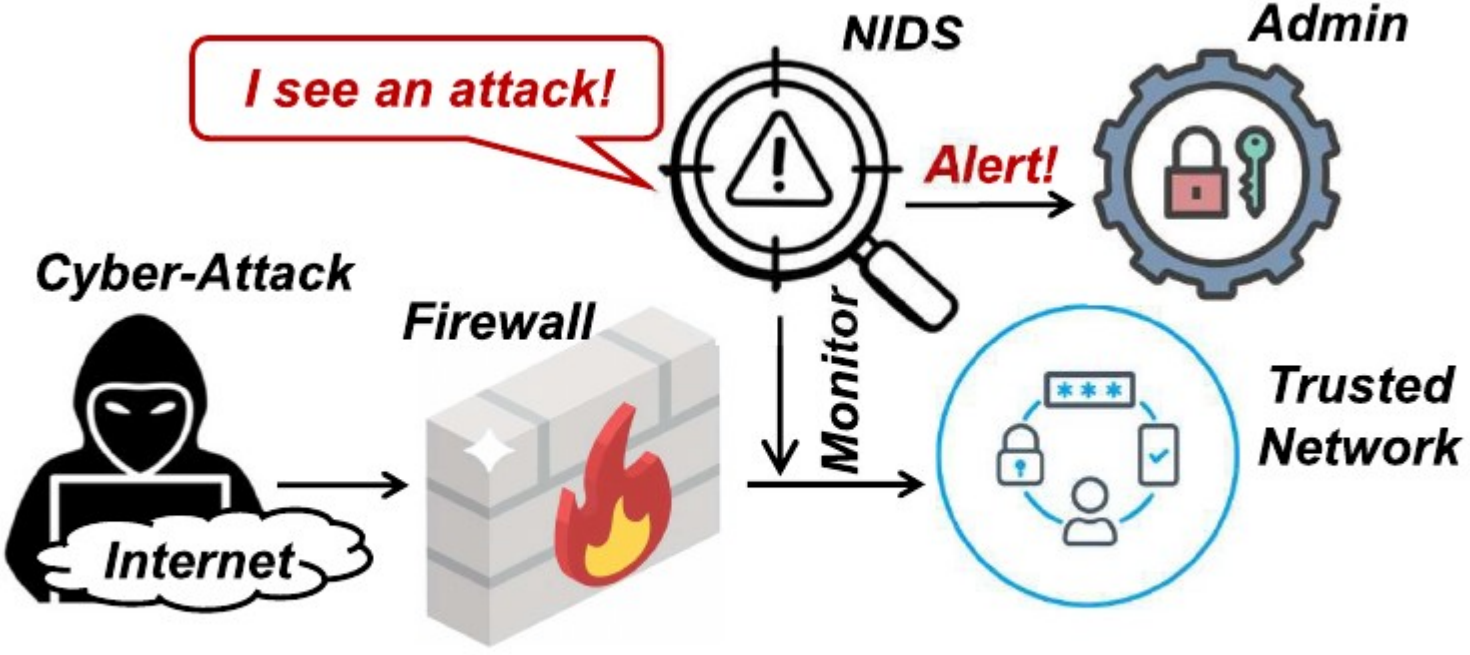


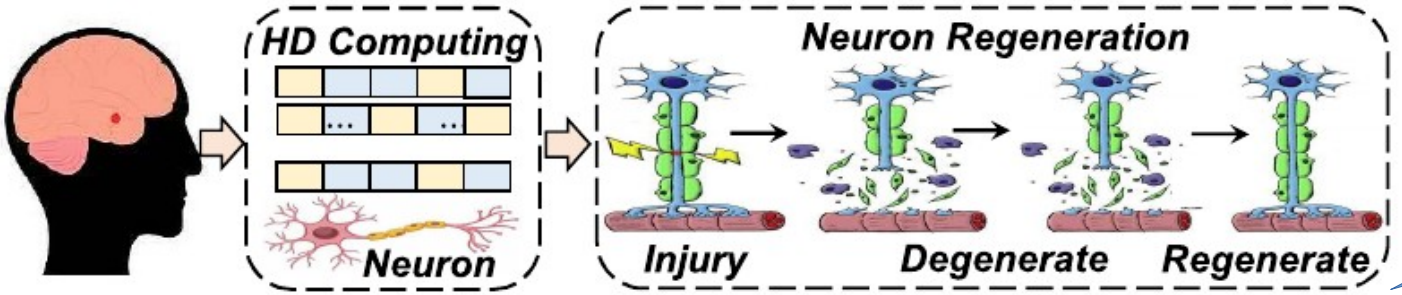
Fig. 1. Model of NIDS.

Contextualização

- Diversos **sistemas de detecção de intrusão em rede (NIDS)**, tipicamente baseados em algoritmos sofisticados de aprendizado de máquina (ML), foram propostos para monitorar o tráfego de rede e detectar atividades maliciosas;
- No entanto, esses designs de NIDS **exigem memória e poder computacional que superam a capacidade dos dispositivos IoT**, e muitas vezes falham em fornecer uma detecção oportuna de ataques de rede;
- Para resolver esse problema, propôs-se o **HyperDetect,** a primeira tentativa de modelagem de NIDS que aproveita as operações altamente eficientes e paralelas da **Computação Hiperdimensional (HDC); e**
- A motivação do framework HyperDetect proposto vem da **regeneração dinâmica de neurônios nos cérebros humanos.**

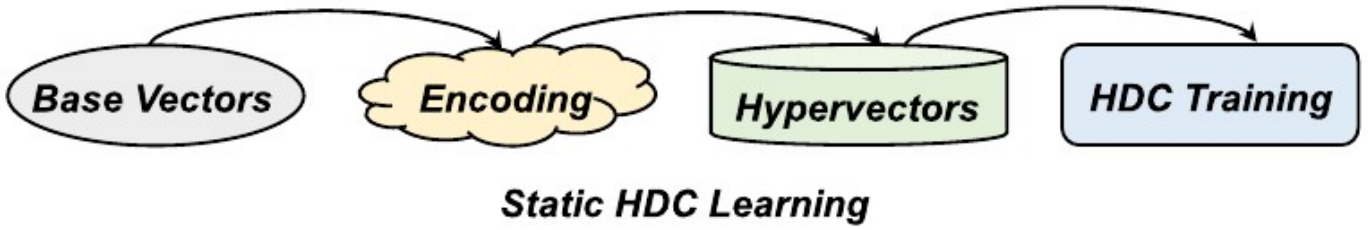
Motivação

- HDC utiliza a **alta dimensionalidade de vetores gerados aleatoriamente para representar informações como um padrão de atividade neural**, porém, as técnicas de **HDCs existentes** dificilmente conseguem suportar um comportamento semelhante; e
- HyperDetect que é um **modelo otimizado** que alcança uma detecção de intrusão eficaz com **menos iterações de treinamento e dimensões mais baixas**, **acelerando** significativamente **tanto o treinamento quanto a inferência ao eliminar cálculos desnecessários.**
  - **Opera de maneira bidirecional**, permitindo que os vetores base e os módulos de codificação sejam adaptáveis e **atualizados em cada iteração de treinamento**



Neurônios nos cérebros humanos mudam e se regeneram dinamicamente o tempo todo e fornecem funcionalidade mais útil ao acessar novas informações.

(a)



O modelo HDC é treinado uma única vez com um conjunto de dados fixo e depois aplicado sem mais ajustes. O modelo gera uma representação hiperdimensional para os dados de entrada que permanecem fixas ao longo de todo o uso.

O modelo HDC atualize suas representações de dados e parâmetros ao longo do tempo, tornando-o adaptativo às mudanças nos padrões de dados

(b)

Objetivo

- HyperDetect oferece, em média, um **treinamento** 5,02× **mais rápido e uma inferência 31,83× mais rápida** em comparação com as abordagens de aprendizado de ponta (SOTA) em uma **ampla gama de tarefas de classificação de intrusões de rede;**
- Inspirado em **técnicas de otimização** que levam em conta **não apenas o valor atual do erro, mas também a direção e intensidade das mudanças anteriores**,pode estabilizar e acelerar o aprendizado;
  - O modelo **não apenas analisa o valor atual dos dados**, mas também observa **como esses valores estão mudando ao longo do tempo;**
  - **Detecta se certas características do tráfego estão aumentando, diminuindo, ou se mantendo estáveis, e usa essa informação para ajustar suas previsões ou decisões; e**
  - Por exemplo, se um **padrão anômalo começa a surgir gradualmente** (como um aumento lento e constante na frequência de certos pacotes), **o modelo pode perceber essa direção de mudança e prever que há uma possibilidade maior de atividade maliciosa antes que o comportamento se torne um problema crítico.**

Objetivo

- Utiliza técnica para **regenerar ou atualizar as "dimensões" do modelo;**
  - Permite que o modelo **"aprenda" ou "reaprenda" de forma incremental, adaptando-se às novas variações no tráfego sem precisar ser treinado do zero**, processo torna a detecção de intrusões mais robusta e responsiva;
- O modelo utiliza um **"design space search"**, ou seja, uma exploração abrangente do espaço de design de configurações e parâmetros , **utilizando um algoritmo genético (GA);**
  - Um algoritmo genético é uma **técnica de otimização** inspirada em processos de evolução natural (seleção, cruzamento e mutação) que **permite identificar as melhores combinações de hiperparâmetros para o modelo de detecção de intrusões.**
  - Essa exploração pode **incluir ajustes de várias variáveis, como taxa de aprendizado, taxa de atualização das dimensões, entre outras.**

Classificação Hiperdimensional

- HDC é um **novo modelo computacional inspirado na neurociência cognitiva**, que usa **vetores de alta dimensão** para processar informações de uma maneira **parecida com o cérebro humano** para lidar com informações de um jeito rápido e eficiente;
  - O **cérebro não guarda as informações em um formato de lista ou de tabela**, mas em um jeito mais complexo, como se fosse um **"mapa mental" onde muitas ideias estão conectadas**
- Um vetor de alta dimensão é como uma **lista enorme de números, que pode ter centenas ou milhares de elementos**. Cada um desses números pode **representar uma parte pequena da informação total**;
- No processamento de informações, o HDC pode **reconhecer padrões rapidamente**
  - Exemplo: Em uma rede IoT, o HDC pode **identificar sinais de intrusão sem precisar processar cada detalhe individualmente**, apenas **comparando o vetor de um momento específico com outros vetores que representam situações normais ou anormais.**

Classificação Hiperdimensional

- Uma **propriedade única do espaço de alta dimensão** é a existência de um grande número de hipervetores **quase ortogonais**
  - Hipervetor é uma **lista enorme de números (ou valores)** que usa milhares de características para **descrever um objeto de forma detalhada**
- Matematicamente, **dois hipervetores aleatórios H1 e H2 com dimensão D**; quando D é grande o suficiente, **o produto escalar H1 · H2 ≈ 0**. Essa propriedade **permite operações altamente eficientes e paralelas**, como **cálculos de similaridade**, agrupamentos **(*bundlings*)** e ligações **(*bindings*)**
  - No caso dos **hipervetores quase ortogonais**, significa que eles são, na prática, **"bem diferentes" uns dos outros,** ou seja, quando os comparamos, eles são **extremamente diferentes em todas as direções ou são quase perpendiculares;**
  - Essa propriedade especial **ajuda a distinguir informações complexas,** porque permite que **vetores representem ideias ou objetos distintos de forma clara e bem separada**.

Encoding

- Funciona como uma **forma de "mapeamento" dos dados** em um espaço onde padrões complexos podem ser mais facilmente identificados;
- Cada **amostra (por exemplo, um pacote de dados ou sequência de tráfego de rede) é representada em um espaço de alta dimensionalidade**, geralmente com milhares de dimensões
  - Esse processo de codificação **expande os dados originais para capturar complexidades** que não seriam visíveis em um espaço de menor dimensão;
- Cada **amostra de dados é multiplicada por uma matriz de projeção composta de vetores base de alta dimensionalidade gerados aleatoriamente**, e cada um representa uma característica diferente do sistema
  - Assim, mesmo elementos de dados semelhantes podem ser projetados para diferentes pontos no espaço hiperdimensional, **o que permite uma diferenciação sutil entre padrões normais e anômalos.**

Encoding

- **O resultado dessa codificação é um hypervector — um vetor hiperdimensional com milhares de elementos que representam a amostra no novo espaço**
  - Cada hypervector encapsula a informação de forma distribuída, **onde cada componente contém uma pequena parte da informação global**.
- Um princípio fundamental da codificação permanece consistente: **a correlação de distância nos dados originais deve ser devidamente preservada durante a codificação**
  - Após a transformação para o espaço hiperdimensional, essa correlação de distância é importante para manter uma certa consistência: **se duas amostras são semelhantes nos dados originais, espera-se que suas representações hiperdimensionais também sejam relativamente próximas**, refletindo essa similaridade
  - Isso garante que a representação codificada mantenha os relacionamentos e estruturas essenciais no conjunto de dados original e, assim, previne a perda de informações
  - **Em detecção de intrusões, isso é útil para que o sistema reconheça padrões similares de comportamento e agrupe comportamentos normais, separando-os dos anômalos.**

Operações Básicas em HDC

- **Similaridade**:
  - Nos sistemas de detecção de intrusão, **o objetivo é identificar comportamentos ou padrões incomuns** que podem indicar uma tentativa de invasão
  - A operação de *similaridade* é uma técnica essencial para isso, **pois ela permite comparar novos dados** (como o tráfego atual da rede) **com padrões conhecidos** (como o comportamento normal da rede)
  - No contexto do HDC, **essa operação de similaridade envolve medir a "distância" entre dois *hipervetores***: um que **representa o dado atual (chamado *hipervetor de consulta*)** e outro que **representa o comportamento normal ou conhecido (chamado *hipervetor de classe*)**.
  - Em um sistema de detecção de intrusão usando HDC, **a operação de similaridade é usada para monitorar continuamente o tráfego da rede**

Operações Básicas em HDC

- ***Bundling (+):***
  - A operação de bundling pode ser usada para ***"lembrar"* informações importantes**, assim como o cérebro humano faz para armazenar memórias
  - **Memória de Padrões**: Ao somar vários hipervetores (que representam diferentes padrões de tráfego de rede ou comportamentos de usuários), **o sistema cria um novo hipervetor que "lembra" esses padrões**. Se um novo tráfego (consultado como um hipervetor) for parecido com esse "pacote" de padrões, o sistema pode identificar que o comportamento é normal.
    - **Exemplo de Detecção de Intrusão**:
      - Digamos que o sistema já tenha registrado dois padrões de tráfego de rede seguros: **um de tráfego de emails e outro de navegação web.**
      - Esses dois padrões (hipervetores) são somados para criar um ***hipervetor combinado*** que representa o tráfego "normal".
      - Quando chega um **novo tráfego**, o sistema verifica se o padrão desse tráfego é semelhante ao *hipervetor combinado*. Se for, significa que é um comportamento esperado. Se não for, o sistema pode alertar sobre uma possível intrusão.

Operações Básicas em HDC

- ***Bundling (+):***
  - *Hbundle = H1 + H2 (generating a hypervector with the same dimension as inputs.)*
  - *δ(Hbundle, H1) >= 0 while δ(Hbundle, H3) ≈ 0 (H3 != H1, H2)*
  - *Bundling models how human brains memorize input information.*

Operações Básicas em HDC

- ***Binding (*):***
  - É uma operação que envolve a **multiplicação de dois hipervetores** para criar um **novo hipervetor** que **representa a associação** dos dois originais;
  - O resultado é um **novo hipervetor, chamado Hbind,** que contém informações dos dois hipervetores, mas de maneira tal que Hbind é **quase ortogonal** a ambos os hipervetores originais.
    - No contexto da matemática, **dois vetores são ortogonais quando o produto interno entre eles é zero**, o que significa que eles estão em direções independentes no espaço de alta dimensão.
    - Quando dizemos que o Hbind é **ortogonal a H1 e H2, isso significa que o novo hipervetor é muito diferente de ambos, mas ainda contém informações deles.**
    - A operação de ***binding* é reversível**, o que significa que, ao multiplicar Hbind por um dos hipervetores originais, podemos recuperar o outro.

Operações Básicas em HDC

- ***Binding (*):***
  - **Exemplo de Detecção de Intrusão**:
    - Quando o sistema recebe **novos dados de tráfego**, ele pode calcular se esses dados se assemelham ao hipervetor combinado Hbind (que representa a associação de tráfego HTTP e e-mail).
    - **Se o tráfego atual se parecer muito com o hipervetor combinado**, o sistema pode concluir que o comportamento é esperado (ou seja, uma combinação normal de tráfego de rede).
    - **Se o tráfego atual não for semelhante ao Hbind,** mas for algo diferente, o sistema pode alertar sobre uma intrusão, porque o padrão não corresponde a um comportamento normal esperado.

Operações Básicas em HDC

- ***Binding (*):***
  - Hbind that is nearly orthogonal to both H1 and H2, i.e., Hbind = H1 ∗ H2 , where δ(Hbind, H1) ≈ 0 and δ(Hbind, H2) ≈ 0
  - Due to reversibility, i.e., Hbind ∗ H1 = H2 , information from both hypervectors can be preserved
  - Binding models how human brains connect input information, i.e., **associating the information** of multiple objects into a single hypervector.

Aprendizado HDC

- **Treinamento:** Descreve como o sistema de detecção de intrusão aprende com exemplos de tráfego de rede, **ajustando seu modelo para classificar corretamente o tráfego como normal ou anômalo (intrusões)**
  - **O primeiro passo do aprendizado é gerar *hipervetores codificados* para cada classe.**
  - Uma ***classe*** pode ser um **tipo de tráfego de rede, como "navegação web", "envio de e-mails", "tráfego de dados normal", e "intrusão"**. Cada classe terá seu próprio conjunto de dados (ou amostras) para treinamento.
  - Após gerar os hipervetores para todas as amostras de uma classe (como tráfego de rede normal ou intrusivo), o sistema **faz um "pacote" de todos esses hipervetores** para formar um **"hipervetor de classe"**, que representa a classe de tráfego de rede.

Aprendizado HDC

- **Inferência:** É o **processo em que o sistema usa o modelo que foi treinado para classificar ou prever o comportamento de novos dados.**
  - **A inferência ajuda o sistema a identificar se o tráfego de rede observado é normal ou se é uma intrusão.**
  - Quando o sistema recebe uma nova amostra de tráfego (como um pacote de rede), ele **precisa converter essa amostra em uma representação que ele consiga entender e comparar com as classes aprendidas durante o treinamento**. **Essa representação é o "hipervetor de consulta"**
  - Depois de **gerar o hipervetor de consulta (Q)**, o próximo passo é comparar esse hipervetor com os hipervetores que representam as diferentes classes aprendidas durante o treinamento.
    - Cada classe pode ser, por exemplo, "tráfego normal", "tráfego de navegação web", "tráfego de e-mail", ou "intrusão".

Metodologia

- **Framework proposto:**

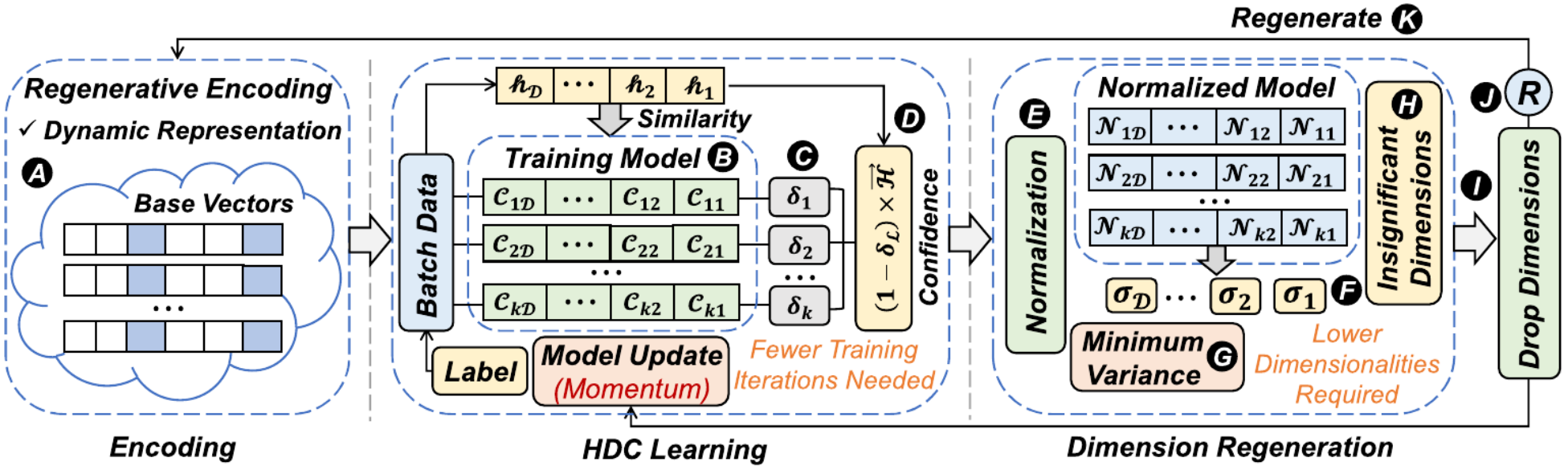


Fig. 3. Overview of the workflow of our proposed HyperDetect. HyperDetect starts with encoding training samples into a high-dimensional space. We then construct HDC models with our innovative model updating technique that explicitly considers the momentum at each data point. Following this, we identify and regenerate dimensions that have minimal impact on the classification task, eventually formulating a highly efficient HDC model.

Metodologia

- O Framework HyperDetect foi **projetado para ser eficiente em termos de recursos,** fator crítico em dispositivos de IoT, que frequentemente têm limitações de processamento, memória e energia;
- Propõe-se duas abordagens inovadoras: **1) Aprendizado HDC (HDC Learning) e 2) Regeneração de dimensões (Dimension Regeneration);**
  - Em 1) propoẽ-se uma abordagem adapatativa de atualização do modelo. Esse método busca resolver dois problemas principais, comuns em sistemas de detecção de intrusão baseados em aprendizado de máquina: **a saturação do modelo e a lentidão na convergência.**
    - **Eliminação da saturação do modelo**: Saturação ocorre quando o **modelo se torna incapaz de assimilar novas informações eficientemente**, pois **novos dados contribuem pouco para melhorar a acurácia**
    - Para resolver isso, o HyperDetect atribui um peso a cada novo dado com base na quantidade de "nova informação" que ele adiciona aos chamados "hipervetores de classe"
    - Se um novo dado for mais relevante, ele recebe um peso maior, garantindo que as informações mais valiosas tenham impacto significativo no modelo.

Metodologia

- **Aceleração da convergência do modelo:** A convergência é o ponto em que o modelo está suficientemente treinado para fornecer previsões precisas e robustas
  - No HyperDetect, para acelerar esse processo, é **calculado um "momento" em cada ponto de dados**, que ajusta o modelo de forma mais dinâmica e eficiente
  - Essa técnica **reduz a quantidade de iterações de re-treinamento necessárias, fazendo com que o modelo aprenda mais rapidamente e se torne utilizável em um período menor.**
- **Regeneração de Dimensões:** é uma técnica usada para **otimizar o modelo ao longo do tempo, garantindo uma alta acurácia com menor custo computacional,**
  - A regeneração de dimensões identifica dimensões insignificantes, permitindo que o modelo saiba quais delas não impactam significativamente a detecção de intrusões.
  - Após identificar as dimensões de menor relevância, o HyperDetect as "regenera" ou substitui, o que significa realocar esses espaços para informações mais úteis
  - Com uma menor quantidade de dimensões, o modelo torna-se mais leve e rápido, necessitando de menos processamento e memória
  - Ao reduzir as dimensões para apenas aquelas que são significativas, o HyperDetect simplifica o processo de aprendizado

Resultados Experimentais

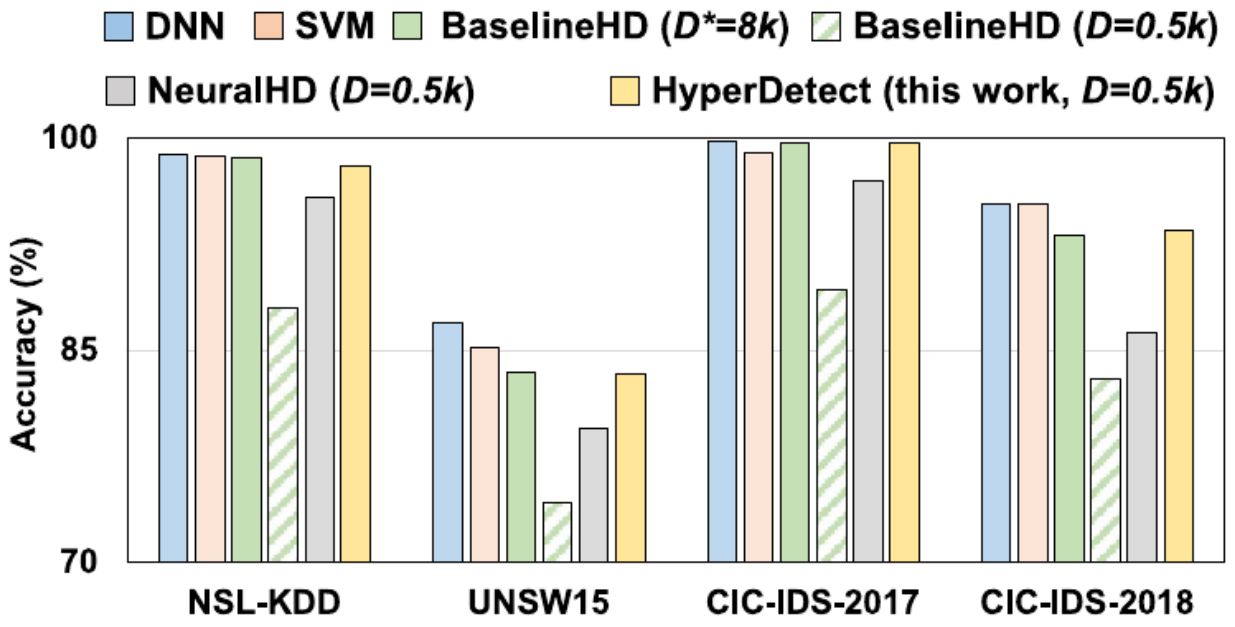


Fig. 5. Comparing accuracy of NIDS classification. HyperDetect demonstrates a comparable classification accuracy to SOTA learning algorithms.

Conclusão

- O HyperDetect é um framework de computação hiperdimensional (HDC) projetado **especificamente para detecção de intrusão em redes de IoT**, oferecendo eficiência de recursos e capacidade de resposta em tempo real;
- O modelo introduz uma técnica de **"regeneração de dimensões"** que identifica e substitui dimensões de menor impacto nas tarefas de classificação, **permitindo reduzir a dimensionalidade do modelo sem perder desempenho;**
- Os testes realizados mostraram que o HyperDetect tem uma **eficiência de aprendizado superior em comparação com outras soluções de HDC e redes neurais profundas de última geração** (SOTA DNNs); e
- Essa superioridade é observada em uma variedade de tarefas de detecção de intrusão de rede, nas quais o **HyperDetect se mostrou mais rápido no aprendizado e mais eficaz no uso de recursos.**

Referências

[1] - Wang, Junyao & Xu, Haocheng & Achamyeleh, Yonatan & Huang, Sitao & Al Faruque, Mohammad Abdullah. (2023). HyperDetect: A Real-Time Hyperdimensional Solution For Intrusion Detection in IoT Networks. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2023.3345279.

[2] - Ge, Lulu & Parhi, Keshab. (2021). Classification Using Hyperdimensional Computing: A Review. IEEE Circuits and Systems Magazine. 20. 30-47. 10.1109/MCAS.2020.2988388.

[3] - Wang, Junyao & Chen, Hanning & Issa, Mariam & Huang, Sitao & Imani, Mohsen. (2023). Late Breaking Results: Scalable and Efficient Hyperdimensional Computing for Network Intrusion Detection. 10.48550/arXiv.2304.06728.


IOT
IOT
IOT