Detecção de Intrusão de Rede para Segurança de IoT com base em Técnicas de Aprendizagem

Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac e Parvez Faruki

Resumo—O crescimento generalizado da Internet das Coisas (IoT) é visível em todo o mundo. O ciberataque Dyn de 2016 expôs as falhas críticas entre as redes inteligentes. A segurança da Internet das Coisas (IoT) tornou-se uma preocupação crítica. O perigo exposto por coisas infestadas conectadas à Internet não afeta apenas a segurança da IoT, mas também ameaça todo o ecossistema da Internet, que pode possivelmente explorar as coisas vulneráveis (dispositivos inteligentes) implantados como botnets. O malware Mirai comprometeu os dispositivos de vigilância por vídeo e paralisou a Internet por meio de ataques distribuídos de negação de serviço (DDoS). No passado recente, os vetores de ataque à segurança evoluíram nos dois sentidos, em termos de complexidade e diversidade. Portanto, para identificar e prevenir ou detectar novos ataques, é importante analisar técnicas no contexto da IoT. Esta pesquisa classifica as ameaças de segurança IoT e os desafios para redes IoT avaliando as técnicas de defesa existentes. Nosso foco principal é em Sistemas de Detecção de Intrusão de Rede (NIDS); Portanto, este artigo revisa as ferramentas e conjuntos de dados de implementação do NIDS existentes, bem como o software de detecção de rede gratuito e de código aberto. Em seguida, levanta, analisa e compara propostas NIDS de última geração no contexto IoT em termos de arquitetura, metodologias de detecção, estratégias de validação, ameaças tratadas A revisão lida com técnicas NIDS tradicionais e de aprendizado de máquina (ML) e discute direções futuras.

Nesta pesquisa, nosso foco está no IoT NIDS implantado via Machine Learning, pois os algoritmos de aprendizado têm uma boa taxa de sucesso em segurança e privacidade. A pesquisa fornece uma revisão abrangente dos NIDSs que implementam diferentes aspectos das técnicas de aprendizado para a Internet das Coisas, ao contrário de outras pesquisas importantes voltadas para os sistemas tradicionais. Acreditamos que o artigo será útil para pesquisas acadêmicas e industriais, primeiro, para identificar ameaças e desafios da IoT, segundo, para implementar seus próprios NIDS e, finalmente, propor novas técnicas inteligentes no contexto da IoT considerando as limitações da IoT. Além disso, a pesquisa permitirá que os indivíduos de segurança diferenciem o IoT NIDS dos tradicionais.

I. INTRODUÇÃO

A Internet das Coisas é considerada a terceira revolução industrial [1]. É definida como “a interconexão, via Internet, de dispositivos de computação embutidos em objetos do cotidiano, permitindo-lhes enviar e receber dados” [2]. O mercado de IoT está crescendo em um ritmo de tirar o fôlego, começando com 2 bilhões de objetos no ano de 2006 e projetando 200 bilhões até 2020 [3], um aumento de 200%. Sensores/ dispositivos IoT geralmente coletam e processam informações espaciais e informações temporais para eventos específicos e ambiente enfrentando vários desafios [4], [5]. Os objetos IoT ou Coisas tornaram-se mais inteligentes, o tratamento é mais inteligente e as comunicações tornaram-se instrutivas. Portanto, a IoT é usada em quase todos os campos: doméstico, educação, entretenimento, distribuição de energia, finanças, saúde, cidades inteligentes, turismo e até transporte [6]. Consequentemente, a indústria, a academia e os indivíduos estão tentando integrar o fluxo de comercialização rápida com pouca atenção à segurança dos dispositivos e redes IoT. Tal negligência pode colocar em risco os usuários da IoT e, por sua vez, interromper o vibrante ecossistema. Por exemplo, casas inteligentes podem ser controladas remotamente por cibercriminosos e veículos inteligentes podem ser sequestrados e controlados remotamente para criar pânico entre os cidadãos.

O perigo exposto por essas coisas conectadas à Internet não afeta apenas a segurança dos sistemas IoT, mas também o ecossistema completo, incluindo sites, aplicativos, redes sociais e servidores, por meio de dispositivos inteligentes controlados como redes de robôs (botnet). Em outras palavras, comprometer um único componente e/ou canais de comunicação em sistemas baseados em IoT pode paralisar parte ou toda a rede Internet. Em 2016, o ataque cibernético Dyn colheu dispositivos conectados instalados em casas inteligentes e os recrutou para “botnets” (também conhecidos como “exército de zumbis”) por meio de um malware chamado Mirai. Além das vulnerabilidades dos sistemas IoT, os vetores de ataque estão evoluindo em termos de complexidade e diversidade.

Consequentemente, mais atenção deve ser dada à análise desses ataques, sua detecção, bem como a prevenção de infecção e recuperação de sistemas após os ataques.

A. Escopo da pesquisa

Como a segurança dos sistemas de IoT pervasivos é crítica, é importante identificar as ameaças de IoT e especificar as estratégias de defesa existentes. Esta pesquisa começa com a classificação das ameaças de IoT para ter uma visão melhor para investigações estratégicas. Para isso, propomos uma classificação binária com: i) Camadas IoT; e ii) encontrou desafios durante o desenvolvimento dos sistemas IoT. Acreditamos que as redes IoT são diferentes das Redes de Sensores Sem Fio (WSN) e dos Sistemas Ciberfísicos (CPS) [7] devido à composição heterogênea de camadas em termos de protocolos, padrões e tecnologias. Além disso, vários desafios encontrados durante a implementação de vários casos de uso mencionados em [8] têm um contexto diferente em comparação com as redes WSN.

Os mecanismos tradicionais de defesa para ataques conhecidos têm uso variado e podem ser eficientes em situações específicas; no entanto,

N. Chaabouni trabalha com LaBRI, Univ. Bordeaux, Bordeaux INP, CNRS France e Atos Innovation Aquitaine Lab, France (e-mail: chaabouni.nadia14@gmail.com).

M. Mosbah e A. Zemmari estão com LaBRI - Laboratório de Pesquisa em Ciência da Computação de Bordeaux, Univ. Bordeaux, Bordeaux INP e CNRS UMR 5800, F33405 França (e-mail: mosbah@u-bordeaux.fr, zemmari@u bordeaux.fr) .

C. Sauvignac é da Atos Innovation Aquitaine Lab (e-mail: cyrille.sauvignac@atos.net).

P. Faruki trabalha no Departamento de Engenharia de Computação, MNIT Jaipur, Índia (e-mail: parvezfaruki.kg@gmail.com)

eles podem não ser completamente seguros. Apesar da disponibilidade de segurança tradicional com criptografia, autenticação, controle de acesso ou confidencialidade de dados, as redes IoT ainda estão sujeitas a ataques de rede que necessitam de uma segunda linha de defesa [9], [10]. Em tais situações, a importância dos Sistemas de Detecção de Intrusão (IDSs) para IoT é relevante. Uma das estratégias populares implantadas entre os sistemas IoT são os IDSs ou Network Intrusion Detection Systems (NIDSs) para coisas inteligentes conectadas. Os NIDSs têm sido objeto de escrutínio para obter sistemas de ciência da computação tradicionais seguros desde a década de 1980 [11]. Assim, NIDS é um campo científico maduro. Infelizmente, as técnicas tradicionais de NIDS podem ser menos eficientes e/ou inadequadas para sistemas IoT devido a mudanças características como recursos restritos, poder limitado, heterogeneidade e conectividade [9], [10]. Os sistemas tradicionais geralmente possuem nós mestres que são poderosos em termos de recursos de computação e espaço de armazenamento/memória. Esses nós monitoram os fluxos de entrada e saída sem grandes restrições de recursos ou largura de banda da rede. No entanto, os sistemas IoT são distribuídos e compostos por um grande número de dispositivos cuja capacidade de computação, espaço de armazenamento/memória e duração da bateria são limitados principalmente em recursos. A IoT também é limitada por sua capacidade de largura de banda de rede. Além disso, a IoT permite a interação entre o ambiente virtual e físico que é imprevisível. Cada nó possui um endereço IP para garantir sua comunicação com a Internet.

Isso causa problemas de confiança e vulnerabilidades específicas. Além dessas limitações, a IoT é baseada na heterogeneidade em termos de protocolos de comunicação e tecnologias coexistentes. Os protocolos e tecnologias não são empregados em redes tradicionais como IEEE 802.15.4, 6LoWPAN1 e CoAP; ou pelo menos não ao mesmo tempo dentro de um único sistema [8]. Finalmente, o ambiente IoT gera volumosos dados críticos que devem ser protegidos. Consequentemente, NIDSs são mais desafiadores e restritivos em redes IoT em comparação com NIDSs em sistemas de computação tradicionais.

Muitos NIDS IoT foram desenvolvidos usando regras de ataques/naturezas de sinais ou especificação de comportamento normal. Infelizmente, esses NIDS têm; i) alto reconhecimento de ataque falso positivo e/ou falso negativo (alarmes falsos); ii) incapacidade de detectar ataques desconhecidos/dia zero. Assim, os pesquisadores exploraram a inteligência artificial (IA) e o aprendizado de máquina (ML) com ênfase nos algoritmos de aprendizado profundo (DL) para melhorar a segurança dos sistemas [13], [14], [15]. De fato, as técnicas de aprendizado têm um impacto significativo na detecção de fraudes, reconhecimento de imagens e classificação de textos. A eficácia do aprendizado de máquina encorajou os pesquisadores a implantar algoritmos de aprendizado entre IDS para melhorar a detecção de ataques cibernéticos, detecção de anomalias e identificar comportamentos anormais entre os IoTs. Portanto, este artigo pesquisa e avalia contribuições notáveis de aprendizado de máquina para IoT NIDSs. No passado recente, a academia e a indústria mudaram seu foco para o desenvolvimento de NIDSs baseados em ML. Eles alcançam resultados interessantes; de 86,53% [16] para mais de 99% [17] na precisão da detecção e uma redução no falso positivo (FP) de cerca de 4% [18] para 0,01% [19].

1 Kasinathan et al. em [12] define 6LoWPAN como “um protocolo padrão projetado pelo IETF como uma camada de adaptação para redes com perdas de baixa potência, permitindo que dispositivos de baixa potência (LLN) se comuniquem com a Internet”.

Para resumir, há três áreas importantes que a pesquisa está direcionando (Fig. 1): i) IoT; ii) mecanismos de segurança; e iii) técnicas de aprendizado de máquina. Nosso foco estará na interseção dos três domínios acima, conforme destacado na Fig. 1: i) a interseção entre IoT e segurança (IDS para IoT); e ii) a interseção entre os três domínios (IoT NIDS implantados por meio de técnicas de aprendizagem).

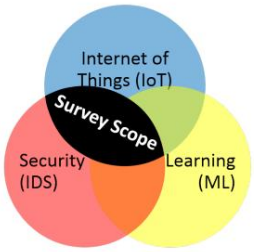

Fig. 1: Escopo da Pesquisa

B. Pesquisas de última geração

Existem várias pesquisas que tratam das outras duas interseções. Para a primeira intersecção entre IoT e domínio de aprendizagem, [20], [21], [22] e [23] têm contribuições significativas. Chen et ai. [20] revisam a mineração de dados (DM) para IoT nos pontos de vista de conhecimento, técnica e aplicação.

Os autores estudam algoritmos de big data e desafios quando implantados em ambiente IoT. Tsai et ai. [21] pesquisam recursos de dados para IoT e recursos para mineração de dados para IoT com uma discussão sobre mudanças, potenciais, questões em aberto e tendências futuras. No entanto, Cui et al. [22] fornecem uma visão geral da aplicação de aprendizado de máquina no domínio IoT. Esta pesquisa é muito recente e concentra-se no progresso das técnicas de aprendizado de máquina para aplicações de IoT. Finalmente, Mahdavinejad et al. [23] apresentam uma taxonomia de algoritmos de aprendizado de máquina enquanto discutem como eles podem ser aplicados aos dados para extrair informações de nível superior.

Além disso, os autores explicam o potencial e os desafios do aprendizado de máquina para análise de dados IoT.
Pesquisas sobre IDS baseadas em técnicas de aprendizado (não especialmente para IoT) como [24], [25], [13], [15], [26] e [27] realizaram avaliações abrangentes para sistemas tradicionais.
Agrawal et al [24] revisam várias técnicas de mineração de dados para detecção de anomalias. Buczak et ai. [25] discutem métodos de aprendizado de máquina (ML) e mineração de dados (DM) para análise cibernética em suporte à detecção de invasões. Ambas as pesquisas referem e resumem a complexidade e os desafios da segurança cibernética de ML/DM. No entanto, Fadlullah et al. [13] concentram-se em aprendizado profundo (DL) para sistemas de controle de tráfego de rede. Hodo et al. [15], propôs taxonomia de redes rasas e profundas e pesquisou sistemas de detecção de intrusão. Além disso, a pesquisa de Wang e Jones [26] revisou trabalhos de DM, ML, DL e Big Data com ênfase em critérios de avaliação subsequentes, como características de fluxo de dados, sistemas de processamento de fluxo, redução de dimensão de recursos e redução de dados. Finalmente, Mishra et al. [27] analisam e comparam as limitações das técnicas de aprendizado de máquina, bem como as restrições para implantação em invasões

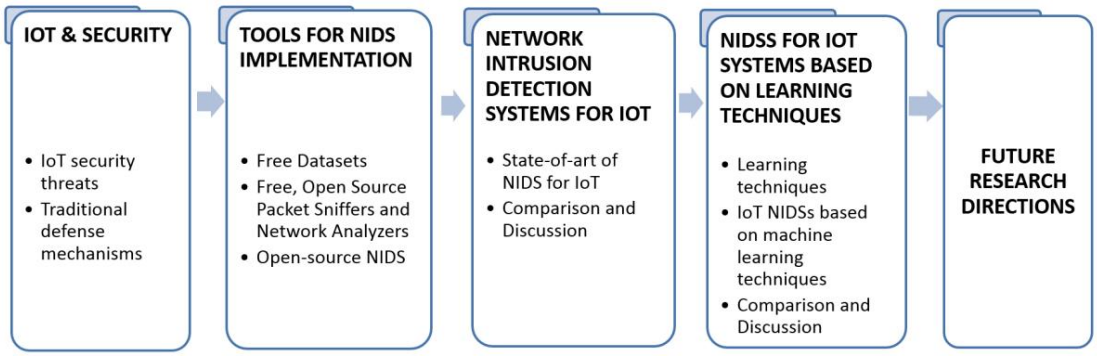

Fig. 2: Fluxo de trabalho da pesquisa

detecção. O fator chave que diferencia o trabalho de Mishra et al. das pesquisas existentes é a avaliação de que nenhuma técnica de detecção de intrusão em particular pode ajudar a detectar todos os tipos de ataques.

A lista final de pesquisas existentes visa a interseção entre IDS e IoT. Do estado da arte recente, dois Surveys sobre detecção de intrusão IoT são identificados: [28] e [8]. Zarpelo et ai. [28] apresentam uma visão geral dos IDS específicos da IoT e introduzem uma taxonomia para classificá-los. Além disso, eles propõem uma comparação detalhada entre os diferentes IDS para IoT com parâmetros como estratégia de posicionamento, método de detecção e estratégia de validação. No entanto, Ben Khelifa et al. [8] concentram-se nos avanços nas práticas de detecção de intrusão em IoT. Eles analisam o estado da arte recente com foco especial na arquitetura IoT. Os autores terminam sua pesquisa com direções futuras para IoT NIDS. Ben Khelifa et ai. pesquisa é mais detalhada revisão crítica clara. No entanto, nenhuma das pesquisas discutidas acima se concentra no uso de aprendizado de máquina para IoT NIDS. Até onde sabemos, nossa proposta é a primeira revisão abrangente que avalia as estratégias de implantação e compara a eficácia do IDS com aprendizado de máquina para a segurança dos sistemas IoT.

C. Critérios de Seleção de Papel e Fluxo de Trabalho de Pesquisa

Os artigos discutidos nas Seções IV e V são baseados nos seguintes critérios:

- Os artigos tratam da detecção de intrusão em IoT. • Os NIDSs visam os sistemas IoT em geral (por exemplo, não apenas redes WSN) com sua heterogeneidade, mobilidade e todos os desafios específicos da IoT. • Os autores apresentam sua arquitetura NIDS em detalhes. • As discussões e comparações são sobre NIDs tradicionais na Seção IV. No entanto, eles dizem respeito a NIDs com base em técnicas de aprendizado na Seção V.

- Os artigos de última geração são principalmente de periódicos indexados e importantes do IEEE, ACM, Elsevier e Springer, e dos principais locais de conferência publicados entre 2013 e outubro de 2018.

O restante da pesquisa está estruturado em quatro blocos, conforme ilustrado na Figura 2. Na Seção II, as categorias de ameaças à segurança da IoT são classificadas e as técnicas de defesa tradicionais são apresentadas com foco nos tipos de IDSs. Listas e discussões da Seção III disponíveis ferramentas que podem ser usadas para desenvolver NIDS; conjuntos de dados gratuitos, sniffers de rede gratuitos e de código aberto e, finalmente, NIDSs de código aberto. Essas ferramentas também podem ser usadas para testar e avaliar o desempenho do NIDS. A Seção IV discute NIDS alimentados por IoT, sua arquitetura, implantação e implicações para os sistemas heterogêneos. Na Seção V, são apresentadas técnicas de aprendizado por meio de classificadores de aprendizado de máquina. Em seguida, NIDSs para sistemas IoT implantados por meio de técnicas de aprendizado são revisados, comparados e avaliados. Discutimos exaustivamente o estado da arte existente, comparamos e avaliamos o desempenho de sistemas de IoT baseados em aprendizado de máquina implantados para proteger as redes.
Finalmente, a pesquisa é concluída com um resumo e uma lista de possíveis direções futuras na Seção VI.

II. TI E SEGURANÇA

Diversidade e heterogeneidade tornam a segurança dos sistemas IoT mais crucial. Os sistemas IoT diferem da segurança dos sistemas tradicionais pelos seguintes motivos: • Os sistemas IoT são limitados em termos de capacidade computacional, capacidade de memória, duração da bateria e largura de banda da rede. Portanto, não é possível implantar soluções de segurança tradicionais existentes, que geralmente consomem muitos recursos.

- Os sistemas IoT são sistemas fortemente distribuídos e heterogêneos. Assim, a solução tradicional centralizada pode não ser adequada. Além disso, o aspecto distribuído da IoT adiciona mais dificuldades e restrições em sua proteção. • Os sistemas IoT são implantados em um ambiente físico que é imprevisível. Assim, os ataques físicos passaram a fazer parte da lista de ameaças tradicionais à segurança. • Os sistemas IoT estão conectados à Internet, pois cada dispositivo tem acesso com seu endereço IP. Portanto, há mais um painel de ameaças relacionadas à Internet. • Os sistemas IoT são compostos por um grande número de objetos restritos que geram uma grande quantidade de dados. Portanto, é fácil inundar e atacar esses pequenos dispositivos, por um lado, e a largura de banda limitada das redes, por outro lado.

- Os sistemas IoT abrangem um grande número de protocolos e tecnologias heterogêneas no mesmo sistema. Portanto, a solução de segurança IoT proposta deve levar em consideração o grande painel desses protocolos e tecnologias na mesma proposta

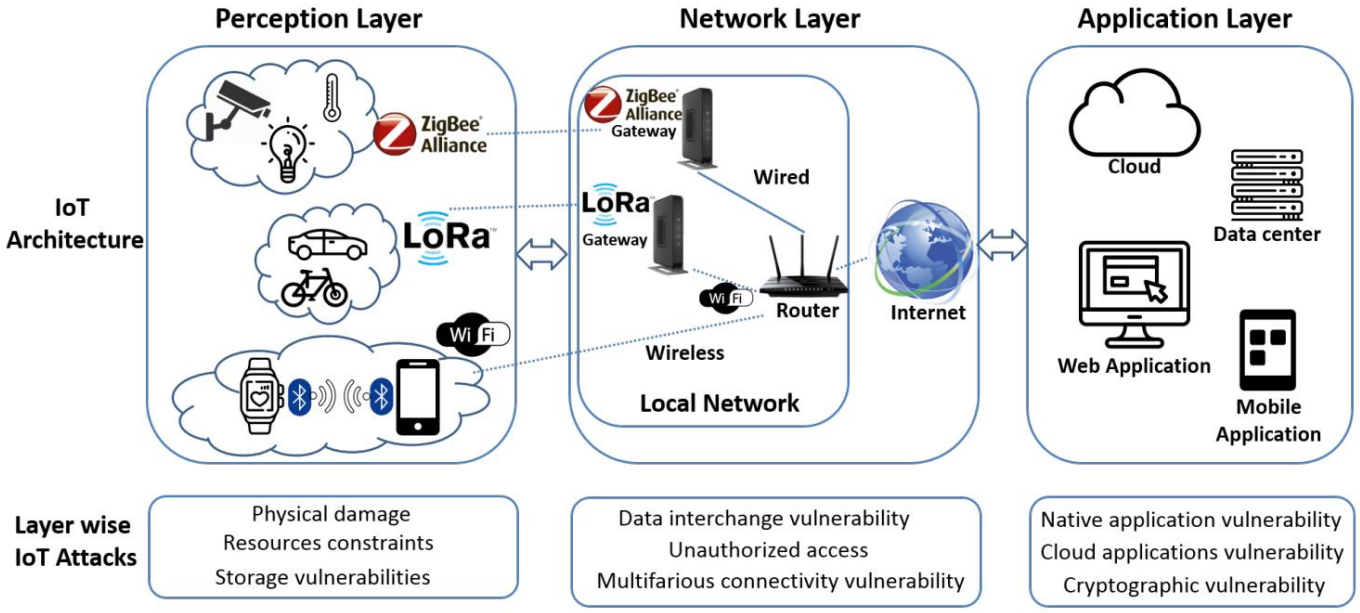

Fig. 3: arquitetura de IoT e ataques inteligentes em camadas

Consequentemente, a classificação de ameaças de sistemas IoT é discutida; em seguida, são introduzidos os mecanismos tradicionais de defesa empregados contra tais ameaças.

A. Classificação de ameaças de IoT

Como os sistemas de IoT são variados e enfrentam vários desafios, as ameaças de IoT podem ser classificadas em dois tipos. O primeiro tipo trata da classificação dependendo das camadas da arquitetura dos sistemas IoT, enquanto o segundo trata da categorização das ameaças IoT com base em seus desafios de design.

1) Classificação das ameaças IoT por camadas: os sistemas IoT relacionam o ambiente físico ao virtual. Uma representação padrão da arquitetura IoT é mostrada na Fig. 3. IoT consiste em três camadas principais [29] que são percepção/camada física, rede/camada de transporte e camada de aplicação.

Primeiro, a camada de percepção é a camada de hardware. É composto por diferentes sensores e atuadores que enviam e recebem dados usando diferentes padrões de comunicação, como Bluetooth, RFID e 6LowPAN. Em segundo lugar, a camada de rede é aquela que garante o roteamento/transmissão eficaz de dados/informações. Ele usa protocolos de comunicação como WiFi, 3G, GSM, IPv6, etc. Em terceiro lugar, a camada de aplicativo, também chamada de camada de software, é a camada superior que fornece aos sistemas a lógica de negócios e oferece as interfaces de usuário (UI) aos usuários finais (monitoramento de tráfego, sala de aula inteligente, etc.).

Cada camada pode representar múltiplas vulnerabilidades conforme ilustrado na Fig. 3 [30], [31]. Como os dispositivos são colocados em locais físicos diferentes, eles podem ser expostos a riscos ambientais, como chuva/neve/vento anormais ou ataques maliciosos ou danos não intencionais. Além disso, os dados armazenados podem ser roubados por meio de acesso físico. Os sensores são coisas minúsculas, portanto, sofrem de problemas de restrição de recursos (recursos computacionais, memória ou energia, etc.). Enquanto os dados/comandos são trocados (camada de rede), eles podem enfrentar diferentes vulnerabilidades de rede, como como vulnerabilidades de intercâmbio de dados (a transferência de dados pode ser encerrada devido a inundações de rede ou acesso de gateway malicioso), acesso não autorizado (ataque de representação, interceptação de comunicação, ataques de adivinhação de senha etc.) -Serviço (QoS), etc.). Além disso, a camada de aplicativo é exposta principalmente a problemas de software, como enumeração de contas, credenciais de conta inseguras e falta de suspensão de contas após um número limitado de tentativas de senha. Aplicações em nuvem [32], [33] podem ser atacadas por vírus, cavalos de Tróia, worms, etc. Como a IoT é baseada em dispositivos de baixa capacidade computacional, a criptografia de transporte às vezes é negligenciada ou usada em uma versão fraca. Portanto, as comunicações são facilmente rastreáveis e facilmente descobertas (ataque somente de texto cifrado, Man In the Middle).

2) Classificação de ameaças de IoT por desafios: Para entender os ataques de segurança de IoT primeiro, apresentamos alguns termos técnicos de ataque de IoT e, em seguida, apresentamos a classificação baseada em desafios de IoT.

a) Termos técnicos dos ataques: Primeiro, **spoofing** [34], [35] ou ataque de personificação rouba credenciais de autenticação para obter acesso não autorizado ao serviço. As credenciais podem ser roubadas diretamente de um dispositivo, espionando o canal de comunicação ou por phishing. A falsificação pode ser categorizada em: i)
falsificação de endereço IP; ii) falsificação de ARP; e iii) falsificação do servidor DNS. A falsificação de endereço IP refere-se à falsificação de conteúdo no cabeçalho IP de origem para mascarar a identidade do remetente ou para lançar um ataque distribuído de negação de serviço (DDoS).
Os ataques de falsificação de ARP normalmente abordam o protocolo de resolução (ARP). O ataque de falsificação resolve endereços IP para endereços MAC (Media Access Control). Quando um invasor envia mensagens ARP falsificadas pela rede local (LAN), o endereço MAC do invasor será vinculado ao endereço IP de um membro legítimo da rede. Consequentemente, malicioso

as partes podem roubar dados, modificar dados em trânsito ou até mesmo interromper o tráfego em uma LAN. A falsificação do servidor DNS modifica um servidor DNS (um sistema que associa a cada nome de domínio um endereço IP) redireciona um nome de domínio específico para um endereço IP não autorizado do servidor infectado.

Em segundo lugar, **os ataques de roteamento** [36] visam protocolos de roteamento onde as informações de roteamento trocadas são falsificadas, alteradas ou repetidas para gerar comportamentos de roteamento fictícios (ou seja, atração de tráfego de rede falsa). **O ataque Sinkhole** [37] diz respeito a um nó malicioso que atrai um grande tráfego ao apresentar um caminho imaginário como um caminho de roteamento ótimo. Em relação ao **ataque de encaminhamento seletivo** [38], é um mau comportamento de encaminhamento de dados em que um invasor encaminha seletivamente pacotes maliciosos enquanto descarta pacotes genuínos e importantes. Além disso, **o ataque de buraco negro** [37] visa interromper o fluxo normal de dados dentro de uma rede. Inicialmente, o ataque disfarça um ou mais nós defeituosos como a(s) melhor(es) rota(s); em seguida, começa a descartar pacotes de dados roteados pelo caminho com falha. Por outro lado, **o ataque de buraco de minhoca** [37] precisa de pelo menos dois nós defeituosos conectados via link com ou sem fio. Esses nós maliciosos encapsulam os pacotes mais rapidamente do que a trilha normal. Além disso, **o ataque de repetição** [34] considera a retransmissão ou o atraso de dados válidos para obter acesso não autorizado dentro de uma sessão já estabelecida.

Em terceiro lugar, o ataque **de adulteração** [34] é classificado como: i) adulteração de dispositivo; e ii) adulteração de dados. A adulteração do dispositivo pode ser facilmente realizada, especialmente quando um dispositivo IoT passa a maior parte do tempo sem vigilância. Ele pode ser facilmente roubado sem ser notado e usado de forma maliciosa. O dispositivo pode ser roubado como hardware ou apenas como software. A adulteração de dados envolve modificação maliciosa de dados, por exemplo, dados armazenados em bancos de dados ou dados em trânsito entre dois dispositivos.

Em quarto lugar, **o repúdio** [39] refere-se a dispositivos que executam uma ação maliciosa e depois negam sua execução. É o caso quando um dispositivo envia um vírus na rede sem deixar rastros que o identifiquem.

Em quinto lugar, **a divulgação de informações** [34] trata do acesso não autorizado à informação. Um invasor consegue o mesmo anexando dispositivos de espionagem, espionando o canal de rede ou obtendo acesso físico a um dispositivo; Por exemplo, Probe [40], [41] é quando os invasores tentam coletar informações sobre um nó de destino e suas vulnerabilidades por meio de varredura de conexões (varredura de porta, etc.). Com a divulgação de informações, ocorre o vazamento de informações confidenciais, como ataque de canal lateral [42].

A sexta ameaça é **o DDoS** [29], o ataque distribuído de negação de serviço executado por vários nós comprometidos juntos de diferentes localizações geográficas. Além disso, o ataque DoS envolve um invasor mal-intencionado que tenta consumir recursos de rede, atingir o tempo de CPU e/ou largura de banda de usuários legítimos, inundando o sistema com tráfego desonesto e amplificado.

Para conduzir um ataque DDoS eficiente, botnets são usados. São redes de dispositivos conectados à Internet infectados/controlados.

Conforme mencionado em [43], os ataques DoS são os ataques mais frequentes, especialmente em redes IoT/Fog relacionadas a IoT social, como cidades inteligentes, etc. Os ataques DDoS podem ser categorizados nos seguintes tipos [44], [45]:

1) Ataques de inundação são baseados em bombardear o sistema da vítima com um grande número de pacotes, principalmente UDP ou Pacotes ICMP [46], que causam comprometimento da rede largura de banda de trabalho. O ataque de inundação pode ser facilmente iniciado usando botnets.
2) Os ataques de amplificação podem ser estabelecidos explorando o mecanismo de reflexão e falsificando as fontes de IP. Os invasores enviam pacotes para servidores refletores com um endereço IP de origem definido para o IP da vítima, portanto, sobrecarregando indiretamente a vítima com os pacotes de resposta.
   Para resumir, os hackers exploram vulnerabilidades em diferentes protocolos para transformar pequenas consultas em um grande número de solicitações para desacelerar e/ou travar o(s) servidor(es) da vítima. Por exemplo, há ataques smurf, fraggle [45] e DNS, amplificação SSDP [47], [48] ataques distribuídos.
3) Os ataques de exploração de protocolo são construídos com base na exploração maliciosa de diferentes protocolos. Como exemplos, há SYN flood [47], TCP reset [47] e ataque de tortura de água [49], [50].

4) Ataques malformados são baseados em pacotes de rede malformados, como usar o mesmo endereço IP para endereços de origem e destino [45], [47].
5) Ataques lógicos/de software são ataques relacionados a protocolos de aplicativos. Por exemplo, Ping of Death [29], onde um invasor envia um pacote de solicitação ICMP ECHO fragmentado simples, maior que o tamanho máximo do pacote IP, de modo que a vítima não consiga remontá-lo. No ataque Teardrop [46], o adversário envia dois fragmentos que não se remontam pelo valor de deslocamento do pacote.

Sétimo, **elevação de privilégio** [39], [34] diz respeito à obtenção ou elevação de privilégios para acessar um dispositivo/serviço sem ter um direito legal. Tal ataque pode levar a uma situação perigosa, especialmente quando o invasor se torna um sistema de parte confiável. User-To-Root (U2R) e Remote-to-Local (R2L) [51], [40] são dois exemplos de ataque de elevação de privilégio. O U2R trata de obter privilégios de root (superusuário) em um nó quando o invasor inicialmente possui apenas uma conta de usuário normal. No entanto, o R2L ocorre quando um invasor não possui uma conta no nó da vítima e, portanto, explora vulnerabilidades para obter acesso local como usuário por meio de adivinhação ou quebra de senha.

Oitavo, **MITM** [34], [35], ataque Man-In-The-Middle que representa a interceptação de comunicação entre dois sistemas por um adversário para escutar uma conversa. Os ataques MITM são classificados como envenenamento de cache ARP, falsificação de DNS, sequestro de sessão, redirecionamento ICMP, roubo de porta, etc.

Nono, **a privacidade do usuário** [34], [52] é como a divulgação de informações. Além disso, um hacker não precisa necessariamente ter acesso a informações não autorizadas para aprender sobre um usuário. isso pode ser feito analisando metadados e tráfego.

Décimo, **nós de clonagem** [53], [54], [55] diz respeito à reintrodução de um clone de um nó na rede ou um componente em um sistema depois de capturar as credenciais e as características do original. Tal ataque permite que o usuário mal-intencionado controle o sistema, insira informações falsas, desabilite funções, etc. Uma vez que um objeto está sob o controle do invasor sem o conhecimento de seu proprietário (botnet), toda a rede pode ser infectada.

b) Classificação de ameaças de IoT por desafios de design: devido aos diferentes desafios relacionados ao design de sistemas de IoT, desenvolvedores e indústrias devem prestar atenção

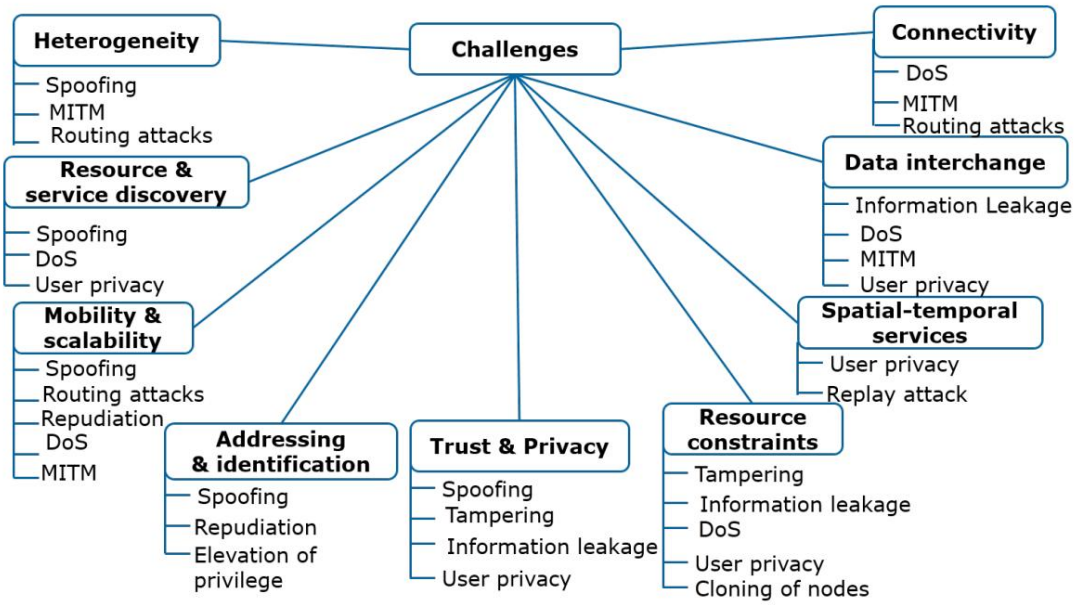

Fig. 4: classificação de ameaças de IoT por desafios de design

a muitas ameaças potenciais. Muitos trabalhos de pesquisa pesquisaram os desafios de segurança da IoT e as oportunidades de pesquisa, como Zhang et al. [56]; eles detalham os desafios de segurança da IoT, como identificação de objetos, autenticação e privacidade da IoT, etc. em redes IoT. Uma classificação baseada nos desafios de design dos sistemas IoT é apresentada na Fig. 4 e detalhada abaixo.

• Heterogeneidade e Interoperabilidade

As soluções backend IoT consideram o uso de sensores, atuadores e gateways fornecidos por diferentes fornecedores e podem ter diferentes versões. Para tanto, é necessária a utilização de um dispositivo que gerencie a interoperabilidade entre dispositivos heterogêneos. Esse componente pode ser bombardeado com solicitações falsas que podem levar a ataques DoS. Em um ambiente tão heterogêneo, spoofing, ataques de roteamento, bem como MITM, são mais prováveis de ocorrer em comparação com os sistemas homogêneos. É mais fácil para um nodo malicioso representar uma Coisa genuína, obter acesso não autorizado a dados e/ou retransmitir a comunicação entre a injeção de mensagem de dois nodos. Como podemos ver na Fig. 3, IoT pode ser considerado como o termo que faz referência a um mundo de grande variedade de protocolos e padrões heterogêneos [57]. Sua consideração torna a solução de segurança IoT cada vez mais complexa. Al-Fuqaha et al. fornecem um bom levantamento sobre essas tecnologias em [58]. • Conectividade Em IoT, a conectividade entre diferentes componentes do sistema é necessária, seja física ou em termos de serviços. Para o primeiro caso, os dados de dispositivos periféricos (sensores, por exemplo) devem ser conectados a uma rede IP com dispositivos de ponte que podem ser a causa de ataques de roteamento, bem como ataques MITM. Para a conectividade em termos de serviços, as alterações na disponibilidade de serviços devem ser notificadas aos respectivos dispositivos para que estes não inundem o sistema inconscientemente com pedidos repetitivos e indisponíveis. Essa inundação pode levar a um ataque DoS. Além disso, a QoS em redes IoT pode ser crucial, especialmente em situações de emergência. Assim, um roteamento robusto de pacotes e uma boa QoS na entrega de dados devem ser assegurados mesmo em topologias altamente dinâmicas [59]. • Mobilidade e Escalabilidade

Dispositivos de sistemas IoT podem estar em mobilidade contínua na área de campo; portanto, eles podem mudar as pontes às quais estão conectados. Isso geralmente causa interrupção de descontinuidade e/ou conexões com serviços não autorizados. Ataques como repúdio, MITM, DoS, sinkhole e wormhole tornam-se potencialmente possíveis. Para mitigar tais riscos, as soluções de segurança não consideram apenas os dispositivos móveis, mas também os componentes de rede, como os switches e os roteadores [60]. • Dispositivos de campo de endereçamento e identificação em aplicações IoT geralmente usam rádios de baixa potência para conexão de curta distância (menos de 1 quilômetro). Para isso, os nós coordenadores alocam endereços locais que não seguem um padrão comum, para dispositivos pares.

Conseqüentemente, esses endereços permanecem ocultos atrás do gateway/ponte FAN; Portanto, comportamentos maliciosos tornam-se indetectáveis. Como resultado, é difícil isolar o(s) nó(s) malicioso(s) e detectar ataques de falsificação e repúdio. Além disso, o nó pode tentar acessar privilégios não autorizados sem ser filtrado da rede externa (elevação de privilégio). • Serviços espaço-temporais

Eventos em IoT podem ser caracterizados pela amplitude do impulso espaço-temporal. Como resultado, os dados dos dispositivos IoT dos mesmos sistemas devem ter comportamento temporal razoável e geolocalização espacial. No entanto, essas tags de espaço temporárias devem ser protegidas de usuários mal-intencionados para evitar ataques de repetição. Além disso, os dados de localização do usuário devem

não sejam revelados a usuários não autorizados.

• Restrições de recursos A maioria dos dispositivos IoT periféricos é pequena, o que significa que eles têm recursos limitados em termos de poder de computação, memória integrada, largura de banda de rede e disponibilidade de energia. A adulteração, o vazamento de informações e a clonagem de nós são ataques possíveis, pois os dispositivos e sensores inteligentes têm recursos limitados. Os recursos limitados limitam a implantação de soluções criptográficas, portanto, soluções leves são a principal preocupação. Por exemplo, em [61], os autores superam essas limitações e propõem uma nova técnica de correção e detecção de erros intitulada “Low Complexity Parity Check (LCPC)”, para melhorar a qualidade das redes IoT futuristas. • Troca de Dados

Antes do início do intercâmbio de dados, ele deve ser criptografado nos nós IoT de origem. Os mecanismos de criptografia, dependendo do tipo de hardware, sua capacidade computacional e capacidade de armazenamento. A seleção inadequada leva a vulnerabilidades de segurança, como vazamento de informações (ou seja, as chaves estão sendo compartilhadas entre vários dispositivos quando os pacotes criptografados são descriptografados e reempacotados em vários pontos da cadeia de comunicação). Além disso, nós que criptografam dados podem ser atacados por meio de ataques de negação de serviço ou esgotamento de recursos. Por esse motivo, a criptografia de ponta a ponta é desejável. • Descoberta de recursos e serviços Em sistemas IoT, mecanismos de descoberta de recursos e serviços devem ser implantados para permitir autonomia e autodescoberta dos dispositivos. Esses mecanismos devem ser protegidos com autenticação bidirecional para evitar falsificação ou restringir o componente de malware de inundar o sistema com solicitações falsas para impedir ataques DoS. • Confiança e privacidade

Dispositivos de sensores inteligentes IoT gerenciam informações privadas/sensíveis do usuário (por exemplo, hábitos do usuário, dados de pacientes, dados de proteção civil, etc.); Portanto, a confidencialidade e a proteção de dados são extremamente importantes. De fato, confiança e privacidade [62], [63] são questões fundamentais para redes baseadas em IoT. Usuários, coisas e dispositivos são obrigados a autenticar por meio de serviços confiáveis para mitigar ataques de falsificação, adulteração e vazamento de informações. Confiança e privacidade estão recebendo mais atenção com smartphones, por exemplo, Android OS [64], [65] e [66].

B. Mecanismos de defesa tradicionais Depois

de detalhar e classificar as ameaças de IoT, discutimos a seguir técnicas de mitigação de ataque que protegem sistemas e redes de IoT existentes. Com o tempo, as soluções convencionais de segurança de TI abrangeram servidores, redes e armazenamento em nuvem. A maioria dessas soluções pode ser implantada para segurança de sistemas IoT. Os mecanismos de defesa podem ser separados ou combinados dependendo das ameaças tratadas [67]. Nesta seção, são descritos os mecanismos tradicionais que podem ser usados para proteger IoT.

Primeiro, **filtrar pacotes** [68], com firewalls e proxies, por exemplo, representam uma importante defesa contra IP spoofing ataques (e consequentemente ataques DDoS). Dois tipos de filtragem são possíveis: i) filtragem de entrada; e ii) filtragem de saída. A filtragem de entrada em pacotes recebidos trata do bloqueio de pacotes de fora da rede com um endereço de origem dentro da rede para proteger contra ataques externos de spoofing. No entanto, a filtragem de saída em pacotes de saída trata do bloqueio de pacotes dentro da rede com um endereço de origem que não está dentro para evitar que um hacker interno ataque máquinas externas.

Em segundo lugar, **adote criptografia** com protocolos criptográficos, criptografia de armazenamento de dados ou redes privadas virtuais (VPNs). O uso de protocolos criptográficos de rede (por exemplo, Transport Layer Security (TLS), Secure Shell (SSH), HTTP Secure (HTTPS) etc.) leva à criptografia de dados/códigos/atualizações antes de enviá-los e autenticá-los. A defesa é baseada em assinaturas/certificados digitais (par de chaves públicas e privadas) para garantir, por um lado, que os dados/código/atualização foram enviados pelo dispositivo/serviço legítimo e nunca modificados. Por outro lado, garante que os dados/códigos/atualizações sejam criptografados e não possam ser lidos ou utilizados por pessoas não autorizadas.

Os protocolos criptográficos de rede podem ser usados para proteger as coisas contra falsificação de IP, adulteração, repúdio, MITM, ataques que comprometem a privacidade do usuário e clonagem de nós. Além disso, criptografar o armazenamento de dados ajuda a evitar a divulgação de informações e mantém a privacidade do usuário. No que diz respeito à VPN (Virtual Private Network), trata-se de um túnel de comunicação seguro entre dois ou mais dispositivos. Ele criptografa a comunicação criando um link privado virtual na rede insegura existente.

A criptografia é uma boa solução para preservar a confidencialidade e a privacidade. No entanto, as redes IoT são vulneráveis, pois o recurso limita os dispositivos. Portanto, o uso das soluções criptográficas leves propostas por Al-Turjman et al. [69] é uma abordagem interessante. Eles propõem uma estrutura baseada em RSSF assistida por nuvem confidencial mantendo a confidencialidade, integridade e privilégios de acesso (CIA). A estrutura ágil proposta garante a integridade dos dados do sensor coletados com criptografia de curva elíptica.

Em terceiro lugar, **empregue esquemas robustos de autenticação de senha.** Além disso, limite o acesso aos dados atribuindo os privilégios apropriados aos recursos. O uso de One-Time Password (OTP) pode ser uma solução interessante. Spoofing, adulteração, divulgação de informações, elevação de privilégios e MITM podem ser evitados pelos mecanismos acima. Para redes IoT, as estratégias de autenticação precisam ser leves, como em Al-Turjman et al. soluções. Em [70] os autores propõem uma estrutura leve para fortalecer a segurança das redes IoT.

Eles introduzem uma autenticação de coletor móvel com suporte em nuvem, uma autenticação segura contínua baseada em curva elíptica e acordo de chave (S-SAKA). No entanto, em [71], os autores propõem uma autenticação baseada em “Hash” e “Global Assertion value”

esquema para a evolução da tecnologia 5G. Sua proposta considera a estrutura de provisionamento de identidade contínua sensível ao contexto (CSIP) para a futurística Internet Industrial das Coisas (IIoT).

Quarto, **atividades de auditoria e log** em servidores web, banco de dados

servidores e servidores de aplicativos. Devido a esses traços, outliers podem ser detectados. Mais especificamente, os principais eventos de log, como transação, login/logout, acesso ao sistema de arquivos ou tentativas de acesso a recursos com falha, podem detectar comportamento anômalo.
Uma boa prática para proteger esses arquivos é fazer backup deles, analisá-los regularmente para detectar atividades suspeitas e realocar os arquivos de log do sistema de seus locais padrão.
Além disso, proteja os arquivos de log usando ACLs restritas (lista de controle de acesso: uma lista de permissões anexadas a um objeto) e criptografe o log de transação. Essas técnicas impedem que os sistemas IoT sofram ataques de repúdio e elevação de privilégios.

Quinto, **detecte invasões usando IDS (Sistema de Detecção de Intrusão).** Um **IDS** [72] é uma combinação de software e hardware que monitora redes ou sistemas para identificar atividades maliciosas e emite alertas imediatos. Eles foram adotados [73] desde 1970 [74]. IDSs são geralmente categorizados de acordo com i) implantação; e ii) metodologia de detecção.

A implantação de IDS é categorizada como i) HIDSs; e ii) NIDSs.
Os sistemas de detecção de intrusão baseados em host (HIDSs) são instalados em uma máquina host (ou seja, um dispositivo ou uma coisa). Eles monitoram e analisam as atividades relacionadas aos arquivos de aplicativos do sistema e ao sistema operacional. HIDSs são preferidos contra dissuasão e prevenção de invasões internas. Os sistemas de detecção de intrusão baseados em rede capturam e analisam o fluxo de pacotes na rede. Em outras palavras, eles estão verificando os pacotes sniffados.
NIDSs são fortes contra ataques de intrusão externa. Como nosso interesse é a segurança de sistemas IoT com recursos limitados, o restante do artigo se concentrará nas soluções NIDSs.

A seguir, discutimos o cenário após a ocorrência da invasão. Um bom sistema de detecção é aquele que identifica a situação comprometida e minimiza a perda identificando rapidamente o(s) ataque(s).

Há uma variedade de IDSs. Em [12], as metodologias de detecção são classificadas como i) detecção de uso indevido; ii) detecção de anomalias, iii) detecção de especificações; e iv) detecção híbrida.

- Detecção de uso indevido ou detecção de assinatura (baseada em conhecimento) é um conjunto de regras predefinidas (como sequência de bytes no tráfego de rede ou sequência de instruções maliciosas conhecidas usadas por um malware) que são carregadas e combinadas com eventos. Quando um evento suspeito é detectado, um alerta é acionado. Este tipo de IDS é eficiente para ataques conhecidos; infelizmente, ele não pode detectar ataques de dia zero [41] / desconhecidos / invisíveis [75] devido à falta de assinaturas.
  As soluções de segurança cibernética preferem a detecção baseada em assinatura, pois é simples de implementar e eficaz para identificar ataques conhecidos (alta taxa de detecção com baixa taxa de falsos alarmes).
- A detecção de anomalias (baseada em comportamento) compara um comportamento normal registrado com a entrada atual. Inicialmente, o comportamento normal da rede e do sistema é modelado. Em caso de desvio do comportamento normal, o detector considera isso um ataque. A anomalia é identificada com análise estatística de dados, mineração e abordagens de aprendizado algorítmico.
  O detector de anomalias é bem-sucedido na prevenção de ataques desconhecidos. No entanto, eles tendem a gerar uma alta taxa de falsos positivos, uma vez que comportamentos não vistos anteriormente (ainda que legítimos) podem ser categorizados como anômalos. Outra vantagem é que as atividades normais do perfil são personalizadas para cada sistema, cada aplicativo e cada rede, o que dificulta as coisas para o invasor. É difícil saber exatamente quais atividades podem não ser detectadas. • A detecção de especificações tem a mesma lógica da detecção de anomalias. Define anomalia como desvio do comportamento normal. Essa abordagem é baseada em especificações de entrada desenvolvidas manualmente para capturar comportamentos legítimos (em vez daqueles vistos anteriormente) e seus desvios.

  No entanto, as especificações exigem que o usuário dê entrada. Este método reduz a alta taxa de falsos alarmes em comparação com os detectores de anomalias.
- A detecção híbrida é uma combinação de métodos anteriores, especialmente detecção baseada em assinatura e anomalia. O detector híbrido melhora a precisão reduzindo eventos falsos positivos. A maioria dos sistemas de detecção de anomalias existentes são, na realidade, híbridos. Eles começam com uma anormalidade de detecção, então tentam relacioná-la com a assinatura correspondente.

Sexto, **evite invasões com IPS (Sistema de Prevenção de Intrusão).** Um **IPS** é um IDS que responde a uma ameaça potencial tentando impedir que ela seja bem-sucedida. Um IPS responde imediatamente e interrompe a passagem do tráfego malicioso antes de responder descartando sessões, redefinindo sessões, bloqueando pacotes ou fazendo proxy de tráfego. No entanto, um IDS responde após detectar ataques passados. Existem muitos tipos de IPS [72], principalmente detecção em linha, comutadores de camada sete, sistemas enganosos, firewalls de aplicativos e comutadores híbridos. Para obter mais detalhes sobre os tipos de IPS, consulte Patil et al. papel [72].

Os mecanismos apresentados acima podem ser usados para proteger sistemas IoT. Alguns deles, como criptografia e autenticação, são insuficientes [9] para proteger IoT, portanto; Os IDS são necessários e são mais adequados para este caso de sistemas. Eles podem ser considerados como a última linha de defesa quando outras ferramentas quebram. Outra vantagem do IDS é que eles são variados e adaptáveis dependendo das necessidades. Eles podem ser dotados de lógica de aprendizado, como aprendizado de máquina e técnicas de inteligência artificial, além de outras tecnologias avançadas. Esse assunto será discutido na próxima seção.

Dos diferentes tipos e categorias de IDSs, esta pesquisa se concentra nos IDSs de redes híbridas e de anomalias (ANIDSs - HNIDSs) para sistemas IoT. Essa escolha foi feita devido ao poder e à capacidade dos IDSs de anomalias e híbridos de detectar ataques desconhecidos. Além disso, o documento se concentra na implantação de rede, pois oferece mais liberdade no desenvolvimento de soluções, ao contrário das implantações de host em IoT, que exigem baixo consumo de energia e recursos limitados. Os sistemas IoT são heterogêneos e muito grandes em termos de número de dispositivos.

Portanto, ter um único/múltiplos sistemas monitorando toda a rede em vez de analisar cada host separadamente (ou seja, a abordagem do HIDS é a segurança por dispositivo) é mais adequado para o caso de segurança de redes IoT. Afinal, IoT é por

definição sobre a interconexão de coisas heterogêneas (dispositivos).

A próxima seção apresenta diferentes ferramentas gratuitas e de código aberto para desenvolver e implementar novos NIDS.

III. FERRAMENTAS PARA IMPLEMENTAÇÃO DE NIDS

Os NIDSs analisam o tráfego de rede para detectar comportamentos maliciosos. Para construir um NIDS, estes são os passos básicos necessários [76] : 1) Coletar os dados de tráfego da rede.

2) Analisar os dados coletados.

3) Identificar eventos de segurança relevantes.

4) Detectar e relatar eventos maliciosos.

Para executar essas etapas, os pesquisadores têm duas opções; utilização das ferramentas existentes para facilitar a implementação de seus próprios NIDS; ou desenvolver uma nova estratégia de detecção. Sobre as ferramentas existentes, uma pessoa pode escolher entre i) conjuntos de dados gratuitos em modo off-line (já que é difícil testar propostas em redes reais e os conjuntos de dados são uma boa solução para benchmarking); ii) sniffers de rede gratuitos de código aberto para capturar seus próprios dados de tráfego de rede; ou iii) NIDS de código aberto gratuito que pode ser usado e adaptado para os objetivos desejados. Para ajudar os pesquisadores a entender as ferramentas disponíveis, começamos com conjuntos de dados gratuitos para NIDS; em seguida, discutimos NIDS e sniffers de rede gratuitos e de código aberto.

Esses três tipos de ferramentas estão correlacionados. Os sniffers de rede são usados para coletar dados de tráfego de rede que serão armazenados no conjunto de dados. A entrada não é rotulada; portanto, NIDS são necessários para diferenciar a instância como um ataque ou comportamento normal.

Os NIDS são geralmente maiores que os sniffers de rede. Eles usam sniffers de rede para capturar dados que são posteriormente usados para diferenciar ataques de comportamentos normais.

A. Conjuntos de dados gratuitos

Conjuntos de dados gratuitos podem ser usados para implementação e/ou validação do NIDS. Infelizmente, não há conjuntos de dados criados especificamente para redes IoT. Assim, duas estratégias são possíveis: baixar um conjunto de dados disponível visando sistemas tradicionais ou implantar software sniffing em redes.

Os conjuntos de dados mais amplamente adotados para NIDS são KDDCUP99 (KDD99) e NSL-KDD, que é uma versão aprimorada do KDD99. UNSW-NB15 [77] parece ser um conjunto de dados interessante para NIDS. Conjuntos de dados públicos como PREDICT, CAIDA, DEFCON, ADFA IDS, KYOTO, ISCX 2012 e conjuntos de dados de ataque ICS estão disponíveis para avaliação e teste. Os mais recentes são compostos de dados não rotulados ou são inacessíveis em alguns países ou são dados de domínio específico. Além disso, os conjuntos de dados sofrem de i) questões de privacidade; ii) a pesada anonimização dos inputs; e iii) o não reflexo dos atuais ataques de segurança.

• KDD99 [51]: é um conjunto de dados usado para detecção de conexões “ruins” das “boas” na Terceira Competição Internacional de Ferramentas de Descoberta de Conhecimento e Mineração de Dados [78] para a construção do NIDS robusto. O conjunto de dados é a versão extraída de recursos do conjunto de dados DARPA (DARPA é um conjunto de dados brutos de base). KDD99 contém registros de ambiente de rede militar com ataques injetados que podem ser categorizados em: i) Negação de Serviço; ii) Remoto ao Usuário; iii) Usuário para Root; e iv) Sondagem.

O KDD99 é baseado em 41 recursos para cada conexão junto com o rótulo de classe usando a ferramenta Bro-IDS (apresentada recentemente).
Os recursos são agrupados em 4 tipos [51]:

- 1-9: Recursos básicos de conexões TCP individuais.
- 10-22: Recursos de conteúdo dentro de uma conexão sugerida gerenciado pelo conhecimento do domínio.
- 23-31: Recursos de tráfego calculados usando uma janela de tempo de dois segundos.
- 32-42: Os recursos do host são projetados para avaliar ataques que duram mais de dois segundos.

KDD99 é popular e é o mais usado por pesquisadores para análise experimental. Diferentes trabalhos [79], [80], [81], [40], [82], [83] foram estabelecidos para reduzir o número de características selecionando as mais relevantes das 41 características iniciais. No entanto, muitos pesquisadores relataram desvantagens de KDD99 como [84], [85].

Alguns dos mais importantes são [73], [77], [86]:

- A distribuição de probabilidade dos conjuntos de teste e treinamento são diferentes. Em outras palavras, KDD99 sofre de métodos de classificação desbalanceados. Devido à adição de novos registros de ataque ao conjunto de testes, o equilíbrio entre os tipos de ataques e o tráfego normal não é mais mantido.
- O conjunto de dados está desatualizado.
- Há evidências de artefatos de simulação que podem resultar em superestimativas dos desempenhos de detecção de anomalias.

• NSL-KDD [77], [87]: é a versão atualizada do KDD99 para superar suas limitações. Primeiro, os registros duplicados nos conjuntos de treinamento e teste são removidos. Em segundo lugar, há uma variedade de registros selecionados do KDD99 original para obter resultados confiáveis de sistemas classificadores. Em terceiro lugar, o problema da distribuição de probabilidade desequilibrada é eliminado. O principal problema que persiste neste conjunto de dados é a falta de cenários modernos de ataque de pegada baixa. • UNSW-NB15 [77]: foi criado em 2015 pelo Cyber Range Lab do Australian Centre for Cyber Security (ACCS) com a ferramenta IXIA PerfectStorm. Seu objetivo é gerar atividades normais modernas reais híbridas e comportamentos sintéticos de ataque contemporâneo. São cerca de dois milhões e 540.044 registros armazenados em quatro arquivos csv. Esses registros são gerados a partir de 100 GB de tráfego bruto capturado com a ferramenta tcpdump [91] (em arquivos pcap). Este conjunto de dados possui nove tipos de ataques, a saber, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode e Worms. A Fig. 5 ilustra as etapas para gerar o conjunto de dados UNSW-NB15.

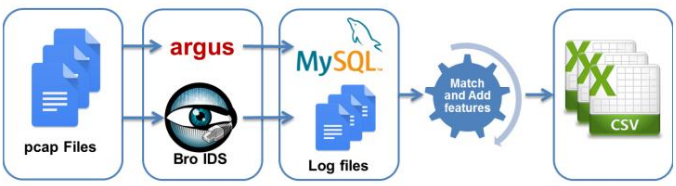

Fig. 5: Como gerar o conjunto de dados UNSW-NB15 [77]

Machine Translated by Google
Este artigo foi aceito para publicação em uma futura edição desta revista, mas não foi totalmente editado. O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/COMST.2019.2896380, IEEE Communications Surveys & Tutorials

- Sivanathan et al. Conjunto de dados IoT [88], [92]: aborda a classificação de dispositivos IoT com base nas características do tráfego de rede. Os autores instrumentam um ambiente inteligente para 28 dispositivos IoT, como câmeras, luzes, plugues, sensores de movimento, aparelhos e monitores de saúde. Além disso, eles sintetizaram rastreamentos de tráfego de rede de sua infraestrutura por um período de seis meses liberados para a comunidade de pesquisa. Sivanathan et ai. Apresente informações valiosas sobre os padrões de tráfego de rede por meio de análise estatística usando atributos como ciclos de atividade, números de porta, padrões de sinalização e conjuntos de cifras.
- Banco de dados CICIDS [89]: é um dos bancos de dados recentes de detecção/prevenção de invasões lançado pelo Instituto Canadense de Segurança Cibernética da Universidade de New Brunswick para refletir as ameaças mais recentes semelhantes aos dados do mundo real. Foi construído com base no comportamento abstrato de 25 usuários com base nos protocolos HTTP, HTTPS, FTP, SSH e e-mail. O conjunto de dados é analisado com CICFlowMeter [93] com fluxos rotulados com base em timestamp, IP inicial e final, portas, protocolos e ataques. Para gerar o tráfego realista, os autores propuseram a abordagem B-Profile [94] para delinear o comportamento nos protocolos HTTP, HTTPS, FTP, SSH e e-mail. Os autores implementaram ataques FTP de força bruta, SSH Heartbleed e DDos durante a captura dos dados. A estrutura de avaliação [95] identificou onze características importantes necessárias para construir um conjunto de dados de referência confiável, ao contrário dos conjuntos de dados IDS tradicionais existentes.
- Banco de dados CSE-CIC-IDS2018 [90]: é um conjunto de dados IDS exclusivo que evoluiu para substituir os conjuntos de dados abaixo do ideal existentes que limitam as avaliações experimentais de IDS/NIDS. Para superar o uso de conjuntos de dados estáticos e únicos, o CSE CIC-IDS2018 é um conjunto de dados gerado dinamicamente com base em anomalias que consiste em intrusão no tráfego de rede. Os autores incluíram sete cenários de ataque, incluindo i) Força bruta; ii) Hemorragia; iii) Botnets; iv) DoS; v) DDoS; vi) Ataques na Web; e vii) Ataques de infiltração de redes locais. A infraestrutura de ataque possui 50 nós e a organização da vítima possui 5 departamentos com 30 servidores e 420 hosts. Os autores extraíram 80 recursos do tráfego de rede e logs de máquina capturados via CICFlowMeter-V3.

A seguir, os conjuntos de dados de rede gratuitos apresentados são discutidos. Conforme mostrado na tabela de comparação I, KDD99 é o conjunto de dados de rede mais popular. É usado desde 1999. Felizmente, está desatualizado. Para superar as limitações do KDD99, o NSL-KDD foi criado. Possui dados balanceados sem registros duplicados. Como o NSL-KDD carece de ataques modernos, o UNSW-NB15 foi proposto. É um conjunto de dados de boa reputação com ataques recentes. Entretanto, é mais complexo que o KDD99 em termos de similaridade entre os novos ataques e os comportamentos normais. Como conjuntos de dados de rede mais recentes, existem i) Sivanathan et al. conjunto de dados; ii) CICIDS e iii) CSE-CIC-IDS2018. Sivanathan et ai. work é o único conjunto de dados de tráfego de rede IoT em comparação com os outros apresentados. No entanto, ele é projetado para proliferação de dispositivos IoT e não para detecção de intrusão. O CICIDS e o CSE CIC-IDS2018 têm registros rotulados, mas não visam a segurança dos sistemas IoT, apesar de sua lista de ataques atualizada.

TABELA I: Comparação entre conjuntos de dados livres

Conjuntos de dados	Vantagens	Desvantagens
KDD99	• KDD99 é popular e o mais usado. [51] • Dados rotulados. • É baseado em 41 recursos para cada conexão junto com o rótulo de classe. • Implementa negação de serviço, remoto para usuário, usuário para root e sondagem ataques. • Fornece tráfego de rede (PCAP). • É uma	• KDD99 sofre de classificação desequilibrada métodos. • O conjunto de dados está desatualizado. • Não para sistemas IoT.
NSL-KDD [87]	versão melhorada do KDD99. • Supera as limitações do KDD99. • Nenhum registro duplicado nos conjuntos de treinamento e teste. • Fornece atividades	• Falta de cenários modernos de ataque de baixa pegada. • Não para sistemas IoT.
UNSW NB15 [77]	normais modernas reais híbridas e comportamentos sintéticos de ataque contemporâneo. • Fornece tráfego de rede (PCAP) e arquivos CSV. • Possui nove tipos de ataques, a saber: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode e Worms.	• É mais complexo do que o conjunto de dados KDD99 devido aos comportamentos semelhantes do ataque moderno e do tráfego de rede normal.
Sivanathan al. conjunto de e dados [88]	• Conjunto de dados IoT de tráfego de rede. • Reflete os sistemas IoT do mundo real. • Fornece tráfego de rede (PCAP) e arquivos CSV.	• Dados não rotulados. • Para proliferação de dispositivos IoT e caracterização de tráfego. • Nenhum dado de ataque.
CICIDOS [89]	• Fluxos de rede rotulados. • Para fins de aprendizado profundo e de máquina. • Fornece tráfego de rede (PCAP) e arquivos CSV. • Implementa ataques como Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet e DDoS.	• Privado. • Não para sistemas IoT.
CSE-CIC IDS2018 [90]	• Fluxos de rede rotulados. • Para fins de aprendizado profundo e de máquina. • Fornece tráfego de rede (PCAP), CSV e arquivos de log. • Implementa força bruta, Heartbleed, Botnet, DoS, DDoS, ataques da Web e ataques de infiltração de rede local. • Conjunto de dados gerado dinamicamente. • É modificável, extensível e reproduzível.	• Privado. • Não para sistemas IoT.

B. Sniffers de rede gratuitos e de código aberto

A seguir, o artigo apresenta os sniffers de rede mais populares e gratuitos, e software de código aberto. Uma ferramenta de sniffing [96] visa monitorar o tráfego de trânsito da rede da origem ao destino. Ele pode ser usado para capturar, examinar, analisar e visualizar pacotes ou quadros.

• Tcpdump [91], [96] é o analisador de pacotes mais popular, poderoso e amplamente utilizado. É uma ferramenta de linha de comando TCP/IP que permite capturar, analisar, salvar e visualizar dados de pacotes. Van Jacobson, Craig Leres e Steven McCanne Desenvolve tcpdump no Lawrence Berkeley Laboratory, UC, Berkeley. Tcpdump captura dados de pacotes ao vivo de uma interface de rede. Uma característica interessante do tcpdump é a possibilidade de salvar os pacotes capturados em um arquivo pcap para posterior análise. Tcpdump usa a biblioteca libpcap para capturar pacotes. Libpcap, é freqüentemente usado por outros programas de captura. A ferramenta Tcpdump está disponível para a maioria dos sistemas operacionais baseados em Linux/Unix. A GUI de código aberto (Graphical User Interface) mais popular baseada no tcpdump é o Wireshark (software de terceiros) que lê arquivos tcpdump pcap, permitindo uma interface amigável e fácil de usar. • Wireshark [102], [96], [98] é um analisador de pacotes popular, gratuito e de código aberto, sob a licença GNU. Ele é usado para detecção e análise de rede. Ele captura ao vivo pacotes de dados de uma interface de rede. Devido a problemas de marca registrada, Wireshark foi renomeado para Ethereal em maio de 2006. O Wireshark é executado em sistemas operacionais semelhantes ao Unix, Solaris e Microsoft Windows. Ele usa libpcap como uma biblioteca para capturar e filtrar pacotes; em seguida, exibe registros com sua GUI. Esta ferramenta permite a leitura das saídas do tcpdump. O Wireshark decodifica um grande painel de protocolos (¿400). Suporta inspeção preliminar de ataques na rede. Sua versão de linha de comando é “tshark”.

• Ettercap [99], [96] é um sniffer de rede multiplataforma. É “um sniffer/interceptor/logger multiuso para LANs comutadas” [99] escrito por Alberto Ornaghi e Marco Valleri. Ettercap é conhecido por sua poderosa capacidade de lançar vários tipos diferentes de ataques man-in-the-middle. Além disso, ele fornece aos usuários muitos ataques clássicos separados e técnicas de reconhecimento em sua interface. Ele detecta conexões ao vivo e filtra pacotes, bem como muitos outros recursos, de maneira ativa ou passiva. • Argus [100], [96] é uma ferramenta para capturar e analisar dados de fluxo de trabalho de rede. Ele roda em vários sistemas operacionais como Linux e Windows. Ele se concentra no desenvolvimento de estratégias de auditoria de atividades de rede. Além disso, o argus trata dados de tráfego ao vivo e capturados para gerar relatórios de status/auditorias sobre fluxos detectados com uma análise semântica. Ele processa os dados do pacote ERF da libpcap e do Endaces para permitir que o usuário tenha uma

TABELA II: Comparação entre sniffers de rede gratuitos e de código aberto

farejadores de rede	**Vantagens**	**Desvantagens**
tcpdump [97], [98], [96]	• Longa vida útil do produto com diversas atualizações e muitos recursos. • Bem documentado com um bom suporte da comunidade. • Fácil acesso remoto com conexão Telnet. • Plataforma cruzada (também foi portado para o Windows). • Menos intrusivo em comparação com o Ethereal. • Captura pacotes de dados ao vivo de uma interface de rede. • Salva dados de pacotes capturados. • Leve em termos de instalação. • Bem documentado com um bom suporte da comunidade. • Plataforma cruzada. • Suporta grande número de protocolos. • Ferramenta gráfica. • Captura pacotes de dados ao vivo de uma interface de rede. • Salva e	• Carece de análise crítica. • Descarta pacotes inválidos (não é útil para detectar pacotes quebrados). • Nenhuma GUI real ou console administrativo.
Wireshark [97], [98], [96]	abre arquivos de dados de pacote. • Fornece informações de protocolo detalhadas. • Plataforma cruzada. • Pode ser usado para técnicas de invasão de LAN. • Decodifica diversos protocolos. • Coleta senhas para vários aplicativos • Manipula a rede eliminando conexões, injetando pacotes e comandos na(s) conexão(ões) ativa(s). • Extensível com plug-ins adicionais. • Plataforma cruzada. • Decodifica diversos protocolos. • Gera relatórios e auditorias sobre a rede. • Sistema de arquivos nativo, bem como suporte a MySQL. • Eficiente na análise de grande	• Nenhuma notificação de comportamento anormal (não é uma IDS). • Reúne informações, mas não pode manipular o rede. • Consumo de recursos em termos de instalação.
Ettercap [99], [96]	quantidade de tráfego de rede. • Exibe gráficos para atividade de rede com um modo de protocolos codificados por cores. • Hosts e links mudam de tamanho com o tráfego. • Pode filtrar pacotes. • Suporta vários quadros e tipos de pacotes. • Suporta arquivo e tráfego de rede em tempo real. • Boa reputação entre a comunidade de administradores de sistema.	• Sniffing é um recurso secundário. • Pode ser usado como uma ferramenta de hacker. • Pode ser detectado por outras ferramentas de rede (por exemplo exemplo pelo próprio ettercap).
Argos [100], [96]		• Não muito óbvio para dominar.
EtherApe [101], [96]		• Suporta apenas Unix OS. • Nenhuma versão de linha de comando. • Captura apenas cabeçalhos de pacotes.

idéia sobre o que está acontecendo em uma rede. Esta ferramenta fornece informações sobre quase todos os parâmetros do pacote, como duração, taxa, carga, retransmissão, atrasos, etc. • EtherApe [101], [96] de autoria de Juan Toledo e Ric cardo Ghetta em 2000 é um sniffer gráfico de pacotes e ferramenta de monitoramento de rede. Ele suporta apenas plataformas Unix.
O EtherApe visa representar pacotes, conexões e fluxos de dados visualmente com hosts codificados por cores e links para os protocolos. A ferramenta também facilita a solução de problemas de rede. Além disso, suporta exibição em tempo real de pacotes de rede por meio de formatos padrão. O tráfego pode ser consultado na própria rede, end-to-end (IP) ou porta-a-porta (TCP).

A seguir, os diferentes sniffers de rede gratuitos e de código aberto são comparados conforme representado em II. Como pode ser notado, o tcpdump é o sniffer de rede mais popular (todos os outros sniffers tentam suportar suas saídas). É um produto de longa duração frequentemente atualizado e ampliado com vários recursos. Está bem documentado e conta com o apoio da comunidade. No entanto, o tcpdump é desenvolvido principalmente para captura de dados, ao contrário de outras ferramentas equipadas para análise de rede. É uma ferramenta de linha de comando sem GUI real. Embora a força do wireshark e do etherApe esteja em seus recursos gráficos, ambos podem exibir arquivos de rede capturados e em tempo real. Wireshark é mais conhecido que etherApe. Além disso, o wireshark é um sniffer de plataforma cruzada, o que não é o caso do etherApe que suporta apenas a plataforma Unix. Além disso, ao contrário do etherApe, o wireshark leva em consideração os detalhes do cabeçalho e da carga útil. Em relação ao Argus, é mais uma ferramenta para auditar as atividades da rede. Decodifica vários protocolos para relatórios e auditorias. Com relação ao ettercap, além dos dados de rede no modo sniffer, interceptor e logger, ele pode manipular a rede e lançar diferentes ataques MITM. Ele é capaz de coletar senhas, matar conexões, injetar pacotes e comandos em conexões ativas. Portanto, pode ser considerado mais uma ferramenta de hacker do que um sniffer de rede. Como os sniffers de rede foram tratados, a próxima parte revisará e discutirá o NIDS de código aberto.

C. NIDS de código aberto

Existem muitas ferramentas NIDS de código aberto gratuitas que são exploradas para farejar, analisar e detectar eventos maliciosos no tráfego de rede, como Snort, Suricata, Bro-IDS, etc. Esta seção apresenta e compara os NIDS gratuitos e de código aberto mais populares.

• Snort [111], [104] é um sistema de prevenção de intrusão leve capaz de análise de tráfego em tempo real e registro de pacotes. Foi lançado pela primeira vez em 1998. Suporta os sistemas operacionais Fedora, Centos, FreeBSD e Windows. Snort é um NIDS baseado em assinatura de thread único. Ele usa Talos, a lista de regras de código aberto mais atualizada e popular. Ele pode ser executado em três modos usando diferentes opções na linha de comando do snort:

- Modo sniffer onde os pacotes são simplesmente lidos da rede e exibidos no console.
- O modo Packet Logger registra os pacotes no disco.
- O modo Network Intrusion Detection System (NIDS) detecta e analisa o tráfego de rede usando um arquivo contendo regras de detecção, como regras para detectar estouros de buffer, varreduras de portas furtivas, etc. O Snort pode detectar ataques na camada de aplicativos, como ataque de injeção de SQL e ataque de script entre sites. •

Suricata [112] é um NIDS multi-thread baseado em assinatura.
Possui vários recursos, principalmente detecção de intrusão em tempo real (IDS), prevenção de intrusão em linha (IPS), monitoramento de segurança de rede (NSM) e processamento pcap offline.
O projeto é propriedade da Open Information Security Foundation (OISF). A primeira versão beta foi lançada em dezembro de 2009. Suporta Linux, FreeBSD, OpenBSD, macOS/Mac OS X e Windows como sistemas operacionais. • A ferramenta Bro-IDS [113] é uma estrutura de análise de rede para inspeção de tráfego de rede contra atividades maliciosas.
Bro é um IDS baseado em assinatura e anomalia. Ele suporta muitos protocolos de camada de aplicação, incluindo DNS, FTP, HTTP, SMTP, etc. O sistema Bro foi projetado e desenvolvido por Vern Paxson do ICSI's Center for Internet Research (ICIR). O Bro-IDS suporta os sistemas operacionais Linux, FreeBSD e Mac OS X.

• Kismet [107] é um detector de rede sem fio, sniffer e IDS. Ele é executado em várias plataformas, como Linux, BSD, Android, Windows (com suporte de hardware restrito), etc. O Kismet funciona com cartões Wi-Fi (IEEE 802.112) e dispositivos alimentados por Bluetooth para digitalizar dispositivos BT e BTLE detectáveis, o rádio RTL-SDR para detectar sensores sem fio, termômetros e interruptores e uma coleção crescente de outros hardwares de captura. • OpenWIPS-ng [108] é um IDS/IPS sem fio modular que monitora o tráfego sem fio para detectar e identificar ataques baseados em assinatura. Ele é desenvolvido por Thomas d'Otreppe de Bouvette, o criador do software Aircrack e roda basicamente em hardware comum. O OpenWIPS ng é composto por sensores para capturar o tráfego sem fio e enviá-lo ao servidor; um servidor para agregar dados de todos os sensores, analisar, detectar invasões e enviar respostas (os ataques são registrados e os alertas são reportados ao administrador) e, finalmente, uma GUI para gerenciar o servidor e exibir informações sobre ameaças. O OpenWIPS-ng possui plug-ins de extensão para maior flexibilidade, mas suporta apenas sistemas Linux.

• Security Onion [109] é uma distribuição Linux para detecção de intrusão, monitoramento de segurança de rede (NSM) e gerenciamento de log. Ele contém um conjunto de ferramentas de segurança específicas, incluindo Snort, Suricata, Sguil, Bro, Elasticsearch, Logstash, etc. que funcionam de forma independente ou em conjunto para detectar atividades maliciosas em vLANs e redes visualizadas.

As principais características do security onion são: i) captura completa de pacotes; ii) NIDS e HIDS; e iii) poderosas ferramentas de análise. • Sagan [110] é um mecanismo de análise e correlação de logs em tempo real desenvolvido pela Quadrant Information Security. Ele é executado no sistema operacional Unix e é escrito em C com uma arquitetura multithread para detectar atividades maliciosas nos níveis de log e rede com alto desempenho. Sagan é

[2]IEEE 802.11 é o padrão mais usado para protocolo Wi-Fi e controle de acesso à mídia (MAC) e camada física (PHY) em redes locais sem fio (WLAN).

TABELA III: Comparação entre NIDS de código aberto

IDS	Vantagens •	Desvantagens
bufar [103]	Detecção de intrusão leve [104]. • Longa vida útil do produto com diferentes atualizações, novos recursos e muitos front-ends administrativos. • Bem documentado com um bom suporte da comunidade. • Bem testado. • Fácil de implantar. • Arquitetura multiencadeada para análise rápida do tráfego de rede. • A inspeção de tráfego de rede	• Nenhuma GUI real ou administração fácil de usar console. • Problemas de perda de pacotes quando o processo atinge taxas de 100-200 megabytes por segundo antes de atingir o limite de processamento de uma única CPU.
Meerkat [105], [106]	pode ser construída usando placas gráficas (aceleração de hardware). • Detecta downloads de arquivos. • Pode usar o script LuaJIT para detectar ameaças complexas de maneira mais fácil e eficiente. • Registra mais de pacotes como certificados TSL/SSL, solicitações HTTP e solicitações DNS. • Possui uma assinatura e métodos de detecção baseados em anomalias. • Assinaturas sofisticadas. • Analisa o tráfego de rede em um nível muito mais alto de abstração. • Armazena informações sobre atividades passadas e as incorpora para análise de novas atividades.	• Requer mais recursos de memória e CPU do que o Snort.
Bro-IDS [105], [106]	• Suporta rede de alta velocidade. • Pode estender sua funcionalidade para redes de outros tipos por meio de plugins. • Permite salto de canal para encontrar o maior número de redes possível. • Indetectável durante a detecção de pacotes (monitora redes sem fio passivamente). • Ferramenta de monitoramento sem fio de código aberto amplamente utilizada e atualizada. • Transmissão ao vivo de capturas em tempo real por HTTP.	• Bro é uma plataforma UNIX. • O Bro-IDS é baseado em arquivos de log sem qualquer GUI, mantidos pelo Bro Project. • Precisa de experiência para configurar.
Kismet [107]		• Não é possível obter diretamente os endereços IP. • Somente para redes sem fio.
OpenWIPS • Software e hardware ng [108] •	Modular e baseado em plug-in (recursos adicionais podem ser integrados). detecção de possíveis problemas considerados por não profissionais. • Maior precisão de	• Somente redes sem fio são consideradas. • O tráfego entre o sensor e o servidor não é criptografado. • Pouco famoso e pouco desenvolvido. • Nenhuma documentação detalhada e nenhum grande suporte da comunidade. • Herda as desvantagens de cada ferramenta constituinte. • Não funciona como um IPS após a
cebola Segurança [109]	• Sistema flexível. • Fácil monitoramento de segurança de rede e análise orientada a eventos por causa da interface gráfica em tempo real Sguil. • Fornece muitas funções e softwares pré-instalados e facilmente configuráveis (instalação do assistente). • Tem atualizações regulares para melhorar os níveis de segurança. • Processamento de log rápido (multi-threading) e em tempo real. • Suporta vários formatos de saída e normalização de log. • Permite a localização geográfica dos endereços IP. •	instalação, mas apenas como um IDS.
Sagan [110]	Pode distribuir seu processamento por vários dispositivos. • CPU leve e recursos de memória. • Ativamente desenvolvido e fácil de instalar.	• É desenvolvido principalmente para análise de log em vez do que para detecção de intrusão.

essencialmente um HIDS e foi estendido para ser considerado também um NIDS baseado em assinatura. É compatível com dados coletados por Snort, Bro, Suricata e outras ferramentas.

A seguir, as vantagens e desvantagens dos NIDSs apresentados são discutidas conforme ilustrado na Tabela III. Por um lado, o Snort é um IDS antigo e atualizado. Está bem documentado e bem testado. Infelizmente, ele sofre de alguns problemas relacionados à perda de pacotes. Por outro lado, o Suricata oferece uma arquitetura multi-threaded, mas requer mais recursos de memória e CPU em comparação com o Snort, o que pode representar problemas especialmente no contexto de IoT. O Bro-IDS analisa o tráfego de rede em um nível mais alto de abstração, mas requer a plataforma UNIX e não possui uma GUI suportada pelo Bro Project. No entanto, o Kismet suporta múltiplas plataformas, mas funciona apenas para redes sem fio [114]. O fato de o Kismet ser indetectável ao farejar redes é uma vantagem importante. Quanto ao OpenWIPS-ng, também é especializado em redes sem fio. Apesar de sua modularidade e escalabilidade, este NIDS não é famoso e não possui suporte da comunidade.

Finalmente, Onion security e Sagan são ferramentas mais importantes. Eles trazem muitas ferramentas em uma solução como Snort e Bro-ids na mesma ferramenta. Eles são considerados mais como Gerenciamento de eventos e informações de segurança (SIEM). Portanto, eles oferecem um rico painel de ferramentas. Eles garantem a facilidade de implantação e utilização dos sistemas integrados. Ambos podem lidar com monitoramento em tempo real, mas Sagan anuncia sua vantagem em termos de consumo leve de CPU e memória, o que é relevante para sistemas IoT.

Nesta Seção, foram apresentadas e discutidas ferramentas que podem ser utilizadas para construir um NIDS; de conjuntos de dados de rede gratuitos a sniffers de rede gratuitos e NIDSs de código aberto gratuitos. O restante do artigo apresentará e comparará o NIDS proposto por pesquisadores especificamente para segurança de sistemas IoT.

4. SISTEMAS DE DETECÇÃO DE INTRUSÃO DE REDE PARA IOT (NIDS)

Para ter uma ideia melhor sobre as arquiteturas e implantações IoT NIDS, a pesquisa discute a seguir, relevantes

funciona a partir do estado da arte da segurança NIDS IoT. Serão tratadas apenas soluções que não sejam baseadas em aprendizado de máquina, pois esse caso especial será o principal assunto da Seção V.

Pesquisas sobre NIDS para IoT demonstram que existem basicamente dois eixos de trabalho: i) NIDS para Redes de Sensores Sem Fio (WSN)3 [115] e ii) NIDS para sistemas IoT em geral. O primeiro eixo está fora do escopo de nosso artigo por dois motivos; primeiro porque muitas pesquisas o trataram em detalhes como [116], [117], [118] e [28]; segundo, porque nossa pesquisa se concentra em sistemas IoT com sua heterogeneidade e todos os desafios apresentados anteriormente e não em NIDS que não levam em conta ataques habilitados para Internet (WSN NIDS) [8].
O segundo eixo trata as ideias e implementações dos pesquisadores como uma continuidade e uma melhoria das pesquisas IoT NIDS mencionadas anteriormente [28] e [8]. Esta parte descreve trabalhos sobre NIDS para sistemas IoT com foco especial em mecanismos de detecção de NIDS, arquiteturas e estratégias de validação. Os papéis revisados do IoT NIDS são importantes no estado da arte.
Permitem acompanhar a evolução do domínio desde a primeira solução proposta [9] até aos dias de hoje. Esta subseção começa com uma descrição detalhada das soluções dos autores. Em seguida, os resume em uma Tabela IV comparativa com as vantagens e desvantagens de cada solução.

A. Estado da arte do NIDS para IoT

Esta seção detalha cada proposta de IoT NIDS com foco especial em arquiteturas, metodologias de detecção e ameaças tratadas.

Raça e outros. [9], [119] projetou e implementou o SVELTE, o primeiro IoT IDS. É um sistema de detecção de intrusão em tempo real baseado em uma assinatura híbrida acompanhada por uma técnica de detecção baseada em anomalias. O trabalho atende aos requisitos da IoT conectada ao IPv6 e concentra-se no roteamento de ataques, como spoofing e sinkhole. A SVELTE considera os desafios da IoT e implanta módulos IDS leves em nós com recursos limitados e módulos IDS com uso intensivo de recursos em
o roteador de borda (BR). Integra três módulos principais: i) 6Mapper (6LoWPAN4 Mapper) que coleta informações sobre a rede RPL5 e reconstrói a rede no 6BR, ii) Componente de Detecção de Intrusão que analisa os dados mapeados para detecção de intrusão; e iii) um mini-firewall (firewall de lista branca para a IoT conectada por IP que usa RPL como um protocolo de roteamento em redes 6LoWPAN) que é distribuído e projetado para descarregar nós filtrando o tráfego indesejado antes que ele entre na rede com recursos limitados. SVELTE foi implementado no Contiki OS. Ele detecta ataques de sumidouro com 90% de taxa de verdadeiro positivo (TPR) em uma pequena rede com perdas e quase 100% de TPR para uma configuração de rede sem perdas.
Infelizmente, os ataques DoS também podem afetar a solução [120]. Como os nós IDS usam a rede para transmitir informações de ataque, uma vez que o DoS afeta a rede, ele falha em detectar ataque de negação de serviço.

Kasinathan et ai. estudado [12], [120] Detecção de DoS para 6LoWPAN desenvolvida como parte do ebbits, um projeto EU FP76 . A arquitetura suportada apresentada na Fig. 6 é baseada no Suricata IDS (Seção III-C). A arquitetura proposta é considerada centralizada apesar das sondas IDS distribuídas. Na verdade, as sondas IDS, que são módulos externos, farejam a rede em modo promíscuo do que enviam dados para o NIDS principal (baseado no Suricata) via conexão com fio. Quando o último corresponde ao tráfego com uma assinatura de ataque, um alerta é lançado para o gerenciador de proteção DoS. O gerenciador de proteção analisa a tentativa com dados adicionais coletados de outros gerenciadores de ebbits, reduz a taxa de alarmes falsos (dados genuínos de detecção incorreta: falsos positivos mais falsos negativos). Essa solução supera as limitações do SVELTE, pois o mecanismo IDS não depende da arquitetura da rede, portanto, não pode ser afetado por ataques DoS contra a rede IoT. O trabalho de estrutura sugerido DEMO em [120], é escalável e aplicável em palavras reais para a maioria dos sistemas IoT. Este trabalho foi avaliado usando um sistema de teste de penetração (PenTest) Scapy [120] que é mais leve que o Metasploit [121]. O IDS adapta as tecnologias de código aberto existentes. Começa com o Suricata, que é um IDS de código aberto, e o modifica com os decodificadores IEEE 802.15.4 e 6LoWPAN. Além disso, um módulo de detecção adicional; FAM (consiste em um gerenciador de agilidade de frequência que analisa os estados de ocupação do canal em tempo real para permitir que a rede tome conhecimento do nível de interferência) e monitora ataques ao Prelude (que é um sistema de gerenciamento de incidentes e eventos de segurança (SIEM) para monitorar os eventos de ataque ou alertas). O mecanismo Suricata aciona alertas com base nas regras programadas; portanto, esta solução pode detectar diferentes ataques dependendo das regras desenvolvidas.

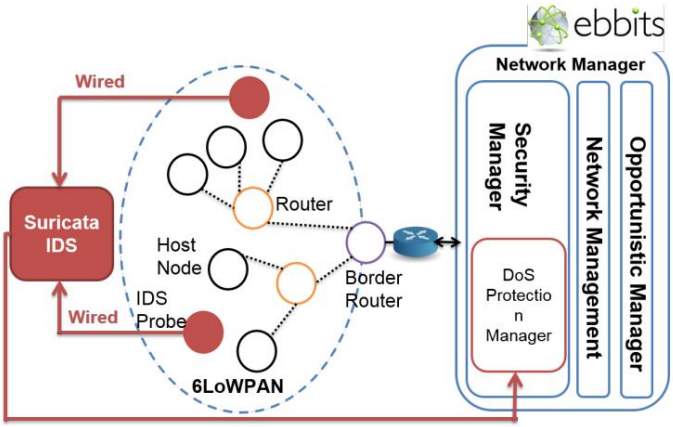

Fig. 6: Arquitetura DEMO [12], [120]

Jun e Chi [122] propuseram um IDS para sistemas IoT baseado na tecnologia Complex Event Processing (CEP), que é uma tecnologia emergente e eficiente para filtrar e processar dados reais.

3A WSN é uma rede sem fio geograficamente distribuída de sensores para Monitore as condições físicas ou ambientais.

4IPv6 sobre rede de área pessoal sem fio de baixa potência

5Protocolo de roteamento para baixa potência

6O projeto ebbits é um projeto de pesquisa europeu que lida com arquitetura, tecnologias e processos para permitir que empresas convencionais incorporem o ecossistema IoT.

eventos de tempo[7]. É uma boa solução para grandes volumes de meses sábios com baixa latência. Portanto, essa tecnologia pode ser adaptada às necessidades de IoT. Jun e Chi avaliaram o desempenho online em vez de offline. A arquitetura desta solução é esquematizada na Fig. 7. O sistema começa com a coleta de dados (tráfego de rede e uso de eventos) de dispositivos IoT, extrai eventos de dados detectados e executa a detecção de eventos de segurança usando o Repositório de Processamento de Eventos EPR[8] e o mecanismo CEP[9]. Finalmente, as ações são executadas pelo mecanismo de ação. Jun e Chin implementaram sua arquitetura IDS de processamento de eventos usando Esper (mecanismo CEP para processamento de eventos complexos e análise de séries de eventos). Sua abordagem é intensiva em CPU, mas consome menos memória. Efetivamente provou melhor desempenho em tempo real.

Por exemplo, para 800k de dados, o IDS baseado em CEP consome 62% da CPU, 730 MB de memória e 422 milissegundos de tempo de processamento. No entanto, o IDS tradicional usa 57% da CPU, 1064MB de memória e 8688ms para processar 800k de dados. É interessante notar que a estrutura é projetada, mas não avaliada para qualquer tipo de detecção de ataque.

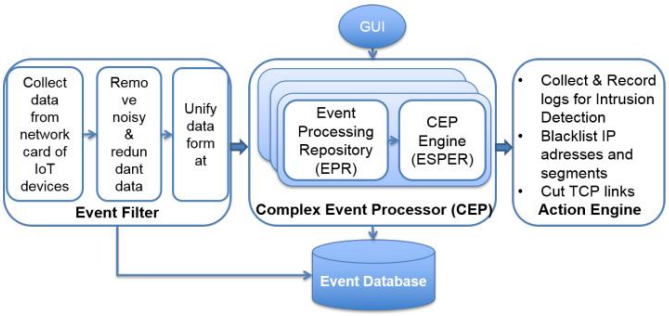

Fig. 7: Arquitetura IDS baseada em CEP para IoT [122]

Cervantes et ai. [123] detectou o perigoso ataque de sumidouro nos serviços de roteamento em IoT. Eles propuseram a detecção de intrusão de ataques SiNkhole em 6LoWPAN para Internet of ThIngs (INTI). Ele combina estratégias de vigilância, reputação e confiança para detecção de invasores. Primeiro, como uma estrutura hierárquica, os nós (agrupados ou separados) são classificados como líderes. Então, os nós podem mudar de função ao longo do tempo com base nos requisitos da rede. Cada nó monitora um número de transmissões realizadas por um nó superior. Se um ataque for detectado, uma mensagem de alerta é transmitida e um isolamento cooperativo do nó malicioso é executado. Cervantes et ai. deu importância para a mobilidade do nó e autorreparação da rede, que são limitações com Raza et al. [9] abordagem. Seus resultados de simulação mostram uma taxa de detecção de sumidouros de 92% no cenário de 50 nós fixos e de 75% para 50 nós móveis.

Além disso, os autores relataram baixos falsos positivos e falsos negativos em comparação com SVELTE.

Surendar e Umamakeswari [124] armaram uma invasão Sistema IoT de Detecção e Resposta (InDReS) com 6LowPAN. O InDReS usa uma técnica de especificação baseada em restrições para detectar ataques de sumidouros em redes RPL. No InDReS, os nós sensores são agrupados em clusters sob a supervisão de um nó observador. Os nós observadores contam os pacotes descartados de seus nós adjacentes e atribuem uma pontuação a cada um deles usando a teoria de Dempster Shafer para detectar o nó malicioso. Este último será anunciado para que todos os nós cooperam para isolá-lo. Finalmente, a rede se reconstrói. A estratégia dos autores melhora a eficiência de algumas métricas críticas de QoS em relação ao esquema INTI existente, que é limitado pelo consumo médio de energia e taxa de queda compactada. Os autores simularam sua proposta no simulador NS2.

Fu et al. [125] apresentou um sistema uniforme de detecção de intrusão considerando os seguintes dois pontos importantes: i) variada heterogeneidade de redes IoT; ii) Sensor IoT e restrição de recursos de dispositivos inteligentes. Os autores afirmam que sua solução é a primeira que se beneficia da teoria dos autômatos para modelar e detectar as invasões de redes IoT. Com base em uma extensão dos sistemas de transição rotulados, eles forneceram uma descrição uniforme dos fluxos de tráfego de rede dos sistemas IoT e compararam os fluxos de ação em tempo real com as bibliotecas de protocolo padrão para detectar e relatar: ataque de bloqueio, ataque falso e ataque de resposta. A abordagem proposta é composta pelos quatro componentes a seguir:

• O Event Monitor coleta o tráfego de rede e transmite dados em arquivos digitais para o IDS Event Analyzer. Este componente deve ser implementado no coordenador PAN (Personal Area Network) ou outros gateways IoT para monitorar o tráfego de rede. • Event Database implementa três bancos de dados armazenados na nuvem: i) Standard Protocol Library (a descrição dos protocolos padrão através do Glued-IOLTS [126]); ii) Abnormal Action Library (fluxos de ações de anomalias reconhecidas pelo sistema); e iii) Normal Action Library (possíveis fluxos de ação criados a partir das Standard Protocol Libraries utilizando as técnicas de Fuzzing [127] e Robustness Testing [128]). • O IDS Event Analyzer é composto pelos três modelos básicos a seguir:

1) Modelo de Aprendizagem da Estrutura de Rede: considera os dados do pacote como entrada, constrói uma visão geral das topologias da rede, distingue IDs de dispositivos IoT e os envia para a Abstração de Fluxos de Ação Modelo,
2) Modelo de Abstração de Fluxos de Ação: classifica os pacotes coletados em tempo real da IoT em sequências de mensagens e, em seguida, traduz essas mensagens em fluxos de ação abstratos com a ajuda do Protocolo Padrão Biblioteca,
3) Modelo de Detecção de Intrusão: compara o resultado do Modelo de Abstração de Fluxos de Ação com a Biblioteca de Ações Anormais. Se corresponder, o fluxo de ação será marcado como intrusivo; caso contrário, um método de detecção de anomalias será aplicado. Para o último, se a sequência de transição de entrada não corresponder às entradas da Biblioteca de ações normais, um manual especializado verifica

As tecnologias 7CEP mesclam dados de várias fontes para interpretar ações em tempo real de eventos ou padrões complicados.

8EPR é um repositório da declaração do Modelo de Processamento de Eventos EPM que é uma coleção de correlação de eventos.

O mecanismo 9CEP analisa uma massa de eventos, identifica os mais importantes e produz ações.

ção é necessária (para evitar o falso positivo). Se finalmente for marcado como seguro, o registro será adicionado à Biblioteca Normal; caso contrário, será adicionado à Biblioteca de Ações Anormais. • Unidade de Resposta:

reporta três tipos de ataques (jam-attack, falso-ataque e resposta-ataque) a uma estação de gerenciamento.

Fu et al. experimentaram sua solução no ambiente de experimento IoT, mas, infelizmente, não apresentaram taxas de detecção.

Midi et ai. [129] propuseram Kalis, que é "a primeira abordagem para detecção de intrusão para IoT que não visa um protocolo ou aplicativo individual e adapta a estratégia de detecção aos recursos específicos da rede". Kalis é um "IDS online baseado em rede, híbrido baseado em assinatura/anomalia, híbrido centralizado/distribuído que se adapta a diferentes ambientes". Ele pode ser implantado como uma ferramenta autônoma em um dispositivo externo separado (para superar o fato de que a maioria dos dispositivos IoT não oferece suporte a alterações de software). É um IDS automático baseado em conhecimento, o que significa que ele escolhe automaticamente as técnicas de detecção dependendo dos recursos coletados da rede. Precisamente, cada ataque pode ser feito apenas em alguns sistemas IoT e não em outros, dependendo dos recursos do sistema, por exemplo, não é possível ter um ataque de replicação em um sistema de salto único.

Kalis identifica ainda a presença, ou não, de técnicas de prevenção como o uso de funções criptográficas.
Assim, Kalis é eficaz e eficiente em relação ao consumo de recursos. Kalis foi implementado usando Java em uma placa de desenvolvimento Odroid xu3 e avaliado com dispositivos IoT do mundo real. O sistema inclui "uma pequena WSN de seis nós TelosB, um Nest Thermostat, um August SmartLock, uma lâmpada inteligente Lifx, um sistema de segurança Arlo e um Amazon Dash Button". Para farejar saltos intermediários de pacotes de dados, Kalis foi localizado perto da porção central da WSN. Midi et ai. Traços reais reproduzidos do tráfego de rede do protótipo e pacotes adicionais adicionados com 50 instâncias de sintomas diferentes para cada ataque. A taxa de detecção de Kalis é de 91% com 100% de precisão, 0,19% de uso da CPU e 13.978,62 KB de consumo de memória. Aqui, é importante observar que o uso tradicional de IDS tem cerca de 48% de taxa de detecção com 75% de precisão, 0,22% de uso de CPU e 23.961,06 KB de RAM.

B. Comparação e Discussão

A seguir, é fornecida uma comparação de propostas revisadas anteriormente para IoT NIDS. Um resumo e uma comparação visual são fornecidos na Tabela IV.
Como pode ser notado, a maioria dos trabalhos utiliza arquiteturas distribuídas [9], [123], [124], [125]. Esse **tipo de implantação** é mais adequado para sistemas IoT do que estratégias centralizadas [120], [122], uma vez que a distribuição de dispositivos é uma característica importante da IoT. No entanto, os IDS centralizados detectam melhor os ataques de segurança que envolvem um grupo de dispositivos operando silenciosamente (sem desligar diretamente a rede) do que os distribuídos. Por exemplo, ataques DDoS são difíceis de detectar em uma implantação distribuída.
Para tais ataques, uma arquitetura híbrida como [129] é mais apropriado. Assim, é garantida uma análise de rede distribuída com uma inspeção geral centralizada (também chamada de estratégia hierárquica). Além disso, a implantação do NIDS no próprio sistema IoT ou em um dispositivo externo separado é importante. Kasinathan et ai. [120] e Midi et al. [129] são os únicos que propuseram seu NIDS como uma ferramenta independente.
A adoção de tal estratégia é considerada limitada, uma vez que as restrições de recursos não representam mais um desafio. Isso supera o problema dos dispositivos IoT sem alterações de software. Isso permite a proteção do sistema IoT inicial contra sobrecarga de rede e dispositivo. Assim, empregar infraestrutura adicional aumenta a complexidade no caso de manutenção de rede e proteção do sistema.

No que diz respeito **à metodologia de detecção,** tanto a detecção de assinaturas quanto a detecção de anomalias são implantadas. Cada método tem suas vantagens e suas desvantagens. A detecção baseada em assinatura é eficiente para ataques conhecidos; no entanto, ele não pode detectar ataques desconhecidos, pois o banco de dados de assinaturas deve ser atualizado e demorado. Quando o tamanho do banco de dados de assinaturas aumenta, o NIDS é solicitado a comparar a entrada com todas as assinaturas existentes. A detecção de anomalias detecta ataques desconhecidos/invisíveis; no entanto, sofre de alarmes falsos altos. Consequentemente, a detecção híbrida, como [125] e [129], foi implantada como soluções práticas.

Em relação à **estratégia de validação,** dois parâmetros importantes são identificados: simulação e emulação. A simulação modela o comportamento do sistema de destino em um ambiente diferente. Ele fornece o comportamento básico de um sistema; pode não necessariamente aderir às regras do sistema original. A emulação duplica exatamente o mesmo comportamento de destino do sistema original operando em um ambiente diferente.
Portanto, a emulação é mais próxima da situação da vida real quando comparada com a simulação [120], [125] e [129]. A simulação é aceitável em IoT, pois a implementação de um sistema IoT requer um grande número de dispositivos físicos para se aproximar da realidade, o que não é uma tarefa fácil para pesquisas experimentais.
O segundo ponto a ser discutido sobre a validação são as métricas de avaliação. Os resultados da revisão mostram que os pesquisadores nem sempre fornecem as mesmas métricas [130], [131] na avaliação de seus trabalhos, o que não permite uma comparação verdadeira e justa;

• Taxa de detecção (DR) é a taxa de detecção de intrusão real ções ao número total de invasões. O DR é diferente da taxa de precisão (taxa preditiva positiva), que representa uma fração de instâncias de dados previstas como positivas que são realmente positivas. Cervantes et ai. [123] relataram taxa de detecção de 92% em uma rede composta por 50 nós fixos e 75% para 50 nós móveis. Midi et ai. [129] alcançou uma taxa de detecção de 91% contra 48% com o IDS tradicional e 89% com a ferramenta snort.

• Precisão é a capacidade de diferenciar intrusões e comportamentos normais corretamente. Representa a proporção de invasões classificadas corretamente para o número total de entradas. Midi et ai. [129] alcançou 100% de precisão, enquanto o IDS tradicional teve 75% e o snort relatou 76%.

• A taxa de falsos positivos (FP) representa o tráfego normal classificado erroneamente como intrusivo. Fu et al. [125] considerou a métrica FP

TABELA IV: Comparação do NIDS para IoT

IDs de referência	de detecção	Implantação metodologia	validação Estratégia	tratado Ameaças	Vantagens	Desvantagens
Raça e outros. [9], [119]	distribuído	Híbrido (baseado em assinatura e anomalia)	simulação	Ataques de roteamento como spoofing e sinkhole, encaminhamento seletivo e alteração de informações	• O desafio das restrições de recursos é levado em consideração • O mini-firewall distribuído para os dispositivos IoT conectados por IP é integrado • Flexível e pode ser estendido para detectar mais ataques	Ataque DoS pode afetar SVELTE
Kasinathan et ai. [12], [120]	Ataque DoS de emulação baseada em assinatura centralizada				• Redução de falsos alarmes • O IDS é implantado em infraestrutura adicional • Cabo escalável e aplicável em palavras reais • Detecção em tempo real • Melhor desempenho em tempo	Ataques detectados dependem de regras declaradas
junho e Qui [122]	Baseada em assinatura centralizada —			—	real • Baixo consumo de memória • IoT Dados maciços são levados em consideração	• Uso intensivo de CPU • Ataques detectados dependem de regras declaradas
Cervantes et ai. [123]	distribuído	Híbrido (estratégia de confiança e reputação)	simulação	ataque de sumidouro	• O INTI leva em consideração a mobilidade dos nós e o auto-reparo da rede • Menos taxa de falsos positivos e falsos negativos do que o SVELTE • O desafio das restrições	• As veiculações de IDS mudam ao longo do tempo, o que pode consumir mais recursos.
Surendar e Uma makeswari [124]	distribuído	Especificação baseada	simulação	ataque de sumidouro	de recursos é levado em consideração • Baixa energia média com consumo • Baixa taxa de perda de pacotes • Resposta instantânea da rede contra ataques detectados • A heterogeneidade dos trabalhos de rede IoT é levada	• Não é possível detectar rumos desconhecidos
Fu et al. [125]	distribuído	Híbrido (baseado em assinatura e anomalia)	Emulação	Jam-ataque, ataque falso e ataque de resposta	em consideração • O desafio das restrições de recursos é levado em consideração • Baixa taxa de falsos positivos	• O algoritmo baseado em estado pode causar "explosão de espaço de estado" • A intervenção humana é necessária para alarmes falsos positivos • O ataque DoS pode afetar a solução
Midi et ai. [129]	Híbrido (centralizado e distribuído)	Híbrido (baseado em assinatura e anomalia)	Emulação	DoS, roteamento e ataques de rede convencionais	• Detecção em tempo real • Leve em termos de requisitos de CPU e RAM • IDS auto-adaptável dinâmico • Automático baseado em conhecimento IDS • Diferentes protocolos e aplicativos de comunicação IoT são levados em consideração • Implantável no roteador de borda ou como uma ferramenta autônoma	• A perspectiva de alto nível pode não ser adequada para objetos de computação restritos. • Kalis propõe implantação em tempo de compilação que pode não ser viável para recursos com sensores sobrecarregados, que podem até mesmo ter recursos limitados em comparação com nós WSN.

enquanto discutem sua solução sem valor concreto. • A taxa de verdadeiro positivo (TP) identifica com sucesso o ataque que se refere ao número de invasões que são detectadas como intrusões. Raça e outros. [9] relataram 90% de TP em uma pequena rede com perdas e 100% em uma sem perdas. • Consumo de energia e queda de pacotes e taxa de entrega de pacotes são as métricas usadas por Surendar et al. [124]. Eles obtiveram melhores resultados em comparação com Cervantes et al. • O uso da CPU e o consumo de memória são levados em consideração em Midi et al. experimentos [129]. Eles consumiu 0,19% de CPU e 13978,62Kb de memória vs 0,22% e 23961,06Kb no IDS tradicional e 6,3% e 101978,24Kb com snort.

Alguns trabalhos não forneceram resultados experimentais como em [120] ou nem mesmo experimentaram sua solução como em [122]. Métricas de avaliação precisam ser fixadas e processadas em cada trabalho para ter uma comparação confiável, mesmo que as métricas utilizadas dependam dos objetivos e aspectos em que cada estudo se concentra.

Sobre as **agressões tratadas** em artigos revisados, conforme Tabela IV,

não há trabalho que leve em consideração todas as ameaças ao mesmo tempo. Normalmente, o NIDS baseado em metodologia de detecção de anomalias ou híbrida deve ser capaz de detectar todos os tipos de ataques, mas nenhum dos trabalhos revisados se concentra na detecção do máximo de tipos de ataque. [9] é o único trabalho que menciona que a solução poderia ser expandida para detectar mais do que os ataques experimentados.

IoT é um ambiente de protocolos e tecnologias coexistentes. Apesar do aspecto de heterogeneidade da IoT, [122], [125] e [129] têm a capacidade de detectar múltiplos protocolos. [9], [120] e [123] focam em intrusões em 6LoWPAN e RPL que são técnicas importantes para redes IoT. Além disso, a complexidade é alta devido à heterogeneidade. Além disso, o desafio das restrições de recursos é considerado por [9], [120], [124], [125] e [129]. A escalabilidade, por outro lado, tem sido objeto de estudo em [120] e [122]. Para [122], era mais sobre escalabilidade de dados.

Finalmente, [123] é a única proposta que considera mobilidade e conectividade.

Os pontos fortes e fracos de cada solução foram identificados conforme ilustrado na Tabela IV.

Esta seção discutiu o NIDS tradicional para sistemas IoT. Detalhes sobre sua arquitetura, metodologias de detecção e resultados experimentais foram fornecidos. Além disso, realizamos uma comparação completa do NIDS para IoT, seus pontos fortes e fracos enquanto avaliamos seus prós e contras. A próxima seção discute, avalia e compara IoT NIDS com base em técnicas de aprendizado.

V. NIDSS PARA SISTEMAS IOT BASEADOS NA APRENDIZAGEM TÉCNICAS

Antes de passar dos NIDSs para IoT para os baseados em técnicas de aprendizado, o artigo discute brevemente as técnicas de aprendizado, as define e discute sua classificação. Por outro lado, NIDSs IoT alimentados por técnicas de aprendizagem são pesquisados. Trabalhos de ponta são comparados e discutidos para extrair pontos fortes e semanas de cada um.

A. Técnicas de aprendizagem

Os avanços tecnológicos alimentados por IoT são complementados por meio de armazenamento em nuvem, mineração de dados e análise de big data. Em 2015, Ben Walker no vouchercloud [132] relatou que os aplicativos da web atuais geram 2,5 quintilhões de bytes por dia. A explosão massiva de dados se deve à relevância das mídias sociais e da IoT na vida cotidiana. Além disso, mais de noventa por cento dos dados não são estruturados e/ou incompreensíveis. Aqui entra o papel do Big Data [133] e as técnicas de aprendizagem. Big Data oferece tecnologias e arquiteturas que visam tratar, organizar e lucrar com big data. Ele permite o tratamento paralelo distribuído. Segundo o Gartner, o Big Data ajuda a lidar com as regras dos 3V: Volume (explosão do volume de dados), Velocidade (a frequência de criação de dados é variada e pode chegar a frações de segundos) e Variedade (os dados são gerados de diferentes fontes e em diferentes formatos estruturados/não estruturados formatar).

Depois de preparar e tratar os dados coletados, começa o papel de análises e algoritmos inteligentes. **A análise de big data** permite descobrir tendências de mercado, preferências do cliente, prever o comportamento do cliente por meio da associação de dependências entre variáveis de entrada ou com ferramentas e técnicas de Data Mining (DM). **A mineração de dados** trata de explorar os dados em busca de padrões ocultos, relacionamentos e correlações anteriormente desconhecidas para prever e reagir. Com a mineração de dados, vem o conceito de Machine Learning (ML). **O aprendizado de máquina** é uma técnica de inteligência artificial amplamente utilizada na mineração de dados. No ano de 1959, Arthur Samuel, o pioneiro do Aprendizado de Máquina (ML), definiu ML como “campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados” [134].

Consiste na implantação de algoritmos para obter uma análise preditiva dos dados (Aprender com exemplos). Existem basicamente dois tipos de algoritmos de ML:

- O aprendizado supervisionado é baseado no aprendizado de dados de treinamento rotulados, o que significa que os dados de treinamento incluem a entrada e os resultados desejados.
- O aprendizado não supervisionado é baseado no agrupamento dos dados de entrada em classes apenas com base em suas propriedades estatísticas. Em outras palavras, descreve a estrutura oculta de dados “não rotulados” (sem classificação ou categorização predefinida nas observações).

ML é baseado em um conjunto de recursos que identificam um estado de um objeto. Os recursos devem ser escolhidos com cuidado e precisão para evitar imprecisões e consumo de tempo desnecessário.
Para obter um bom aprendizado de dados com resultados precisos e eficientes, recursos não relacionados ou irrelevantes devem ser removidos do conjunto. Tal processo é conhecido como redução de recursos ou redução de dimensionalidade. É uma ferramenta necessária para analisar dados ruidosos de alta dimensão [135]. Aplicar uma redução de dimensão em um dado original de alta dimensão em termos de recursos preservaria (seleção de recursos [136]) ou geraria apenas recursos importantes.

Deep Learning (DL) [137] faz parte de uma família mais ampla de métodos de ML baseados no aprendizado de representações abstratas de alto nível. DL agrupa algoritmos genéricos que imitam o funcionamento biológico de um cérebro sem serem destinados a uma tarefa específica. Tecnicamente, DL é a aplicação de redes neurais artificiais (ANNs) que contêm várias camadas ocultas.

Os pesquisadores começam a colocar mais energia na exploração de algoritmos de aprendizado de máquina no NIDS por vários motivos:

- Ataques desconhecidos / dia zero evitam os NIDSs tradicionais baseados em assinatura; Considerando que os algoritmos de aprendizado de máquina supervisionados têm um potencial interessante na detecção de novos ataques.

- Os NIDSs tradicionais sofrem com uma alta taxa de falso reconhecimento que pode ser reduzida com técnicas de aprendizado de máquina. Comparado ao IDS de assinatura/ sem assinatura, um IDS/NIDS equipado com ML emprega estatística, genética e heurística ou uma combinação deles para disseminar um padrão de ataque complexo para melhorar a taxa de detecção com falsos negativos reduzidos. • As soluções tradicionais sofrem de propriedades complexas de ataque

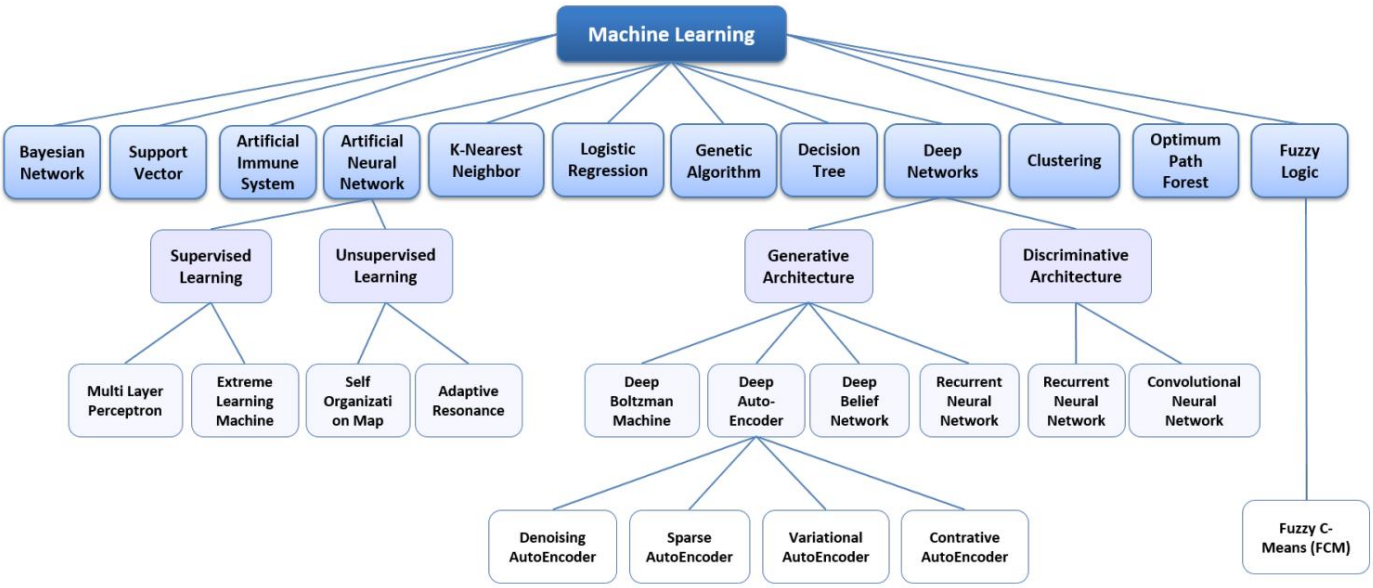

Fig. 8: algoritmos de ML

captura de laços, enquanto o aprendizado de máquina pode melhorar a precisão e a velocidade da detecção.

• Pequenas variações nos ataques não podem ser efetivamente detectadas com NIDSs tradicionais. Mesmo um detector heurístico pode ser evitado invertendo o padrão de ataque. No entanto, os IDSs equipados com ML aprendem o padrão de tráfego recente continuamente; Portanto, eles identificam efetivamente pequenas variações no padrão de tráfego. Em outras palavras, os algoritmos de ML são eficientes contra a detecção de variantes [27].

• Os cibercriminosos implantam padrões de ataque em evolução para escapar dos detectores. Os NIDSs tradicionais, especialmente os NIDSs baseados em assinatura, requerem atualizações contínuas. No entanto, NIDSs de ML baseados em clustering e detecção de outliers não precisam de atualizações regulares. • As soluções tradicionais e mais precisamente os IDSs baseados em assinaturas combinam cada assinatura com o banco de dados IDS. O processo consome CPU (grande banco de dados de assinaturas que cresce exponencialmente); enquanto o IDS baseado em ML consome processamento de baixo a médio. Portanto, o elemento de processamento pode ser usado de forma eficaz.

Consequentemente, as técnicas de aprendizagem parecem ser uma solução adequada, especialmente com os bons resultados que alcançam nos diferentes domínios.

Uma classificação para os algoritmos de ML mais populares é estabelecida na Fig. 8. A taxonomia de ML é apresentada neste artigo para ajudar o leitor a identificar o tipo de algoritmo usado nos NIDSs revisados posteriormente. No entanto, a descrição detalhada do algoritmo de ML está fora do escopo de nosso artigo. Várias pesquisas, como [24], [25], [13], [15], [26] e [27] trataram ML para detecção de intrusão. Uma descrição sobre cada pesquisa é detalhada.

B. IoT NIDSs baseados em técnicas de aprendizado de máquina

Com a evolução, complexidade e diversidade dos ataques de segurança, os pesquisadores estão se concentrando no uso de inteligência artificial e aprendizado de máquina para ameaças à segurança detecção. Para fazer isso, o IDS deve incorporar a inteligência da máquina e melhorar os recursos de tomada de decisão [138].

Muitos estudos aplicam ML em IDSs e fornecem resultados promissores. Agrawal et al., Buczak e Guvan, Fadlullah et al., Hodo et al., Wang e Jones e Mishra et al. apresentam diferentes IDSs de aprendizado em suas pesquisas [25], [15], [26] onde algoritmos de ML, como árvores de decisão (DT), máquinas de vetor de suporte (SVM), bayes naive (NB), redes neurais artificiais (ANN), k -means clustering, lógica difusa, algoritmos genéticos, codificador automático empilhado (SAE), foram implantados separadamente e combinados [139], [140] para melhorar o resultado em sistemas gerais. Conforme evocado em [19], a ciência começou com a aplicação de cada algoritmo de inclinação de máquina separadamente, do que ocorre a combinação dos algoritmos no mesmo sistema.

Uma vez que nosso foco principal é a implantação de IDS inteligentes em IoT, agora discutiremos singularmente NIDSs para IoTs empregando técnicas de aprendizado. O restante do artigo apresenta uma descrição detalhada de cada proposta e, na próxima subseção, serão discutidas as escolhas dos pesquisadores e os resultados.

Hodo et al. [17] usaram Multi-Layer Perceptron (MLP), que é um tipo de Rede Neural Artificial (ANN) supervisionada em um IDS IoT off-line. Sua análise é construída em rastreamentos de pacotes de internet e tende a detectar ataques DoS e DDoS na rede IoT. Suas características MLP são:

• Rede neural retroalimentada e retroalimentada de três camadas. • função de transferência sigmoide unipolar em cada um dos neurônios das camadas oculta e de saída. • algoritmo de aprendizado estocástico com erro quadrático médio função.

O NIDS foi testado em uma simulação composta por quatro nós clientes e um nó servidor relay. Os ataques DOS/DDoS foram executados no nó do servidor com 10 milhões de pacotes UDP enviados de um único host para ataque DoS e com três hosts na velocidade do fio para DDoS. O conjunto de dados de treinamento foi composto por 2313 amostras, das quais 496 amostras foram implantadas

para validação e 496 amostras foram usadas para teste. A precisão geral da detecção de ataques foi de 99,4% com 0,6% de falsos positivos.
Tais resultados garantem uma boa estabilidade da rede.

Nobakht et ai. [141] propuseram uma estrutura IDS baseada em host IoT-IDM para dispositivos inteligentes escolhidos pelo usuário em ambiente de casas. O IoT-IDM monitora o tráfego que passa pelos dispositivos para identificar ameaças. A estrutura se beneficia da arquitetura de rede definida por software (SDN) com técnicas de aprendizado de máquina para detectar hosts comprometidos e mitigar esses ataques, pressionando as ações apropriadas (como bloquear o intruso ou redirecionar o tráfego malicioso) para roteadores/switches subjacentes. Nobakht et ai. implantou o protocolo OpenFlow, que é um padrão de rede para implantar a implementação de SDN. SDN abstrai serviços de rede. Separa o plano de controle (o tomador de decisão sobre o encaminhamento de dados) do plano de dados (o responsável pelo envio dos dados). Essa tecnologia oferece a oportunidade de gerenciar remotamente a segurança que leva a fornecer ao usuário do IoT-IDM um Security as a Service (SaaS). A solução de Nobakht et al. é caracterizada pela modularidade no design: é composta por cinco módulos separados (gerenciador de dispositivos, elemento sensor, extrator de recursos, unidade de detecção e unidade de mitigação). Consequentemente, há uma flexibilidade para escolher o algoritmo de aprendizado de máquina a partir de um conjunto de técnicas fornecidas. Os algoritmos de ML usam padrões de assinatura aprendidos de ataques conhecidos para treinar o modelo. Além da ampla gama de ataques detectados, uma das desvantagens do IoT-IDM é que tecnicamente ele não pode pesquisar todos os dispositivos IoT domésticos devido ao alto volume de tráfego de rede com o detalhe de que os elementos do sensor estão posicionados no topo do controlador SDN. Consequentemente, IoT-IDM só pode inspecionar dispositivos IoT escolhidos que não sobrecarreguem o controlador SDN. Nobakht et ai. testou o IoT-IDM em um dispositivo IoT real que é a lâmpada inteligente (luzes Hue) e comparou técnicas de aprendizado de máquina de regressão logística e SVM (máquinas de vetor de suporte). Na detecção não autorizada, o primeiro fornece 94,25% de taxa de precisão e 85,05% de taxa de recuperação contra 98,53% e 95,94% para SVM.

Hossein pour et ai. [18] propuseram um novo IDS em tempo real, distribuído e leve, baseado no Sistema Imunológico Artificial (AIS), uma combinação eficaz de borda, névoa e computação em nuvem. A Cisco introduziu o conceito de computação em névoa para estender a computação em nuvem na camada de rede. A camada de névoa está entre os sensores IoT e a nuvem. A camada é equipada com capacidade de computação em névoa (processamento inteligente de dados em um nível intermediário) para eficiência e redução do transporte de dados para a nuvem. Consequentemente, o processamento ocorre em hubs, roteadores ou gateways. Essa tecnologia permite a detecção de ataques distribuídos; eficiente em termos de escalabilidade, autonomia na detecção de ataques locais, aceleração no treinamento de dados próximos às fontes e compartilhamento de parâmetros dos vizinhos. Os autores avaliaram detectores na camada de borda, alertas de intrusão tratados com o conceito de dados inteligentes na camada de névoa. Um agrupamento de tráfego de rede primário e treinamento de detector é realizado na nuvem. Seguem os importantes reforços do trabalho: i) Fog computing possibilitou qualidade de serviço com baixa latência na análise de dados; ii) Combinação de análise leve na camada de névoa com análise avançada na nuvem; iii) Detecção de ataques silenciosos, como ataques de botnet usando estratégia de dados inteligentes ("uma estrutura de dados ativa e inteligente que facilita o gerenciamento de Big Data em IoT" [18]); e iv) detecção de ataques desconhecidos e de dia zero via AIS com base em um método de autotreinamento online com aprendizado de máquina não supervisionado. O algoritmo AIS do IDS é composto das três partes a seguir:

1) Um mecanismo de treinamento: aprende a partir de um conjunto de dados de aprendizado inicial e treina detectores (fase de inicialização do AIS). Esta etapa é tratada na camada de nuvem, pois necessita de unidades de processamento complexas e poderosas.
2) Um mecanismo analisador: analisa as anomalias relatadas pelos detectores para alertar e rejeitar os sinais falsos positivos.
   Os autores usam detectores de células de memória e algoritmos genéticos como apresentados em trabalhos anteriores [142] e [143] para melhorar a precisão. Esta etapa requer mais comunicação entre os nós de borda infectados e o mecanismo principal, portanto, o mecanismo do analisador é implantado na camada de névoa.
3) Sensores detectores: a lógica de detecção é inserida em cada nó monitorando a rede. O IDS proposto é dotado de uma detecção inteligente e distribuída onde cada tipo de ataque pode ser detectado por vários detectores diferentes. Se um limite for atingido, a anomalia será relatada ao mecanismo do analisador, gerando um alerta de intrusão profunda.

Dois conjuntos de dados foram usados para avaliar a eficiência leve do IDS, que são KDD-Cup99 e SSH Brute Force do conjunto de dados ISCX [144]. De acordo com os resultados experimentais, a solução proposta de três camadas atinge 3,51% de taxa de falsos positivos com 98,35% de exatidão e 97,83% de precisão.

Bostani e Sheikhan [37] sugeriram um híbrido em tempo real de IDS baseado em anomalia e baseado em especificação.
Ele permite a detecção de ataques de sumidouro e encaminhamento seletivo em redes 6LowPAN e pode ser estendido para detectar ataques de buraco negro, classificação e buraco de minhoca. Este IDS funciona principalmente em duas etapas: detecção de especificação no nível do roteador e detecção de anomalias no nível raiz. Para o primeiro, os roteadores analisam os recursos localmente do tráfego de rede e dos nós de host. Os resultados da primeira etapa são enviados para o nó raiz da segunda etapa e removidos dos roteadores para garantir menor consumo de memória e ciclos de CPU. A segunda etapa é a detecção de intrusão global, onde a análise baseada em anomalias é realizada nos pacotes de dados recebidos no nó raiz. Esta etapa emprega o algoritmo de floresta de caminho ideal (OPF) não supervisionado para criar modelos de agrupamento para cada roteador de nó de origem. Com uma plataforma de arquitetura MapReduce, é assegurada uma execução paralela e distribuída da detecção de anomalias de acordo com modelos de agrupamento. A decisão final sobre marcar um comportamento suspeito como um ataque é feita com um mecanismo de votação. O sistema proposto não usa mensagens de controle adicionais, nem faz uso de infraestrutura adicional. Conseqüentemente, ele economiza em custos de comunicação e configuração em comparação com outros IDS. Os autores avaliaram a proposta

técnica em sua própria ferramenta de simulação. Eles provam resultados apropriados de detecção em tempo real com três experimentos principais, cada um feito com dez simulações: o primeiro experimento lida com valores de critérios de avaliação, o segundo experimento aborda a escala das redes (pequeno e médio porte) para confirmar a escala independente -network IDS e o terceiro comprova a possibilidade de estender os ataques detectados, como wormhole.
Os resultados experimentais dos cenários simulados mostraram que, quando os ataques de sumidouro e encaminhamento seletivo foram lançados simultaneamente, o método híbrido proposto pode atingir taxa de verdadeiros positivos de 76,19% e taxa de falsos positivos de 5,92%. No entanto, para ataque de buraco de minhoca as taxas são de 96,02% e 2,08%, respectivamente.

Bostani e Sheikhan resumiram em [145], [146] a mesma arquitetura fornecida em [37] (ou seja, com base no Modelo MapReduce distribuído). Eles propuseram uma anomalia e agentes de uso indevido com modelo de floresta de caminho ótimo supervisionado e não supervisionado, em vez de detecção baseada em anomalia e especificação.
Eles também reduziram os recursos do conjunto de dados com um algoritmo de seleção de recursos híbridos, construído sobre informações mútuas e algoritmo de busca gravitacional binária.

Pajouh et al. [147] apresentou um IDS de anomalia construído com redução de dimensão de duas camadas e classificação de duas camadas (TDTC) para backbone IoT. Eles se concentraram principalmente em ataques comuns de baixa frequência: ataques de usuário para root e remoto para local, enquanto seus experimentos foram baseados no conjunto de dados NSL-KDD. Pajouh et al. implantou uma redução de dimensão de duas camadas para limitar a alta dimensionalidade do conjunto de dados: • A primeira camada se beneficia de uma técnica não supervisionada que é a Análise de Componente Principal (PCA) para redução de dimensão de recurso (combina recursos de conjunto de dados para construir novos). Portanto, para o NSL-KDD, a complexidade do overhead foi reduzida no TDTC, pois apenas 35 dos 41 recursos do conjunto de dados foram usados.

• A segunda camada usa uma técnica supervisionada: Análise Discriminante Linear (LDA) para tornar os recursos reduzidos do PCA melhores para classificação e para melhorar a velocidade de detecção de intrusão. Depois de analisar as classes do conjunto de dados, o LDA termina com um conjunto de dados de duas dimensões para NSL-KDD.

Essa redução de dimensão diminui a taxa de detecção de falsos positivos e a complexidade computacional. A segunda etapa é a classificação multicamadas onde o TDTC usa Naive Bayes (NB) e a versão de Fator de Certeza do K-Nearest Neighbor (KNN) para classificar as entradas. Pajouh et al. iniciado com NB para detecção de anomalias, então os resultados são refinados com CF-KNN. Seu trabalho provou uma redução de computação de cerca de dez vezes com detecção mais rápida e menos requisitos de recursos. Eles alcançaram uma taxa de detecção de cerca de 84,86% para classificação binária com 4,86% de alarme falso.

Lopez-Martin et al. [148] propuseram uma anomalia não supervisionada NIDS para IoT baseada no Conditional Variational AutoEncoder (CVAE). Seu método é único devido à sua capacidade de realizar reconstrução de recursos, ou seja, ele pode recuperar recursos ausentes de conjuntos de dados de treinamento incompletos. Os autores alegaram atingir 99% de precisão na recuperação de características categóricas. Além disso, esse recurso torna a proposta sistema relevante para redes IoT, pois são mais sensíveis em termos de problemas de conexão e detecção de erros que afetam os dados enviados/recebidos. Como entradas para o CVAE de detecção de intrusão (ID-CVAE), eles usaram não apenas recursos de intrusão (como no Variational AutoEncoder VAE), mas também rótulos de classe de intrusão. Apesar do aspecto não supervisionado em seu NIDS, eles se beneficiaram dos rótulos de classe na fase de treinamento para implantar um NIDS baseado em desvio empregando uma estrutura discriminativa (em vez de limite) que associa o rótulo de baixo erro de reconstrução a uma amostra de entrada. Outra força importante em seu estudo é que o ID-CVAE executa apenas uma única etapa de treinamento para gerar apenas um modelo de vários treinamentos, dependendo do número de rótulos diferentes, como no VAE. Essa característica torna o ID-CVAE uma opção adequada para sistemas IoT devido à eficiência no tempo de computação, flexibilidade e precisão dos resultados. O conjunto de dados selecionado para treinamento e teste do ID-CVAE foi uma versão refinada do NSL-KDD. Terminou com 116 recursos e 23 rótulos possíveis. Eles provaram experimentalmente que seu trabalho é menos complexo em comparação com outros NIDS não supervisionados, com melhor precisão de classificação do que algoritmos conhecidos, como máquina de vetor de suporte linear e perceptron multicamada, etc. Os autores obtiveram uma precisão de 99%, 92% e 71% quando o modelo recupera características categóricas ausentes com, respectivamente, três, 11 e 70 valores.

Thing [149] analisou ameaças de rede IEEE 802.11 e propôs um IDS de rede de anomalias para detectar e classificar ataques em redes IEEE 802.11. Este trabalho é considerado como o primeiro trabalho que emprega algoritmos de aprendizado profundo para o padrão IEEE 802.11. A Thing experimentou a arquitetura Stacked Auto-encoder (SAE) com duas e três camadas ocultas.
O autor experimentou diferentes funções de ativação para os neurônios ocultos. Para testar sua estratégia, ele usou um conjunto de dados gerado a partir de uma infraestrutura de Small Office Home Office (SOHO) emulada em laboratório. Ele alcançou uma precisão geral de 98,66% em uma classificação de 4 classes (tráfego legítimo, ataques do tipo flooding, ataques do tipo injeção e ataques de personificação).

Diro et ai. [150] recomendaram o uso de fog computing em sistemas IoT para detectar invasões. Fog computing é equipar a camada de fog (hubs, roteadores ou gateways) com um processamento inteligente de dados em um nível intermediário com o objetivo de melhorar a eficiência e reduzir os dados transportados para a nuvem. Tal tecnologia permite detecção de ataques distribuídos mais eficiente em termos de escalabilidade, autonomia na detecção de ataques locais, aceleração no treinamento de dados próximos às fontes e compartilhamento de parâmetros dos vizinhos.

Os autores propuseram uma abordagem de aprendizado profundo para detectar ataques de intrusão conhecidos e não vistos. Os ataques conhecidos representam 99% o que leva a afirmar que os ataques de dia zero são elaborados com pequenas mutações nos antigos. Portanto, as redes profundas multicamadas aprimoram a percepção de pequenas mudanças (em um algoritmo autodidata com recursos de compactação) em comparação com os classificadores de aprendizado superficial. A abordagem de aprendizado profundo distribuído é baseada na distribuição do conjunto de dados para treinar cada subconjunto de dados localmente e rapidamente do que compartilhar e coordenar os parâmetros de aprendizado com os vizinhos. Então a arquitetura

IDS mestre que atualiza os valores dos parâmetros dos IDSs distribuídos abaixo e mantém a sincronização. Os estudos mostram que a abordagem de aprendizado profundo paralelo distribuído obtém melhores resultados em precisão do que o NIDS de aprendizado profundo centralizado e também do que os algoritmos de aprendizado de máquina superficial. Para treinar os modelos e avaliar o IDS, Diro et al. usou o conjunto de dados NSL-KDD depois de adicionar algumas modificações para terminar com 123 recursos de entrada e 1 rótulo. Como resultados, eles obtiveram detecção multiclasse composta por 4 rótulos (normal, DoS, Probe, R2L.U2R) para atingir 96,5% de taxa de detecção e 2,57% de falsos alarmes para o modelo profundo em comparação com o classificador raso, atingindo 93,66% de detecção e 4,97 % taxa de detecção falsa. Eles também notaram um aumento na precisão geral da detecção ao adicionar o número de nós de névoa de 96% para 99%.
A abordagem proposta levou mais tempo de treinamento; no entanto, a detecção real foi rápida e precisa.

Prabavathy et ai. [151] propuseram uma nova técnica de detecção de intrusão baseada em computação de neblina usando a Máquina de Aprendizado Extremo Sequencial Online (OS-ELM). O mecanismo de segurança distribuído (garantido pela ideia de computação em névoa) respeita os aspectos de interoperabilidade, flexibilidade, escalabilidade e heterogeneidade dos sistemas IoT. O sistema proposto é composto por duas partes principais:

1) Detecção de ataque em nós de névoa: Prabavathy et al. use o algoritmo OS ELM para detectar intrusões em nós de névoa. A rede IoT é dividida em clusters virtuais onde cada cluster corresponde a um grupo de dispositivos IoT sob um único nó de névoa. O OS-ELM classifica os pacotes recebidos como normais ou um ataque. ELM é uma rede neural feed-forward de camada oculta única caracterizada por sua fase de aprendizado rápido. Os pesos da camada de entrada e os valores de viés da camada oculta são selecionados aleatoriamente para deduzir analiticamente os pesos de saída usando cálculos de matrizes simples. No entanto, a natureza online do OS-ELM favorece uma detecção de streaming de ataques IoT.
2) Resumo no servidor de nuvem: para se ter uma ideia geral sobre o estado de segurança global do sistema IoT, as invasões detectadas são enviadas do nó de névoa para o servidor de nuvem. Após a análise e visualização do estado atual, Prabavathy et al. propor duas ações; i) prever a próxima ação do atacante usando a abordagem de reconhecimento do plano do atacante; ou ii) identificar vários estágios baseados na posição geográfica do nó de névoa e ataques DDoS. Portanto, uma resposta de intrusão pode ser ativada.

Prabavathy et ai. Propôs uma prova de conceito para avaliar sua proposição no processador DUALCORE, 1 GB RAM e 200 GB HDD como nós de névoa. Os autores implantaram o serviço de nuvem Azure (4 X Dual-Core AMD Opteron 2218 @2,6 GHz, 8 núcleos, 32 GB de RAM, 6146 GB de HDD) para configuração experimental.
Eles implementaram o OS-ELM usando MATLAB e NSL-KDD como conjunto de dados de referência. Os autores alegaram alta precisão e tempo de resposta. Eles atingem 97,36% de precisão com taxa de alarme falso reduzida de 0,37%. A taxa de detecção com a estratégia de nó de névoa foi 25% mais rápida quando comparada com a implementação baseada em nuvem. Uma vantagem importante é que novos dados online podem ser incorporados ao processo de aprendizagem, o que não é o caso de ANN e NB.

Rathore et Park [16] implantaram um novo detector de nevoeiro usando NIDS Semi-supervisionado Fuzzy C-Means (ESFCM) baseado em ELM. Esse IDS distribuído lida com detecção de IoT geograficamente distribuída e de baixa latência para recursos e rede limitados por meio de computação em névoa. O ML supervisionado não detecta ataques desconhecidos, apesar de sua boa precisão. O ML não supervisionado tem menor precisão, mas tem a capacidade de detectar ataques desconhecidos/dia zero. Portanto, Rathore et Park propuseram uma abordagem semi-supervisionada usando o ML supervisionado e não supervisionado para entradas rotuladas e não rotuladas. Para o aprendizado não supervisionado, Fuzzy C-Means (FCM) foi o algoritmo escolhido (um dos amplamente utilizados em clustering). O FCM seleciona dados não rotulados e atribui cada entrada a um ou mais clusters com vários graus de associação. Enquanto a parte supervisionada implanta o Extreme Learning Machine (ELM) para detecção eficaz e eficiente. Assim, os autores propuseram uma classificação ESFCM onde Semi-supervisioned Fuzzy C-Means (SFCM) funciona com o classificador ELM para uma detecção mais rápida de ataques conhecidos e desconhecidos. O IDS começa gerando um modelo (M) depois de treinar o classificador ELM no conjunto de dados rotulado. Então, o algoritmo SFCM aprende com os dados rotulados e não rotulados para atribuir um grau de pertinência às entradas não rotuladas. As instâncias não rotuladas que têm uma melhor oportunidade de pertencer a uma classe são classificadas usando o modelo treinado M e adicionadas aos dados rotulados de acordo com o limite definido. Além disso, os dados não rotulados restantes são agrupados novamente com SFCM e treinados novamente com ELM até que todas as instâncias sejam atribuídas. Finalmente, um modelo treinado é gerado para dados rotulados e não rotulados.

Dois tipos de avaliação para o algoritmo proposto foram estabelecidos usando o conjunto de dados NSL-KDD após escalonamento e pré-processamento; i) uma comparação entre a solução distribuída dos autores e um framework centralizado baseado em nuvem; e ii) a eficácia do ESFCM foi comparada com métodos tradicionais de aprendizado de máquina em termos de medidas padrão. Os resultados mostram melhor desempenho de 11 ms em termos de tempo de detecção e 86,53% de precisão.

Moustafa et ai. [19] propuseram uma rede ensemble na técnica de detecção de intrusão com base em recursos de fluxo estatístico estabelecidos para mitigar eventos maliciosos, particularmente ataques de botnet contra protocolos DNS, HTTP e MQTT usados em redes IoT. Sua solução pode ser dividida em:

1) Um conjunto de recursos é extraído dos protocolos de tráfego de rede MQTT, HTTP e DNS por meio de uma análise profunda do modelo TCP/IP. Os autores empregam a ferramenta Bro IDS para os recursos básicos e desenvolvem um novo módulo extrator (que funciona simultaneamente com o Bro-IDS) para gerar recursos estatísticos adicionais dos fluxos transacionais.

2) Uma etapa de seleção de recursos onde o coeficiente de correlação é aplicado nos recursos do resultado para extrair os mais importantes. Esta etapa possibilita a redução do custo computacional do NIDS.
3) Um método de conjunto onde os dados da rede são distribuídos com o algoritmo AdaBoost. Em seguida, Árvore de Decisão (DT), Naive Bayes (NB) e Rede Neural Artificial

(ANN) Algoritmos ML são implantados para detectar ataques.
A escolha das técnicas de classificação é justificada pelo cálculo da medida de correntropia. O método AdaBoost (Adaptive Boosting) melhora o desempenho da detecção em comparação com algoritmos de aprendizado de máquina separados. Ele pode lidar com as pequenas diferenças dos vetores de recursos por meio do cálculo de uma função de erro.
A função de erro é atribuída a cada instância dos dados de entrada distribuídos para aprender e decidir quais alunos podem classificar corretamente cada instância.

Para extrair as melhores características e avaliar a técnica de ensemble proposta, Moustafa et al. usou os conjuntos de dados de rede de bots UNSW-NB15 e NIMS com dados de sensor IoT simulados.

Os resultados dos experimentos têm alta taxa de detecção (DR) e baixa taxa de falsos positivos (FPR) em comparação com as técnicas de ponta existentes. A estratégia de conjunto alcançou entre 95,25% e 99,86% DR e 0,01% a 0,72% FPR.

C. Comparação e Discussão

Conforme apresentado na seção anterior, muitos pesquisadores dão um interesse especial para NIDS alimentados por IoT por meio de algoritmos de aprendizado de máquina. Uma comparação entre as propostas detalhadas anteriormente é ilustrada na Tabela V, onde focamos principalmente na implantação do IDS, metodologia de detecção, conjunto de dados usado, ameaças tratadas e algoritmos de ML usados.

Ao revisar [17], [141], [147], [148] e [149], observamos uma falta de detalhes sobre as **implementações da arquitetura.**
As propostas acima concentram-se nos mecanismos de ML para detecção de intrusão sem discutir os projetos de arquitetura. Soluções de Hosseinpour et al. [18], Bostani e Sheikhan [37], [145], [146], Diro et al. [150], Prabavathy et al [151], Rathore e Park [16] e Moustafa et al. [19] implantou arquitetura distribuída para detecção de intrusão, mais adequada para as necessidades de IoT. Na verdade, os sistemas IoT são distribuídos, pois são compostos por nós geograficamente distribuídos.

Tal critério desempenha um papel importante na escolha do algoritmo de aprendizado de máquina. Dependendo de onde e como queremos implantar nosso NIDS, os pesquisadores devem prestar atenção e identificar o algoritmo para objetos inteligentes com recursos limitados. Portanto, treinar um algoritmo intensivo em um nó limitado pode não ser viável. No entanto, a tarefa intensiva pode ser transferida para a nuvem. A melhor estratégia é processar tarefas que consomem recursos na parte de nuvem/servidor e executar partes leves na borda da IoT. É o caso de NIDS baseados em fog computing, como em [18], [37], [146], [150] e [151]. As soluções propostas aproveitam a camada de nuvem para treinamento de modelo de aprendizado de máquina e usam nós de névoa para detecção de intrusão. A detecção de intrusão baseada em névoa permite a coordenação para uma melhor detecção de baixa latência (perto da fonte de dados). Reduz o consumo de largura de banda da rede, pois os dados parciais são enviados na nuvem. Apenas alguns detalhes são relatados à parte centralizada e com uso intensivo de fontes do sistema IoT para resumir e detectar ataques distribuídos. O conceito de névoa permite a detecção de ataques distribuídos autônomos e paralelos [150]. [16] e [18] afirmaram que suas propostas podem ser implantadas em sistemas IoT distribuídos. Autores concentrados mais sobre a distribuição de dados da rede de tráfego.

Com relação aos **conjuntos de dados,** informações recentes são necessárias para treinar e avaliar IoT NIDS. Propostas como [147], [148] e [150] são baseadas no conjunto de dados NSL-KDD; conjunto de dados não IoT. As propostas não suportam protocolos IoT como 6LowPAN, Zigbee, CoAP, nem arquitetura IoT e princípios como mobilidade e heterogeneidade. Infelizmente, **não existe nenhum conjunto de dados NIDS dedicado à IoT** que explique o uso do NSL-KDD. [17], [141] e [149] avaliam sua proposta com seus próprios dados. No entanto, [18], [37], [151], [16] e [19] usaram uma combinação de dados reais e sintéticos ou simularam o ambiente.

Além disso, a maioria das pesquisas estudadas são feitas para proteger os sistemas IoT de **tipos precisos de ataques,** principalmente DoS, U2R, R2L e Probe, pois são inspirados no conjunto de dados NSL KDD. Coisa é a única proposta que se concentra especialmente em ataques IEEE 802.11. No entanto, as soluções de aprendizado de máquina não supervisionadas e semissupervisionadas são avaliadas contra ataques específicos. No entanto, eles são capazes de detectar ataques desconhecidos, como em [18], [146], [148], [16].
O último ponto importante a ser discutido é sobre **os algoritmos usados em ML.** Hodo et al. [17] usam Multi-Layer Perceptron (MLP), uma parte da família de Redes Neurais Artificiais (ANN) para detecção off-line. A partir de RNA, avançamos para as soluções com deep machine learning (DL) propostas por [148], [150] e [149]. DL é implantado em uma rede neural multicamada.
Os bons resultados de detecção das propostas usando DL conforme representado na Fig. 9 são devidos a i) estabilidade de treinamento e generalização de DL; ii) sua capacidade de atingir uma alta taxa de precisão se houver dados e tempo suficientes [23]; iii) DL é um algoritmo de autoaprendizagem, o que significa que não necessita de engenharia manual de características [16]; iv) DL extrai recursos hierárquicos complexos e não lineares de dados de treinamento de alta dimensão [150]. Lopez-Martin et al. [148] usou o algoritmo de autoencoder variacional condicional (CVAE), que é um modelo generativo baseado em conceitos de autoencoder variacional (VAE). O CVAE depende de duas entradas: i) os recursos de intrusão e ii) os rótulos de classe de intrusão, em vez de usar apenas os recursos de intrusão como entrada como no VAE. CVAE é melhor em flexibilidade e desempenho. Os autores escolheram o CVAE por sua capacidade de reconstrução de recursos, sua capacidade de recuperar recursos ausentes de conjuntos de dados incompletos. Apesar de seu uso para um algoritmo DL não supervisionado, eles se beneficiam de dados rotulados na fase de treinamento para um NIDS baseado em desvio. Martin-Lopez et al. garantiu um bom tempo computacional com uma boa flexibilidade gerando um modelo a partir de vários treinamentos em apenas uma única etapa de treinamento. Eles provaram uma melhor precisão de 80% do que os algoritmos lineares SVM (75%), MLP (78%) e Random Forest (73%). Enquanto isso, Diro et al. usou o algoritmo DL multicamadas em uma estratégia distribuída que fornece melhores resultados em comparação com o DL centralizado (99% vs 96% de precisão). É verdade que a fase de treinamento leva mais tempo, mas a detecção em tempo real é mais rápida e precisa. Diro et ai. [150] escolheu DL multicamadas, pois é a forma mais prevalente de DL. Ele mostra estabilidade de treinamento com uma escalabilidade significativa no conceito de big data. Além disso, Diro et al. comparou DL a ML

TABELA V: Resumo do NIDS para IoT baseado em Técnicas de Aprendizagem

Referências	Metodologia de detecção	de implantação de IDS	usado conjunto de dados	Ameaças Tratadas	algoritmos de ML
Hodo et al. [17]	—	Baseado em anomalia	Simulação DoS / DDoS		Perceptron multicamada (MLP)
Nobakht et ai. [141]	—	Baseado em host de anomalia	Dispositivos IoT reais (Luzes matiz)	Acesso não autorizado	Regressão logística vs. SVM
Hosseinpour et al. [18]	Distribuído	baseado em anomalia	KDD99 e SSH bruto força de ISCX	ataque botnet	Sistema Imunológico Artificial (AIS)
Bostani e Sheikhan [37], [145], [146]	Centralizado / Distribuído (Grande Arquitetura de dados MapReduce)	Híbrido: Baseado em anomalia para a peça centralizada e baseado em especificação para a distribuída proprietária [37]	Parte simulador+ NSL-KDD em uso	Dolina / Seletiva Encaminhamento em 6LoWPAN e pode ser estendido para classificação Blackhole e Wormhole	Floresta de caminho ideal não supervisionado (OPF) em [37]
		Híbrido: baseado em anomalia para a parte centralizada e baseado indevido para a parte distribuída [145], [146]			Supervisionado e não supervisionado ideal Caminho da Floresta (OPF) em [145], [146]
Pajouh et al. [147]	—	baseado em anomalia	NSL-KDD	Ataques de baixa frequência (como U2R, R2L)	{Componente principal não supervisionado Análise (PCA) + Linear Supervisionado Análise discriminante (LDA)} para redução de recursos e {Naive Bayes (NB) + Versão do Fator de Certeza de K Vizinhos Mais Próximos (CF KNN)} para classificação
López Martin e outros. [148]	—	Baseado em anomalia	NSL-KDD	DoS / R2L / U2R / tentar	AutoEncoder Variacional Condicional (CVAE)
Coisa [149]	—	Baseado em anomalia	Conjunto de dados de ataques IEEE 802.11 gerados (inundação, injeção e representação de laboratório) SOHO		Codificador automático empilhado (SAE)
Diro et ai. [150]	distribuído	baseado em anomalia	NSL-KDD	DoS / R2L.U2R / Probe	Multi-Layer Deep Learning
Prabavathy et ai. [151]	distribuído	baseado em anomalia	Emulação + NSL-KDD	Sonda / R2L / U2R / Dois	Aprendizagem Sequencial Extrema Online Máquina (OS-ELM)
Rathore e Parque [16]	distribuído	baseado em anomalia	simulação +NSL kdd	Sonda / R2L / U2R / Dois	Fuzzy semi-supervisionado baseado em ELM C-Médias (ESFCM)
Moustafa et ai. [19]	distribuído	baseado em anomalia	UNSW NB15+ NIMS + simulação	ataque botnet	Método ensemble AdaBoost usando três técnicas de DT, NB e ANN

no contexto distribuído e provou que a precisão do modelo profundo é maior que a do modelo raso (aumento da precisão da detecção multiclasse de 96,75% para 98,27%) e a taxa de falsos alarmes é menor para DL (de 4,97% para ML para 2,57% com DL na detecção multiclasse). Em relação à versão DL usada do Thing [149], experimentei o Stacked Auto-Encoder (SAE) com duas e três camadas ocultas, mas não fornece nenhum argumento de escolha primária. SAE é uma rede neural construída empilhando várias camadas de codificadores automáticos esparsos. A saída de cada camada forma a entrada da camada seguinte. Suas camadas ocultas reduzem a dimensionalidade do recurso e produzem um novo conjunto de recursos [152]. Esses novos recursos são aprendidos em profundidades de cascata para melhorar a precisão. Os nós na entrada e na camada de saída do SAE são os mesmos [27]. A solução proposta obteve bons resultados de precisão (98,66%) em comparação com o J48 (uma implementação do DT).

O modelo de 2 camadas ocultas teve um melhor desempenho em relação ao Modelo de 3 camadas ocultas.

Tratava-se de propostas de DL. Outra estratégia no uso de algoritmos de ML que vem ganhando cada vez mais atenção é a combinação de diferentes algoritmos no mesmo sistema [147], [19], [16] e [145], [146]. Pajouh et al. [147] usou duas técnicas simples de ML que são Naive Bayes (NB) e redes K-mais próximas (KNN) para rótulos de classe mais exatos. NB é aplicado em primeiro lugar para identificar anomalias.

Em seguida, os comportamentos normais serão analisados com KNN para refinar as instâncias normais. NB assume a independência de todas as características de cada amostra no rótulo de classe dado. Ele tem a capacidade de medir boas semelhanças de instâncias raras com o objetivo de lidar com dados desbalanceados. KNN usa uma técnica de bucketing [153] para acelerar a tarefa de classificação. Por outro lado, Pajouh et al. redução de dimensão aplicada antes de executar a classificação. Para isso, eles implantaram a Análise Discriminante Linear (LDA) (uma análise supervisionada

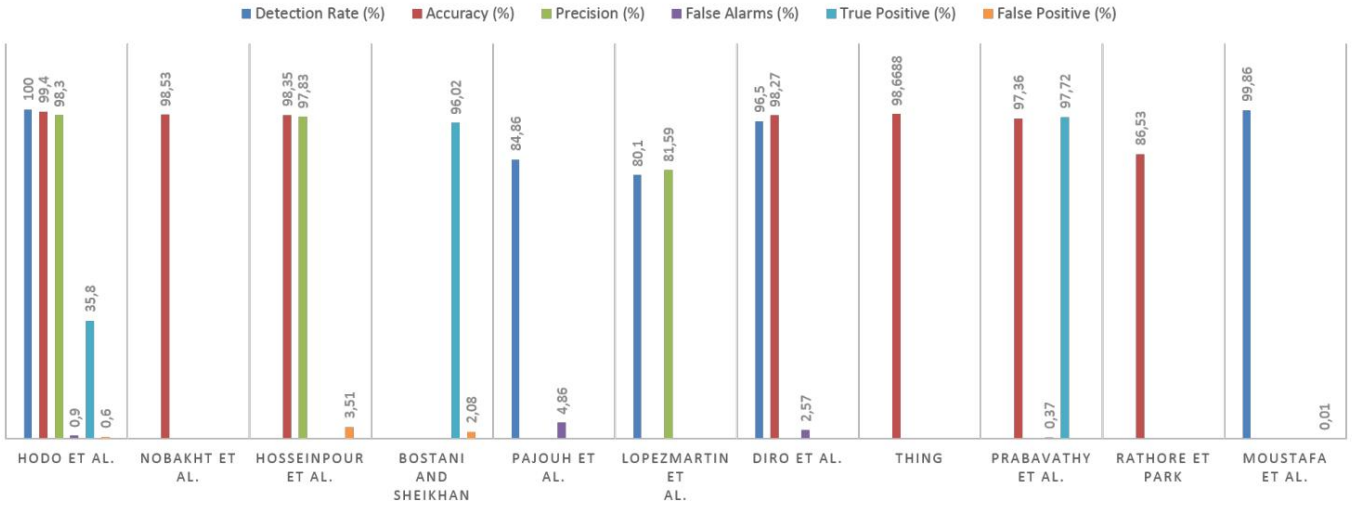

Fig. 9: Resultados de detecção de intrusão de última geração

técnica de redução de dimensão) e Análise de Componentes Principais (PCA) (uma técnica de redução de dimensão não supervisionada). O PCA fornece um espaço de recursos menor, gerando recursos não correlacionados a partir dos correlacionados iniciais. O LDA reduz a dimensão de grandes conjuntos de dados de trabalho examinando os rótulos de classe. Portanto, essas técnicas de redução de duas dimensões representam uma boa estratégia para i) reduzir as necessidades computacionais, o que é perfeito para sistemas IoT e ii) acelerar a detecção com menos erros, o que é perfeito para detecção de intrusão.

Pajouh et al. alcançou 84,86% de taxa de detecção em NSL-KDD, porém Moustafa et al. [19] conseguiu ter 99,86%, o que é um valor impressionante. A ideia deles é baseada em um método de aprendizado de conjunto AdaBoost que usa três técnicas de ML, ou seja, Árvore de Decisão (DT), Naive Bayes (NB) e Rede Neural Artificial (ANN). A combinação desses algoritmos é feita de forma paralela distribuída. Os dados são divididos em N conjuntos (de acordo com uma função de erro) e cada subconjunto de dados será tratado com um algoritmo escolhido para finalmente atualizar a distribuição. Essa lógica é garantida graças ao fluxo AdaBoost. Moustafa et ai. aplicados também seleção de recursos antes de iniciar a classificação. Ao analisar os ataques tratados usando correntropia, os autores notaram que existem pequenas variações entre vetores legítimos e suspeitos. Assim, os algoritmos de ML a serem utilizados devem classificar essas pequenas diferenças. Foi assim que DT, NB e ANN foram escolhidos.

DT [19] tem múltiplas vantagens ao classificar dados de rede. Ele seleciona recursos importantes, prepara pontos de dados de aprendizado com facilidade e manipula diretamente os valores dos recursos.
Mesmo que existam relações não lineares entre os parâmetros, o desempenho do DT não é afetado. NB [19] é conhecido por sua boa detecção de entradas anormais. Ele precisa de menos dados de treinamento e escala linearmente preditores e valores de recursos. É simples na otimização de parâmetros. ANN [19] tem muitos méritos. Exige treinamento estatístico menos formal e define correlações não lineares complexas entre variáveis dependentes e independentes. Além disso, permite a detecção de todas as possíveis interações entre preditores e variáveis. O terceiro tipo de combinação de algoritmos de ML é apresentado em [16]. Rathore e Park integrou um algoritmo Semi-supervisionado Fuzzy C-Means (SFCM) com o classificador Extreme Learning Machine (ELM) para compor o método Semi-Supervisioned Fuzzy C-Means (ESFCM) baseado em ELM. O SFCM é baseado no algoritmo Fuzzy C-Means (FCM) não supervisionado que agrupa os dados de entrada.
É uma das técnicas amplamente utilizadas no aprendizado não supervisionado. Ele captura estruturas de dados ocultas e visíveis.
No entanto, o algoritmo ELM [154] foi originalmente criado para treinar redes neurais feedforward (SLFNs) de camada oculta única.
O ELM é eficiente e dotado de capacidade de aprendizado rápido em ambientes altamente dinâmicos como sistemas IoT. Consequentemente, Rathore e Park alcançaram uma detecção mais rápida (11ms) com uma taxa de precisão melhor de 86,53% em comparação com o ML tradicional em sua estrutura com a vantagem de classificação de dados rotulados e não rotulados. Prabavathy et ai. [151] tiraram proveito do algoritmo ELM em sua proposta de detecção de intrusão. Eles operaram uma versão online do ELM (OS-ELM) para uma análise em tempo real. Em comparação com ANN e NB, os autores obtiveram melhor precisão (97,36%) com menor taxa de falsos positivos (0,37%) em um período de tempo menor (25% mais rápido). Uma grande vantagem do OS-ELM é que ele pode incorporar novos dados online para aprendizado, o que não é possível com os outros algoritmos comparados.

Sobre as soluções de Sheikhan e Bostani, eles usaram em seus trabalhos [37], [145] e [146] principalmente o algoritmo Optimum-Path Forest (OPF), que é um ML baseado em grafos eficiente. Eles usaram duas variantes de OPF; i) OPFC (OPF Clustering) que é um ML não supervisionado e ii) MOPF (Modified OPF) que é um algoritmo supervisionado. A força principal do OPF [155] é que ele não faz nenhuma suposição sobre a forma das classes.

Os autores usam OPFC para projetar modelos de agrupamento em uma arquitetura MapReduce. O MOPF é usado em um mecanismo de detecção baseado em uso indevido com um módulo de seleção de recursos. As soluções propostas são classificadores simples e rápidos, são independentes de parâmetros e originalmente suportam problemas multi-classe [155].

Além disso, Nobakht et al. [141] executou uma redução de características heuristicamente e experimentou dois algoritmos de ML; logística

Regressão (LR) e SVM para detecção de intrusão. LR é gradiente descendente que visa descobrir os parâmetros ótimos de um modelo LR. A precisão do modelo linear obtido com LR foi menos interessante do que o modelo não linear de SVM (96,2% para LR enquanto SVM atinge 100%).

Finalmente, Hosseinpour et al. [18] usou o Sistema Imunológico Artificial (AIS), que é um algoritmo de ML não supervisionado inspirado no sistema imunológico humano. É caracterizada por uma estrutura de proteção multicamadas. A primeira linha de defesa responde imediatamente a problemas vistos anteriormente, então uma proteção não específica para ataques desconhecidos é processada. Não precisa de conhecimento prévio de estranhos específicos. Outro ponto importante para o AIS é o aspecto da memória; O AIS é eficiente na detecção de ataques desconhecidos. Os autores alcançam 98,35% de acurácia e 97,83% de precisão, resultados notáveis conforme observado na Fig. 9. No entanto, o treinamento AIS necessita de recursos, razão pela qual Hosseinpour et al. prossiga-o na camada de nuvem.

Como visto anteriormente, muitos trabalhos de ML para detecção de intrusão em redes IoT foram desenvolvidos. Cada proposta de estado da arte tem seus argumentos, suas vantagens e suas desvantagens dependendo da arquitetura de IDS escolhida; se é centralizado ou distribuído ou os dois combinados. A estratégia de detecção em termos de usar apenas detecção de anomalias ou combiná-la com detecção baseada em assinatura também desempenha um papel. Além disso, cada estudo do pesquisador é baseado em um conjunto de dados, uma vez que as técnicas de ML são construídas no banco de dados. Os conjuntos de dados podem ser rotulados ou não, a partir de um sistema online ou de um repositório pré-existente. Além disso, diferentes algoritmos de ML foram experimentados; de estratégias supervisionadas para não supervisionadas, para estratégias semi-supervisionadas. Os algoritmos foram implantados de forma autônoma ou combinados. E mesmo a combinação é em paralelo ou em cascata. Muitas combinações são possíveis e cada uma dá resultados diferentes. Comparar os NIDS do estado da arte é difícil, pois cada um trata ataques especiais, em uma arquitetura especial, com conjunto de dados diferentes, usando vários algoritmos de ML em diferentes estratégias. Para ter uma representação gráfica da eficiência de detecção em IoT NIDS com base em ML, apresentamos um histograma na Fig. 9 onde tendemos a resumir os desempenhos e não realmente comparar os resultados um a um devido às diferenças nas estratégias implantadas. É verdade que o ambiente IoT é condicionado por muitos desafios apresentados na Seção II, por exemplo, restrições de recursos, portanto, a aplicação de algoritmos de ML em IoT nem sempre é óbvia. Por exemplo, o treinamento de modelos de ML pode ser uma tarefa computacionalmente intensiva para pequenos dispositivos IoT. Por um lado, treinar o modelo em um servidor ajudará com problemas de dispositivos de baixa potência. Por outro lado, esta solução exigirá a transferência de todos os dados coletados no dispositivo local para o servidor externo para processamento, o que nos coloca diante da conectividade limitada de dispositivos de baixa potência. No entanto, conforme mostrado na Seção V, as estratégias de ML superam esses problemas de IoT/ML e alcançam resultados interessantes em IoT NIDS. Consequentemente, ao implementar um NIDS para IoT, uma boa estratégia para superar IoT, ML e limites de segurança deve ser bem estudada. Processos intensivos devem ser executados em dispositivos poderosos, e o tipo oposto devem ser colocados em pequenos dispositivos. Os dados enviados entre as diferentes camadas da IoT (Seção II) devem ser bem escolhidos. Os dados precisam ser pré-processados para acelerar os resultados e aumentar a precisão das taxas de detecção enquanto diminuem os alarmes falsos. Tratamentos desnecessários devem ser removidos. A dependência/independência de recursos, bem como o desequilíbrio de dados, devem ser profundamente estudados e as características da IoT devem ser exploradas; por exemplo, o aspecto distribuído IoT. Conforme mostrado nesta seção, muitas soluções de ML são possíveis, os pesquisadores precisam apenas fazer as escolhas certas em termos de implantação, metodologia de detecção, conjuntos de dados usados e algoritmos de ML desenvolvidos. Na próxima seção, apontaremos possíveis direções futuras de pesquisa.

SERRA. INSTRUÇÕES DE PESQUISA FUTURA

Com a explosão da IoT, surgem dois novos paradigmas: **edge computing e fog computing.** Ambos tendem a empurrar o emprego de inteligência e lógica de processamento para perto de fontes de dados (o que significa o mais próximo possível de sensores e atuadores) para reduzir a largura de banda da rede necessária para comunicar dados da camada de percepção para data centers onde as análises geralmente são processadas . A principal diferença entre arquitetura de borda e arquitetura de névoa está no local onde o processamento inteligente e o poder de computação estão localizados.

A computação de borda os leva aos extremos da rede, como gateways e dispositivos de borda (por exemplo, PACs de controladores de automação programáveis). No entanto, a computação em névoa tende a colocá-los no nível da rede local da arquitetura de rede, o que significa hubs, roteadores ou gateways (nós de névoa). Esses dois conceitos devem ser profundamente explorados e explorados para a futura arquitetura IoT IDS. Eles permitem que o processo de detecção de intrusão seja distribuído. Consequentemente, essa estratégia deve permitir a detecção de intrusão com menos necessidade de recursos, o que é adequado para IoT. Como exemplo, Al-Turjman propõe em [156] uma abordagem de baixa substituição de cache baseada na lógica de computação em névoa. Ele retém o valor do cache dos nós sensores ativos no SDN por mais tempo e melhora a eficácia da rede. Na mesma linha, **Big Data** [133] é uma solução para sanar problemas relacionados ao grande volume de tráfego de rede gerado por redes IoT. Portanto, como trabalhos futuros na implantação da arquitetura IDS, computação de borda e névoa, bem como métodos de Big Data, devem ser profundamente explorados para IoT NIDS, prestando mais atenção na proteção dos próprios IDSs em caso de falha do sistema IoT.

Além disso, o IoT NIDS precisa de um **conjunto de dados dedicado à IoT do mundo real.** Um conjunto de dados dedicado comum do mundo real ajudaria com uma comparação real e eficiente entre as diferentes pesquisas. Um benchmark de conjunto de dados permite treinar, validar e avaliar estudos com diferentes algoritmos de ML.

Além disso, segundo Sommer e Paxson [14], os IDS baseados em técnicas de aprendizagem sofrem de “uma lacuna semântica entre os resultados e sua interpretação operacional”. Infelizmente, os IDS baseados em técnicas de aprendizado geralmente são avaliados com taxas como precisão, falso positivo e falso negativo. Acreditamos que apresentar apenas essas métricas não é suficiente. Os pesquisadores devem interpretar os resultados e entender a semântica do

escolha de características e o processo de detecção. A semântica também ajudaria a diferenciar entre comportamentos anormais e maliciosos. Portanto, **a relação semântica entre detecção e processo de aprendizagem** parece ser uma trilha interessante a ser explorada.

Além disso, **as escolhas de recursos ,** bem como **a reconstrução de recursos e a redução de dimensões de recursos** podem ser mais inspecionadas para IoT NIDS com base em técnicas de aprendizado. Essas técnicas podem ajudar a superar os desafios de restrição de recursos da IoT. **Técnicas de deep learning** usadas sozinhas ou combinadas também devem ser mais experientes, pois algoritmos como autoencoders são eficientes na reconstrução de recursos e redução de dimensão.

Além disso, técnicas como **acelerador de software,** para baixa potência dos algoritmos de aprendizado em dispositivos minúsculos, podem ser experimentadas no ambiente de segurança IoT como em [157]. Nicholas D. Lane em al. projetou e implementou o DeepX, um acelerador de software para execução de deep learning que reduz significativamente os recursos do dispositivo (ou seja, memória, computação, energia) necessários para o deep learning. Pesquisadores de segurança podem se inspirar em trabalhos como Ravi et al. em [158] onde eles apresentaram uma abordagem de otimização para permitir o uso de aprendizado profundo em tempo real em dispositivos de baixa potência. Os autores usaram uma representação de espectrograma dos dados de entrada inerciais para fornecer invariância contra mudanças no posicionamento do sensor, amplitude ou taxa de amostragem, permitindo assim um projeto de método mais compacto.
Os mesmos autores propuseram em [159] uma combinação de recursos aprendidos superficiais de uma abordagem de aprendizado profundo para permitir a classificação precisa e em tempo real da atividade. Tal proposta supera algumas limitações para aprendizado profundo quando a computação no nó é necessária.

Por último, mas não menos importante, para o futuro, mais esforços precisam ser feitos para detectar **ataques desconhecidos e de dia zero** em redes IoT e desenvolver IDSs que possam atualizar automaticamente a lista dos ataques considerados quando novos aparecerem. IoT NIDS precisa ser experimentado com algoritmos de ML e estratégias de big data [160], [133] para atualizar seu modelo de treinamento em **tempo real,** em uma **detecção de streaming.** Por exemplo, o campo **Incremental ML** na detecção de intrusão deve ser experimentado. O aprendizado incremental trata de treinar novamente o modelo em dados vistos anteriormente e não vistos para construir novos modelos. Visa garantir a continuidade do processo de aprendizagem por meio da atualização regular do modelo com base apenas no novo lote de dados disponível.

Essa ideia se junta à abordagem de tornar os IDSs mais inteligentes e independentes do ser humano na tomada de decisões.

Por fim, a IoT está sendo implantada cada vez mais em sistemas industriais, operações militares, ambiente de saúde e muitas outras áreas sensíveis que são IoT centradas no ser humano com base cognitiva. Dados sensíveis e informações privadas são trocados entre os objetos que viajam em um contexto que coloca a vida das pessoas em risco, por um lado, e onde os comportamentos humanos afetam os sistemas IoT, por outro lado. Portanto, mais atenção de segurança precisa ser dada a esses sistemas IoT baseados em humanos.

VII. CONCLUSÃO

Coisas conectadas (IoTs) tornaram-se difundidas para todos os indivíduos. De fato, os benefícios da IoT fazem com que a vida humana evolua com as Coisas. IoT está em cidades inteligentes (por exemplo, estacionamento inteligente), ambiente inteligente (por exemplo, para poluição do ar), em medição inteligente (por exemplo, rede inteligente), em controle industrial (por exemplo, para diagnóstico automático de veículos), etc. Eles estão em todos os domínios, mesmo nos críticos, como militar, saúde e segurança de edifícios. Infelizmente, as indústrias estão focando em inovar e desenvolver produtos mais conectados sem verificar muito sua qualidade e segurança. Nesta fase, observamos que a IoT é uma arma de dois gumes. Este exército de dispositivos conectados pode ser hackeado e usado contra a humanidade. Um nó comprometido pode afetar toda a rede IoT. Um usuário mal-intencionado torna-se capaz de quebrar sistemas de automação residencial e assim roubá-los ou pode controlar veículos remotamente para atingir pessoas inocentes nas estradas.

Um dos mecanismos poderosos para garantir a segurança da rede IoT são os NIDSs. Eles ajudam a detectar invasões nos sistemas.
Para aumentar sua eficiência, eles estão sendo fornecidos com técnicas de aprendizagem.

Até onde sabemos, nossa pesquisa é a primeira proposta com discussão abrangente de NIDSs baseados em aprendizado para sistemas IoT. Neste artigo, o campo de segurança IoT foi apresentado com uma comparação entre pesquisas anteriores. Além disso, as ameaças de IoT e as técnicas de detecção sobre os mecanismos de defesa tradicionais foram classificadas. Em seguida, foi apresentada uma avaliação abrangente das ferramentas de implementação do NIDS; Começando com conjuntos de dados de rede gratuitos, para sniffers de rede gratuitos e de código aberto, para NIDS de código aberto que podem ser usados por pesquisadores e industriais para implementar e avaliar sua própria solução NIDS sofisticada. Além disso, foi dada uma visão geral sobre NIDS em sistemas IoT com foco em sua arquitetura, implantações, metodologias de detecção e ameaças tratadas. Os prós e contras de cada proposta são minuciosamente avaliados. Por último, mas não menos importante, continuamos com o aprendizado de NIDSs para o ecossistema de IoT, onde as terminologias de aprendizado foram introduzidas e o estado da arte do aprendizado de NIDS de IoT foi detalhado. Cada trabalho foi resumido separadamente; em seguida, as estratégias adotadas foram comparadas para chegar a pontos fortes e táticas ideais para NIDS de ML e não ML. O estado da arte apresenta resultados interessantes; até 99% de precisão de detecção e 0,01% de falso positivo.

Por fim, as principais propostas de IoT NIDS foram comparadas com foco em algoritmos de ML e as direções de pesquisa futuras foram detalhadas.

Nos próximos tempos, as soluções baseadas em IoT vão explodir. Acreditamos que uma das necessidades mais importantes a tratar é a melhoria da estratégia de validação; Mais especificamente, o desenvolvimento de um conjunto de dados de benchmark público para trocas de rede de sistemas IoT. Deve incluir diferentes protocolos de IoT com as diferentes ameaças de IoT. Este conjunto de dados permitiria uma comparação clara, prática e conveniente dos diferentes NIDS desenvolvidos. Além disso, também é importante se concentrar no desenvolvimento de IoT NIDS que detectam ataques conhecidos e desconhecidos sem depender de protocolo. Para concluir, uma combinação de abordagens de edge computing e fog computing pode ser cada vez mais explorada para arquiteturas IoT NIDS. Essas abordagens permitem a detecção de intrusão IoT com menor consumo de recursos, portanto, com relação aos desafios da IoT.

REFERÊNCIAS

[1] J. Rifkin, “The Zero Marginal Cost Society: The Internet of Things, the Collaborative Commons, and the Eclipse of Capitalism: Book”, abril de 2014.

[2] A. Grau, “A Internet das Coisas Seguras O que é Realmente Necessário para Proteger a Internet das Coisas? | Icon Labs”, março de 2014. [On-line]. Disponível: http://www.iconlabs.com/prod/internet-secure things-%E2%80%93-what-really-needed-secure-internet-things [3] UN IDC, Intel, “A Guide to the Infográfico da Internet das Coisas,” Fevereiro de 2015. [Online]. Disponível: https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html [4] J. Gubbi, R. Buyya, S. Marusic, e M. Palaniswami, “Internet das coisas (IoT): uma visão, elementos arquitetônicos e direções futuras,” Sistemas de computador de geração futura, vol. 29, não. 7, pág. 1645–1660, setembro de 2013. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[5] D. Singh, G. Tripathi e AJ Jara, “A survey of Internet-of-Things: Future vision, Architecture, Challenges and Services,” em 2014 IEEE World Forum on Internet of Things (WF-IoT), março de 2014, pág. 287–292.

[6] O. Vermesan e P. Friess, "Internet of Things Applications - From Research and Innovation to Market Deployment Book", River Publishers, junho de 2014. [Online]. Disponível: http://www.internet-of-things research.eu/pdf/IERC Cluster Book 2014 Ch.3 SRIA WEB.pdf [7] R. Mitchell e IR Chen, “A Survey of Intrusion Detection Techniques for Cyber- Sistemas físicos,” ACM Comput. Surv., vol. 46, nº. 4, pág. 55:1–55:29, março de 2014. [Online]. Disponível: http://doi.acm.org/10.1145/2542049

[8] E. Benkhelifa, T. Welsh e W. Hamouda, “A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Towards Universal and Resilient Systems,” IEEE Communications Surveys Tutorials, pp. 1–1, junho de 2018.

[9] S. Raza, L. Wallgren e T. Voigt, “SVELTE: Detecção de intrusão em tempo real na Internet das Coisas,” Ad Hoc Networks, vol. 11, não. 8, pág. 2661–2674, novembro de 2013. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S1570870513001005 [10] E. Bertino e N. Islam, “Botnets and Internet of Things Security,” Computadores, vol. 50, não. 2, pág. 76–79, fevereiro de 2017.

[11] JP Anderson, “Monitoramento e vigilância de ameaças à segurança do computador,” James P. Anderson Company, Relatório Técnico, 1980.

[12] P. Kasinathan, C. Pastrone, MA Spirito e M. Vinkovits, "Detecção de negação de serviço na Internet das coisas baseada em 6lowpan", na 9ª Conferência Internacional IEEE sobre Computação Móvel e Sem Fio, Redes e Comunicações, 2013, pp. pp. 600–607.

[13] ZM Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue e K. Mizutani, “Aprendizagem profunda de última geração: evolução da inteligência de máquina para o tráfego de rede inteligente de amanhã Control Systems,” IEEE Communications Surveys Tutorials, vol. 19, não. 4, pág. 2432-2455, 2017.

[14] R. Sommer e V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection”, em 2010 IEEE Symposium on Security and Privacy, maio de 2010, pp. 305–316.

[15] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis e R. Atkinson, "Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey", arXiv:1701.02145 [cs], janeiro de 2017, arXiv: 1701.02145. [On-line]. Disponível: http://arxiv.org/abs/1701.02145 [16] S. Rathore e JH Park, “Estrutura de detecção de ataques distribuídos baseada em aprendizagem semi-supervisionada para IoT,” Applied Soft Computing, vol. 72, pág. 79–89, novembro de 2018. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1568494618303508 [17] E. Hodo, X. Bellekens, A. Hamilton, PL Dubouilh, E. Iorkyase, C. Tachtatzis e R. Atkinson, “Análise de ameaças de redes IoT usando sistema de detecção de intrusão de rede neural artificial”, em 2016 Simpósio Internacional de Redes, Computadores e Comunicações (ISNCC), maio de 2016, pp. 1–6.

[18] F. Hosseinpour, P. Vahdani Amoli, J. Plosila, T. Hmlinen e H. Tenhunen, “An Intrusion Detection System for Fog Computing and IoT based Logistic Systems using a Smart Data Approach,” Jornal Internacional de Tecnologia de Conteúdo Digital e suas Aplicações, vol. 10, dez. 2016. [Online]. Disponível: https: //jyx.jyu.fi/handle/123456789/54088

[19] N. Moustafa, B. Turnbull e KR Choo, “An Ensemble Intrusion Detection Technique based onposed Statistical Flow Features for Protecting Network Traffic of Internet of Things,” IEEE Internet of Things Journal, pp. 1–1, setembro de 2018.

[20] F. Chen, P. Deng, J. Wan, D. Zhang, AV Vasilakos e X. Rong, “Data Mining for the Internet of Things: Literature Review and Challenges,” International Journal of Distributed Sensor Networks, vol . 11, não. 8, pág. 431047, agosto de 2015. [Online]. Disponível: https://doi.org/10.1155/2015/431047

[21] CW Tsai, CF Lai, MC Chiang e LT Yang, “Data Mining for the Internet of Things: A Survey,” IEEE Communications Surveys Tutorials, vol. 16, não. 1, pág. 77–97, janeiro de 2014.

[22] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu e J. Qin, “A survey on application of machine learning for the Internet of Things,” International Journal of Machine Learning and Cybernetics , vol. 9, não. 8, pág. 1399–1417, agosto de 2018. [Online]. Disponível: https://doi.org/10.1007/s13042-018-0834-5 [23] MS Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi e AP Sheth, “Machine learning for internet de análise de dados de coisas: uma pesquisa,” Digital Communications and Networks, vol. 4, não. 3, pág. 161–175, agosto de 2018. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S235286481730247X [24] S. Agrawal e J. Agrawal, “Survey on Anomaly Detection using Data Mining Techniques,” Procedia Computer Science, vol. 60, pág. 708–713, janeiro de 2015. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S1877050915023479 [25] AL Buczak e E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” IEEE Communications Tutoriais de pesquisas , vol. 18, não. 2, pág. 1153-1176, 2016.

[26] L. Wang e R. Jones, “Análise de Big Data para Detecção de Intrusão de Rede: Uma Pesquisa,” International Journal of Networks and Communications, vol. 7, não. 1, pág. 24–31, 2017.

[27] P. Mishra, V. Varadharajan, U. Tupakula e ES Pilli, “A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection,” IEEE Communications Surveys Tutorials, pp. 1–1, junho de 2018.

[28] BB Zarpelo, RS Miani, CT Kawakani e SC de Alvarenga, “Uma pesquisa de detecção de intrusão na Internet das Coisas”, Journal of Network and Computer Applications, vol. 84, pág. 25–37, abril de 2017. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804517300802

[29] S. Krushang e H. Upadhyay, “Uma Pesquisa: Ataque DDOS na Internet das Coisas,” International Journal of Engineering Research and Development, vol. Volume 10, nº. Edição 11, pág. 58–63, novembro de 2014. [On-line]. Disponível: www.ijerd.com

[30] IB Ida, A. Jemai e A. Loukil, “A survey on security of IoT in the context of eHealth and clouds,” em 2016 11º Simpósio Internacional de Testes de Design (IDT), Hammamet, Tunísia, dezembro de 2016, pp. 25–30.

[31] S. Babar, A. Stango, N. Prasad, J. Sen e R. Prasad, "Estrutura de segurança incorporada proposta para Internet das Coisas (IoT)", em 2011, 2ª Conferência Internacional sobre Comunicação Sem Fio, Tecnologia Veicular, Teoria da Informação e Tecnologia de Sistemas Eletrônicos Aeroespaciais (Wireless VITAE), fevereiro de 2011, pp. 1–5.

[32] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel e M. Rajarajan, "Uma pesquisa sobre técnicas de detecção de intrusão na nuvem", Journal of Network and Computer Applications, vol. 36, nº. 1, pág. 42–57, janeiro de 2013. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804512001178 [33] P. Mishra, ES Pilli, V. Varadharajan e U. Tupakula, “Técnicas de detecção de intrusão em ambiente de nuvem: uma pesquisa,” Journal of Network and Computer Applications, vol. 77, pág. 18–47, janeiro de 2017. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804516302417

[34] AW Atamli e A. Martin, “Análise de Segurança Baseada em Ameaças para a Internet das Coisas,” em 2014 International Workshop on Secure Internet of Things, setembro de 2014, pp. 35–43.

[35] M. Conti, N. Dragoni e V. Lesyk, "Uma pesquisa sobre ataques do homem no meio", IEEE Communications Surveys Tutorials, vol. 18, não. 3, pág. 2027–2051, 2016.

[36] C. Karlof e D. Wagner, "Roteamento seguro em redes de sensores sem fio: ataques e contramedidas", em Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003., maio de 2003, pp. 113–127.

[37] H. Bostani e M. Sheikhan, “Híbrido de IDS baseado em anomalia e baseado em especificação para Internet das Coisas usando OPF não supervisionado com base na abordagem MapReduce,” Computer Communications, vol. 98, nº. Suplemento C, p. 52–71, janeiro de 2017. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0140366416306387 [38] L. Wallgren, S. Raza e T. Voigt, “Routing Attacks and Countermeasures in the RPL-Based Internet of Things,” Internacional

Journal of Distributed Sensor Networks, vol. 9, não. 8, pág. 794326, agosto de 2013. [Online]. Disponível: https://doi.org/10.1155/2013/794326 [39] Microsoft, "The STRIDE Threat Model," 2005. [Online]. Disponível: https://msdn.microsoft.com/fr-fr/en-en/enus/library/ee823878(v=cs.20).aspx

[40] N. Arajo, R. d. Oliveira, E. Ferreira, AA Shinoda e B. Bhargava, "Identificando características importantes no conjunto de dados de detecção de intrusão KDD99 por seleção de recursos usando uma abordagem híbrida", em 2010 17ª Conferência Internacional de Telecomunicações, abril de 2010, pp. 552–558.

[41] E. Bou-Harb, M. Debbabi e C. Assi, "Cyber Scanning: A Comprehensive Survey," IEEE Communications Surveys Tutorials, vol. 16, não. 3, pág. 1496–1519, 2014.

[42] S. Anwar, Z. Inayat, MF Zolkipli, JM Zain, A. Gani, N.B. Anuar, MK Khan e V. Chang, "Ataques de canal lateral baseados em cache entre VMs e mecanismos de prevenção propostos: uma pesquisa," Journal of Network and Computer Applications, vol. 93, pág. 259–279, setembro de 2017. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804517302205

[43] C. Kolias, G. Kambourakis, A. Stavrou e S. Gritzalis, "Detecção de intrusão em redes 802.11: avaliação empírica de ameaças e um conjunto de dados público," IEEE Communications Surveys Tutorials, vol. 18, não. 1, pág. 184–208, 2016.

[44] V. Zlomisli, K. Fertalj e V. Sruk, "Ataques de negação de serviço, defesas e desafios de pesquisa," Cluster Computing, vol. 20, não. 1, pág. 661–671, março de 2017. [Online]. Disponível: https://link.springer.com/article/10.1007/s10586-017-0730-x

[45] B. Prabadevi e N. Jeyanthi, "Distributed Denial of Service Attacks and its Effects on Cloud Environment- A Survey", no Simpósio Internacional de Redes, Computadores e Comunicações de 2014, junho de 2014, pp. 1–5.

[46] Cisco, "Um guia da Cisco para defesa contra ataques distribuídos de negação de serviço", outubro de 2012. [Online]. Disponível: http://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html [47] K. Hengst, "DDoS through the Internet of Things Uma análise que determina o potencial poder de um ataque DDoS usando dispositivos IoT", Jul. 2016. [Online]. Disponível: http://referaat.cs.utwente.nl/ [48] C. Rossow, "Amplification hell: Revisiting network protocols for DDoS abuse," Network and Distributed System Security Symposium, fevereiro de 2014. [Online]. Disponível: https://dud.inf.tu-dresden.de/ystrufe/ rn lit/rossow14amplification.pdf [49] S. Liron, "Mirai: The IoT Bot that Took Down Krebs and Launched a Tbps Attack on OVH, " Out. 2016. [Online]. Disponível: https://f5.com/labs/articles/threat-intelligence/ddos/mirai-the-iot bot-that-took-down-krebs-and-launched-a-tbps-attack-on-ovh- 22422 [50] D. Holmes, "Qual é a solução para ataques DDoS de IoT?

| SecurityWeek.Com", outubro de 2016. [Online]. Disponível: http://www.securityweek.com/whats-fix-iot-ddos-attacks [51] S. Hettich e S. Bay, "KDD Cup 1999 Data - The UCI KDD Archive. Irvine, CA: Universidade da Califórnia, Departamento de Informação e Ciência da Computação." 1999. [On-line]. Disponível: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [52] MR Asghar, G. Dn, D. Miorandi e I. Chlamtac, "Smart Meter Data Privacy: A Survey," IEEE Communications Surveys Tutorials, vol. 19, não. 4, pág. 2820–2835, 2017.

[53] S. Game e C. Raut, "Protocolos para detecção de ataque de replicação de nó em rede de sensores sem fio", Journal of Computer Engineering, vol. 16, não. 1, pág. 01–11, janeiro de 2014. [On-line]. Disponível: http://www.iosrjournals.org/iosr-jce/papers/Vol16-issue1/Version-2/A016120111.pdf?id=8539 [54] L. Sujihelen, C. Jayakumar e CS Singh, "Detecting Ataques de replicação de nós em redes de sensores sem fio: pesquisa," Indian Journal of Science and Technology, vol. 8, não. 16, jul. 2015. [Online]. Disponível: http://www.indjst.org/index.php/indjst/article/view/54150 [55] W. Ben Jaballah, M. Conti, G. Fil, M. Mosbah e A. Zemmari, "Whac -A-Mole: Posicionamento de nó inteligente em ataque de clone em redes de sensores sem fio," Computer Communications, vol. 119, pág. 66–82, abril de 2018. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S0140366416307381 [56] Z. Zhang, MCY Cho, C. Wang, C. Hsu, C. Chen e S. Shieh, "IoT Security: Desafios contínuos e oportunidades de pesquisa", em 2014 IEEE 7ª Conferência Internacional sobre Computação e Aplicações Orientadas a Serviços, novembro de 2014, pp. 230–234.

[57] J. Granjal, E. Monteiro e JS Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," IEEE Communications Surveys Tutorials, vol. 17, não. 3, pág. 1294-1312, 2015.

[58] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari e M. Ayyash, "Internet das coisas: uma pesquisa sobre tecnologias facilitadoras, protocolos e aplicativos", Tutoriais de pesquisas de comunicações do IEEE, vol. 17, não. 4, pág. 2347–2376, 2015.

[59] F. Al-Turjman, "Estrutura de entrega de dados QoSaware para multimídia inspirada em segurança em IoT veicular integrada," Computer Communications, vol. 121, pág. 33–43, maio de 2018. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S0140366417306060 [60] SA Alabady, F. Al-Turjman e S. Din, "A Novel Security Model for Cooperative Virtual Networks in the IoT Era, " International Journal of Parallel Programming, julho de 2018. [Online]. Disponível: https://doi.org/10.1007/s10766-018-0580-z [61] SA Alabady e F. Al-Turjman, "Low Complexity Parity Check Code for Futuristic Wireless Networks Applications," IEEE Access, vol. 6, pág. 18 398–18 407, abril de 2018.

[62] S. Sicari, A. Rizzardi, LA Grieco e A. Coen-Porisini, "Segurança, privacidade e confiança na Internet das Coisas: O caminho à frente," Redes de Computadores, vol. 76, pág. 146–164, janeiro de 2015. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1389128614003971 [63] Y. Yang, L. Wu, G. Yin, L. Li e H. Zhao, "A Survey on Security and Privacy Questões na Internet das Coisas," IEEE Internet of Things Journal, vol. 4, não. 5, pág. 1250–1258, outubro de 2017.

[64] AK Sikder, H. Aksu e AS Uluagac, "6º sentido: um detector de ataque baseado em sensor sensível ao contexto para dispositivos inteligentes", 26º Simpósio de Segurança USENIX (Segurança USENIX 17), p. 19 de agosto de 2017.

[65] P. Faruki, V. Ganmoor, V. Laxmi, MS Gaur e A. Bharmal, "AndroSimilar: Robust Statistical Feature Signature for Android Malware Detection," em Proceedings of the 6th International Conference on Security of Information and Networks, ser. PECADO '13. Nova York, NY, EUA: ACM, novembro de 2013, p. 152–159. [On-line]. Disponível: http://doi.acm.org/10.1145/2523514.2523539 [66] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, MS Gaur, M. Conti e M. Rajarajan, "Android Security: Uma pesquisa de problemas, penetração de malware e defesas", IEEE Communications Surveys Tutorials, vol. 17, não. 2, pág. 998–1022, 2015.

[67] RE Crossler, F. Blanger e D. Ormond, "A busca pela segurança completa: uma análise empírica da proteção multicamada dos usuários contra ameaças à segurança", Information Systems Frontiers, abril de 2017. [Online]. Disponível: https://doi.org/10.1007/s10796-017-9755-1 [68] M. Tanase, "IP Spoofing: An Introduction | Symantec Connect Community", março de 2003. [On-line]. Disponível: https://www.symantec.com/connect/articles/ip-spoofing-introduction [69] F. Al-Turjman e S. Alturjman, "Confidential smart-sensing framework in the IoT era," The Journal of Supercomputing , vol. 74, nº. 10, pág. 5187–5198, outubro de 2018. [On-line]. Disponível: https://doi.org/10.1007/s11227-018-2524-1 [70] F. Al-Turjman, YK Ever, E. Ever, HX Nguyen e DB David, "Seamless Key Agreement Framework for Mobile- Mergulhe em redes de sensores de segurança pública seguras centradas na nuvem baseadas em IoT," IEEE Access, vol. 5, pág. 24 617–24 631, outubro de 2017.

[71] F. Al-Turjman e S. Alturjman, "Acesso sensível ao contexto em aplicações de saúde industriais da Internet das Coisas (IIoT)", IEEE Transactions on Industrial Informatics, vol. 14, não. 6, pág. 2736–2744, junho de 2018.

[72] S. Patil, P. Kulkarni, P. Rane e B. Meshram, "IDS vs. IPS", Jornal Internacional de Redes de Computadores e Comunicações Sem Fio, vol. V 2, não. Edição 1, 2012. [Online]. Disponível: http://www.ijcnwc.org/papers/vol2no12012/16vol2no1.pdf [73] M. Ahmed, A. Naser Mahmood e J. Hu, "Uma pesquisa de técnicas de detecção de anomalias de rede," Journal of Network and Aplicações de Computador, vol. 60, pág. 19–31, janeiro de 2016. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804515002891 [74] W. Haider, J. Hu, J. Slay, BP Turnbull e Y. Xie, "Generating realistic intrusion detection system dataset based sobre modelagem qualitativa difusa," Journal of Network and Computer Applications, vol. 87, pág. 185–192, junho de 2017. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804517301273 [75] P. Casas, J. Mazel e P. Owezarski, "Unsupervised Network Intrusion Detection Systems: Detecting the Unknown without Knowledge," Computer Comunicações, vol. 35, não. 7, pág. 772–783, abril de 2012. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0140366412000266

[76] S. Sharma e M. Dixit, "Uma revisão sobre o sistema de detecção de intrusão de rede usando o Snort de código aberto", International Journal of Database Theory and Application, vol. 9, não. 4, pág. 61–70, abril de 2016. [Online]. Disponível: http://www.earticle.net/article.aspx?sn=272711

[77] N. Moustafa e J. Slay, "UNSW-NB15: um conjunto de dados abrangente para sistemas de detecção de intrusão de rede (conjunto de dados de rede UNSW-NB15)", na Conferência Militar de Comunicações e Sistemas de Informação (MilCIS), novembro de 2015 , pp. 1–6.

[78] C. Xiang e SM Lim, "Design de classificador híbrido de nível múltiplo para sistema de detecção de intrusão", em 2005 IEEE Workshop sobre aprendizado de máquina para processamento de sinal, setembro de 2005, pp. 117–122.

[79] N. Chandolikar e V. Nandavadekar, “Seleção de Recurso Relevante para Classificação de Ataque de Intrusão analisando KDD Cup 99,” MIT International Journal of Computer Science & Information Technology, vol. 2, não. 2, pág. 85–90, agosto de 2012.

[80] N. Kayacik e M. Heywood, “Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets”, na 3ª Conferência Anual sobre Privacidade, Segurança e Confiança (PST), 2005.

[81] SK Sahu, S. Sarangi e SK Jena, "Uma análise detalhada sobre conjuntos de dados de detecção de intrusão", em 2014 IEEE International Advance Computing Conference (IACC), fevereiro de 2014, pp. 1348–1353.

[82] H. Nguyen, K. Franke e S. Petrovic, "Improving Effectiveness of Intrusion Detection by Correlation Feature Selection", na Conferência Internacional de 2010 sobre Disponibilidade, Confiabilidade e Segurança, fevereiro de 2010, pp. 17–24.

[83] AO Adetunmbi, SO Adeola e OA Daramola, "Análise do Conjunto de Dados de Detecção de Intrusão KDD 99 para Seleção de Recursos de Relevância", em Anais do Congresso Mundial de Engenharia e Ciência da Computação, vol. 1, São Francisco, EUA, outubro de 2010.

[84] P. Gogoi, MH Bhuyan, DK Bhattacharyya e JK Kalita, “Packet and Flow Based Network Intrusion Dataset,” in Contemporary Computing, ser. Comunicação em Ciência da Computação e da Informação. Springer, Berlim, Heidelberg, agosto de 2012, p. 322–334.
[On-line]. Disponível: https://link.springer.com/chapter/10.1007/978-3- 642-32129-0 34 [85] AR Vasudevan, E. Harshini e S. Selvakumar, “SSENet-2011: A Network Intrusion Detection Conjunto de dados do sistema e sua comparação com o conjunto de dados KDD CUP 99”, em 2011 Second Asian Himalayas International Conference on Internet (AH-ICI), novembro de 2011, pp. 1–5.

[86] MV Mahoney e PK Chan, “An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection,” in Recent Advances in Intrusion Detection, ser. Notas de aula em Ciência da Computação. Springer, Berlim, Heidelberg, setembro de 2003, p. 220–237. [On-line]. Disponível: https://link.springer.com/chapter/ 10.1007/978-3-540-45248-5 13

[87] “NSL-KDD | Conjuntos de dados | Pesquisa | Instituto Canadense de Segurança Cibernética | UNB”, 2016. [Online]. Disponível: http://www.unb.ca/cic/research/conjuntos de dados/nsl.html

[88] A. Sivanathan, D. Sherratt, HH Gharakheili, A. Radford, C. Wi jenayake, A. Vishwanath e V. Sivaraman, “Caracterizando e classificando o tráfego de IoT em cidades e campi inteligentes”, em 2017 IEEE Conference on Workshops de Comunicação por Computador (INFOCOM WKSHPS), maio de 2017, pp. 559–564.

[89] CI para Cibersegurança (CIC), “IDS 2017 | Conjuntos de dados | Pesquisa | Instituto Canadense de Segurança Cibernética | UNB”, 2017. [Online]. Disponível: https://www.unb.ca/cic/datasets/ids-2017.html [90] “CSE-CIC-IDS2018 | Conjuntos de dados | Pesquisa | Instituto Canadense de Segurança Cibernética | UNB”, 2018. [Online]. Disponível: https://www.unb.ca/cic/datasets/ids-2018.html [91] tcpdump, “Repositório público Tcpdump/Libpcap,” 2017. [Online]. Disponível: http://www.tcpdump.org

[92] A. Sivanathan, A. Hamza, H. Habibi e V. Sivaraman, "UNSW Proliferation Dataset". [On-line]. Disponível: https://iotanalytics.unsw.edu.au/index [93] A. Habibi Lashkari, G. Draper Gil, MSI Mamun e AA Ghorbani, “Characterization of Tor Traffic using Time based Features:,” in Proceedings of a 3ª Conferência Internacional sobre Segurança e Privacidade dos Sistemas de Informação. Porto, Portugal: SCITEPRESS - Publicações de Ciência e Tecnologia, 2017, pp. 253–262. [On-line]. Disponível: http://www.scitepress.org/DigitalLibrary/ Link.aspx? doi=10.5220/0006105602530262 [94] I. Sharafaldin, A. Habibi Lashkari e AA Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:,” in Proceedings of the 4th International Conference on Information Systems Security and Privacy. Funchal, Madeira, Portugal: SCITEPRESS - Publicações de Ciência e Tecnologia, 2018, p. 108–116. [On-line]. Disponível: http://www.scitepress.org/ DigitalLibrary/Link.aspx?doi=10.5220/0006639801080116 [95] A. Gharib, I. Sharafaldin, AH Lashkari e AA Ghorbani, “An Evaluation Framework for Intrusion Detection Dataset, em 2016 Conferência Internacional sobre Ciência da Informação e Segurança (ICISS), dezembro de 2016, p. 1–6.

[96] N. Hoque, MH Bhuyan, RC Baishya, DK Bhattacharyya e JK Kalita, “Ataques de rede: taxonomia, ferramentas e sistemas,” Journal of Network and Computer Applications, vol. 40, pág. 307–324, abril de 2014. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1084804513001756 [97] F. Fuentes e DC Kar, “Ethereal vs. Tcpdump: Um estudo comparativo sobre ferramentas de detecção de pacotes para fins educacionais”, J. Comput. Sci. Coll., vol. 20, não. 4, pág. 169–176, abril de 2005. [On-line]. Disponível: http://dl.acm.org/citation.cfm?id=1047846.1047873 [98] P. Asrodia e H. Patel, “Análise de várias ferramentas de detecção de pacotes para monitoramento e análise de rede,” International Journal of Electrical, Electronics e Engenharia da Computação, vol. 1, não. 1, pág. 55–58, 2012.

[99] “Página inicial do Ettercap.” [On-line]. Disponível: https://www.ettercap project.org/ [100] “ARGUS- Auditing Network Activity,” 2017. [Online]. Disponível: https://qosient.com/argus/ [101] “EtherApe, um monitor gráfico de rede.” [On-line]. Disponível: https://etherape.sourceforge.io/ [102] “Wireshark Go Deep.” [On-line]. Disponível: https://www.wireshark.org/ [103] J. Schreiber, “Ferramentas de detecção de intrusão de código aberto: uma visão geral rápida”, 2014. Jan. [On-line]. Disponível: https://www.alienvault.com/blogs/security-essentials/open source-intrusion-detection-tools-a-quick-overview [104] M. Roesch, “Snort Lightweight Intrusion Detection for Networks,” p. 11 de 1999.

[105] B. Cusack e M. Alqahtani, “Aquisição de evidências de sistemas de detecção de intrusão de rede”, na Conferência Australiana de Forense Digital, dezembro de 2013. [Online]. Disponível: http://ro.ecu.edu.au/adf/118

[106] K. Thongkanchorn, S. Ngamsuriyaroj e V. Visoottiviseth, "Estudos de avaliação de três sistemas de detecção de intrusão sob vários ataques e conjuntos de regras", em 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), outubro de 2013, pp. 1–4.

[107] "Kismet sem fio", 2018. [On-line]. Disponível: https://www.kismetwireless.net/index.shtml

[108] “OpenWIPS-ng,” 2012. [Online]. Disponível: http://openwips-ng.org/index.html

[109] “Security Onion”, 2018. [Online]. Disponível: https://securityonion.net/ [110] “Solução Sagan, SIEM Gerenciado, Análise de Log, Análise de Rede, Monitoramento, Alerta, MSSP | Segurança da Informação do Quadrante”, 2018. [On-line]. Disponível: https://quadrantsec.com/sagansolution/

[111] S. Team, “Snort - Network Intrusion Detection & Prevention System,” 2017. [On-line]. Disponível: https://www.snort.org/ [112] TOIS Foundation, “Suricata,” 2017. [Online]. Disponível: https://suricata-ids.org/ [113] P. Vern, “Bro: Um sistema para detecção de intrusos de rede em tempo real,” Redes de computadores, vol. 31, nº. 23-24, p. 2435-2463, 1999.
[On-line]. Disponível: http://www.icir.org/vern/papers/bro-CN99.pdf [114] C. Kolias, V. Kolias e G. Kambourakis, “TermID: uma abordagem baseada em inteligência de enxame distribuído para intrusão sem fio detecção,” Jornal Internacional de Segurança da Informação, vol. 16, não. 4, pág. 401–416, agosto de 2017. [On-line]. Disponível: https://doi.org/10.1007/s10207-016-0335-z

[115] Y. Zhang, W. Lee e YA Huang, "Técnicas de detecção de intrusão para redes sem fio móveis", Redes sem fio, vol. 9, não. 5, pág. 545–556, janeiro de 2003. [Online]. Disponível: http://dl.acm.org/citation.cfm?id=942545.942556 [116] A. Abduvaliyev, AK Pathan, J. Zhou, R. Roman e W. Wong, “On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks,” IEEE Communications Surveys Tutorials, vol. 15, não. 3, pág. 1223-1237, 2013.

[117] I. Butun, SD Morgera e R. Sankar, "Uma pesquisa de sistemas de detecção de intrusão em redes de sensores sem fio", Tutoriais de pesquisas de comunicações IEEE, vol. 16, não. 1, pág. 266–282, 2014.

[118] O. Can e OK Sahingoz, “A survey of intrusion detection systems in wireless sensor networks,” em 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), maio de 2015, pp. 1–6.

[119] A. Aris e SF Oktug, “Poster: State of the Art IDS Design for IoT,” em Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks, ser. EWSN 17. EUA: Junction Publishing, fevereiro de 2017, p. 196–197. [On-line]. Disponível: http://dl.acm.org/citation.cfm?id=3108009.3108037

Este artigo foi aceito para publicação em uma futura edição desta revista, mas não foi totalmente editado. O conteúdo pode mudar antes da publicação final. Informações de citação: DOI 10.1109/COMST.2019.2896380, IEEE Communications Surveys & Tutorials

[120] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone e MA Spirito, "DEMO: An IDS Framework for Internet of Things Empowered by 6lowpan," in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Segurança, seja. CCS '13. Nova York, NY, EUA: ACM, novembro de 2013, p. 1337–1340. [On-line]. Disponível: http://doi.acm.org/10.1145/2508859.2512494 [121] "Metasploit | Software de teste de penetração, segurança de teste de caneta", 2017. [On-line]. Disponível: https://www.metasploit.com/ [122] C. Jun e C. Chi, "Design of Complex Event-Processing IDS in Internet of Things," em 2014 Sexta Conferência Internacional sobre Tecnologia de Medição e Automação Mecatrônica, janeiro 2014, pág. 226–229.

[123] C. Cervantes, D. Poplade, M. Nogueira e A. Santos, "Detecção de ataques de sumidouros para apoiar o roteamento seguro em 6lowpan para Internet das Coisas", em 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), maio de 2015, p. 606–611.

[124] M. Surendar e A. Umamakeswari, "InDRes: An Intrusion Detection and response system for Internet of Things with 6lowpan", na Conferência Internacional de 2016 sobre Comunicações Sem Fio, Processamento de Sinal e Rede (WiSPNET), março de 2016, pp. . 1903–1908.

[125] Y. Fu, Z. Yan, J. Cao, O. Kon e X. Cao, "Um método de detecção de intrusão baseado em autômatos para Internet das Coisas", maio de 2017. [Online]. Disponível: https://www.hindawi.com/journals/misy/2017/1750637/abs/

[126] Y. Fu e O. Kon, "Segurança e Robustness by Protocol Testing," IEEE Systems Journal, vol. 8, não. 3, pág. 699–707, setembro de 2014.

[127] P. Tsankov, MT Dashti, e D. Basin, "SecFuzz: Fuzz-testing Security Protocols," em Proceedings of the 7th International Workshop on Automation of Software Test, ser. AST '12. Piscataway, NJ, EUA: IEEE Press, junho de 2012, pp. 1–7. [On-line]. Disponível: http://dl.acm.org/citation.cfm?id=2663608.2663610 [128] B. Lei, X. Li, Z. Liu, C. Morisset e V. Stolz, "Teste de robustez para componentes de software, "Ciência da Programação de Computadores, vol. 75, nº. 10, pág. 879–897, outubro de 2010. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S0167642310000328 [129] D. Midi, A. Rullo, A. Mudgerikar e E. Bertino, "Kalis A System for Knowledge Driven Adaptive Intrusion Detection for the Internet of Things," em 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), junho de 2017, pp. 656-666.

[130] G. Kumar, "Métricas de avaliação para sistemas de detecção de intrusão - um estudo", no. 11, pág. 7 de novembro de 2014.

[131] G. Francia, L. Ertaul, LH Encinas, E. El-Sheikh e K. Daimi, Computer and Network Security Essentials. Springer Publishing Company, Incorporated, 2018.

[132] VCloudNews, "Every Day Big Data Statistics 2,5 quintilhões de bytes de dados criados diariamente (http://www.vcloudnews.com/every-day-big data-statistics-2-5-quintillion-bytes-of-data- criado-diariamente/)", abril de 2015. [On-line]. Disponível: http://www.vcloudnews.com/every-day-big-data statistics-2-5-quintillion-bytes-of-data-created-daily/ [133] E. Ahmed, I. Yaqoob, IAT Hashem , I. Khan, AIA Ahmed, M. Imran e AV Vasilakos, "O papel da análise de big data na Internet das Coisas," Redes de Computadores, vol. 129, pág. 459–471, dezembro de 2017. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S1389128617302591 [134] J.

F. Puget, "O que é aprendizado de máquina? (o segredo mais bem guardado de TI é a otimização)", maio de 2016. [On-line]. Disponível: https://www.ibm.com/developerworks/community/blogs/jfp/entry/What is Machine Learning

[135] MA Ambusaidi, X. He, P. Nanda e Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," Transações IEEE em Computadores, vol. 65, não. 10, pág. 2986–2998, outubro de 2016.

[136] KK Gupta, B. Nath e R. Kotagiri, "Abordagem em camadas usando campos aleatórios condicionais para detecção de intrusão", IEEE Transactions on Dependable and Secure Computing, vol. 7, não. 1, pág. 35–49, janeiro de 2010.

[137] D. Silver, A. Huang, CJ Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel e D. Hassabis, "Dominando o jogo de Go com redes neurais profundas e pesquisa em árvore," Natureza, vol. 529, nº. 7587, pág. 484–489, janeiro de 2016. [On-line]. Disponível: https://www.nature.com/articles/nature16961 [138] L. Deng, D. Li, X. Yao, D. Cox e H. Wang, "Detecção de intrusão de rede móvel para sistema IoT baseado em transferência algoritmo de aprendizado", Cluster Computing, janeiro de 2018. [Online]. Disponível: https://doi.org/10.1007/s10586-018-1847-2 [139] W. Feng, Q. Zhang, G. Hu e JX Huang, "Mineração de dados de rede para detecção de intrusão através da combinação de SVMs com formigas colônia redes," Future Generation Computer Systems, vol. 37, pág. 127–140, julho de 2014. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0167739X13001416

[140] WC Lin, SW Ke e CF Tsai, "CANN: Um sistema de detecção de intrusão baseado na combinação de centros de cluster e vizinhos mais próximos," Knowledge-Based Systems, vol. 78, pág. 13–21, abril de 2015. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S0950705115000167

[141] M. Nobakht, V. Sivaraman e R. Boreli, "A Host-Based Intrusion Detection and Mitigation Framework for Smart Home IoT Using OpenFlow," em 2016 11ª Conferência Internacional sobre Disponibilidade, Confiabilidade e Segurança (ARES), agosto 2016, pág. 147–156.

[142] F. Hosseinpour, A. Meulenberg, S. Ramadass, PV Vahdani Amoli e Z. Moghaddasi, "Modelo baseado em agente distribuído para sistema de detecção de intrusão baseado em sistema imunológico artificial", JDCTA Int. J.Digit. Tecnologia de Conteúdo. seu Appl, vol. 7, pág. 206–214, maio de 2013.

[143] F. Hosseinpour, PV Amoli, F. Farahnakian e J. Plosila, "Detecção de intrusão baseada no sistema imunológico artificial: imunidade inata usando uma abordagem de aprendizagem não supervisionada", JDCTA Int. J. Digit. Tecnologia de Conteúdo. seu Appl., vol. 8, não. 5, pág. 1–12, outubro de 2014.

[144] A. Shiravi, H. Shiravi, M. Tavallaee e AA Ghorbani, "Para o desenvolvimento de uma abordagem sistemática para gerar conjuntos de dados de referência para detecção de intrusão," Computers & Security, vol. 31, nº. 3, pág. 357–374, maio de 2012. [On-line]. Disponível: http://www.sciencedirect.com/science/article/pii/S0167404811001672

[145] M. Sheikhan e H. Bostani, "Uma arquitetura de detecção de intrusão híbrida para a Internet das coisas", em 2016 8º Simpósio Internacional de Telecomunicações (IST), setembro de 2016, pp. 601–606. [146] ——, "Um mecanismo de segurança para detecção de intrusões na Internet das coisas usando recursos selecionados com base no MABOSA," Jornal Internacional de Pesquisa em Tecnologia da Informação e comunicação, vol. 9, não. 2, pág. 53–62, outubro de 2017. [On-line]. Disponível: http://journal.itrc.ac.ir/index.php/ijictr/article/view/261 [147] HH Pajouh, R. Javidan, R. Khayami, D. Ali e KKR Choo, "A Two Redução de dimensão de camada e modelo de classificação de duas camadas para detecção de intrusão baseada em anomalia em redes de backbone IoT,"

IEEE Transactions on Emerging Topics in Computing, vol. PP, não. 99, pág. 1–1, novembro de 2016.

[148] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas e J. Lloret, "Autoencoder Variacional Condicional para Predição e Recuperação de Recursos Aplicados à Detecção de Intrusão em IoT," Sensors, vol. 17, não. 9, pág. 1967, agosto de 2017. [Online]. Disponível: http://www.mdpi.com/1424-8220/17/9/1967

[149] VLL Thing, "IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach," em 2017 IEEE Wireless Communications and Networking Conference (WCNC), março de 2017, pp. 1–6.

[150] AA Diro e N. Chilamkurti, "Esquema de detecção de ataque distribuído usando abordagem de aprendizagem profunda para Internet das Coisas," Future Generation Computer Systems, setembro de 2017. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0167739X17308488 [151] S. Prabavathy, K. Sundarakantham e SM Shalinie, "Design of cognitiva fog computing for intrusion detection in Internet of Things," Jornal de Comunicações e Redes, vol. 20, não. 3, pág. 291–298, junho de 2018.

[152] ME Aminantoa e K. Kimb, "Deep Learning in Intrusion Detection System: An Overview," na Conferência Internacional de Pesquisa em Engenharia e Tecnologia (2016 IRCET). Fórum do Ensino Superior, 2016., 2016. [Online]. Disponível: /paper/Deep-Learning-in-Intrusion-Detection-System-%3A-An Aminantoa-Kimb/c0fa578c1fae002e02834806a576d811002cb4a4 [153] JH Friedman, JL Bentley e RA Finkel, "An Algorithm for Finding Best Expected Matches in Logarithmic Tempo", ACM Trans. Matemática. Softw., vol. 3) Não. 3, pág. 209–226, setembro de 1977. [Online].

Disponível: http://doi.acm.org/10.1145/355744.355745 [154] GB Huang, QY Zhu e CK Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, não. 1, pág. 489–501, dezembro de 2006. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0925231206000385

[155] JP Papa e AX Falco, "Um Algoritmo de Aprendizagem para o Classificador de Floresta de Caminho Ideal", em Representações Baseadas em Gráficos em Reconhecimento de Padrões, ser. Notas de Palestra em Ciência da Computação, A. Torsello, F. Escolano e L. Brun, Eds. Springer Berlin Heidelberg, maio de 2009, p. 195–204.

[156] F. Al-Turjman, "Cache baseado em neblina em redes centradas em informações definidas por software", Computers & Electrical Engineering, vol. 69, pág.

54–67, julho de 2018. [Online]. Disponível: http://www.sciencedirect.com/science/article/pii/S0045790618311856

[157] ND Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro e F. Kawsar, "DeepX: A Software Accelerator for Low-power Deep Learning Inference on Mobile Devices", em Proceedings of the 15th International Conference on Information Processing in Sensor Networks, ser. IPSN '16. Piscataway, NJ, EUA: IEEE Press, abril de 2016, p. 23:1–23:12. [On-line]. Disponível: http://dl.acm.org/citation.cfm?id=2959355.2959378 [158] D. Ravi, C. Wong, B. Lo e G. Yang, "Aprendizado profundo para o reconhecimento da atividade humana: um recurso eficiente implementação em dispositivos de baixa potência", em 2016 IEEE 13ª Conferência Internacional sobre Redes de Sensores Corporais vestíveis e implantáveis (BSN), junho de 2016, pp. 71–76.

[159] D. Ravi, C. Wong, B. Lo e G.-Z. Yang, "Uma abordagem de aprendizado profundo para análise de dados de sensores no nó para dispositivos móveis ou vestíveis", IEEE Journal of Biomedical and Health Informatics, vol. 21, pág. 56–64, janeiro de 2017. [On-line]. Disponível: http://doi.org/10.1109/JBHI.2016.2633287 [160] H. Hromic, DL Phuoc, M. Serrano, A. Antoni, IP arko, C. Hayes e S. Decker, "Real time analysis of dados do sensor para a Internet das Coisas por meio de agrupamento e processamento de eventos", em 2015 IEEE International Conference on Communications (ICC), junho de 2015, pp. 685–691.

Nadia Chaabouni concluiu o mestrado em ciência da computação pela Universidade de Bordeaux, em 2016. Atualmente, ela está cursando o doutorado. na Universidade de Bordeaux e no AtoS Innovation Aquitaine Lab sob a supervisão de Mohamed Mosbah e Akka Zemmari da University of Bordeaux, França, e Cyrille Sauvignac do Atos innovation Aquitaine Lab, França. Seus interesses de pesquisa incluem a área de Segurança de Sistemas de Internet das Coisas e técnicas de aprendizado de máquina.

Mohamed Mosbah é professor titular de ciência da computação no Instituto Politécnico de Bordeaux, na França. Ele obteve seu Ph.D. pela Universidade de Bordeaux, em 1993. Ele desenvolve sua pesquisa no LaBRI, um laboratório de pesquisa em ciência da computação comum com a Universidade de Bordeaux e o CNRS, onde atualmente é o Diretor Adjunto. Seus interesses de pesquisa incluem algoritmos e sistemas distribuídos, modelos formais, segurança e redes ad hoc e de sensores.
Participei em vários projetos de investigação nacionais e europeus, incluindo colaborações com a indústria. Ele escreveu mais de 60 artigos de pesquisa publicados em revistas internacionais e anais de conferências e está envolvido em vários comitês de programas técnicos e organizações de muitas conferências internacionais.

Akka Zemmari recebeu seu Ph.D. pela Universidade de Bordeaux, França, em 2000. É professor associado em ciência da computação desde 2001 na Universidade de Bordeaux, França. Seus interesses de pesquisa incluem algoritmos e sistemas distribuídos, grafos, algoritmos aleatórios, aprendizado de máquina e segurança.

Cyrille Sauvignac é Gerente de Projetos Profissional Certificado e Gerente de Inovação (ITEA2 Usenet, ITEA2 A2Nets, Smart Services para projetos ITS de veículos conectados), com sólida experiência em tempo real e sistemas aviônicos, portais corporativos, padronização de interoperabilidade M2M e dispositivos embarcados. Agora, Cyrille é responsável pela inovação da Atos Aquitaine Lab para desenvolver sistemas complexos de ITS e IoT Unid.

Parvez Faruki recebeu M.Tech e PhD em Ciência da Computação e Engenharia do Malaviya National Institute of Technology Jaipur Índia em julho de 2012 e março de 2016, respectivamente. Em 2012, ele foi premiado com uma bolsa CFAIT Common Wealth para pesquisas futuras. Ele fez parte da equipe de pesquisa em nome da NIT Jaipur para um projeto DST RFBR conjunto com o Centro Educacional e de Pesquisa do Sul da Rússia para Segurança de TI na SFU Rússia. Visitei o LaBRI-Laboratoire Bordelais de Recherche en Informatique Bordeaux, França, para realizar pesquisas em 2015. Publiquei mais de 19 artigos, alguns entre os principais periódicos internacionais revisados por pares e boas conferências de segurança, incluindo IEEE Communication Surveys & Tutorials e ACM Computing Surveys.