Detecção de Intrusão em Sistemas IoT Baseada em Comitê de Classificadores

Davyson S. Ribeiro[1], Erik J. F. Nascimento[1], Juliana L. Garça [1], Márcio E. F. Maia[1], José M. da S. M. Filho[1], José D. C. Neto[2], Nicksson C. A. de Freitas[2], Emanuel B. Rodrigues[1], Jarelio G. da S. Filho[2]

[1]Universidade Federal do Ceará (UFC)
Av. da Universidade, 2853 – CEP 60020-181 – Fortaleza – CE – Brasil

[2]SiDi
Av. República do Líbano, 251 – CEP 51110-160 – Recife – PE – Brasil

{davysonribeiro, erikjhonesf, julianagarca}@alu.ufc.br,
marcioefmaia@ufc.br, monteiro@dc.ufc.br,
{j.carneiro, nicksson.a}@sidi.org.br,
emanuel@dc.ufc.br, j.filho@sidi.org.br

Abstract. *IoT applications are generally vulnerable to attacks from malicious users due to their lower robustness as well as the simplicity of use and ubiquity of devices. On the other hand, Intrusion Detection Systems (IDS) have been successfully used to analyze information from a monitored system and detect events of malicious behavior, making it possible to alert administrators and take corrective measures promptly. This work presents a new approach to intrusion detection called PaC (Preprocessing and Committee), which is based on the use of a committee of classifiers. The PaC approach has shown superior results compared to the state of the art, achieving higher accuracy, precision, recall, and F1-score metrics in detecting attacks in IoT applications.*

Resumo. *Aplicações IoT são, em geral, vulneráveis a ataques de usuários maliciosos devido à sua menor robustez atrelada à simplicidade de uso e onipresença dos dispositivos. Por outro lado, Sistemas de Detecção de Intrusão (IDS) têm sido utilizados com sucesso com a finalidade de analisar informações de um determinado sistema monitorado e detectar sinais de comportamento malicioso, o que torna possível alertar os administradores e adotar medidas corretivas de forma ágil. Este trabalho apresenta uma nova abordagem para detecção de intrusão denominada PaC (Preprocessing and Committee), a qual baseia-se na utilização de um comitê de classificadores. O PaC apresentou resultados superiores ao estado da arte, alcançando melhores valores de acurácia, precisão, recall e F1-score na detecção de ataques em aplicações IoT.*

1. Introdução

A Internet das Coisas (IoT) é um paradigma que integra um conjunto de dispositivos por meio de uma rede de interação e cooperação. Dispositivos e sistemas IoT estão sendo utilizados em diferentes áreas de aplicação, como automação residencial e industrial. À medida que as aplicações IoT vão se difundindo, maior também é a quantidade de vulnerabilidades sendo exploradas [Balaji et al. 2019].

Essas vulnerabilidades podem comprometer diretamente a integridade, disponibilidade e confidencialidade das informações. Nesse contexto, ataques de intrusão, como negação de serviço (DoS), sequestro de dados (*ransomware*), e manipulação de dados (*injection*), podem resultar em prejuízos significativos, tanto em termos financeiros quanto em relação à reputação das organizações afetadas [Asharf et al. 2020]. Por exemplo, um ataque DoS pode resultar na interrupção dos serviços essenciais fornecidos por um Sistema Industrial IoT (IIoT), levando a perdas financeiras substanciais devido à paralisação da produção, indisponibilidade de serviços ou danos físicos em infraestruturas críticas [de Souza et al. 2021].

Por outro lado, Sistemas de Detecção de Intrusão (IDS - *Intrusion Detection Systems*) têm sido utilizados com sucesso para analisar informações de um determinado sistema monitorado e detectar anomalias, o que possibilita alertar os administradores e adotar medidas corretivas de forma ágil. Essencialmente, IDS são capazes de monitorar um certo sistema em busca de eventos que possam violar suas regras de segurança [Asharf et al. 2020].

Em geral, IDS podem ser categorizados em dois grupos: i) baseados em assinaturas de padrões de intrusão e ii) baseados na detecção de anomalias em relação ao comportamento dos usuários. IDS baseados em assinatura possuem um banco de dados contendo informações sobre formas de ataques conhecidos (assinaturas). Eles funcionam de forma reativa, ou seja, monitoram determinadas informações e procuram por padrões que coincidam com assinaturas de ataques, emitindo um alerta caso o sistema detecte um ataque. Já os IDS baseados na detecção de anomalias modelam os dados referentes ao comportamento legítimo (normal). Em seguida, qualquer atividade que não possua comportamento semelhante ao modelo estabelecido será classificada como anomalia. Nos últimos anos, os métodos baseados em aprendizado de máquina (ML - *Machine Learning*) ganharam grande destaque e consolidaram-se como as principais estratégias para detecção de anomalias. Todavia, os modelos de aprendizado de máquina precisam ser treinados com grandes volumes de dados para alcançar altas taxas de acurácia, precisão, *recall* e *F1-score* [do Nascimento et al. 2021].

Nesse contexto, a utilização de técnicas avançadas para detectar e prevenir ameaças é fundamental para garantir os requisitos básicos de segurança da informação em sistemas IoT. Recentemente, diversas abordagens foram propostas para detectar ataques de intrusão em sistemas IoT [Mandal et al. 2020, Mallet et al. 2022]. A grande maioria dessas abordagens baseia-se em redes neurais profundas [Tareq et al. 2022] ou na intensiva transformação dos dados [Kumar et al. 2021].

Este trabalho apresenta uma nova abordagem para detecção de intrusão baseada em aprendizado de máquina denominada PaC (*Preprocessing and Committee*), que combina uma exploração detalhada dos conjuntos de dados para o treinamento de múltiplos classificadores, com uma estratégia de classificação múltipla chamada comitê. Essa estratégia, também conhecida como Sistema de Classificação Múltipla (MCS), é uma abordagem que envolve a combinação de múltiplos classificadores individuais para melhorar o desempenho da classificação em tarefas de aprendizado de máquina. Em vez de confiar em um único classificador, o MCS utiliza a diversidade e a complementaridade dos classificadores individuais para tomar decisões mais precisas e robustas [Géron 2019]. Essa técnica pode se destacar no processo de detecção de intrusão, uma vez que a combinação de múltiplos

classificadores apresenta melhor acurácia do que um sistema decisório único, além de reduzir a variância de classificação, e minimizar drasticamente a instabilidade inerente dos algoritmos de aprendizado de máquina.

O principal objetivo deste trabalho consiste em mostrar que a combinação de múltiplos classificadores pode ser uma ferramenta poderosa na detecção de ataques de intrusão em dispositivos e sistemas IoT. De fato, a abordagem PaC apresentou resultados superiores ao estado da arte, alcançando melhores valores para as métricas de acurácia, precisão, *recall* e *F1-score* na detecção de ataques em aplicações IoT, tanto em tarefas de classificação binária quanto nas tarefas de classificação multi-classe.

O restante deste artigo está organizado da seguinte forma. Na Seção 2, são apresentados os trabalhos relacionados encontrados na literatura. Em seguida, na Seção 3, é descrita a metodologia adotada para implementação da abordagem PaC. A Seção 4 apresenta os experimentos realizados durante este estudo. Logo após, na Seção 5, são apresentados e discutidos os resultados. Por fim, na Seção 6, são apresentadas as conclusões deste trabalho.

2. Trabalhos relacionados

Nesta seção são apresentas as principais soluções de detecção e identificação de intrusões propostas para ambientes baseados em sistemas IoT, encontradas por meio de uma pesquisa bibliográfica nas bases de dados IEEE Xplore, *Applied Sciences* e *ScienceDirect* no período de junho até dezembro de 2022.

Alsedi et al. [Alsaedi et al. 2020] abordaram a necessidade do desenvolvimento de IDS eficazes para IoT, e destacaram que nesse cenário, conjuntos de dados antigos, como KDDCUP99, NSL-KDD [Tavallaee et al. 2009], e ISCX [Sharafaldin et al. 2018], não são adequados por não incluírem características específicas destes sistemas como, por exemplo, dados de leitura de sensores e tráfego de rede. Adicionalmente, foi apresentada a construção de um novo conjunto de dados denominado ToN-IoT [Moustafa 2021], utilizado para treinar um conjunto de classificadores binários para sete dispositivos diferentes. O desempenho desses modelos foi descrito por meio das métricas de acurácia, precisão, *recall* e *F1-score*. No entanto, diferente da abordagem proposta neste trabalho (PaC), os autores não exploraram a seleção ou otimização de hiper-parâmetros dos modelos de aprendizado de máquina, bem como não apresentaram resultados para o problema da classificação multi-classe aplicada a cada um dos dispositivos presentes no conjunto de dados.

Os trabalhos de Sarhan et al. [Sarhan et al. 2021], Gad et al. [Gad et al. 2021] e Kumar et al. [Kumar et al. 2021] apresentaram análises semelhantes sobre detecção de intrusão em sistemas IoT. Todos eles consideraram que métricas como acurácia, precisão, *recall* e *F1-score* são indicadores importantes para avaliar a eficácia dos classificadores na detecção de intrusões. Vale destacar que Sarhan et al. [Sarhan et al. 2021] desenvolveu seu próprio conjunto de dados, combinando características semelhantes de diferentes conjuntos de dados, visando identificar tipos específicos de ataques de intrusão. Por sua vez, Gad et al. [Gad et al. 2021] propôs uma modelagem bem definida para seus classificadores, onde abordou de forma bem detalhada as etapas de pré-processamento de dados, balanceamento de classes, seleção de atributos, normalização dos dados e treinamento dos modelos. Já Kumar et al. [Kumar et al. 2021] preocupou-se também com a privacidade dos dados em redes IoT e criou um *framework* baseado na computação

em névoa, projetado para proteger dados confidenciais e detectar instâncias maliciosas. Um ponto em comum entre esses trabalhos é a ausência de uma etapa de seleção de hiper-parâmetros para os modelos de aprendizado de máquina. Além disso, esses trabalhos não exploraram sistemas de classificação múltipla, como os comitês. Por fim, Tareq et al. [Tareq et al. 2022] consideraram a utilização de dois modelos, *Densnet121* e *Inception Time*, ambos baseados em redes neurais profundas, para desenvolver um IDS para redes IoT. Os autores exploraram técnicas de seleção de atributos.

Na Tabela 1 é ilustrada uma comparação entre os trabalhos relacionados e a abordagem proposta (PaC). A análise comparativa é realizada por meio de diferentes critérios, tais como: i) a presença ou não de uma análise de atributos bem definida, ii) se o trabalho explorou ou não a seleção de hiper-parâmetros, iii) se a solução foi aplicada ou não a problemas de classificação binária, iv) se a abordagem foi aplicada ou não a problemas de classificação multi-classe e v) se utilizou ou não técnicas de comitê (classificação múltipla). Observando Tabela 1, pode-se notar que os trabalhos encontrados na literatura não exploraram a seleção de hiper-parâmetros e não utilizaram técnicas de comitê, sendo estas duas as principais contribuições da abordagem proposta neste artigo (PaC).

	Análise de atributos bem definida	**Seleção de hiper-parâmetros**	**Classificação binária**	**Classificação multi-classe**	**Utilização de técnicas de comitê**
[Alsaedi et al. 2020]	✗	✗	✓	✗	✗
[Sarhan et al. 2021]	✓	✗	✗	✓	✗
[Gad et al. 2021]	✓	✗	✓	✓	✗
[Kumar et al. 2021]	✓	✗	✗	✓	✗
[Tareq et al. 2022]	✓	✗	✗	✓	✗
PaC	✓	✓	✓	✓	✓

3. Abordagem Proposta

Considerando a crescente popularização dos sistemas IoT, suas deficiências de segurança, e as pesquisas relacionadas sobre métodos de detecção e identificação de intrusão nesses ambientes, neste trabalho é proposta uma nova abordagem baseada em combinar múltiplos classificadores tradicionais via *Ensemble* comitê, para melhorar a detecção de intrusão em sistemas IoT sem aumentar drasticamente o custo computacional. Essa abordagem é treinada e validada utilizando alguns dos mais modernos conjuntos de dados relacionados à detecção de intrusão em dispositivos e sistemas IoT.

As restrições e características da IoT têm implicações diretas no projeto e implementação de sistemas IDS baseados em ML. Os algoritmos de ML devem ser otimizados para funcionar de forma eficiente em dispositivos com recursos limitados. É necessário encontrar um equilíbrio entre a precisão da detecção e os recursos necessários para executar o IDS em dispositivos IoT. Além disso, a heterogeneidade dos dispositivos e protocolos de comunicação requer uma abordagem flexível que possa se adaptar a diferentes contextos e ambientes.

Uma das principais vantagens de se utilizar IDS com técnicas de ML é que é possível lidar com ataques desconhecidos. Enquanto o IDS tradicional podem identificar apenas ameaças conhecidas com base em regras predefinidas, o IDS baseado em ML pode detectar anomalias e padrões não usuais que podem indicar um ataque em desenvolvimento. Isso torna o IDS mais adaptável a ameaças emergentes e evasivas.

Os conjuntos de dados utilizados neste trabalho possuem nove rótulos de diferentes tipos de ataques cibernéticos direcionados à camada de rede de sistemas IoT, sendo eles: ***Backdoor***: Um ponto de acesso secreto é deixado intencionalmente no sistema, permitindo que um invasor tenha controle remoto e acesso não autorizado; **Negação de Serviço** (DoS): Um invasor sobrecarrega um sistema ou recurso, tornando-o inacessível aos usuários legítimos; **Negação de Serviço Distribuído** (DDoS): Múltiplos dispositivos são utilizados para inundar um sistema com tráfego malicioso; ***injection***: Um invasor insere comandos ou código malicioso em um sistema ou aplicação; ***Man-in-the-Middle*** (MITM): É realizada uma interceptação e manipulação de comunicações entre duas partes legítimas; ***Cracking* de senhas**: Um invasor utiliza técnicas para descobrir senhas por meio de força bruta; ***Ransomware***: Um tipo de malware que criptografa dados e exige um resgate em troca da chave de criptografia; ***Scanning***: Mapeia e identifica sistemas vulneráveis; ***Cross-Site Scripting*** (XSS): Um invasor injeta código malicioso em uma aplicação, permitindo que ele execute scripts mal-intencionado.

Esses ataques fornecem diferentes níveis de ameaças para os sistemas IoT, incluindo a violação de dados, interrupção dos serviços, danos financeiros e comprometimento da segurança geral. Portanto, é fundamental implementar medidas de segurança como IDS baseado em aprendizado de máquina para proteger os sistemas IoT contra esses ataques.

3.1. Modelo PaC

Com base nas motivações apresentadas anteriormente, foi projetado um IDS para sistemas IoT chamado de *Preprocessing and Committee* (PaC). O PaC é um *pipeline* formado por uma sequência de etapas executadas em ordem para processar e analisar dados de sistemas IoT a fim de realizar tarefas de detecção de intrusão. O PaC inicia com fases de pré-processamento de dados e termina com detecção dos ataques via técnica de comitê.

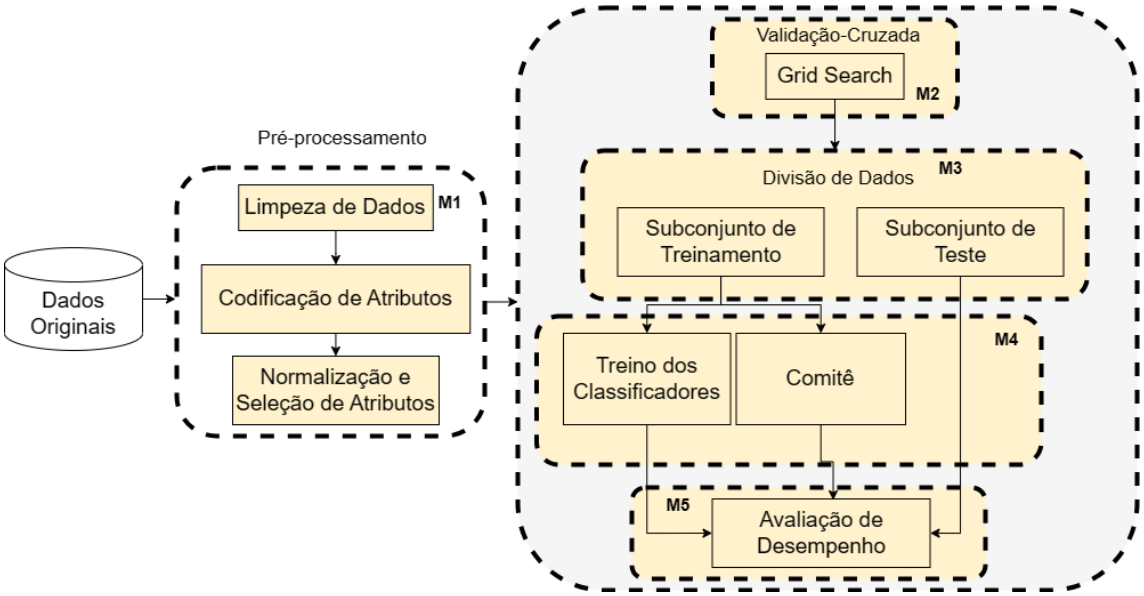
Como pode ser visto de forma mais detalhada na Figura 1, o PaC é composto por duas fases: a primeira relacionada ao tratamento que os dados receberam, e a segunda pelos processos de aprendizado de máquina. A arquitetura contém cinco módulos principais, o primeiro módulo (M1) é o de pré-processamento de dados, composto pela limpeza dos dados, codificação dos atributos, normalização e seleção dos atributos. Nessa etapa, os dados são limpos, tratados e preparados para o treinamento do modelo. Isso envolve a remoção de dados ausentes, balanceamento de classes e tratamento de características categóricas, além de conter os processos de normalização e seleção de atributos.

O segundo módulo (M2) está inserido dentro do processo de validação cruzada, e contém a busca pelos melhores hiper-parâmetros para os modelos de aprendizado de máquina através da técnica de Grid Search. Essa etapa é crucial para que os classificadores possam alcançar boas taxas de acurácia, precisão, recall e f1-score nas tarefas de detecção de intrusão. O terceiro módulo (M3) é onde os dados são divididos em subconjuntos de treinamento, validação e teste. O conjunto de treinamento é usado para treinar os modelos, o de validação para validar os hiper-parâmetros, enquanto o conjunto de teste é usado para avaliar o desempenho final dos modelos.

No quarto módulo (M4) é onde ocorre o treinamento dos classificadores e do comitê de classificadores com os dados processados. Os classificadores foram configurados com os melhores hiper-parâmetros encontrados no segundo módulo. A combinação de vários modelos individuais via comitê torna a detecção de intrusão final mais robusta. Por fim,

o último módulo (M5) é responsável por calcular as métricas para avaliação tanto dos classificadores individuais quanto para o comitê.

Levando em consideração a heterogeneidade dos dados utilizados, podemos afirmar que os módulos utilizados pelo PaC é o que possibilita um resultado equilibrado e eficiente nos processos de detecção de intrusão.



4. Experimentos

Nesta seção são apresentados os estudos e experimentos realizados com o PaC para obter a maior acurácia possível nas tarefas de detecção de intrusão em dispositivos e sistemas IoT.

Testar IDS voltados especificamente para sistemas IoT requer datasets com informações representativas capturadas dos sensores e do tráfego entre esses sistemas. No entanto, há uma carência desse tipo de informação, principalmente quando se trata de dados de ataques. Portanto, para realizar os experimentos do PaC, foram utilizados dois subconjuntos de dados derivados do dataset ToN-IoT [Moustafa 2019], sendo eles *ToN-IoT Network* e *ToN-IoT Devices*, que foram construídos para avaliar modelos de aprendizado de máquina no contexto de IDS para IoT. O dataset pode ser encontrado em sua forma original no repositório `https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=%2FRaw_datasets`.

Todas as implementações foram desenvolvidas utilizando a biblioteca Scikit-learn da linguagem Python 3.10. Além disso, os experimentos foram executados sobre um dispositivo com as seguintes configurações: Windows 11 64-bits, CPU intel Core™ i7-12700H 12^a geração, e 16GB RAM DDR5. Os códigos utilizados para realizar os experimentos, e o espaço de hiper-parâmetros que foram buscados neste trabalho estão disponibilizados no repositório `https://github.com/metodoPAC/MetodoPaC`.

Devido aos recursos limitados da maioria dos dispositivos e sistemas IoT, o IDS deve possuir baixa complexidade computacional. A seleção de atributos é um dos pontos cruciais para lidar com esse problema. Além disso, ao remover recursos redundantes, podemos melhorar ainda mais os processos de classificação. Para obter um subconjunto ideal

de atributos, foram realizadas análises envolvendo métodos de extração de características como testes de correlação de Spearman, testes de normalidade, medidas de variabilidade, medidas de tendência central, e visualizações gráficas.

Outro detalhe importante é que os conjuntos de dados utilizados neste trabalho possuem diversos atributos categóricos. Como os algoritmos de aprendizado de máquina são projetados para processar somente dados com atributos numéricos, os atributos categóricos foram transformados utilizando codificadores de dados e.g., *One-Hot Encoding*, *Label Encoding*, *M-Estimator Encoding* e *Ordinal Encoding*. Por fim, os dados processados foram utilizados para treinamento e avaliação do PaC.

Na prática, o tipo de pré-processamento, a escolha do codificador, e os tipos de classificadores que compõem o comitê, estão intimamente atrelados ao tipo de conjunto de dados utilizado para treinar o IDS. Neste trabalho, foram analisados diversos classificadores: *MultiLayer Perceptron* (MLP), *Logistic Regression* (LR), *K-Nearest Neighbors* (KNN), *Naive Bayes* (NB), *Linear Discriminant Analysis* (LDA), *Decision Tree Classifier* (DTC) e *Random Forest* (RF) para saber qual a melhor combinação para compor o PaC. A seguir são descritos os detalhes dos experimentos dos dois conjuntos de dados utilizados.

ToN-IoT Network. Esse subconjunto de dados apresenta a coleta de nove tipos diferentes de ataques (*backdoor*, DoS, DDoS, *injection*, *cracking* de senhas, *ransomware*, *scanning*, XSS e *man-in-the-middle*). Possui 45 atributos, sendo 43 de caracterização e 2 de rotulação.

Ao todo, 20 desses atributos possuem até 90% de dados faltantes (e.g., *ssl_cipher*, *ssl_resumed*, *ssl_established*, *ssl_subject*, *ssl_issuer*). Após a etapa de análise exploratória detalhada dos atributos, foi constatado que a remoção deles não afetava a capacidade discriminativa do conjunto de dados. Além disso, os atributos *timestamp*, *src_ip*, *dest_ip*, *src_port* e *dest_port*, também foram removidos, pois não estão relacionados aos rótulos de classificação. Usuários mal-intencionados tendem a lançar ataques para computadores de usuários legítimos, o que sugere que os atributos *src_ip*, *dest_ip*, *src_port* e *dest_port* não ajudam na identificação de invasões e podem resultar em problemas de *overfitting* para a classificação. Contendo valores discretos, o atributo *timestamp* é incapaz de contribuir para o reconhecimento de um ataque, portanto também foi removido. Ao fim desta etapa, o dataset ficou com 18 atributos.

Na etapa de pré-processamento, os dados categóricos foram codificados utilizando *M-estimator Encoding*, um tipo de codificador que atribui um peso a cada categoria com base no desvio da frequência geral dos rótulos. Este codificador foi escolhido porque ele consegue reduzir a taxa de *overfitting* e *underfitting*, consegue lidar com *outliers* e com classes desbalanceadas, além de codificar bem distribuições não-normais e não alterar a dimensão do conjunto de dados, o que não aumenta o tempo para o treinamento do classificador. Os dados resultantes foram normalizados via *MinMaxScaller*.

Como esse conjunto de dados possui desbalanceamento do total de amostras por classe, foi aplicado um método de *data augmentation* (DA) com *Variational Autoencoders* (VAE). DA-VAE é um método utilizado para aumentar o tamanho do conjunto de dados de treinamento, gerando variações artificiais dos dados existentes. Esses dados artificiais são gerados por amostragem da distribuição aprendida pelo VAE. Essa abordagem de aumento de dados se mostrou relevante, reduzindo o *overfitting* e ajudou a melhorar o desempenho

do PaC. Por fim, após diversos experimentos comparativos envolvendo a combinação entre os classificadores, os modelos RF, DTC e MLP foram selecionados para compor o comitê do PaC para esse conjunto de dados. Aqui, foi escolhido o comitê *voting regressor*, um tipo de comitê que combina as saídas dos classificadores via média ponderada. Para mais detalhes acesse o link do GitHub.

ToN-IoT Devices. Esse subconjunto de dados apresenta a coleta de oito tipos diferentes de ataques (*backdoor*, DoS, DDoS, *injection*, *cracking* de senhas, *ransomware*,*scanning* e XSS) em sete sensores IoT (sensor de geladeira, sensor de porta de garagem, sensor GPS, sensor modbus, sensor de movimentação e iluminação, sensor termostato e sensor climático). Cada sensor possui seu próprio conjunto de dados e características específicas.

O sensor da geladeira possui os atributos *fridge_temperature* e *temp_condition*; a porta da garagem tem os atributos *door_state* e *sphone_signal*; o GPS usa atributos de latitude e longitude; o modbus possui 4 atributos, sendo um para ler um registrador de entrada, um para ler um valor discreto, um para ler um registrador de retenção e o último para ler uma bobina; o sensor de movimento e iluminação possui um atributo para cada uma dessas funções; o termostato possui um atributo de status e outro para a temperatura; e o sensor climático possui 3 atributos, para as leituras de temperatura, pressão e umidade. Além dessas características específicas, os datasets de todos os sensores possuem atributos de *timestamp*, data, hora, e os atributos de rotulação dos dados: label, para identificar acesso normal e malicioso; e type para descriminar acesso normal ou o tipo específico do ataque.

Inicialmente cada sensor foi analisado de forma individual, mas posteriormente todos os sensores foram combinados gerando um conjunto de dados único que também foi analisado e será apresentado na seção de resultados. O primeiro passo foi verificar o balanceamento dos conjuntos de dados de todos os sensores, pois como vimos anteriormente, o desbalanceamento de classes pode interferir nos processos de decisão dos modelos de aprendizado de máquina e levar a resultados ruins no processo de detecção de ataques. Cerca de 58% das leituras representam acessos normais e 42% delas representam ataques. Concluiu-se que esse cenário pode ser caracterizado como um desbalanceamento leve, visto que foi constatado que técnicas de balanceamento não apresentaram melhoras significativas nas métricas de avaliação para os sensores.

O próximo passo foi analisar a relevância das informações temporais para o processo de aprendizagem. Para isso foi realizada a divisão dos atributos *date* e *time* em outros mais específicos. O atributo de data deu origem a novos três (dia, mês e ano), enquanto o atributo de tempo foi separado em outros três (hora, minuto e segundo). Após essa modificação, toda a correlação entre os atributos foi analisada novamente, com o intuito de prevenir que os modelos de aprendizado de máquina ficassem enviesados. Foi constatado que a melhor configuração possível seria excluir os atributos de mês e ano para realizar o processo de treinamento, pois estes geravam *overfitting* por não estarem relacionados aos rótulos de ataques. Esse processo permitiu os algoritmos apresentarem resultados mais satisfatórios. Ao final dessa etapa, os datasets da geladeira, porta da garagem, GPS, sensor de movimento e iluminação e termostato ficaram com 8 atributos cada, o dataset do sensor climático ficou com 9 atributos e o do modbus com 10.

Alguns dispositivos apresentam dados categóricos, tais como porta da garagem, sensor de movimento e iluminação, e o termostato, sendo necessária a codificação desses atributos. Foram realizados experimentos com os codificadores *Ordinal Encoder* e o *One-Hot Encoder*. O *Ordinal Encoder* apresentou melhores resultados em 70% dos experimentos, devido ao fato dele preservar a informação da ordem das categorias, a qual mostrou-se relevante para o problema. Por exemplo, em um problema de classificação com categorias "ligado" e "desligado", a codificação ordinal podia refletir a relação de ordem natural entre essas categorias, o que era benéfico para determinados algoritmos de classificação como, por exemplo, árvores de decisão [Géron 2019]. No entanto, é importante ressaltar que nem sempre o *Ordinal Encoder* apresentou melhores resultados. Em alguns casos, quando não havia uma ordem inerente entre as categorias ou quando o algoritmo de classificação não era sensível a essa ordem, o *One-Hot Encoder* era preferível para evitar qualquer viés decorrente de suposições de ordem.

Após o tratamento dos atributos categóricos, a técnica de padronização *StandardScaler* foi utilizada para dimensionar os dados não numéricos. A padronização dos dados é especialmente importante para algoritmos sensíveis à escala das características. Inicialmente todos os sensores possuíam em média 5 atributos. A partir do tratamento que foi dado, essa média subiu para 9. Sendo assim, outro ponto significativo para a análise foi a utilização de seleção de *features* para selecionar as características mais relevantes ou informativas do conjunto de dados, visando melhorar o desempenho dos modelos de classificação [Imad et al. 2022].

Outro aspecto bastante importante e decisivo para aperfeiçoar a precisão de todos os processos em ambos subconjuntos de dados foi a utilização conjunta do *grid search*, o qual realiza a busca pelos melhores hiper-parâmetros para cada um dos modelos, e a técnica de comitê. Embora essa busca exaustiva possa ser computacionalmente custosa, sem a utilização dos hiper-parâmetros ideais para cada modelo, as métricas de acurácia, precisão, *recall* e f1-score têm uma queda significativa. Uma vez que os melhores hiper-parâmetros foram encontrados, foi essencial utilizá-los para treinar o comitê de modelos. A utilização dessas técnicas em conjunto garantiu que todos os modelos fossem otimizados para os conjuntos de dados em questão, permitindo uma generalização melhor e um aumento significativo no desempenho tanto em tarefas de classificação binária quanto multiclasse.

O subconjunto de dados *ToN-IoT Devices* foi treinado pelo comitê resultante formado pelos classificadores LR, KNN, NB, LDA e RF. Além disso, com o intuito de diversificar ainda mais os comitês, foram utilizados diferentes hiper-parâmetros para cada um dos modelos. Para o treinamento do subconjunto de dados *ToN-IoT Devices*, o *grid search* consistiu em uma busca exaustiva por combinações de hiper-parâmetros pré-definidos em um espaço de busca, pois cada sensor inserido no conjunto de dados continha características próprias que necessitavam de parâmetros específicos. Os detalhes das configurações utilizadas nos experimentos podem ser acessadas no link do GitHub.

5. Resultados

Nesta seção são apresentados os resultados dos experimentos descritos previamente. Para fazer uma avaliação DOS modelos, além da acurácia, foram analisadas as métricas de precisão, revocação (*recall*) e f1-score. Essas métricas desempenham um papel crucial na avaliação do desempenho de um modelo de detecção de intrusão em sistemas IoT, pois

elas mostram com clareza a capacidade do modelo em identificar corretamente os eventos de intrusão e distinguir entre atividades normais e maliciosas.

A escolha das métricas deve-se ao fato de que penas a acurácia não é suficiente para avaliar adequadamente a eficácia do modelo, especialmente em casos de desequilíbrio entre as classes de intrusão e atividades normais. Nesse sentido, a precisão e o *recall* são métricas distintas e importantes.

A precisão mede a proporção de exemplos classificados como intrusão que realmente são intrusões, ou seja, a capacidade do modelo em minimizar os falsos positivos. Uma alta precisão indica que o modelo comete poucos erros ao classificar eventos normais como intrusões. Por outro lado, o *recall* mede a proporção de exemplos de intrusão corretamente identificados pelo modelo, ou seja, a capacidade de capturar efetivamente a maioria das intrusões e minimizar os falsos negativos. Um alto recall indica que o modelo está identificando a maioria das intrusões presentes no conjunto de dados.

Além disso, o f1-score é uma métrica que combina a precisão e o *recall* em uma única medida, fornecendo uma avaliação geral do desempenho do modelo. Ele é particularmente útil quando é necessário encontrar um equilíbrio entre precisão e *recall*. Ao avaliar e comparar diferentes modelos de detecção de intrusão em sistemas IoT, é essencial levar em consideração essas métricas de desempenho. Um modelo com altos valores em ambas as métricas é capaz de identificar eficientemente os eventos de intrusão, minimizando tanto os falsos positivos quanto os falsos negativos.

A fim de garantir uma comparação justa, foram seguidas as configurações utilizadas nos trabalhos relacionados. Utilizamos um particionamento de 80%/10%/10% (treino, teste e validação) para ambos os subconjuntos de dados *ToN-IoT Network* e *ToN-IoT Devices*. Além disso, utilizamos uma avaliação do tipo k-*fold cross validation*.

5.1. ToN-IoT Network

Esta subseção apresenta os resultados relacionados ao subconjunto de dados *ToN-IoT Network* e servirá para verificação da efetividade do PaC em detectar intrusões em sistemas IoT. Foi realizado um estudo comparativo entre o modelo proposto neste trabalho e alguns métodos recentes encontrados na literatura para classificação do subconjunto de dados *ToN-IoT Network*. O PaC também é comparado contra os classificadores RF, DT e MLP. Esses classificadores foram treinados utilizando os melhores hiper-parâmetros encontrados.

A Tabela 2 mostra os resultados de um problema de classificação multiclasse, onde várias classes de ataques estão presentes. A acurácia do modelo PaC é comparada com os resultados dos classificadores RF, DT e MLP, bem como os classificadores apresentados no trabalho [Kumar et al. 2021]. No geral, o PaC supera de forma majoritária os classificadores RF, DT e MLP, especialmente na detecção de ataques dos tipos *backdoor*, DDoS, *injection*, *Man-In-The-Middle* (MITM), *craking* de senhas (*password*), *ransomware*, *scanning* e *Cross-Site Scripting* (XSS). Considerando o estado-da-arte, pode-se notar que o PaC é altamente competitivo contra o método P2IDF* proposto em [Kumar et al. 2021]. A diferença é mais expressiva considerando os ataques MITM e DDoS, onde o PaC supera o P2IDF* em 10.4% e 4.7%, respectivamente. Pode-se notar que ainda que o método DT de Kumar et al. teve 0% de acurácia na detecção de ataques de *injection* e MITM. Uma possível explicação para isso é a baixa quantidade de amostras para essas classes de ataques, que fez com que o método não conseguisse atingir uma boa generalização.

	Método	Backdoor	DDoS	DoS	Injection	MITM	Normal	Password	Ransomware	Scanning	XSS
Kumar et al.	NB	99.2	26.8	91.7	92.9	95.1	100.0	75.3	79.9	96.9	19.0
	DT	100.0	**100.0**	100.0	0.0	0.0	100.0	**100.0**	**100.0**	**100.0**	**100.0**
	RF	99.9	90.4	91.9	93.5	0.0	100.0	97.8	99.4	95.7	85.4
	P2IDF	99.8	94.2	98.2	95.4	92.0	100.0	99.8	99.6	1.0	97.8
	P2IDF*	99.8	94.2	98.2	98.1	88.8	100.0	98.8	99.8	99.0	99.8
Nosso	RF	100.0	96.4	**100.0**	86.5	43.9	99.1	99.2	99.3	98.7	91.6
	DT	100.0	95.5	99.2	96.9	53.7	100.0	99.6	99.5	97.6	94.3
	MLP	99.5	85.9	100.0	62.7	41.7	97.6	71.2	96.3	95.8	77.9
	PaC	**100.0**	98.9	98.8	**99.7**	**99.2**	**100.0**	99.8	99.7	99.7	99.8

Na Tabela 3, é apresentada uma comparação dos métodos do estado-da-arte contra o PaC. Os resultados foram calculados por meio de uma média ponderada das classes de ataque. Os valores reportados para os métodos concorrentes foram adquiridos dos próprios artigos. De forma geral, o PaC supera métodos mais complexos que ele, como por exemplo *Inception Time*, *DenseNet* e P2IDF. Aqui, foi apresentado P2IDF, em vez do P2IDF*, porque ele possui melhores resultados. O ótimo desempenho do PaC demonstra que não é necessária uma alta quantidade de parâmetros para alcançar um alto índice de *F1-score*. Todos os resultados foram adquiridos dos próprios artigos comparados com as Tabelas 2 e 3 foram adquiridos dos próprios artigos citados. A não disponibilização do código fonte e baixa riqueza de detalhes na metodologia tornaram impossíveis a reprodutibilidade dos experimentos.

Autor	Método	Acurácia	Precisão	Recall	F1-score
Sarhan, et al.	NB	96.78 ± 1.03	98.01 ± 1.12	98.03 ± 1.21	98.10 ± 0.98
	DT	97.29 ± 0.98	99.01 ± 1.23	99.03 ± 1.05	99.01 ± 0.97
Gad, et al.	XGBoost	98.30 ± 1.02	98.32 ± 1.00	98.33 ± 0.98	98.30 ± 1.01
Tareq, et al.	Inception	99.65 ± 0.97	99.68 ± 0.99	99.64 ± 0.89	99.67 ± 0.97
	DenseNet	98.57 ± 1.20	98.59 ± 1.09	98.57 ± 1.01	98.57 ± 1.02
Kumar, et al.	P2IDF	99.12 ± 0.87	99.15 ± 0.89	97.23 ± 0.98	98.25 ± 1.02
Nosso	RF	97.35 ± 0.78	97.51 ± 0.98	98.32 ± 0.89	98.52 ± 0.86
	MLP	93.52 ± 0.54	93.74 ± 0.59	93.52 ± 0.58	93.53 ± 0.52
	DT	99.02 ± 0.27	99.03 ± 0.28	99.02 ± 0.18	99.02 ± 0.22
	PaC	$\mathbf{99.98 \pm 0.78}$	$\mathbf{99.89 \pm 0.62}$	$\mathbf{99.87 \pm 0.88}$	$\mathbf{99.97 \pm 0.57}$

5.2. ToN-IoT Devices

Neste trabalho também foram conduzidos experimentos com o conjunto de dados *ToN-IoT Devices*. Os modelos foram avaliados com um dataset que unificou os registros de todos os dispositivos. Nesses experimentos foram avaliados 5 modelos de aprendizado de máquina para classificação binária e multiclasse: LR, KNN, NB, LDA e RF. A classificação binária consiste em identificar se um acesso é legítimo ou pode ser considerado um ataque, enquanto na classificação multiclasse, os acessos são classificados entre normal, DDoS, *backdoor*, injeção de código malicioso (*injection*), *cracking* de senha (*password*), *ransomware*, *scanning* e XSS.

Cada dispositivo recebeu tratamento seguindo a ordem do *pipeline* definido, ou seja, os datasets estão preservados (dados desbalanceados), a codificação dos atributos

categóricos é realizada pelo *Ordinal Encoder*, a padronização é feita através do *Standard Scaler*, seguido por uma fase de seleção de atributos em conjunto com os melhores parâmetros encontrados pelo processo de *grid search*.

Neste estudo, realizamos uma comparação da efetividade do PaC com a abordagem de [Alsaedi et al. 2020] denominada PDDE (*Per-Device Dataset Evaluation*). Na tabela 4 pode ser vista a comparação dos resultados das médias ponderadas das métricas de acurácia, precisão, *recall* e F1-score dos classificadores binários e multiclasse aplicados ao dataset *ToN-IoT Devices* unificado utilizando a técnica de comitê (PaC) em comparação com a técnica PDDE.

Método	**Acurácia**				**Precisão**				**Recall**				**F1-score**			
	PDDE		**PaC**		**PDDE**		**PaC**		**PDDE**		**PaC**		**PDDE**		**PaC**	
	B	**M**	**B**	**M**	**B**	**M**	**B**	**M**	**B**	**M**	**B**	**M**	**B**	**M**	**B**	**M**
LR	0.61	0.61	0.75	0.46	0.37	0.38	0.81	0.74	0.61	0.62	0.69	0.46	0.46	0.47	0.69	0.52
KNN	0.84	0.72	**1.00**	**0.97**	0.85	0.71	**1.00**	**0.97**	0.84	0.73	**1.00**	**0.97**	0.84	0.70	**1.00**	**0.97**
NB	0.62	0.54	0.62	0.61	0.63	0.59	0.62	0.44	0.62	0.51	0.52	0.61	0.51	0.52	0.53	0.46
LDA	0.68	0.62	0.79	0.65	0.74	0.46	0.75	0.58	0.68	0.63	0.74	0.65	0.46	0.51	0.75	0.56
RF	0.85	0.71	1.00	0.97	0.87	0.69	1.00	0.97	0.68	0.72	1.00	0.97	0.62	0.67	1.00	0.97

No geral, os algoritmos KNN e RF demonstraram um desempenho consistente e eficaz tanto no processo de classificação binária, quanto multiclasse, obtendo resultados bons e em alguns casos resultados perfeitos. Optamos por dar destaque ao KNN devido o processo da busca pelos melhores hiper-parâmetros ter ocorrido de maneira mais rápida, o que levou a um tempo de processamento menor que o RF.

Como pode ser observado, o PaC, o qual é baseada na utilização combinada de *grid search* com comitê de classificadores, foi capaz de superar a abordagem PDDE proposta em [Alsaedi et al. 2020] para todos os modelos de ML considerados.

É essencial ressaltar que os resultados obtidos a partir de experimentos conduzidos estão intrinsecamente delimitados pelo escopo e pelas condições definidas durante os testes. Caso o modelo em questão seja aplicado em dispositivos implantados em ambiente real, é crucial reconhecer a necessidade de uma readaptação do modelo. As nuances do ambiente real podem diferir significativamente das situações simuladas, o que pode impactar a acurácia do modelo. Portanto, a contínua coleta de dados e ajustes iterativos são indispensáveis para garantir que o modelo se adeque e mantenha sua eficácia em condições dinâmicas e variáveis.

Em comparação com os trabalhos encontrados na literatura existente, observamos que nosso método pode ser considerado competitivo em relação ao estado da arte para cada um dos dispositivos. Embora a utilização combinada da técnica de grid search com o comitê melhore as taxas de avaliação no processo de identificação de ataques, vale ressaltar que o processo de busca por hiper-parâmetros na fase de treinamento pode ser demorado, pois são analisadas todas as combinações possíveis definidas em um espaço de busca delimitado.

6. Considerações Finais

Neste trabalho foi apresentada o modelo *Preprocessing and Committee* (PaC), uma nova abordagem de IDS para redes IoT. O PaC é constituído por etapas de processamento

detalhado dos dados e um modelo de detecção de ciberataques via comitê de classificadores. A combinação dessas duas características tornaram o PaC altamente competitivo contra os mais recentes IDS que utilizam redes neurais profundas. Experimentos utilizando os datasets *ToN-IoT Network* e *ToN-IoT Devices* mostraram que o PaC superou técnicas do estado-da-arte na detecção de ataques de intrusão em sistemas IoT considerando todas as métricas apresentadas. O ganho do PaC foi mais expressivo na detecção de ataques do tipo *Man-In-The-Midlle* e injeção de DDoS, onde ele superou o estado-da-arte P2IDF* em 10.4% e 4.7%, respectivamente.

Devemos levar em consideração que overhead em dispositivos IoT é uma preocupação central devido às suas limitações de recursos computacionais. A computação em borda poderá surgir como uma solução promissora para mitigar esse problema, permitindo o processamento local de dados antes do envio à nuvem, por exemplo. A implementação de IDS na borda pode ser uma estratégia eficaz. Ao analisar os dados de tráfego e identificadores de possíveis ameaças ou anomalias na própria borda, a sobrecarga de comunicação tende a ser reduzida, uma vez que somente informações relevantes poderão ser transmitidas à nuvem para uma análise mais aprofundada.

7. Agradecimentos

Parte dos resultados apresentados neste trabalho foram obtidos através do projeto "RESIDÊNCIA EM SEGURANÇA DA INFORMAÇÃO", executado pela UFC, em parceria com o SiDi e financiado pela Samsung Eletrônica da Amazônia Ltda., no âmbito da Lei de Informática no. 8.248/91.

Referências

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., and Anwar, A. (2020). TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8:165130–165150.

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., and Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics*, 9(7):1177.

Balaji, S., Nathani, K., and Santhakumar, R. (2019). IoT technology, applications and challenges: a contemporary survey. *Wireless personal communications*, 108:363–388.

de Souza, C., Cardoso, J., and Westphall, C. (2021). Multiclass decomposition and artificial neural networks for intrusion detection and identification in internet of things environments. In *Anais do XXI SBSeg*, pages 85–98, Porto Alegre, RS, Brasil. SBC.

do Nascimento, E. J. F., Souza, A. H., and Mesquita, D. (2021). Improving graph variational autoencoders with multi-hop simple convolutions. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 105–110.

Gad, A. R., Nashat, A. A., and Barkat, T. M. (2021). Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access*, 9:142206–142217.

Géron, A. (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books.

Imad, M., Abul Hassan, M., Hussain Bangash, S., and Naimullah (2022). A comparative analysis of intrusion detection in IoT network using machine learning. In *Big Data Analytics and Computational Intelligence for Cybersecurity*, pages 149–163. Springer.

Kumar, P., Kumar, R., Srivastava, G., Gupta, G. P., Tripathi, R., Gadekallu, T. R., and Xiong, N. N. (2021). PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. *IEEE Transactions on Network Science and Engineering*, 8(3):2326–2341.

Mallet, J., Pryor, L., Dave, R., Seliya, N., Vanamala, M., and Sowells-Boone, E. (2022). Hold on and swipe: a touch-movement based continuous authentication schema based on machine learning. In *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, pages 442–447. IEEE.

Mandal, K., Rajkumar, M., Ezhumalai, P., Jayakumar, D., and Yuvarani, R. (2020). Improved security using machine learning for IoT intrusion detection system. *Materials Today: Proceedings*.

Moustafa, N. (2019). New generations of internet of things datasets for cybersecurity applications based machine learning: TON-IoT datasets. In *Proceedings of the eResearch Australasia Conference, Brisbane, Australia*, pages 21–25.

Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON-IoT datasets. *Sustainable Cities and Society*, 72:102994.

Sarhan, M., Layeghy, S., Moustafa, N., and Portmann, M. (2021). Netflow datasets for machine learning-based network intrusion detection systems. pages 117–135. Springer.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *4th International Conference on Information Systems Security and Privacy (ICISSp)*, 1:108–116.

Tareq, I., Elbagoury, B. M., El-Regaily, S., and El-Horbaty, E.-S. M. (2022). Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT datasets using DL in cybersecurity for IoT. *Applied Sciences*, 12(19):9572.

Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. IEEE.