FamilyGuard: Uma arquitetura de segurança para detecção de anomalias em redes domésticas

Pedro Henrique Aparecido Damaso de Melo

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2022

Pedro Henrique Aparecido Damaso de Melo

FamilyGuard: Uma arquitetura de segurança para detecção de anomalias em redes domésticas

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Pedro Frosi Rosa
Coorientador: Rodrigo Sanches Miani

Uberlândia
2022

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

M528 Melo, Pedro Henrique Aparecido Damaso de, 1991-
2022 FamilyGuard: Uma arquitetura de segurança para detecção de anomalias em redes domésticas [recurso eletrônico] / Pedro Henrique Aparecido Damaso de Melo. - 2022.

Orientador: Pedro Frosi Rosa.
Coorientador: Rodrigo Sanches Miani.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2022.587
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Rosa, Pedro Frosi,1959-, (Orient.). II. Miani, Rodrigo Sanches,1983-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese de Doutorado 18/2022, PPGCO				
Data:	04 de outubro de 2022	Hora de início:	09:00	Hora de encerramento:	13:06
Matrícula do Discente:	11713CCP006				
Nome do Discente:	Pedro Henrique Aparecido Damaso de Melo				
Título do Trabalho:	FamilyGuard: Uma arquitetura de segurança para detecção de anomalias em redes domésticas				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Sistemas de Computação				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Prof. Dr. Rafael Pasquini - FACOM/UFU; Prof. Dr. Flávio de Oliveira Silva - FACOM/UFU; Prof. Dr. Bruno Bogaz Zarpelão - Universidade Estadual de Londrina (UEL); Prof. Dr. Cesar Augusto Cavalheiro Marcondes - Instituto Tecnológico de Aeronáutica (ITA); Prof. Dr. Rodrigo Sanches Miani - FACOM/UFU - coorientador; e Prof. Dr. Pedro Frosi Rosa - FACOM/UFU - orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Bruno Bogaz Zarpelão - Londrina/PR; Cesar Augusto Cavalheiro Marcondes - São José dos Campos/SP; Rodrigo Sanches Miani, Rafael Paquini, Flávio de Oliveira Silva e Pedro Frosi Rosa - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Pedro Frosi Rosa, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato por unanimidade:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Pedro Frosi Rosa**, **Coordenador(a)**, em 04/10/2022, às 13:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Bruno Bogaz Zarpelão**, **Usuário Externo**, em 06/10/2022, às 13:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rafael Pasquini**, **Professor(a) do Magistério Superior**, em 06/10/2022, às 21:09, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 07/10/2022, às 10:38, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Cesar Augusto Cavalheiro Marcondes**, **Usuário Externo**, em 20/10/2022, às 22:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Flávio de Oliveira Silva**, **Professor(a) do Magistério Superior**, em 25/10/2022, às 08:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3969095** e o código CRC **3D2B2DCE**.

Referência: Processo nº 23117.074626/2022-64 SEI nº 3969095

Dedico este trabalho aos meus pais Pedro e Lusia,
pelo apoio incondicional e constante incentivo ao longo da minha vida.

Agradecimentos

Agradeço em primeiro lugar a Deus, que iluminou o meu caminho durante essa jornada e por ter me dado força e coragem para desbravar esse período de muito estudo e aprendizado. Aos meus pais, um grande e sincero obrigado, que sempre me apoiaram com muito carinho para que eu pudesse alcançar os meus sonhos. Agradeço a Ana Caroline pelo apoio, incentivo, compreensão e companheirismo durante os momentos dessa caminhada. Aos amigos que encontrei durante todo o percurso, na Universidade Federal do Triângulo Mineiro (UFTM), Universidade Federal de Uberlândia (UFU) e Centro de Tecnologia da Informação e Comunicação (CTIC), seja no âmbito acadêmico, profissional ou pessoal, a troca de experiências foi inestimável para o meu crescimento, foi ótimo poder contar com vocês. Aos professores da Faculdade de Computação, pelo apoio e ensinamentos, que foram muito além dos conteúdos programáticos, com os quais foi possível adquirir aprendizados importantes para a vida toda. Por fim, gostaria de agradecer aos orientadores e amigos, o Prof. Rodrigo Sanches Miani e Prof. Pedro Frosi Rosa, pelo convívio, apoio, compreensão e palavras regadas de sabedoria.

"Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível."

(Charles Chaplin)

Resumo

A evolução de protocolos e dispositivos inteligentes para o ambiente residencial está ocorrendo em ritmo acelerado. Novos dispositivos e aplicações são criados por diversos fabricantes e disponibilizados no mercado todos os dias. Essa transformação tecnológica está impactando diversas áreas, uma delas é o ambiente doméstico que incorpora dispositivos inteligentes conectados à Internet para fornecer comodidade para as pessoas. Apesar de todos os benefícios adquiridos pela adoção de dispositivos inteligentes no ambiente residencial existe uma preocupação da comunidade científica em relação aos mecanismos de segurança, pois trata-se de um ambiente heterogêneo e com dispositivos que possuem recursos de hardware e software limitados. Baseado nisso, este trabalho propõe uma arquitetura de segurança que fornece uma camada de proteção adicional para redes domésticas por meio do uso de modelos de detecção de anomalias. Três aspectos da arquitetura em questão, batizada de FamilyGuard, foram avaliados neste trabalho: a implementação dos componentes em um hardware de baixo custo, a capacidade dos modelos de aprendizado de máquina para detectar ameaças e o tempo de resposta gasto durante o processo de detecção de anomalias. Para criar os modelos de detecção de anomalias também foi necessário definir e adaptar os conjuntos de dados para representar o tráfego de rede presente em um ambiente residencial. Após a definição do conjunto de dados de tráfego da rede doméstica, foram realizados experimentos para verificar a viabilidade de criar modelos não-supervisionados para detectar anomalias e usar algoritmos supervisionados para classificar as ameaças identificadas. Em vista disso, avaliou-se o desempenho de doze algoritmos do tipo *one-class* em três casos de teste; os melhores modelos apresentaram uma Área Sob Curva (AUC) superior a 94%, o que indica a utilidade da adoção de algoritmos de aprendizado não supervisionado na identificação de tráfego anômalo em redes domésticas. Os resultados também indicam que os modelos supervisionados são capazes de classificar as ameaças e que a arquitetura proposta pode fornecer uma camada adicional de segurança ao cenário residencial.

Palavras-chave: Cibersegurança, Internet das coisas, Detecção de Anomalias.

Abstract

The evolution of protocols and smart devices for the residential environment is taking place quickly; several manufacturers have created new devices and applications and made them available in the market daily. This technological transformation is impacting several areas, one of which is the home environment that incorporates smart devices connected to the Internet to provide convenience for people. Despite all the benefits of adopting smart devices in the residential environment, there is a concern in the scientific community regarding the security mechanisms since it is a heterogeneous environment with devices that have security capabilities, limited hardware, and software. This work proposes a security architecture that provides an additional layer of protection for home networks through anomaly detection models. Three aspects of the architecture in question, named FamilyGuard, were evaluated in this work: the implementation of components on low-cost hardware, the ability of machine learning models to detect threats, and the response time during the anomaly detection process. To create the anomaly detection models, it was also necessary to define and adapt the data sets to represent the network traffic in a residential environment. After defining the home network traffic dataset, experiments were carried out to verify the feasibility of creating unsupervised models to detect anomalies and using supervised algorithms to classify the identified threats. The performance of twelve one-class algorithms was evaluated in three test cases; the best models presented an area under the curve (AUC) greater than 94%, which indicates the usefulness of adopting unsupervised learning algorithms in the identification of anomalous traffic in home networks. The results also indicate that supervised models can classify threats and that the proposed architecture can provide an additional layer of security to the residential scenario.

Keywords: Cybersecurity, Internet of Things, Anomaly Detection.

Lista de ilustrações

Lista de tabelas

Lista de siglas

ANN Artificial Neural Network

AIWO AI Workflow Orchestrator

AKNN Average KNN

CPD Conditional Probability Distribution

CSS Cloud Security Service

CIC Canadian Institute for Cybersecurity

CBB Customized Blockchain-Based

CBLOF Clustering-Based Local Outlier Factor

COPOD Copula-Based Outlier Detection

DoS Denial of Service

DDoS Distributed Denial of Service

FTP File Transfer Protocol

FB Feature Bagging

GRNN General Regression Neural Network

GHOST Safe guarding home IoT environments with personalised realtime risk control

HAN Home Area Network

HP Hewlett-Packard

HTTPS Hyper Text Transfer Protocol Secure

HS Home Switch

HNR HomeNetRescue

IoT Internet of Things

ISP Internet Service Provider

IF Isolation Forest

IMAP Internet Message Access Protocol

IPFIX Internet Protocol Flow Information Export

KNN k-Nearest Neighbor

LOF Local Outlier Factor

LS-SVM Least-squares support-vector machine

LSVM Lagrangian Support Vector Machine

LEA Lightweight Encryption Algorithm

LODA Lightweight On-line Detector of Anomalies

MITM Man In the Middle

ML Machine Learning

MCD Minimum Covariance Determinant

MKNN Median KNN

NNRW Neural network with random weights

NB Naive Bayes

OCC One Class Classification

OCSVM One-Class Support Vector Machines

OCNB One Class Naive Bayes

PMI Project Management Institute

PCAP Packet Capture Data File

P2P Peer-to-peer

POP3 Post Office Protocol

PCA Principal Component Analysis

RBAC Role-Based Access Control

SVM Support Vector Machine

SDN Software Defined Networking

SHSec Secure Smart Home Environment

SMTP Simple Mail Transfer Protocol

SSL Secure Sockets Layer

SSH Secure Shell

SECaaS Security-as-a-Service

SS Standard Scaler

SVC C-Support Vector Classification

TCP Transmission Control Protocol

UDP User Datagram Protocol

VPN Virtual Private Network

VoIP Voice over Internet Protocol

Sumário

Capítulo 1

Introdução

Com a evolução dos dispositivos Internet of Things (IoT) surge o ambiente doméstico inteligente (*smart home*), no qual, objetos físicos como eletrodomésticos, relógios e lâmpadas se conectam à Internet e fornecem serviços inteligentes para proporcionar facilidades para seus residentes (FERNANDES; JUNG; PRAKASH, 2016). Acrescenta-se que, as tecnologias IoT permitem monitorar e controlar o ambiente físico. Dessa forma, é possível observar, processar e analisar dados gerados por dispositivos, sensores e objetos inteligentes (ATZORI; IERA; MORABITO, 2010).

Paralelamente, com a evolução de todo ecossistema IoT surge um crescimento acelerado de aplicativos e dispositivos, que contribui significativamente para o aumento das ameaças encontradas em ambientes residenciais (DACIER et al., 2017).

Trabalhos recentes demonstram a preocupação dos pesquisadores com os desafios encontrados para manter o ambiente residencial seguro (AMMI; ALARABI; BENKHELIFA, 2021) (MASCARENHAS et al., 2021). Zaidan e Zaidan (2020), por exemplo, enumeram seis preocupações principais: problemas de segurança, gerenciamento de dados, conectividade dos dispositivos, limitação de hardware, métodos de aprendizado de máquina e comportamento do usuário.

A comunidade científica está empenhada em encontrar soluções para os problemas de segurança em redes domésticas. Os principais trabalhos nessa linha incluem: uma arquitetura chamada *Secure Smart Home Environment (SHSec)* (SHARMA et al., 2019), o projeto de pesquisa europeu GHOST (COLLEN et al., 2018) e a arquitetura *HomeNetRescue (HNR)*.

Sharma et al. (2019) propõe uma arquitetura chamada *SHSec*, que busca prevenir ataques *Denial of Service (DoS)/Distributed Denial of Service (DDoS)*, solucionar problemas de violação de segurança em assistentes de voz digital e também detectar *malwares*.

O projeto de pesquisa europeu *Safe guarding home IoT environments with personalised realtime risk control* (GHOST) (COLLEN et al., 2018) define uma arquitetura conceitual incorporada a um dispositivo inteligente, adaptado para redes domésticas, que objetiva realizar uma análise avançada do fluxo de dados e classificá-los em perfis de usuários e

dispositivos, os quais são utilizados em uma avaliação automatizada de riscos em tempo real (COLLEN et al., 2018).

Ao propor soluções para os problemas de segurança e a fim de lidar com um ambiente mais dinâmico, diversos trabalhos sugerem integrar o paradigma *Software Defined Networking (SDN)* nas soluções para redes domésticas. Hafeez, Ding e Tarkoma (2017) propõem uma arquitetura que disponibiliza soluções de segurança como serviço, orientada a nuvem e faz uso de SDN para melhorar o monitoramento, a segurança e o gerenciamento da rede. Alves et al. (2018) apresentam a arquitetura *HomeNetRescue (HNR)*, que utiliza SDN para realizar o gerenciamento autônomo de redes domésticas.

Ao integrar o SDN nas soluções, a rede deixa de ser estática e passa a ser programável, tornando-se mais dinâmica e ágil, portanto, é capaz de abstrair o controle do encaminhamento, permitindo ajustes nos fluxos do tráfego de toda a rede, atendendo às necessidades de constante mudança. Dessa forma, possibilita aos administradores obterem vantagens para configurar dinamicamente os caminhos de rede, avaliar o desempenho e realizar diagnósticos que possam contribuir para alcançar um maior nível de segurança.

As propostas apresentadas anteriormente (HAFEEZ; DING; TARKOMA, 2017), (HAFEEZ; DING; TARKOMA, 2017), (ALVES et al., 2018) exibem avanços no que tange a segurança de um ambiente residencial, no entanto, elas carecem de estudos mais específicos sobre como elaborar sistemas de detecção de anomalias direcionados às redes domésticas. Existem problemas de pesquisa que não foram devidamente investigados na literatura, pois não envolvem as características e limitações de todo o ambiente. Além disso, também não descrevem como e onde eles podem ser implementados e posicionados na rede.

Adicionalmente, é importante destacar que as soluções apresentadas (HAFEEZ; DING; TARKOMA, 2017), (HAFEEZ; DING; TARKOMA, 2017), (ALVES et al., 2018) não levam em consideração a completude do tráfego de rede presente em uma residência. As soluções investigadas assumem que os sistemas irão inspecionar somente tráfego IoT, contudo, desconsideram que os ambientes possuem dispositivos multifuncionais como *smartphones* e computadores que podem prejudicar os sistemas de detecção. Ao propor uma solução para o ambiente residencial é necessário pensar de que forma a detecção de anomalias contribui para a segurança do ambiente e quais técnicas podem ser utilizadas, bem como analisar os efeitos da aplicação do paradigma *SDN* frente a segurança em redes domésticas.

Portanto, este trabalho apresenta a **FamilyGuard**, uma arquitetura que fornece uma forma de definir e implantar mecanismos que atendam aos requisitos de segurança de um ambiente de casa inteligente. FamilyGuard usa o paradigma SDN para analisar e gerenciar fluxos de rede doméstica proporcionando flexibilidade ao lidar com questões de segurança, como monitoramento de tráfego para identificar e mitigar ameaças. A arquitetura usa algoritmos de aprendizado de máquina (ML) para identificar comportamentos anômalos

com base no tráfego de rede para fornecer proteção adicional aos ambientes residenciais em termos de segurança da informação.

1.1 Motivação

Com o crescimento dos crimes virtuais, a Segurança da Informação é cada vez mais relevante para os ambientes residenciais e organizações. No Brasil, foram registradas mais de 8,4 bilhões de tentativas de ataques cibernéticos durante o ano de 2020 (LABS, 2020). Apesar do número de tentativas ser extremamente alto, existe uma preocupação adicional com o nível de sofisticação das ferramentas e tecnologias que são utilizadas por cibercriminosos para desenvolver ataques otimizados e com maior chance de sucesso.

Desta forma, existem grandes desafios para garantir a confidencialidade, disponibilidade e integridade dos dados. A cada dia surgem novas vulnerabilidades e, consequentemente, existe a probabilidade de acontecer ataques aos ambientes residenciais. Um exemplo é a *botnet* Mirai, identificado pela primeira vez em agosto de 2016, por um grupo de pesquisa de segurança chamado *MalwareMustDie*. O Mirai vasculha a Web em busca de dispositivos domésticos inteligentes que tenham nomes de usuário e senhas padrão e, em seguida, alista os dispositivos em ataques que lançam tráfego indesejado em um alvo online (KOLIAS et al., 2017).

Portanto, a principal motivação do presente trabalho é apresentar uma arquitetura de segurança que ajude a transformar o ambiente residencial estático em um ambiente dinâmico. Em outras palavras, a FamilyGuard fornecerá meios para lidar com mudanças no ambiente e responder a ameaças. Ademais, é um incentivo para o presente trabalho fornecer uma arquitetura acessível no que tange ao valor financeiro para mitigar riscos e melhorar a privacidade do ambiente residencial, levando em consideração todos os dispositivos presentes e suas limitações.

1.2 Hipótese

Considerando que as redes domésticas diferem das redes de computadores convencionais e possuem aspectos como o tráfego de dados pessoais, a heterogeneidade de dispositivos, aplicações inteligentes e sistemas avançados de automação, acredita-se que o desenvolvimento e avaliação de sistemas de detecção de anomalias para redes domésticas devem obedecer as características e limitações de tais ambientes. Além disso, o uso de sistemas especializados de detecção de anomalias permite que diferentes tipos de ameaças orientadas aos ambientes residenciais sejam devidamente detectadas e mitigadas.

1.3 Objetivos e Desafios da Pesquisa

Nas seções anteriores foram apresentadas discussões sobre os desafios de segurança enfrentados em ambientes domésticos, a importância de manter os dados e as informações que são geradas de forma segura e a necessidade de prever possíveis ataques a rede. Dessa maneira, este trabalho busca, através de técnicas de aprendizado de máquina, demonstrar que a detecção de anomalias contribui para segurança em redes domésticas. Nesse contexto, o principal objetivo deste trabalho é aprimorar o nível de segurança do ambiente residencial através da arquitetura FamilyGuard para detecção de anomalias baseada na tipificação do comportamento do tráfego da rede. Os objetivos específicos são listados a seguir:

- ❏ Gerar bases de dados que possam representar o ambiente residencial através dos fluxos de rede.
- ❏ Definir e implementar os componentes de uma arquitetura de segurança para o ambiente residencial.
- ❏ Avaliar os resultados dos modelos de detecção de anomalias em cenários experimentais.
- ❏ Validar a arquitetura através de uma prova de conceito em um ambiente residencial.

1.4 Requisitos Funcionais

Os seguintes requisitos são considerados durante o processo de elaboração da arquitetura FamilyGuard:

- ❏ A arquitetura não deve causar impacto na estrutura dos ambientes residenciais e ter uma implantação simplificada.
- ❏ Utilizar dispositivos de hardware que tenham baixo custo aquisição
- ❏ Usar modelos de aprendizado de máquina para detectar ameaças no ambiente doméstico.
- ❏ Capacidade de combinar modelos e técnicas de aprendizado de máquina

1.5 Organização da Tese

Este documento está estruturado da seguinte forma:

- ❏ Capítulo 2: define os termos e conceitos usados ao longo deste trabalho e revisa o progresso atual na compreensão do contexto doméstico, e os seus desafios;

- ❏ Capítulo 3: apresenta as principais características e a evolução atual das soluções que buscam garantir um maior nível de segurança;
- ❏ Capítulo 4: apresenta os aspectos conceituais da arquitetura proposta neste trabalho bem como as suas camadas e os serviços utilizados para detecção de anomalias;
- ❏ Capítulo 5: apresenta as bases de dados elaboradas, os experimentos e resultados; e
- ❏ Capítulo 6: traz as considerações finais do trabalho, dificuldades e os próximos passos da pesquisa.

Capítulo 2

Fundamentação Teórica

Este capítulo analisa termos e conceitos usados ao longo deste trabalho. A Seção 2.1 apresenta o paradigma IoT e suas características. A Seção 2.2 expõe definições de segurança e apresenta os tipos de ataque, medidas de proteção e desafios presentes no ambiente residencial. A Seção 2.4 descreve os conceitos de aprendizado de máquina, detecção de anomalias, algoritmos e técnicas que pode ser utilizado para prover uma segurança adicional para o ambiente residencial. Por fim, a Seção 2.5 descreve o paradigma SDN que torna o ambiente residencial mais dinâmico para lidar com os desafios que surgem no ambiente.

O ambiente residencial era desprovido de tecnologias, mas com a evolução de dispositivos e sistemas surge o termo "Casas Inteligentes", que pode ser definido sob duas perspectivas: a primeira,concerne a residências equipadas com Tecnologias da Informação e Comunicação (TIC), incluindo aparelhos conectados e que podem ser monitorados e controlados remotamente para atender às necessidades dos moradores. A segunda, em uma escala mais ampla, concerne a casas e outros edifícios e construções inteligentes, como elementos de sistemas de energia conectados de forma flexível e interativa. Essas duas perspectivas são vistas em termos de utilidade para as próprias famílias e para o sistema elétrico como um todo (GRAM-HANSSEN; DARBY, 2018).

No ambiente doméstico, a "inteligência" pode integrar dispositivos e serviços elétricos como: aquecimento, iluminação, segurança, geração fotovoltaica e carregamento de veículos elétricos, que são controlados remotamente pelos ocupantes ou algum outro agente. Além disso, sensores e atuadores também podem obter e aplicar conhecimento sobre uma casa, operando de forma independente da ação humana direta. Os dispositivos elétricos presentes nas residencias têm o potencial de auxiliar no gerenciamento da rede. Dessa forma, as casas inteligentes podem promover eficiência energética (DARBY, 2017).

Existem questionamentos sobre quando e quais tecnologias são necessárias para qualificar uma casa como inteligente. Uma TV inteligente ou um *smartphone* podem ser vistos como uma forma de classificar uma casa como inteligente, pois permitem a comunicação entre a casa com o mundo exterior, no entanto, para os objetivos deste trabalho,

um alto nível de conectividade de dispositivos dentro e fora do ambiente residencial são vistos como relevantes para definir se uma casa pode ser chamada de "inteligente". A definição apresentada aqui está associada a fatores físicos e operacionais e presume que as funcionalidades estão além dos limites comuns de uma casa tradicional (DAS; COOK, 2005).

Uma casa inteligente, portanto, é composta por uma rede de comunicações que conectam sensores, aparelhos, atuadores e outros dispositivos para permitir o monitoramento e controle remoto por ocupantes e outros, com o objetivo de fornecer serviços frequentes aos residentes e ao sistema elétrico. Assim, conclui-se que tais conceitos estão intimamente ligados ao termo Internet das Coisas (IoT) devido ao fato de prover facilidades para os ocupantes do ambiente residencial e interconectar objetos, sensores e atuadores.

2.1 Internet das Coisas

A IoT é um paradigma tecnológico criado como uma rede global de máquinas e dispositivos capazes de interagir entre si (LEE; LEE, 2015). É reconhecida como uma das áreas mais importantes da tecnologia do futuro e está ganhando atenção da indústria.

A adoção dessa tecnologia está ganhando força rapidamente à medida que as pressões tecnológicas, sociais e competitivas forçam as empresas a inovar e se transformar (LEE; LEE, 2015). Conforme apontado por Atzori, Iera e Morabito (2010) a IoT causa impacto na vida cotidiana e no comportamento dos usuários. Do ponto de vista dos usuários, os efeitos mais óbvios da introdução da IoT serão visíveis nas atividades desempenhadas no trabalho e no ambiente doméstico.

Apesar do crescimento acelerado, a IoT é vulnerável a ataques por várias razões. Primeiro, seus componentes passam a maior parte do tempo sem vigilância, portanto, é fácil atacá-los fisicamente. Segundo, a maioria das comunicações é sem fio, o que torna a escuta de tráfego extremamente simples. Por fim, a maioria dos componentes da IoT é caracterizada por baixos recursos em termos de energia e recursos de computação e comunicação, consequentemente, esses componentes não podem implementar esquemas complexos que suportam segurança (ATZORI; IERA; MORABITO, 2010).

Com toda essa evolução é fundamental que os mecanismos de segurança consigam acompanhar o avanço das tecnologias para proteger as informações que são trafegadas entre esses dispositivos e aplicações. Para entender os mecanismos utilizados para proteger o ambiente residencial, a Seção 2.2 apresenta o conceito de segurança, para que seja possível entender os mecanismos utilizados para proteger o ambiente residencial.

2.2 Conceito de Segurança

Existem diversas definições para o termo Segurança, sendo uma delas aquela apresentada como a ausência de ameaças aos valores adquiridos (WOLFERS, 1952). Entretanto, essa definição é genérica. A aplicação do conceito de segurança varia com base em seu domínio, por exemplo, segurança corporativa, segurança da informação e segurança residencial. Em (BALDWIN, 1997), o autor apresenta questões que colaboram para definir e aplicar a segurança em um domínio específico:

- Segurança para quem? O autor destaca que a segurança deve especificar o objeto, entidade ou indivíduo referente para que seja compreensível, a exemplo na segurança da informação em que o objeto que se deseja proteger são os dados e informações gerados por usuários, sistemas e aplicativos.

- Segurança para quais valores? Esses valores são ações que podem garantir a segurança de um determinado objeto. A título de exemplo, na segurança da informação é possível definir valores como (ANDRESS, 2014):

  - Confidencialidade: medidas para garantir que apenas pessoas autorizadas tenham permissão para acessar as informações.
  - Integridade: garantir que as informações permaneçam inalteradas durante o armazenamento, a transmissão e o uso, sem envolver modificações nas informações.
  - Disponibilidade: medidas para proteger os componentes do sistema e garantir que a informação esteja disponível.

- Quanta segurança? Determinar o que é suficiente para proteger um ativo é desafiador, pois essas definições dependem do ambiente. Por exemplo, uma casa que possua somente um sensor de temperatura detém requisitos de segurança diferentes de uma casa com computadores e *smartphones* conectados à Internet.

- De quais ameaças? Para proteger os ativos de forma efetiva, é necessário entender as ameaças de segurança no domínio estabelecido. A falta de conhecimento dessas ameaças pode causar dificuldades na aplicação de medidas de segurança apropriadas. O conceito de ameaça refere-se a qualquer coisa que represente perigo a um ativo.

- Por quais meios? Podem existir diversas maneiras de proteger um ativo contra uma ameaça, sendo assim, é importante avaliar o motivo por que determinadas medidas foram escolhidas.

- A que custo? A busca por segurança sempre envolve custos, ou seja, o sacrifício de outros propósitos que poderiam ter sido atingidos com os recursos dedicados

à segurança. Essa relação de custo-benefício colabora para selecionar as medidas adequadas.

- Por quanto tempo? Ao definir uma política de segurança, é necessário entender por quanto tempo ela será aplicada, para evitar que as políticas de curto prazo não entrem em conflito com as de longo prazo.

Baseado nessas perguntas é possível definir o domínio em que a segurança será aplicada e extrair características que colaboram para definir os mecanismos de segurança, os quais podem ser utilizados para mitigar ameaças e neutralizar tentativas de ataques aos ativos. Dessa forma, a Seção 2.3 explora o ambiente residencial como um domínio para que seja possível identificar as propriedades e definir mecanismos de segurança para o ambiente.

2.3 Ambiente Residencial

É importante compreender o ambiente residencial para definir ou melhorar a sua segurança. Diante disso, a Subseção 2.3.1 descreve possíveis ataques. A Subseção 2.3.2 caracteriza as medidas de proteção que podem ser encontradas e, por fim, são detalhados os desafios de segurança na seção 2.3.3.

2.3.1 Tipos de ataques

Em uma casa inteligente existem muitas vulnerabilidades que podem ser exploradas e nenhum mecanismo defensivo é capaz de lidar com todas essas ameaças existentes (SAXENA; SODHI; SINGH, 2017). A seguir são descritos ataques que podem ocorrer em uma rede doméstica inteligente:

- *Eavesdropping*: um invasor monitora o tráfego de dados entre dois nós da rede doméstica. Com isso, o objetivo é descobrir qual é a transferência de dados que está ocorrendo e qual é o padrão dos dados. As informações capturadas pelo invasor podem ser confidenciais e isso pode causar um perigo para a rede (KARLOF; WAGNER, 2003).

- *Masquerading*: um invasor usa identidade falsa com o objetivo de obter acesso não autorizado à rede através da identificação legítima de acesso. O principal objetivo nesse ataque é obter dados secretos ou adquirir serviços autênticos (KARLOF; WAGNER, 2003).

- *Replay Attack*: um invasor intercepta e repete uma transmissão de dados maliciosa entre dois nós, sendo seu objetivo conseguir a mesma autenticidade que um nó original possui, para que ele possa monitorar facilmente dados (SAXENA; SODHI; SINGH, 2017).

- *Message Modification*: um invasor tenta capturar dados transmitidos entre dois nós e modifica esses dados para que não exista autenticidade da mensagem, sendo que essas modificações podem conter códigos maliciosos (KARLOF; WAGNER, 2003).

- *DoS*: nesse tipo de ataque o invasor ataca um nó da rede, sobrecarregando e enviando continuamente mensagens para que o destinatário fique ocupado e indisponível para processar as solicitações de outros nós (KARLOF; WAGNER, 2003).

- *Man In the Middle (MITM)*: um invasor ataca o meio de comunicação, interceptando o que está ocorrendo entre duas partes e, como resultado, elas se comunicam acreditando que ninguém está ouvindo sua comunicação (SAXENA; SODHI; SINGH, 2017).

- *Sinkhole*: o intruso atrai os nós vizinhos com informações de roteamento forjadas e, em seguida, realiza o encaminhamento seletivo ou altera os dados que passam por ele. Este ataque pode acorrer sobre o protocolo 6LoWPAN para Internet das Coisas, que pode estar presente em diversas *smart-homes* (SONI; MODI; CHAUDHRI, 2013).

- *Sybil Attack*: um invasor obtém o endereço IP e o endereço MAC de um usuário legítimo, roubando a identidade do usuário real. *Sybil* é forma avançada de *Impersonate Attack*. Neste ataque um atacante rouba várias identidades e executa atividades maliciosas na rede (SAXENA; SODHI; SINGH, 2017).

- *De-Synchronization*: o invasor busca modificar os sinalizadores de controle e o número de sequência para alterar os pacotes ou mensagens que ocorrem entre dois dispositivos ou pontos finais. Devido a isso, limita o usuário legítimo a enviar e receber dados ou mensagens, podendo gerar uma situação de impasse que requer muita energia dos dispositivos (SAXENA; SODHI; SINGH, 2017).

A seção 2.3.2 apresenta medidas de proteção que podem ser implementadas em ambientes residenciais para evitar estes tipos de ataques, no entanto, o novo cenário com dispositivos inteligentes traz consigo diversos desafios apresentados na Subseção 2.3.3. Este trabalho busca auxiliar as medidas de segurança existentes, implementando um ambiente mais dinâmico e fornecendo um serviço de detecção de anomalias como uma medida de proteção adicional ao ambiente residencial.

2.3.2 Medidas de proteção

Para analisar as medidas de proteção utilizadas atualmente no contexto residencial é necessário analisar as informações do espaço social e suas relações. Existem dois atores que podem prover a segurança dos dados para o ambiente doméstico, que são: os internos e os externos. Os interessados internos envolvem as pessoas que vivem no ambiente e

os externos são representados pelos prestadores de serviços e fornecedores que desejam garantir uma relação comercial, como os *Internet Service Providers (ISPs)*.

Nthala e Flechais (2018) apresentam práticas de segurança para o contexto residencial, incluindo o uso de tecnologias relacionadas a segurança e comportamentos das pessoas que vivem no ambiente. As seguintes medidas foram descritas: *firewall*, *backup* dos dados, criptografia, *adblocker*, autenticação (senha, biometria e identificação por dois fatores), gerenciador de senhas, *Virtual Private Network (VPN)*, exclusão de softwares e aplicativos não utilizados, remoção de *cookies*, histórico do navegador, aplicação de *patches* de correção e atualizações e hábitos cautelosos como visitar apenas sites conhecidos e verificar se os sites possuem conexão segura.

Existem algumas práticas que podem ser aplicadas tanto por interessados internos quanto externos como o controle dos pais, atualização de *firmware* do roteador e monitoramento de tráfego de entrada e saída. Essas práticas e medidas de segurança podem contribuir para mitigar problemas enfrentados no ambiente residencial, sendo que, no entanto, não é o suficiente para manter um ambiente seguro (KARIMI; KRIT, 2019).

A cada dia surgem novas ameaças e, consequentemente, existe a probabilidade de surgirem diferentes ataques à rede da sua residência. Novos dispositivos e aplicações são lançados no mercado todos os dias com falhas de segurança (SPINELLIS; KOKOLAKIS; GRITZALIS, 1999) (HE, 2002). Mediante o exposto, é fundamental saber identificar e corrigir pontos de vulnerabilidade e tudo isso exige estudos e soluções mais acuradas para a segurança em redes domésticas (ALI et al., 2017).

2.3.3 Desafios

Gerenciar a segurança no contexto residencial é uma tarefa complexa, pois existe uma grande quantidade de informações pessoais e novos dispositivos surgem todos os dias (LEE et al., 2014). Existem algumas particularidades nos dispositivos inteligentes que impedem o uso de mecanismos de segurança padrão, que são adotados nas redes convencionais como:

- ❏ Restrições de recursos: grande parte dos dispositivos domésticos inteligentes são projetados para funcionar com hardware de baixo consumo e tamanho reduzido e por isso, restringe o desempenho de computação e os recursos de armazenamento.

- ❏ Protocolos de comunicação heterogêneos: existem diversos protocolos utilizados para interconectar os dispositivos em uma *smart-home*, que exigem o uso de um *gateway* intermediário, e isso dificulta a implementação de soluções de segurança de ponta a ponta entre os dispositivos e os aplicativos de internet.

- ❏ Comunicações não confiáveis: os pacotes podem ser danificados devido a erros do canal de transmissão ou podem ser descartados em nós altamente congestionados, pois as retransmissões e os algoritmos de tratamento de erros requerem grande

sobrecarga que não são toleráveis em dispositivos de redes de baixa potência (SEN, 2010).

- ❏ Restrições de energia: os dispositivos inteligentes possuem recursos de energia limitados para as suas comunicações, armazenamento e cálculos. Tais restrições fornecem vulnerabilidades a ataques de esgotamento de recurso que forçam os dispositivos a permanecer ativos.

- ❏ Acesso físico: os dispositivos ficam acessíveis fisicamente o tempo todo, tornando-se alvos fáceis. Um invasor que tenha acesso físico aos dispositivos pode extrair informações como as chaves de criptografia predefinidas e outras informações confidenciais.

Portanto, para melhorar a segurança nas residências inteligentes, são necessárias soluções que consigam adaptar automaticamente a heterogeneidade dos dispositivos e protocolos.

Os mecanismos de segurança de rede existentes, como o *firewall* normalmente operam no roteador doméstico e buscam proteger a rede contra ataques não autorizados que estão do lado de fora da rede. Contudo, para conexões que estão dentro da rede, esses mecanismos de segurança não são efetivos, o que permite que dispositivos infectados ataquem facilmente outros dispositivos dentro da rede (SERROR et al., 2018). Baseado nisso, diversas pesquisas estão buscando fornecer proteção adicional para as redes domésticas inteligentes por meio de técnicas de aprendizado de máquina para detectar comportamentos anômalos (MAHDAVINEJAD et al., 2018).

2.4 Inteligência Artificial

Russel, Norvig et al. (2013) descrevem que o campo da Inteligência Artificial, ou IA, busca o entendimento da inteligência e a construção de entidades inteligentes. A Inteligência Artificial está preocupada com o comportamento inteligente, que envolve percepção, raciocínio, aprendizado, comunicação e atuação em ambientes complexos (NILSSON; NILSSON, 1998).

O Teste de Turing, proposto por Turing (2009), foi projetado para fornecer uma definição satisfatória de inteligência. Um computador passa no teste se um interrogador humano, após fazer algumas perguntas escritas, não puder dizer se as respostas escritas vêm de uma pessoa ou de um computador. Um computador para passar no teste precisaria possuir as seguintes capacidades: processamento de linguagem natural, para permitir que ele se comunique com sucesso; representação do conhecimento, para armazenar o que ele sabe ou ouve; raciocínio automatizado, para usar as informações armazenadas visando responder perguntas e tirar novas conclusões; e aprendizado de máquina, para se adaptar a novas circunstâncias e detectar e extrapolar padrões (RUSSEL; NORVIG et al., 2013).

Neste trabalho será utilizado o aprendizado de máquina, caracterizado por usar algoritmos capazes de aprender a partir de exemplos. Seus princípios são utilizados nas demais áreas de AI e podem ser aplicadas na resolução de problemas em diversas áreas.

2.4.1 Aprendizado de Máquina

O aprendizado de máquina é uma área da AI que tem como objetivo o desenvolvimento de sistemas capazes de aprender a partir de dados, usando algoritmos e técnicas que automatizam as soluções de problemas complexos e difíceis de programar através dos métodos de programação convencionais (REBALA; RAVI; CHURIWALA, 2019).

Ao utilizar o aprendizado de máquina, é possível obter informações sobre estrutura e padrões em grandes quantidades de dados e criar modelos que aprendam com os conjuntos de dados existentes para prever resultados ou comportamentos. A seguir são descritos alguns exemplos de problemas de aprendizado de máquina (REBALA; RAVI; CHURIWALA, 2019):

- Classificação: classificar algo em uma ou mais categorias (classes). A título de exemplo, identificar se um *e-mail* é um *spam* ou não.
- Agrupamento: busca dividir os dados em alguns grupos, de modo que os pontos dentro de um grupo tenham propriedades em comum. Diferentemente da classificação, o número de grupos pode não ser conhecido antecipadamente.
- Regressão: baseado em dados históricos são construídos modelos usados para prever um valor futuro. Por exemplo, a demanda por um produto específico durante a temporada de férias.

O processo de construção de um modelo de aprendizado de máquina pode ser dividido em sete etapas (JAMAL et al., 2018): a coleta de dados; a preparação dos dados que verifica as informações coletadas e avalia a sua qualidade, também realiza ajustes se for necessário e aplica uma normalização nos dados; escolha do modelo de aprendizado; treinamento; avaliação que tem como objetivo testar o modelo com informações não utilizadas na fase de treinamento para verificar se o modelo realmente foi capaz de aprender; ajuste dos parâmetros para melhorar a qualidade e a eficiência do modelo que está sendo utilizado; por fim a fase de predição que disponibiliza o modelo para ser usado.

Dependendo da situação, os algoritmos de aprendizado de máquina funcionam usando mais ou menos intervenção humana. Os quatro principais modelos utilizados são descritos a seguir: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado semi-supervisionado e aprendizado de reforço (REBALA; RAVI; CHURIWALA, 2019).

- Aprendizado supervisionado: é utilizado um conjunto de treinamento com os dados rotulados. Dessa forma, os algoritmos buscam aprender a prever a saída apropriada

para um determinado vetor de entrada. Quando os rótulos de destino consistem em um número limitado de categorias discretas são conhecidas como tarefas de classificação (BENGIO; GOODFELLOW; COURVILLE, 2015). Caso os rótulos de destino sejam compostos de uma ou mais variáveis contínuas são conhecidas como tarefas de regressão.

- ❏ Aprendizado não-supervisionado: não são necessários rótulos para o conjunto de treinamento. Dada uma quantidade de dados, o algoritmo busca identificar tendências de similaridade nos dados.

- ❏ Aprendizado semi-supervisionado: fica entre o treinamento supervisionado e não supervisionado. O algoritmo recebe um conjunto de dados, no qual, apenas alguns dados são rotulados. O algoritmo pode usar técnicas de agrupamento para identificar grupos dentro do conjunto de dados fornecido e usar os poucos pontos de dados rotulados.

- ❏ Aprendizado por reforço: lida com o problema de aprender uma ação ou determinada sequência de ações a serem executadas para uma dada situação, que pode receber recompensas ou penalidades, onde com o objetivo final é de maximizar a recompensa total.

Neste trabalho será utilizado o aprendizado não-supervisionado para detectar anomalias presentes no ambiente residencial, pois não é necessário ter uma amostra do tráfego malicioso para que ele possa ser identificado. Dessa forma, existe a possibilidade que os modelos consigam identificar ameaças que ainda possam surgir no futuro.

2.4.2 Detecção de Anomalias

A detecção de anomalias se refere ao problema de encontrar padrões que não estão em conformidade com o comportamento esperado (CHANDOLA; BANERJEE; KUMAR, 2009). Bakar et al. (2016) discutem que a detecção de anomalias é uma área de pesquisa popular com várias aplicações, principalmente no ambiente doméstico inteligente. Os recursos exclusivos dos sensores domésticos inteligentes desafiam os métodos tradicionais de análise de dados. No entanto, o reconhecimento de atividades anômalas ainda é incipiente para *smart homes* quando em comparação com outros domínios, como detecção de defeitos na fabricação de produtos, processamento de imagens médicas, etc.

É possível definir quatro categorias de detecção de anomalias para o contexto residencial, que são descritas a seguir:

- ❏ Comportamento de pessoas: são sistemas de detecção que utilizam dados gerados pelas pessoas através de sensores, como o monitoramento de frequência cardíaca e pressão sanguínea ou uma interação da pessoa com o ambiente residencial, por

exemplo, sensores de movimentos. Zhu, Sheng e Liu (2015) apresentam uma estrutura para detectar anomalias comportamentais na vida diária humana usando sensores vestíveis que podem ajudar nos cuidados de pessoas idosas ao identificar uma queda ou sonambulismo. Novák, Biňas e Jakab (2012) apresentam um método que utiliza sensores para detectar comportamento anômalo, como longos períodos de inatividade, presença incomum e mudanças no cotidiano dos padrões de atividade.

- ❏ Comportamento dos dispositivos: engloba a detecção de anomalias que utiliza dados dos dispositivos como *smartphones* e computadores. Podem ser utilizadas informações como o uso da memória, processador e o dispositivo de armazenamento.

- ❏ Comportamento do ambiente: baseada nas informações capturadas pelos sensores que possuem a responsabilidade de monitorar informações que estão diretamente envolvidas com o ambiente residencial, por exemplo, sensor de umidade, energia, temperatura, etc. Ul Alam et al. (2016), por exemplo, definem uma abordagem analítica que ajuda a detectar anomalias no uso de energia.

- ❏ Comportamento da rede: anomalias identificadas através da comunicação realizada pelos dispositivos IoT e não IoT presentes na residencia. Pode ocorrer por meio de uma análise de pacotes ou fluxos da rede. Nguyen-An et al. (2019) apresentam uma abordagem para para detectar anomalias baseado no tráfego IoT em ambiente doméstico inteligente.

Este trabalho busca melhorar a segurança da informação usando o comportamento da rede para identificar anomalias. Nesse cenário, as anomalias podem ser provocadas por diferentes problemas, que logram ser classificadas em duas categorias. A primeira engloba anomalias onde não há a presença de agentes maliciosos e a segunda abrange anomalias causadas por ataques, que normalmente são elaborados por agentes que visam violar a segurança da rede de um determinado alvo.

A primeira categoria de anomalias pode ser caracterizada por um congestionamento da rede e consequentemente um aumento brusco de tráfego em um determinado ponto, causando dificuldade de processamento dos pacotes, os quais são entregues com atraso ou são descartados. A queda de um enlace, por exemplo, pode dificultar o escoamento do tráfego que chega a determinados pontos da rede. *Bugs* de softwares de roteamento podem não estar preparados para lidar com pacotes que chegam com determinados problemas. Erros em configurações de *firewalls*, podem barrar a entrada de uma grande quantidade de pacotes, causando mudanças não esperadas no comportamento da rede.

A segunda categoria de anomalias é representada por ataques oriundos de agentes maliciosos, ou seja, trata-se de uma intrusão. A intrusão engloba uma ação ou conjunto de ações que objetivem comprometer a integridade, confidencialidade e disponibilidade de um determinado sistema.

Um exemplo são os ataques de negação de serviço (DoS), que visa tornar um serviço da empresa inoperante através de uma sobrecarga. Nesses ataques, o atacante envia uma grande quantidade de requisições ao servidor de forma que todos os recursos de memória e processamento sejam consumidos. Outro exemplo desta categoria são os worms que possuem a capacidade de se espalhar e executar cópias em máquinas remotas ao longo da rede, conseguindo infectar muitos alvos em uma rede. Por fim, o último exemplo trata-se de escaneamento de portas (*port scan*). Apesar de ser uma técnica usada por administradores de rede para executar tarefas para gerenciar a rede, também pode ser utilizado alguém que deseja descobrir vulnerabilidades nas redes escolhidas como alvos.

A detecção de intrusão pode ser classificada em duas categorias (PATCHA; PARK, 2007) que são descritas nas subseções 2.4.2.1 e 2.4.2.2.

2.4.2.1 Detecção baseada em assinaturas

A detecção baseada em assinatura é uma técnica que depende de um conjunto predefinido de características que são usadas para produzir assinaturas, que serão armazenadas em uma base de dados. A principal vantagem deste método é que os ataques presentes na base de dados são detectados de maneira efetiva, com baixas taxas de falsos positivos. Outra vantagem é que o sistema pode proteger a rede assim que for implantado, pois não há período de treinamento ou aprendizado a respeito do funcionamento da rede.

No entanto, possui algumas desvantagens, a primeira é que não possuem a capacidade de detectar ataques com assinaturas desconhecidas da sua base. A segunda desvantagem é a necessidade de constante atualização no conjunto de assinaturas. Por último, uma desvantagem que dificulta a detecção dos ataques é que grande parte das assinaturas são muito específicas. Dessa forma, ataques variantes não são detectados.

O *Snort* (CHAKRABARTI; CHAKRABORTY; MUKHOPADHYAY, 2010) é um sistema de detecção de intrusão *open source* capaz de realizar análise de tráfego e captura de pacotes. Sua detecção é baseada em assinaturas utilizando uma linguagem flexível de regras para analisar o tráfego coletado. Estas assinaturas/regras são atualizadas diariamente pela sua equipe de desenvolvimento.

2.4.2.2 Detecção baseada em comportamento

A detecção baseada em comportamento cria um perfil fundamentado em características das atividades consideradas normais. Após uma investigação das características é definido um perfil de normalidade. Posteriormente, qualquer atividade que se desvie do perfil de normalidade é tratada como uma possível anomalia.

A principal vantagem deste método é a capacidade em detectar anomalias desconhecidas. Entretanto, existem desvantagens como a alta porcentagem de falsos alarmes, pois existem variações na rede que podem ser encaradas como desvio de comportamento. Outra desvantagem é o período de treinamento utilizado para definir o perfil de normalidade.

Uma das soluções utilizadas para criação dos sistemas de detecção baseados no comportamento normal da rede é a técnica de aprendizado, que consiste em aprender sobre o histórico de dados reais coletados da rede para gerar os perfis de tráfego normais. Dentre as técnicas de aprendizado utilizadas, é possível citar os algoritmos estatísticos e modelos de aprendizado de máquina (MAHDAVINEJAD et al., 2018).

Diversas soluções utilizam o aprendizado de máquina no contexto residencial para detectar anomalias através do monitoramento do tráfego da rede. Em (YAMAUCHI et al., 2019) é apresentado um método para detectar ataques na rede com base no comportamento do usuário e o (RAMAPATRUNI et al., 2019) busca identificar atividades anômalas que podem ocorrer em um ambiente doméstico inteligente.

Existem diversas técnicas que podem ser utilizadas para detectar anomalias, todavia os algoritmos de *One Class Classification (OCC)* (KHAN; MADDEN, 2013) têm apresentado resultados promissores para detectar ataques cibernéticos (BEZERRA et al., 2019), pois o conjunto de treinamento é representado pelo tráfego da rede sem a presença de nenhum *malware* e no momento em que a rede apresentar algum procedimento diferente do normal o OCC classifica esse comportamento como anômalo (KHAN; MADDEN, 2010).

Novas ameaças são criadas todos os dias e isso torna o OCC interessante, pois ele não precisa de um conjunto de treinamento das novas ameaças que surgem como em outras abordagens (LI et al., 2003).

Percebe-se que identificar o comportamento anômalo não é o suficiente. Além disso, é necessário agir de forma rápida e efetiva para que o ataque não se propague na rede ou cause algum dano. Afim de adquirir essa flexibilidade é possível utilizar o conceito de *SDN* (KREUTZ et al., 2015a) para controlar a rede de maneira centralizada por meio de aplicativos de software, uma breve discussão sobre SDN é apresentada na Seção 2.5.

2.4.3 Algoritmos e Técnicas

Para os experimentos de detecção de anomalias, será utilizado o método de aprendizado não-supervisionado através dos OCCs. Tais algoritmos são úteis para a detecção de anomalias, pois utilizam na fase de treinamento somente amostras do tráfego normal da rede. Dessa forma, elimina a preocupação de representar as anomalias, que trata-se de uma tarefa difícil pelo fato de que novas ameaças e vulnerabilidades são criadas todos os dias. Também será usado o método de aprendizado supervisionado para tentar classificar possíveis ameaças.

A Tabela 1 apresenta informações dos algoritmos utilizados neste trabalho. É possível observar a sigla, nome, classe, tipo, ano em que foi criado e uma referência, onde é possível encontrar informações detalhadas sobre o funcionamento de cada um deles.

Na etapa de pré-processamento é possível usar três técnicas. A primeira trata-se de normalizar os dados, para isso, é possível usar a Standard Scaler (SS), que patroniza os dados removendo a média e dividindo pela variância. A segunda refere-se a Principal

Component Analysis (PCA), usada para reduzir a dimensão dos dados, porém mantendo-se suas informações e características, como os valores de variância. Por fim, para lidar com dados desbalanceados na base de treinamento a técnica chamada *Oversampling* pode ser aplicada. Trata-se de uma técnica para aumentar a quantidade de amostras da classe com menor frequência até que a base de dados apresente uma quantidade equilibrada entre as classes da variável alvo.

Tabela 1 – Algoritmos usados nos experimentos

Nome	Tipo	Ano	Referência
Minimum Covariance Determinant (MCD)	Modelo Linear	1999	(ROUSSEEUW; DRIESSEN, 1999) (HARDIN; ROCKE, 2004)
C-Support Vector Classification (SVC)	Modelo Linear	1999	(PLATT et al., 1999)
Local Outlier Factor (LOF)	Baseado em Proximidade	2000	(BREUNIG et al., 2000)
k-Nearest Neighbor (KNN)	Baseado em Proximidade	2000	(RAMASWAMY; RASTOGI; SHIM, 2000) (ANGIULLI; PIZZUTI, 2002)
K-Nearest Neighbors	Baseado em Proximidade	2000	(RAMASWAMY; RASTOGI; SHIM, 2000) (ANGIULLI; PIZZUTI, 2002)
One-Class Support Vector Machines (OCSVM)	Modelo Linear	2001	(SCHöLKOPF et al., 2001)
Average KNN (AKNN)	Baseado em Proximidade	2002	(ANGIULLI; PIZZUTI, 2002)
Median KNN (MKNN)	Baseado em Proximidade	2002	(ANGIULLI; PIZZUTI, 2002)
Clustering-Based Local Outlier Factor (CBLOF)	Baseado em Proximidade	2004	(HE; XU; DENG, 2003)
Naive Bayes (NB)	Probabilístico	2004	(ZHANG, 2004)
Feature Bagging (FB)	Conjuntos de Valores Atípicos	2005	(LAZAREVIC; KUMAR, 2005)
Isolation Forest (IF)	Conjuntos de Valores Atípicos	2008	(LIU; TING; ZHOU, 2008)
Lightweight On-line Detector of Anomalies (LODA)	Conjuntos de Valores Atípicos	2016	(PEVNỲ, 2016)
Copula-Based Outlier Detection (COPOD)	Probabilístico	2020	(LI et al., 2020)

2.5 Redes Definidas por Software

As Redes Definidas por Software (KREUTZ et al., 2015b) são definidas como um paradigma que fornece mudanças em relação às limitações de infraestrutura das redes atuais. Um dos principais objetivos da SDN é criar uma rede ágil e flexível que suporte mudanças rápidas com base nas necessidades e demandas das aplicações.

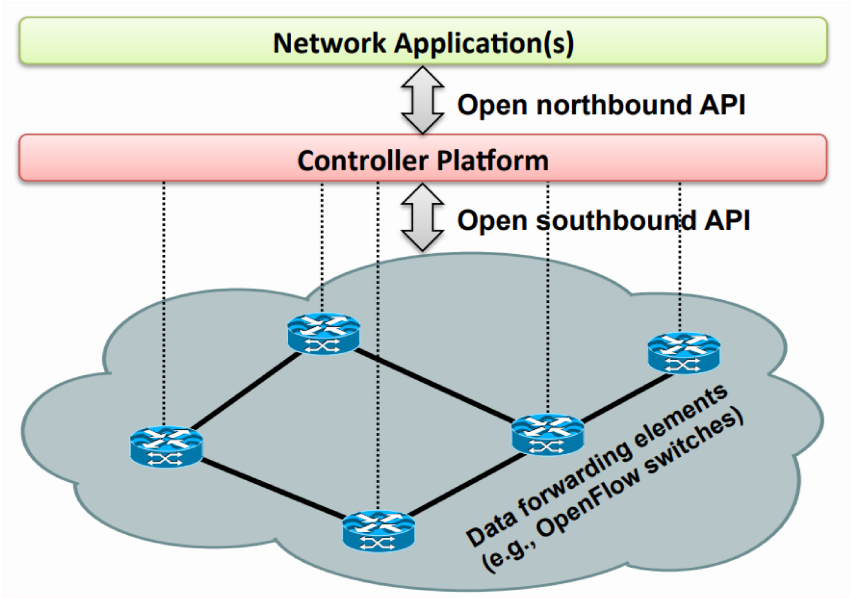


Figura 1 – Visualização simplificada de uma arquitetura SDN

Fonte: (KREUTZ et al., 2015a)

A Figura 1 apresenta uma visão simplificada da arquitetura SDN, que possui o plano de controle separado do plano de dados. Basicamente, a rede é controlada por uma plataforma de software logicamente centralizada, separada do *hardware* que é responsável por encaminhar os pacotes de dados da rede. Um controlador pode ser responsável por vários *switches*, ou seja, eles compartilham o mesmo plano de controle. Com isso, é possível alterar as regras na definição de fluxos dos *switches* a partir do plano de controle.

Na Figura 2 é possível observar a arquitetura SDN e identificar o motivo de ser considerada um paradigma de rede independente de hardware e fornecedor. O SDN possui dois componentes principais: controlador e os *switches*. Controlador SDN é responsável pelo gerenciamento de toda a rede enquanto os *switches* são responsáveis por operar com base nas instruções implantadas através do controlador SDN (RAWAT; REDDY, 2017).

O SDN é materializado através do protocolo OpenFlow, ou seja, é encarregado pela separação do plano de controle e plano de dados. O OpenFlow é um conjunto de especificações mantidas pelo Open Networking Foundation (ONF) (FOUNDATION, 2017).

O OpenFlow (MCKEOWN et al., 2008) atrai uma atenção expressiva tanto da academia quanto da indústria. Um grupo de operadores de rede, prestadores de serviços e fornecedores criaram a ONF (FOUNDATION, 2017), uma organização para promover

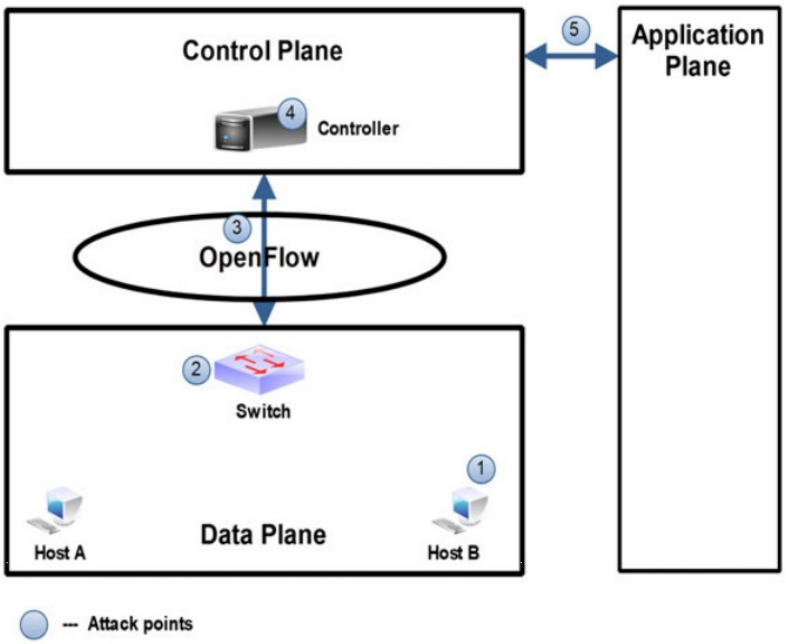


Figura 2 – Arquitetura SDN. Extraído de (DAYAL et al., 2016)

SDN e padronizar o protocolo OpenFlow. O *switch* é controlado por um elemento externo, o controlador, a comunicação entre eles é realizada através do protocolo OpenFlow (MCKEOWN, 2009).

O protocolo Openflow define algumas mensagens para realizar essa comunicação, por exemplo, PACKET_IN, PACKET_OUT e FLOW_MOD. A mensagem do tipo PACKET_IN é enviada do *switch* para o Controlador quando um pacote recebido não combina com nenhuma entrada na tabela de fluxos do *switch*. A mensagem PACKET_OUT é usada quando o Controlador envia um pacote através de uma ou mais portas do *switch*. Por fim a mensagem do tipo FLOW_MOD é enviada do Controlador para o *switch* com o objetivo de instruí-lo a adicionar um fluxo em particular na sua tabela de fluxos.

Os controladores são uma parte fundamental do *OpenFlow*, já que são responsáveis por definir a lógica de encaminhamento através das regras que irão configurar nos *switches*. Atualmente, existem diversos controladores disponíveis para o uso, por exemplo, Floodlight (Big Switch Networks, 2014), Hyperflow (TOOTOONCHIAN; GANJALI, 2010), Maestro (NG; CAI; COX, 2010), entre outros.

Com a detecção de anomalias e o paradigma SDN trabalhando em conjunto é possível gerar um ambiente que consiga suportar a heterogeneidade dos dispositivos e seus protocolos, bem como lidar com novas ameaças que possam surgir (GALEANO-BRAJONES et al., 2020) (FILHO et al., 2020).

Portanto, este trabalho utiliza o paradigma SDN para adquirir mais flexibilidade e controle do ambiente residencial. Ao identificar uma ameaça um alerta é emitido e com uma decisão rápida pode-se enviar uma regra ao controlador para que ele faça o bloqueio do dispositivo que contém a ameaça ou crie uma regra que impeça a sua propagação no

ambiente.

Capítulo 3

Trabalhos Relacionados

Este capítulo descreve na Seção 3.1 as soluções que buscam melhorar a segurança do ambiente residencial. A seção 3.2 apresenta uma discussão sobre as soluções apresentadas. Por fim, na Seção 3.3 descreve métodos utilizados na área que provaram ser eficazes para detectar anomalias.

3.1 Soluções orientadas ao ambiente residencial

O objetivo desta seção é apresentar as principais características e o progresso atual das soluções que buscam melhorar a segurança do ambiente residencial. Uma revisão sistemática da literatura foi realizada para identificar os principais trabalhos e também as lacunas de pesquisa. Primeiro, foi realizada a definição das palavras-chave, como *smart home*, *security mechanisms* e *anomaly detection*, que são amplamente utilizadas em artigos de periódicos e conferências. Tais palavras-chave foram adotadas como entrada para motores de busca como *Google Scholar*, *IEEE Xplore* e *ACM Digital Library*.

Com base nos resultados dos motores de busca foram estabelecidos os seguintes critérios para seleção de artigos: o número de citações, o fator de impacto da revista e a maturidade das conferências. Nas subseções seguintes, são apresentadas as características de cada trabalho. A Tabela 2 resume o processo sistemático de literatura, onde as publicações selecionadas estão organizadas de acordo com as seguintes características: ano de publicação, estratégia, tipo de validação, possíveis ameaças, objetivos de segurança comprometidos, contramedidas, técnicas utilizadas e a existência de recursos SDN.

3.1.1 *Customized Blockchain-Based (CBB)*

A solução busca evoluir os aspectos de segurança para sistemas domésticos inteligentes usando *blockchain*, que pode ser usada como tecnologia para proteger dispositivos IoT. Também propõe o mapeamento de atributos de uma casa inteligente, mapeamento que permita personalizar a solução para atender os requisitos de segurança de casas in-

teligentes baseadas em IoT. A arquitetura foi implementada e testada para melhorar a integridade, confidencialidade, disponibilidade, autorização e privacidade do ambiente doméstico (AMMI; ALARABI; BENKHELIFA, 2021).

3.1.2 *Urban Patrol*

Pretende mitigar ataques a uma rede doméstica inteligente e, assim, criar um sistema de patrulha. O projeto proposto consiste em *proxies* de dispositivos individuais. O *hub* coleta dados da camada de aplicativos, de cada dispositivo IoT, de cada *proxy* de dispositivo na rede. Os dados coletados são usados como entradas para treinar o *hub* central para detectar intrusões na rede doméstica inteligente. A arquitetura usa XGBoost para classificar e detectar anomalias e também implementa um mecanismo de acesso baseado em funções para validar a autorização e os controles de acesso (MASCARENHAS et al., 2021).

3.1.3 Towards Secure and Privacy-Preserving (TSP2)

Fornece uma arquitetura que pode dar suporte a vários casos de uso doméstico inteligente habilitados para IoT, com um nível especificado de segurança e preservação de privacidade. Foi apresentado um estudo para fornecer informações sobre a adequação de vários algoritmos criptográficos, para uso na proteção dos dispositivos IoT usados na arquitetura proposta. Os resultados obtidos mostram que muitos algoritmos modernos de criptografia simétrica leve, como CLEFIA e TRIVIUM, são otimizados para implementações de hardware e podem consumir até 10 vezes mais energia do que as técnicas legadas quando implementados em software (ABU-TAIR et al., 2020).

3.1.4 *Extended Generalized Role Based Access Control (EGRBAC)*

Apresenta um modelo de controle de acesso para casas inteligentes. Os autores fornecem uma definição formal para controle de acesso baseado em funções generalizadas estendidas (EGRBAC) e apresentam os recursos da solução com um caso de uso. Uma demonstração de prova de conceito utilizando *AWS-IoT Greengrass* é discutida no apêndice. EGRBAC é uma primeira etapa no desenvolvimento de uma família abrangente de modelos de controle de acesso para casa inteligente IoT (AMEER; BENSON; SANDHU, 2020).

3.1.5 *Secure Smart Home Environment (SHSec)*

A arquitetura *SHSec* utiliza o paradigma SDN com o controlador POX dentro do ambiente residencial. Três estudos de caso demonstram que a arquitetura pode colaborar com

o ambiente residencial, sendo que: o primeiro é em relação a privacidade dos usuários; o segundo é o uso da arquitetura para prevenir ataques DoS/*DDoS*; e o terceiro demonstra que a arquitetura pode solucionar problemas de violação de segurança em assistentes de voz digital. Foi avaliada a sobrecarga e o desempenho da SHSec utilizando a taxa de perda de pacotes, o tempo de configuração do fluxo e o tempo de resposta do controlador. Também foram apresentados testes para detectar *malware* utilizando *Conditional Probability Distribution (CPD)* no conjunto de dados Sherlock. Esse conjunto de dados contém dez bilhões de registros, contendo dados de trinta usuários coletados em um período de dois anos, e vinte usuários adicionais por dez meses (SHARMA et al., 2019).

3.1.6 *HomeNetRescue (HNR)*

A arquitetura HomeNetRescue apresenta um serviço que utiliza SDN para realizar o gerenciamento autônomo de redes domésticas, sem fio ou com fio, e tem como base o gerenciamento de falhas e configurações. A solução pode ser usada por ISPs para reduzir o tempo de inatividade das redes de clientes, sendo que com esse controle, os ISPs possuem uma gama mais ampla de operações para solucionar problemas remotamente e, portanto, reduzir o número de manutenção no local. Foi realizada a avaliação de um protótipo simulando falhas comuns em redes domésticas. Como resultado, a arquitetura conseguiu aumentar a taxa de transferência e reduzir o atraso da rede (ALVES et al., 2018).

3.1.7 *Safe guarding home IoT environments with personalised realtime risk control* (GHOST)

O projeto de pesquisa europeu *Safe guarding home IoT environments with personalised realtime risk control (GHOST)* define uma arquitetura incorporada a um dispositivo inteligente, adaptado para redes domésticas e projetado para ser independente do fornecedor. O projeto conceitual da arquitetura envolve análise avançada dos fluxos de dados, que podem ser classificados em perfis de usuários e dispositivos, os quais, são utilizados em uma avaliação automatizada de riscos em tempo real. Comparações e correspondências com padrões de fluxos de dados seguros são efetuadas utilizando uma abordagem de autoaprendizagem. Técnicas de análise e visualização de dados são implantadas para garantir maior percepção do usuário e entendimento do status de segurança, ameaças potenciais, riscos, impactos associados e diretrizes de mitigação. A estratégia de validação definida é baseada em uma visão tripla que combina testes de laboratório, bancos de ensaio realistas e testes pilotos em um ambiente real (COLLEN et al., 2018).

3.1.8 *Securebox*

Securebox é uma arquitetura que disponibiliza soluções de segurança como serviço, orientada para a nuvem e faz uso de SDN para melhorar o monitoramento, a segurança e o gerenciamento da rede. Os secureboxes, instalados em diferentes residências, atuam como sensores para o *Cloud Security Service (CSS)* e coletam estatísticas no nível de rede, que podem ser usadas para realizar análises com o objetivo de detectar *botnets*, *malwares* e outros *insights* sobre a rede. O *Cloud Manager* é o componente central da arquitetura, sendo que sua responsabilidade é gerenciar solicitações de clientes, recursos e gerenciar tarefas de análise de tráfego. Foi apresentada uma discussão sobre a avaliação do protótipo em diferentes cenários, como a latência, detecção colaborativa de ameaças, a privacidade dos usuários quando as estatísticas no nível da rede são compartilhadas com um terceiro, eficiência em relação aos custos da solução, escalabilidade, ataques contra a arquitetura e largura de banda (HAFEEZ; DING; TARKOMA, 2017).

3.1.9 *HanGuard*

O *design* da arquitetura HanGuard é inspirado parcialmente no paradigma SDN, que foi concebida com a ideia de ser implantada facilmente em uma *Home Area Network (HAN)*. Para atender a esse propósito, a arquitetura apresenta o controle de segurança distribuído, que inclui um *Controller*, instalado em um roteador presente na residência para imposição de políticas, e um *Monitor* no telefone do usuário para coletar a situação em tempo de execução e tomar decisões de acesso. Para evitar alteração no sistema operacional móvel, o *Monitor* está na forma de um aplicativo no espaço do usuário. Ele detecta o aplicativo, que está fazendo a comunicação em rede e sua conformidade com as políticas de segurança, e envia a permissão de acesso ao Controlador do roteador por um canal de controle seguro. O roteador utiliza essas informações para impor as políticas, sendo que somente o tráfego com uma permissão do Monitor pode acessar dispositivos IoT (DEMETRIOU et al., 2017).

3.1.10 *CommunityGuard*

A arquitetura CommunityGuard possui dois componentes principais: o primeiro é o *Guardian Node* que se refere a um dispositivo responsável por observar todo o tráfego da rede doméstica e o segundo componente é o *Community Outpost*, que se refere ao servidor em nuvem, que faz interface com os *Guardian Nodes* e realiza o monitoramento do tráfego. Foi criado um protótipo inicial utilizando o sistema *Snort IPS*, com o objetivo de gerenciar de forma automática sua configuração. Juntos, esses componentes fornecem monitoramento de segurança colaborativo, que pode ser gerenciado remotamente. Cada *Guardian Node* é capaz de monitorar e enviar informações ao servidor *Community Outpost*. Ao fazer isso, o servidor possui vários pontos de vista e a capacidade de detectar algo em

um local para ajudar a proteger outro. O *Community Outpost* realiza análises usando todas as informações e implanta as defesas conforme necessário enviando as mudanças nas configurações do *Snort* (STEWART; VASU; KELLER, 2017).

3.2 Discussão

As arquiteturas apresentadas na Seção 3.1 buscam melhorar a segurança e o gerenciamento da rede no ambiente doméstico. A seguir são descritos os critérios analisados durante o estudo:

- ❏ Nome: nome do projeto ou da arquitetura.
- ❏ Ano: o ano em que a solução foi proposta, pois isso pode influenciar nas tecnologias empregadas em sua estrutura.
- ❏ Estratégia: as arquiteturas propõem o monitoramento do tráfego e gerenciamento da rede para melhorar algum aspecto do ambiente residencial. É importante avaliar onde esses dados estão sendo processados e de onde estão sendo gerenciados. Caso esse processamento/gerenciamento ocorra dentro da casa é considerada a estratégia de *edge computing* e quando são processados fora do ambiente residencial consideramos a estratégia de *cloud computing*.
- ❏ Validação: a forma de validação da arquitetura é importante para analisarmos o quanto a solução foi explorada. Uma validação **Hipotética** possui uma relação distante de um ambiente real; **Empírica** utiliza informações de configurações operacionais, experiências e observações; **Experimental** busca reproduzir o cenário onde a solução está sendo aplicada; **Teórica** utiliza argumentos teóricos para apoiar os resultados.
- ❏ Ameaças: ao propor uma solução de segurança, os trabalhos descrevem as ameaças que buscam mitigar. Por exemplo, uma solução que pretende conter um ataque de negação de serviço apresenta seus componentes e suas validações para evitar que os recursos de um sistema fiquem indisponíveis para os seus utilizadores.
- ❏ Objetivos de segurança comprometidos: impactos gerados caso a solução falhe ao mitigar as ameaças analisadas. A título de exemplo, caso uma solução forneça mecanismos de proteção contra negação de serviço e a solução não funcione, a disponibilidade seria afetada diretamente. Dessa forma, é importante destacar cada objetivo de segurança e o impacto gerado em caso de falha da solução.
- ❏ Contramedidas: identificar a principal contribuição criada pelas soluções para mitigar uma determinada ameaça, por exemplo, uma medida possível para evitar um ataque de negação de serviço seria criar um mecanismo de detecção de anomalia.

- ❏ Técnicas/Ferramentas: apresenta as técnicas/ferramentas usadas para criar as contramedidas com o objetivo de melhorar a segurança do ambiente residencial. Por exemplo, ao propor um mecanismo de detecção de anomalias é importante entender o que foi usado para a sua concepção como os algoritmos e técnicas necessárias para alcançar a proteção desejada.

- ❏ SDN: identifica se a referência selecionada adotou recursos SDN.

As características das arquiteturas são apresentadas na Tabela 2. Baseadas nas necessidades atuais do ambiente residencial, as arquiteturas buscam fornecer maior flexibilidade e agilidade para lidar com os novos desafios e ameaças que surgem a cada dia. Diante disso, o paradigma SDN pode colaborar para que o ambiente se torne mais dinâmico. Por esse motivo, grande parte das arquiteturas empregam o uso de SDN em sua estrutura.

Dentre as arquiteturas apresentadas, grande parte optam pelo processamento das informações dentro do ambiente residencial. Com isso, é possível obter um melhor desempenho e agilidade comparado com soluções que fazem esse processamento em uma *cloud*, pois todo o tráfego ou fluxos da rede precisam ser enviados para que a solução realize o processamento das informações. Outro fator importante a ser analisado é como a privacidade dos usuários é mantida em uma solução que precisa usar todas as informações de comunicação da rede para desempenhar a sua função fora do ambiente residencial.

Para implantar uma solução em um ambiente residencial, existem diversas preocupações tais como o custo para aderir à solução e a complexidade de implantação. Diante disso, é importante analisar como é feita a implantação e como os mecanismos de segurança são usados para fornecer proteção adicional ao ambiente residencial.

As soluções Customized Blockchain-Based (CBB) (AMMI; ALARABI; BENKHELIFA, 2021), Urban Patrol (MASCARENHAS et al., 2021) e EGRBAC (AMEER; BENSON; SANDHU, 2020) apresentam uma preocupação somente com os dispositivos IoT e descartam outros dispositivos presentes no ambiente. As soluções também carecem de detalhes sobre a implantação no ambiente real.

A HomeNetRescue (ALVES et al., 2018) busca melhorar o gerenciamento da rede doméstica focando no gerenciamento de falhas e configurações. Os experimentos apresentados no trabalho demonstram que ao usar a arquitetura é possível aumentar a taxa de transferência da rede e reduzir o atraso e instabilidade da transmissão sem fio. As métricas utilizadas nos experimentos foram *throughput*, *delay* e *jitter*.

A arquitetura SHSec (SHARMA et al., 2019) apresenta uma solução para detecção de anomalias. Para um dos experimentos é utilizado o *dataset* Sherlock (Sherlock, 2016). No entanto, a proposta não apresenta detalhes de como o modelo foi criado. As métricas usadas para avaliar o modelo foram a precisão e sensibilidade. Apesar de lidar com a detecção de anomalias, este trabalho não deixa claro como é a abordagem de treinamento, como foi criado o modelo ou como ele será atualizado. Dessa forma, é possível depreender

que a arquitetura está em sua concepção e que as possíveis anomalias presentes em um ambiente residencial inteligente não foram exploradas.

Tabela 2 – Características das arquiteturas para o contexto residencial

Nome	Ano	Estratégia	Validação	Ameaças	Objetivos de segurança comprometidos	Contramedidas	Técnicas/ Ferramentas/ Algoritmos	SDN
FamilyGuard	2022	Edge/ Cloud	Empírica Experimental	DDoS Replay attack Eavesdropping	Disponibilidade Integridade Confidencialidade	DA	ML(LOF, OCSVM, IF)	•
CBB	2021	Edge/ Cloud	Experimental	Message Modification Replay attack Eavesdropping	Disponibilidade Integridade Autenticidade Confidencialidade	Blockchain	Chaincode	○
Urban Patrol	2021	Edge/ Cloud	Experimental	Sinkhole Attack Worm Attack Side Channel Attack	Disponibilidade Confidencialidade Integridade Autenticidade	DA Blockchain RBA	ML(XGBoost)	○
TSP2	2020	Edge	Teórica	Replay Attacks Data Leakage Eavesdropping	Confidencialidade	Criptografia	LEA	○
EGRBAC	2020	Edge/ Cloud	Experimental	Man-in-the-Middle Identity misbinding	Autenticidade	RBAC	Modelo personalizado	○
GHOST	2019	Edge/ Cloud	Teórica	Impersonation Replay attack	Integridade Disponibilidade	DA Blockchain	-	•
SHSec	2019	Edge	Experimental	DDoS	Disponibilidade Confidencialidade	DA	Distribuição de probabilidade condicional	•
HNR	2018	Edge/ Cloud	Experimental	DDoS	Disponibilidade	Gestão Autônoma	Detecção de Falha	•
Securebox	2017	Edge/ Cloud	Experimental	DDoS	Integridade Disponibilidade	IPS	SNORT	•
HanGuard	2017	Edge	Empírica	Man-in-the-Middle Identity misbinding	Autenticidade	Políticas de Controle	Modelo personalizado	○
CG	2017	Edge/ Cloud	Experimental	DDoS	Disponibilidade	IPS	SNORT	○

Legenda: Detecção de Anomalias (DA); *Role-Based Access Control (RBAC) Lightweight Encryption Algorithm (LEA)*

A solução Securebox (HAFEEZ; DING; TARKOMA, 2017) apresenta uma análise de tráfego e o ganho de desempenho usando a abordagem colaborativa para detecção de ameaças. No experimento foi realizado o monitoramento de um ataque de *port scanning*, no qual foram usados 20 nós para gerar tráfego e 15 nós zumbis (controlado pelo invasor), cada um varrendo 1000 portas aleatórias, para atacar três segmentos de rede, cada um com três *hosts* conectados. Também foram criados dois protótipos para avaliar o desempenho do Securebox através do tráfego Voice-over-IP (VoIP) usando um Fit-PC3 pro-Linux (fitPC) e Raspberry PI (R-Pi). Os resultados mostram que o Securebox apresenta um aumento na latência da experiência do usuário. Outro fator importante ao analisar a solução é que o componente *Cloud Manager* é o ponto central de todo o sistema, podendo ser foco de ataque.

A abordagem colaborativa também é usada na arquitetura CommunityGuard, no qual um *Guardian Node* é capaz de monitorar e enviar informações ao servidor do *Community Outpost*. Dessa forma, conseguem detectar algo em um local para ajudar a proteger outro. O experimento realizado foi o bloqueio de tráfego DDoS na saída da rede, também foi descrito a proteção adicional que a solução fornece a rede tanto para ataques de entrada quanto de saída. A solução utiliza o Snort. Portanto, é necessário avaliar os requisitos mínimos para colocar a solução em execução no ambiente residencial, o custo de implantação pode ser alto comparado com as outras soluções (STEWART; VASU; KELLER, 2017).

As soluções apresentadas não buscam substituir mecanismos de segurança utilizados atualmente e sim fornecer uma proteção adicional, que visa contribuir significativamente com a segurança e privacidade das pessoas que vivem em determinado ambiente. Ao observar as arquiteturas, é possível notar que algumas soluções utilizam o paradigma SDN, colocam um novo dispositivo na rede para analisar o tráfego, apresentam uma forma de gerir as políticas de segurança e possivelmente enviar informações para um servidor em *cloud* para criar uma malha de colaboração com o objetivo de auxiliar na detecção de novos ataques. Conclui-se que essa estrutura base precisa existir, sendo que, no entanto, ao comparar esses trabalhos, é possível observar que o benefício adquirido ao implantar determinada solução depende dos serviços que estão disponíveis para serem utilizados, como a detecção de anomalias, gerenciamento de configurações, detecção de falhas e políticas de segurança.

Alguns serviços disponibilizados pelas arquiteturas estão em sua concepção inicial e precisam ser explorados para fornecer valor para as pessoas que os utilizam. Nossa solução traz benefícios em relação aos trabalhos analisados na Seção 3.1. A estrutura da arquitetura FamilyGuard é robusta e flexível para atender a heterogeneidade dos ambientes residenciais. Sua estrutura permite que os provedores de serviços de segurança ofereçam modelos customizados para cada ambiente de acordo com a demanda do cliente. Outro fator importante em nossa proposta é que a *FamilyGuard* considera todos

os dispositivos presentes em ambientes residenciais, não apenas dispositivos IoT, como a maioria das arquiteturas analisadas. Um dos componentes da arquitetura *AI Workflow Orchestrator* (AIWO) pode configurar e gerenciar vários fluxos de trabalho para atender o HAN. Quando detecta uma ameaça, outro componente chamado *Security Orchestrator* (SECOR) é executado para mitigar ameaças. Dessa forma, a solução proposta se aproxima de um ambiente real comparado as arquiteturas apresentadas.

3.3 Detecção de Anomalias

Apesar das soluções apresentadas na Seção 3.1 disponibilizarem serviços de detecção de anomalias para o ambiente residencial, tais serviços não exploram métodos que já são utilizados na área de redes de computadores e que provaram ser eficientes.

A Tabela 3 foi criada com base nos trabalhos (ZARPELãO et al., 2017) (BOUTABA et al., 2018) (FERNANDES et al., 2019) que analisam os mecanismos mais utilizados para detecção de anomalias tanto no cenário tradicional de redes quanto para o IoT. A Tabela exibe uma comparação entre trabalhos para detectar anomalias, apresentando o ano de sua publicação de cada trabalho, as técnicas utilizadas para criar o modelo, o conjunto de dados utilizado nos experimentos e as métricas que foram utilizadas para avaliar a eficiência do modelo.

Os conjuntos de dados dos trabalhos apresentados na Tabela 3 não retratam a realidade de um ambiente residencial, que possui tráfego de dispositivos IoT e multifuncionais. Ao propor uma solução para tal ambiente, é necessário pensar no contexto geral e suas características. A análise do ambiente como um todo não foi feita pelas solução apresentadas na Seção 3.1 e nem pelos trabalhos citados na Tabela 3. Desse modo, este trabalho pretende definir um conjunto de dados que consiga representar o tráfego de um ambiente residencial. Após a definição do conjunto de dados serão criados modelos que consigam detectar anomalias no ambiente.

Com base nos trabalhos apresentados, apenas um deles utilizou algoritmos do tipo OCC (DEMERTZIS; ILIADIS; SPARTALIS, 2017). O objetivo desses algoritmos é criar um modelo de classificação com exemplos de uma única classe. Dessa forma, para detectar anomalias em um ambiente residencial, o tráfego normal pode ser usado para o treinamento do modelo. Contudo, tal tipo de tráfego deve ser bem definido e sem a presença de nenhum comportamento anômalo (KHAN; MADDEN, 2013). Desse modo, o tráfego identificado fora do padrão dos dados de treinamento é classificado como anomalia.

Essa abordagem pode obter bons resultados ao ser utilizada no contexto residencial, pois novas vulnerabilidades e *malwares* surgem todos os dias. Diante disso, ao criar um modelo que consiga representar o tráfego normal do ambiente, existe a possibilidade que ele consiga lidar melhor com novas ameaças do que as abordagens adotadas pelas arquiteturas apresentadas na Seção 3.1 e na Tabela 3, ao usar técnicas que precisam de

amostras dos *malwares* gera uma complexidade grande para gerir um modelo que precisa ser treinado ou atualizado sempre que uma nova ameaça é encontrada.

Mediante o exposto, este trabalho, inicialmente, aplicará algoritmos do tipo OCC para identificar anomalias no ambiente residencial. É importante ressaltar que essa abordagem não foi utilizada por nenhuma das arquiteturas apresentadas na Seção 3.1. Para avaliar os modelos gerados, serão usadas as métricas exploradas pelo trabalhos apresentados na Tabela 3.

Tabela 3 – Abordagens de Detecção de Anomalias

Artigo	Ano	Técnicas	Datasets	Ref.	Métricas
(HASAN et al., 2019)	2019	Regressão logística Árvore de decisão Floresta aleatória ANN SVM	DS2OS	(PAHL; AUBET, 2018)	Acurácia Precisão Revocação F1-Score
(ALRASHDI et al., 2019)	2019	Floresta aleatória	UNSW-NB 15	(MOUSTAFA; SLAY, 2015)	Acurácia Especificidade Precisão Revocação
(DOSHI; APTHORPE; FEAMSTER, 2018)	2018	LSVM KNN Árvore de decisão Floresta aleatória ANN	PVEIoT	(APTHORPE; REISMAN; FEAMSTER, 2017)	Acurácia Precisão
(WANG; GU; WANG, 2017)	2017	SVM	NSL-KDD	(TAVALLAEE et al., 2009)	Acurácia Precisão
(KABIR et al., 2018)	2017	LS-SVM	KDD99	(KDDCUP, 1999)	Acurácia Precisão Revocação
(ASHFAQ et al., 2017)	2017	NNRW	NSL-KDD	(TAVALLAEE et al., 2009)	Acurácia
(DEMERTZIS; ILIADIS; SPARTALIS, 2017)	2017	OCC-SVM OCC-eSNN OCC-CD/CPE	Águe Gás Eletricidade	(MORRIS; THORNTON; TURNIPSEED, 2015)	Acurácia Precisão Revocação F1-Score
(BROWN; ANWAR; DOZIER, 2016)	2016	GRNN	UNB ISCX	(SHIRAVI et al., 2012)	Revocação Especificidade
(SWARNKAR; HUBBALLI, 2016)	2016	OCNB	IIT Indore HTTP attack	(MCHUGH, 2000)	Especificidade

Legenda: *KNN*; *Artificial Neural Network (ANN)*; *Support Vector Machine (SVM)*; *Lagrangian Support Vector Machine (LSVM)*; *Least-squares support-vector machine (LS-SVM)*; *Neural network with random weights (NNRW)*; *General Regression Neural Network (GRNN)*; *One Class Naive Bayes (OCNB)*

Capítulo 4

Especificação da Arquitetura FamilyGuard

O objetivo deste capítulo é apresentar o método de pesquisa, os aspectos conceituais e estruturais da arquitetura FamilyGuard, especificando suas camadas e seus serviços providos para detecção de anomalias em redes domésticas.

4.1 Método

Esta seção apresenta o método utilizado, que descreve a abordagem lógica adotada nesta tese para alcançar os objetivos declarados na Seção 1.3 e, desse modo, o método consiste nas etapas ilustradas na Figura 3. Adicionalmente, nesta pesquisa foi adotada uma abordagem incremental que se baseia inicialmente na estruturação do projeto e etapas adicionais, que são retroalimentadas.

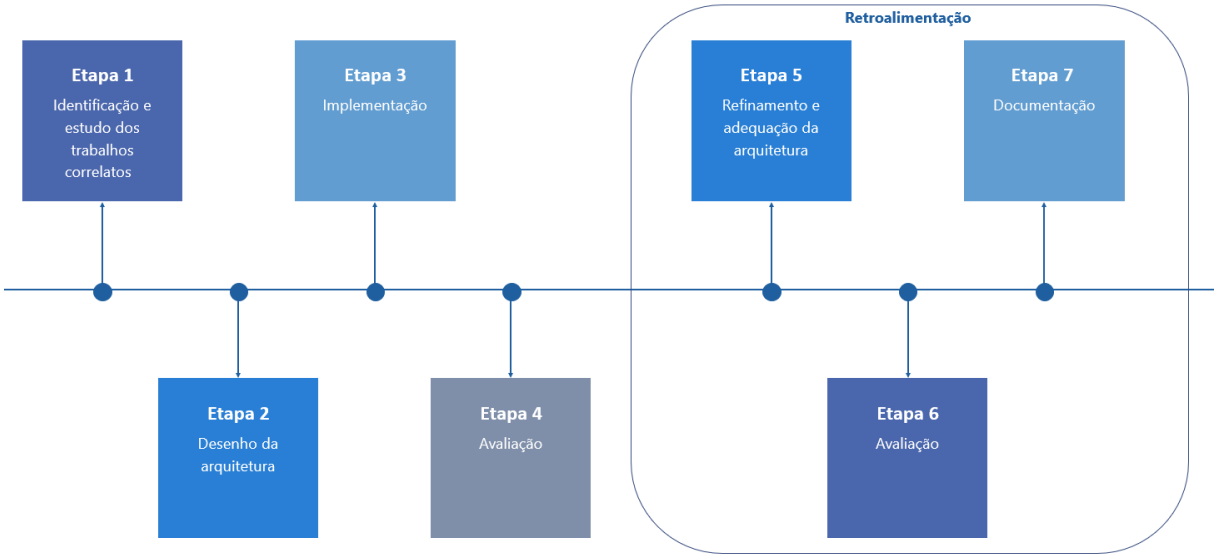


Figura 3 – Método de Pesquisa e Desenvolvimento

A primeira etapa, conforme ilustrado na Figura 3, consiste no resgate de conceitos para a fundamentação teórica e soluções relacionadas a segurança da informação direcionadas ao contexto residencial. Essa etapa considerou uma abrangente pesquisa bibliográfica, usando palavras-chaves, amplamente utilizadas em conferências e artigos de revistas para identificar as características relevantes apontadas pelo estado-da-arte. Ao concluir, foram estruturados os atributos dos trabalhos relacionados. Dentre as particularidades dos trabalhos, foram escolhidas as qualitativamente mais desejáveis para a detecção de anomalias em ambientes residenciais.

Na segunda etapa, "Desenho da arquitetura", foram realizadas atividades de engenharia de *software* para a especificação de funcionalidades. Nessa etapa, os aspectos de desenvolvimento e a parametrização da arquitetura foram relacionados com o estado da arte e analisados com o objetivo de produzir um documento lógico e estrutural da arquitetura, a saber: a estrutura analítica do projeto ilustrada na Figura 4, com base nas práticas definidas pelo *Project Management Institute (PMI)* (ROSE, 2013).

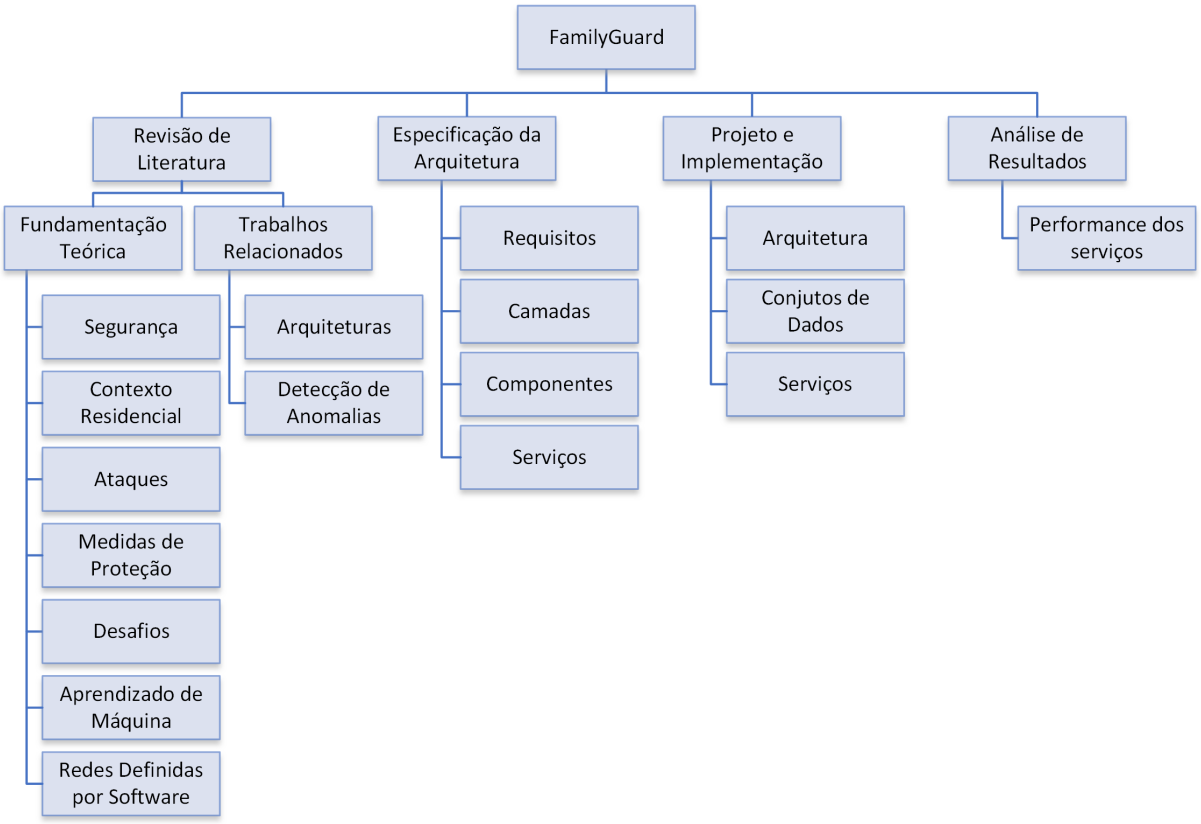


Figura 4 – Estrutura Analítica do Projeto

A etapa "Implementação" consiste na codificação da estrutura da arquitetura para receber os fluxos de tráfego de rede e modelos de inteligencia artificial. Nesta etapa também são previstos testes de funcionalidade que englobam os testes de integração e sistema. Ao concluir essa etapa considera como entregáveis o código fonte em repositório especifico.

A quarta etapa, "Avaliação", consiste em testar as funcionalidades e a eficácia da arquitetura para lidar com os problemas apontados nas soluções do estado da arte. Nessa etapa, são definidas métricas de comparação, configuração do ambiente, *hardware* utilizado pela solução, experimentação, validação e avaliação da arquitetura em relação aos trabalhos correlatos.

As etapas seguintes consistem em melhorar a estrutura inicial da arquitetura, que são retroalimentadas. Essas etapas exercem a função de refinamento e adequação da solução para questões incipientes do estado da arte bem como ajustes remanescentes. Na quinta etapa é realizado o refinamento e adequação da arquitetura, entende-se que essas melhorias possam envolver a estrutura da arquitetura, conjuntos de dados utilizados para criar os modelos de inteligencia artificial e algorítimos de detecção de anomalias. A melhoria realizada na quinta etapa é avaliada na sexta etapa e posteriormente documentada na sétima etapa. Com base no método proposto a próxima seção descreve a arquitetura FamilyGuard.

4.2 FamilyGuard

FamilyGuard é uma arquitetura de segurança para ambientes residenciais, que tem como objetivo oferecer proteção adicional ao ambiente residencial no que se refere à segurança da informação. A sua estrutura flexível permite lidar com heterogeneidade dos dispositivos e protocolos presentes em uma *smart home* e também contribui de forma significativa para solucionar os desafios apresentados na Subseção 2.3.3.

Com base nas informações presentes no Capítulo 3, o ambiente doméstico carece de soluções de segurança confiáveis. A maioria das residências possui apenas software antivírus instalado nos computadores e dificilmente possuem defesas de perímetro instaladas em suas redes, como um sistema de detecção de intrusão (IDS) ou um *firewall* na rede (NTHALA; FLECHAIS, 2018) (KARIMI; KRIT, 2019).

A Figura 5 apresenta uma visão de alto nível dos locais nos quais a arquitetura atua, que possui serviços atuando na rede doméstica, nos provedores de serviços de segurança e um aplicativo para acessar as configuração do ambiente.

Os provedores de serviços de segurança são responsáveis por oferecer recursos de gerenciamento e configuração que ajudem a proteger as informações presentes em uma HAN, por exemplo, modelos de *Machine Learning (ML)* para identificar anomalias no comportamento da rede. O usuário pode acessar os serviços de segurança dos provedores, por meio de um aplicativo em seu dispositivo móvel, e contratar um *Security-as-a-Service (SECaaS)* que atenda às suas necessidades. Uma HAN pode enviar informações e alertas aos provedores de serviços caso identifique situações anômalas na rede.

Cada família possui diferentes tipos de necessidades, então, além da arquitetura proposta lidar com a heterogeneidade dos dispositivos, é necessário atender às particularida-

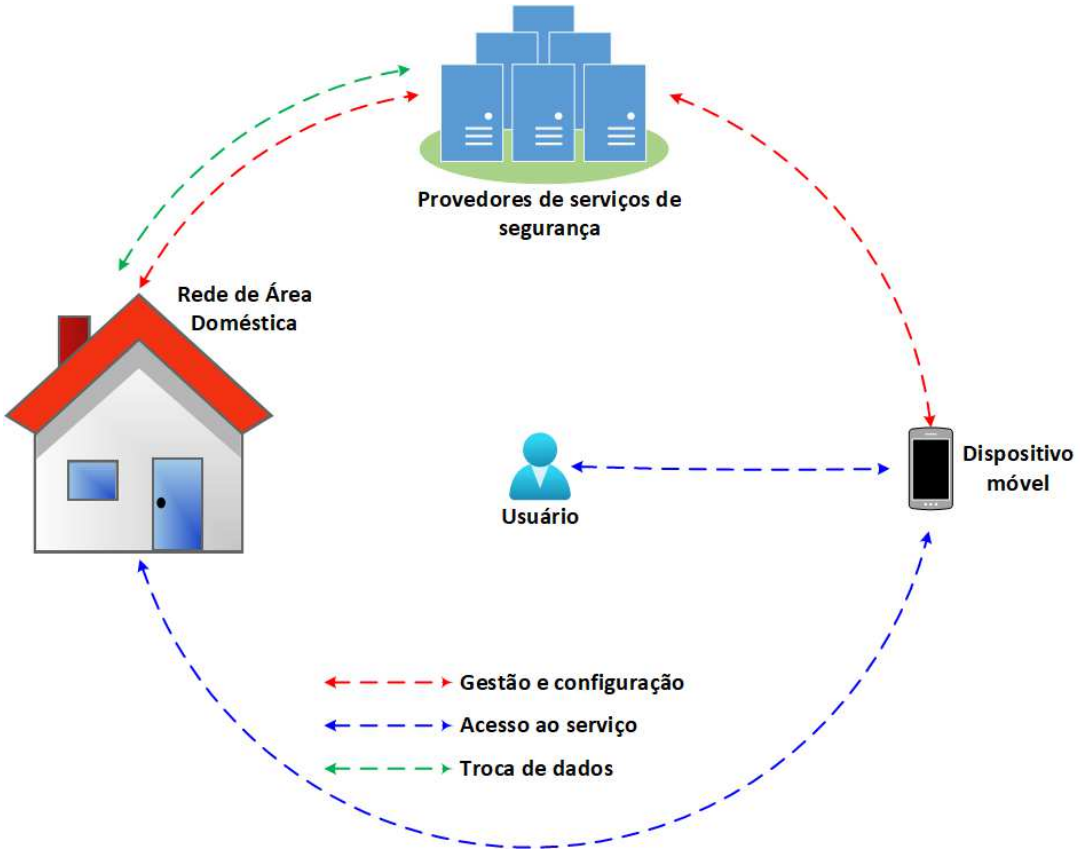


Figura 5 – Visão geral da arquitetura FamilyGuard

des de cada família. Deste modo, para oferecer proteção adicional ao ambiente residencial, a arquitetura oferece serviços que atuam em dois locais, sendo: (i) o ambiente residencial, que possui serviços para detecção de anomalias e mitigação de ameaças; e (ii) provedores de serviços de segurança, que fornecem serviços especializados para melhor mitigar ameaças nos ambientes residenciais.

A Figura 6 apresenta a comunicação entre os componentes da arquitetura. Os provedores de serviços de segurança possuem um componente chamado Assistente Central de Segurança (*Central Security Assistant - CSA*). O ambiente residencial possui três componentes: Unidade de Vigilância Domiciliar (*Home Surveillance Unit - HSU*), Gerador de fluxo de rede (*Network flow generator - NFG*) e o Controlador SDN. A funcionalidade de cada componente é descrita nas próximas subseções e a Figura 6 ilustra o posicionamento de cada um deles no ambiente, bem como a sua comunicação, que pode ser dividida entre fluxos de controle, dados e gestão.

4.2.1 Unidade de Vigilância Domiciliar (HSU)

A HSU é responsável por receber os fluxos da rede, realizar tarefas de análise de tráfego e participar do gerenciamento da rede em conjunto com o controlador. A HSU utiliza uma estrutura de fluxos de trabalho ou *Workflow* para tratar os fluxos do tráfego de rede recebidos e realizar análises. Um fluxo de trabalho pode conter vários modelos

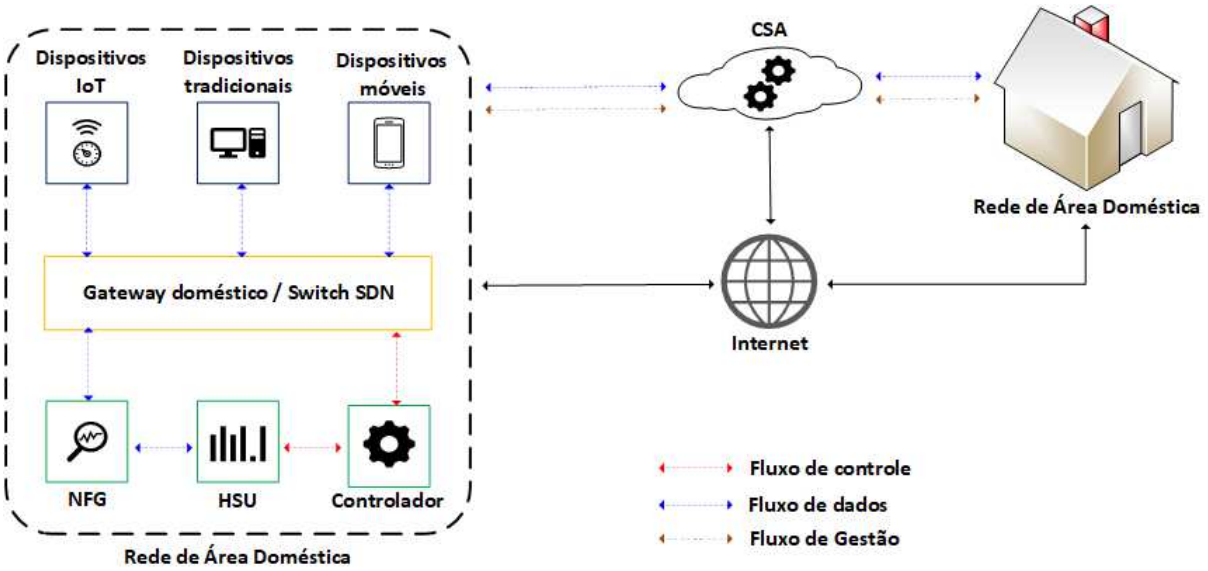


Figura 6 – Componentes da arquitetura FamilyGuard

de análise de dados para tratar os fluxos e realizar análises de forma sequencial. Por exemplo, a HSU recebe um fluxo e encaminha para o *Workflow A*, que inicialmente prepara o fluxo removendo as características desnecessárias, outro modelo identifica o tipo de tráfego do fluxo e encaminha para um último modelo para verificar se o fluxo possui um comportamento anômalo. A HSU possui dois componentes principais: o Orquestrador de segurança (*Security Orchestrator* - SECOR) e o Orquestrador de fluxo de trabalho de IA (*AI Workflow Orchestrator - AIWO*), mostrados na Figura 7 e descritos abaixo:

- SECOR: responsável pelo gerenciamento, configuração e notificação das informações relacionadas a políticas de segurança aplicadas no HAN. Ao receber o resultado de uma previsão do AIWO, o Mecanismo de Decisão (*Decision Engine - DE*) verifica as políticas de segurança que podem ser aplicadas para mitigar a ameaça detectada e, em seguida, notifica o controlador para que seja realizada a alteração na tabela de fluxo com o objetivo de bloquear ou limitar o acesso ao dispositivo onde a ameaça foi detectada. O usuário também pode alterar as políticas de segurança, ao optar por bloquear o dispositivo infectado imediatamente, ou ser notificado para tomar alguma medida para mitigar a ameaça.

- AIWO: responsável por receber uma solicitação de previsão e gerar um resultado para o SECOR. Também é responsável por gerenciar e administrar os serviços disponíveis para previsões, classificados como Serviços de Preparação de Dados, Serviços de Detecção de Anomalias e Serviços de Classificação de Anomalias.

A Figura 8 exemplifica a estrutura do AIWO, que permite definir vários fluxos de trabalho usando um conjunto de modelos disponíveis. Com essa estrutura provedores de serviços de segurança e usuários são livres para criar e definir fluxos de trabalho para

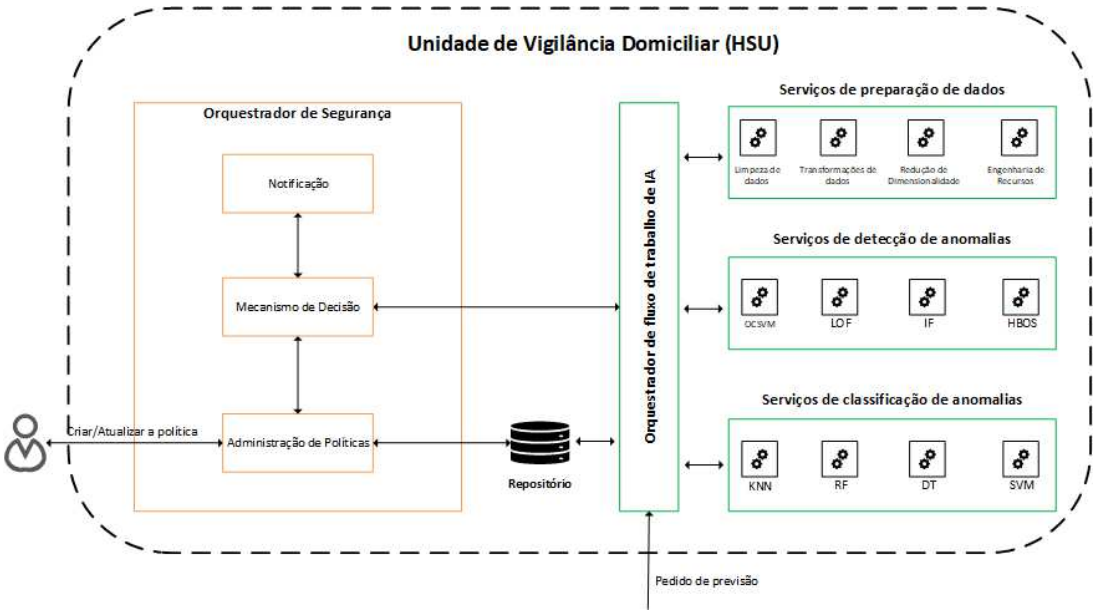


Figura 7 – Componentes da Unidade de Vigilância Domicilia

atender às necessidades específicas de uma HAN. Por exemplo, é possível ter fluxos de trabalho para identificar anomalias em cenários específicos, como detectar anomalias em dispositivos IoT e ignorar dispositivos domésticos tradicionais, a exemplo de notebooks e computadores pessoais.

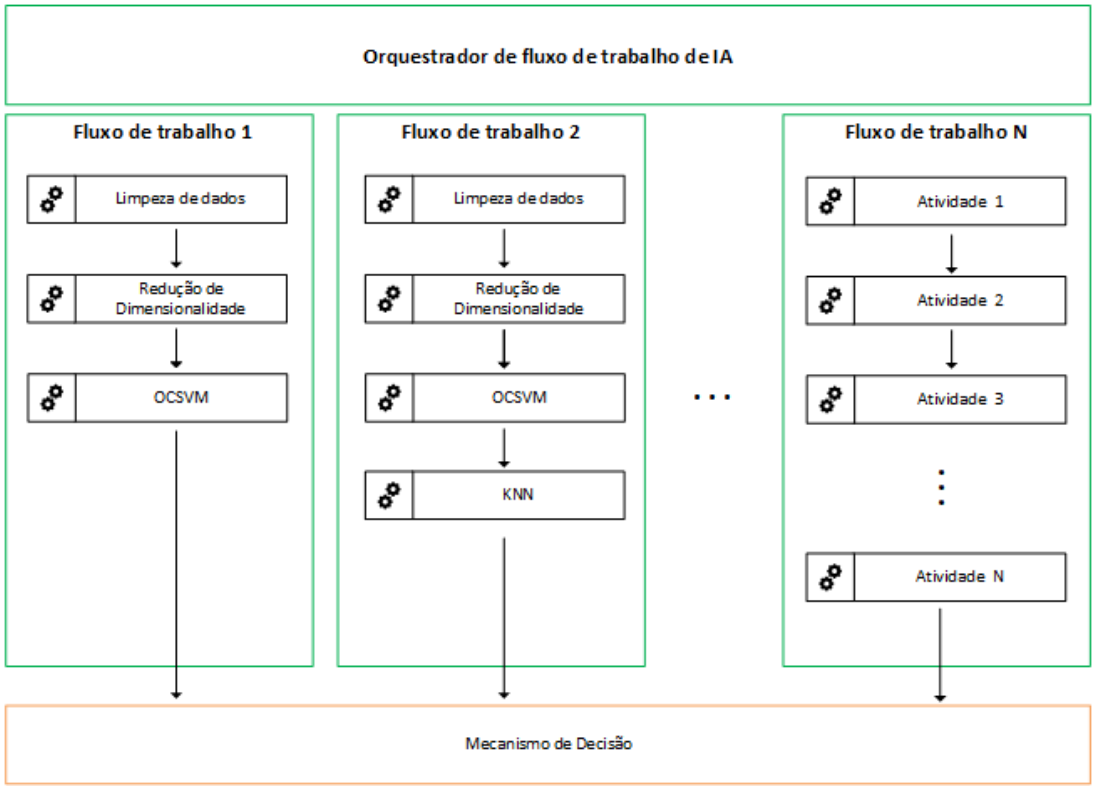


Figura 8 – Exemplos de fluxos presentes no AIWO

É fundamental destacar a existência de vários fluxos de trabalho, no entanto, uma solicitação de previsão não precisa passar por todos os fluxos de trabalho. O AIWO permite

configurar que os fluxos de determinados dispositivos sejam roteados para fluxos de trabalho específicos. Dessa forma, os modelos podem identificar anomalias em determinados tipos de fluxos, por exemplo, processar somente fluxos originados dos dispositivos móveis e ignorar todos os outros dispositivos do ambiente.

4.2.2 Assistente Central de Segurança (CSA)

O CSA visa a prestação de serviços que colaboram para o desempenho das atividades de HSUs. Adicionalmente, o CSA pode ser disponibilizado por algum ISPs ou por prestadores de serviços interessados em fornecer uma estrutura adequada para a gestão da segurança em ambientes residenciais.

A flexibilidade topológica no posicionamento do CSA permite que a arquitetura seja empregada não apenas no contexto de ambientes residenciais inteligentes, mas também em diferentes ambientes de IoT, como *smart grid*, saúde, entre outros. No entanto, é necessário avaliar, entre outras coisas, a quantidade de dispositivos implantados na rede e o tráfego gerado por eles para definir os recursos de hardware necessários para atender o ambiente.

A Figura 9 mostra o diagrama de fluxo de dados presente no CSA. O processo de notificação recebe notificações fornecidas pela HSU e salva no Repositório de Notificações. O CSA também possui o processo de Estatística, que é responsável pela criação de métricas com base nas notificações recebidas pela HSU.

Além das notificações, o CSA é responsável por manter os modelos de detecção de anomalias utilizados pelas HSUs. O processo de Construção de Modelos é responsável por construir modelos de inteligência artificial e salvá-los no repositório. O processo chamado Gerente de atualização é responsável por realizar consultas para verificar se houve atualizações nos modelos e enviar uma mensagem notificando a HSU, caso haja uma atualização.

4.2.3 Gerador de Fluxo de Rede (NFG)

O NFG coleta pacotes de uma determinada rede ou segmento de rede para gerar fluxos. Os objetivos podem variar desde a solução de problemas de conectividade até o planejamento de alocação de largura de banda futura. No caso deste trabalho, os fluxos ajudam a identificar e corrigir problemas de segurança. Depois do fluxo gerado pelo NFG, ele é direcionado para a HSU para que possa ser analisado pelos modelos de detecção de anomalias.

O monitoramento de fluxos de rede fornece informações sobre como uma rede opera, uso de aplicativos, possíveis gargalos e anomalias que possam sinalizar ameaças à segurança. Para isso, vários padrões e formatos diferentes são usados no monitoramento de fluxos de rede, incluindo NetFlow (CLAISE et al., 2004), sFlow (GIOTIS et al., 2014) e

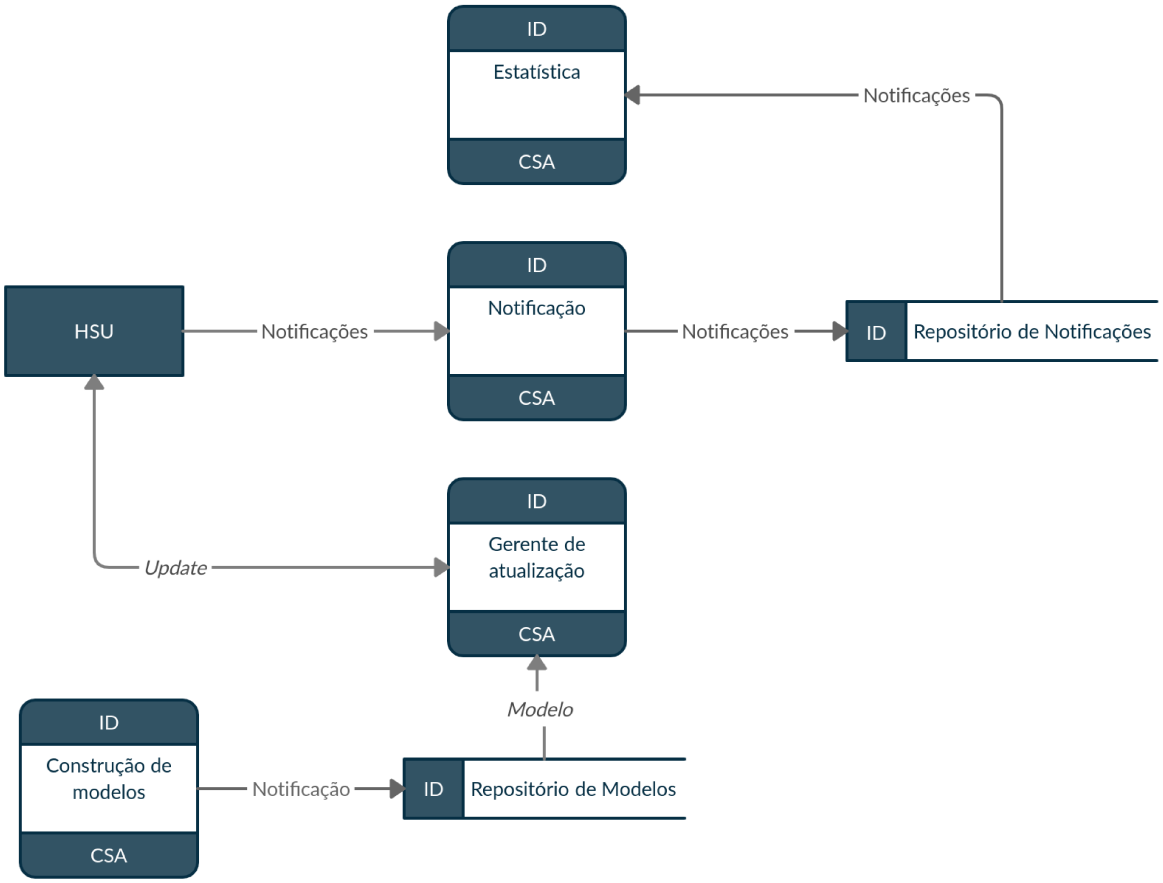


Figura 9 – Fluxo de dados do CSA

Internet Protocol Flow Information Export (IPFIX) (HOFSTEDE et al., 2014). Os fluxos utilizados nos experimentos apresentados na seção 5 foram gerados pela ferramenta CICFlowMeter (DRAPER-GIL et al., 2016) (LASHKARI et al., 2020).

As informações presentes no cabeçalho de pacotes IP fornecem a base para gerar os fluxos de rede. Essas informações são resumidas e a quantidade de dados processados pelo sistema de detecção de anomalias é reduzida. Outro fator que colabora para a decisão de usar fluxos de rede para a análise é a quantidade de aplicativos de rede que utilizam criptografia de ponta a ponta, não sendo possível inspecionar dados criptografados em um local intermediário por um sistema de detecção de intrusão baseado em pacotes.

Diante disso, a detecção de anomalias baseada em fluxos se torna interessante, pois não é necessária nenhuma varredura de dados de pacotes. Por fim, existem menos preocupações com a privacidade do que a inspeção baseada em pacotes, pois as informações dos usuários são protegidas de qualquer varredura intermediária. Apesar dos benefícios mencionados, a detecção de intrusão baseada em fluxo também tem algumas desvantagens. Por exemplo, fluxos possuem informações generalizadas de rede e, por conseguinte, torna-se mais complexo distinguir um ataque usando o método generalizado (UMER; SHER; BI, 2017).

Logo, a adoção de uma solução para geração de fluxos de rede é incentivada pelos benefícios alcançados pela detecção de tráfego anômalo e outras ameaças à segurança da rede. O monitoramento de fluxo também é útil para medir o impacto de um novo aplicativo, configuração de rede ou alteração no número de usuários que acessam a rede.

4.2.4 Controlador SDN

O controlador SDN é responsável por receber as instruções da HSU e transmiti-las ao *Home Switch (HS)*, ao qual todos os dispositivos da casa se conectam por meio de enlaces com ou sem fio. A estratégia utilizada neste trabalho foi colocar a HSU e o controlador no mesmo dispositivo.

A principal vantagem de instalar a HSU junto com o controlador SDN reside em alcançar um cenário de implantação simples em uma casa inteligente, evitando o uso de outro dispositivo para executar as instruções da HSU. Outra característica que contribui para essa decisão é o tempo gasto na troca de mensagens entre o controlador e a HSU, que é menor se os serviços estão no mesmo dispositivo.

O controlador SDN no ambiente residencial oferece maior agilidade, programabilidade, controle de dados centralizado, complexidade reduzida, operações simplificadas e melhor uso dos recursos de rede. Desse modo, a HSU aproveita do dinamismo gerado pelo uso do controlador para agir de forma ativa na rede, com o objetivo de corrigir ou mitigar ameaças que poderiam ferir a privacidade dos usuários. Ao identificar uma ameaça a HSU pode enviar uma regra de bloqueio para o controlador que se propaga nos equipamentos de redes.

Adicionar um controlador SDN dentro de casa pode ter uma desvantagem financeira para o usuário. No entanto, para o cenário de casa inteligente, um *Raspberry Pi* ou outro pequeno dispositivo de baixo custo pode suprir essa necessidade. A complexidade adquirida para os usuários finais em lidar com um controlador SDN é resolvida através da arquitetura FamilyGuard, que realiza toda a comunicação com o controlador e fornece APIs para que sejam desenvolvidas páginas web ou aplicativos móveis para controlar seus dispositivos e informações.

O *design* conceitual da arquitetura segue uma abordagem voltada para detecção de anomalias, o qual permite lidar com heterogeneidade existente dos dispositivos IoT implantados em um ambiente residencial inteligente, concentrando-se na análise de atividades geradas na rede. Diante disso, foi adotada uma estrutura em multicamadas que permite o desenvolvimento independente dos componentes (DAY; ZIMMERMANN, 1983). A Seção 4.3 descreve as camadas lógicas da arquitetura FamilyGuard.

4.3 Organização da Arquitetura FamilyGuard

A arquitetura FamilyGuard é composta por quatro camadas, sendo: Camada de Gerenciamento, Camada de Detecção, Camada de Rede, Camada de Dispositivo. A Figura 10 ilustra a relação entre elas. A HSU atua nas camadas de Rede e Detecção, já o controlador SDN é responsável por desempenhar suas funções na camada de Rede, por fim, os serviços responsáveis pela detecção de anomalias estão associados à camada de Detecção. O CSA atua na camada de Gerenciamento, colaborando para que diversas HSUs consigam realizar suas tarefas. Por fim, a camada de Gerenciamento também conta com aplicações que auxiliam no controle e configuração dos serviços locais presentes no ambiente residencial. As características de cada uma das camadas são descritas nas próximas subseções.

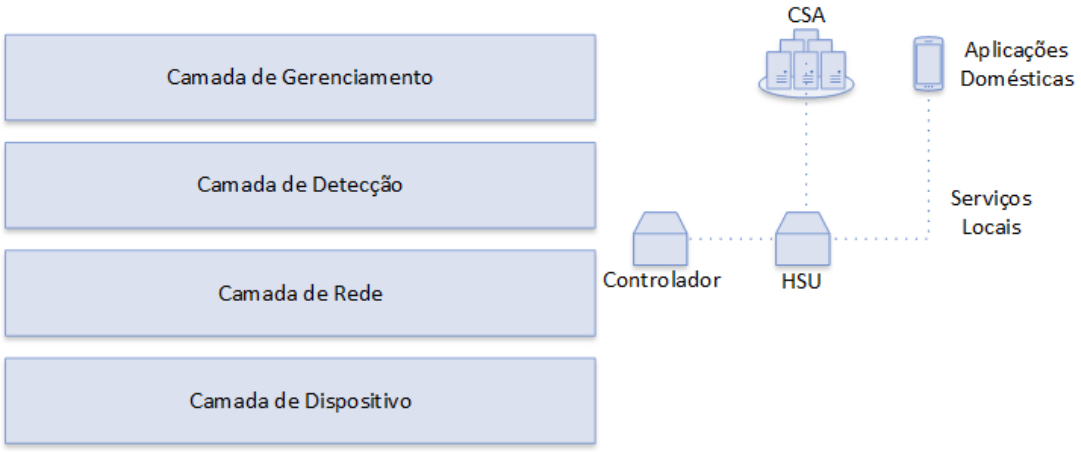


Figura 10 – Camadas da arquitetura FamilyGuard

4.3.1 Camada de Dispositivo

A camada de Dispositivo representa os dispositivos que possuem capacidade de se comunicar dentro do ambiente residencial, incluindo computadores pessoais, *smartphones* e dispositivos inteligentes como sensores de temperatura, presença e fumaça. Entretanto, existe uma grande diversidade de dispositivos inteligentes presentes no mercado que são criados por diversos fabricantes, sendo assim, o ambiente residencial torna-se heterogêneo e complexo ao gerenciar os riscos e ameaças.

4.3.2 Camada de Rede

Responsável por prover conectividade para diferentes protocolos e pelo recebimento de dados transmitidos da rede e pela Camada de Dispositivo. Estão inclusas nessa camada conexões *Wi-Fi*, *Ethernet*, *Wireless Sensor Network* (WSN) e *Machine-to-Machine* (M2M).

4.3.3 Camada de Detecção

A função principal desta camada é realizar a detecção de anomalias por meio dos serviços apresentados na Figura 11. Esses serviços possuem funções bem definidas que vão desde o início do processo, ou seja, na recepção do tráfego de rede transmitido pela Camadas de Rede até a notificação gerada para a Camada de Gerenciamento ao classificar um determinado fluxo como anomalia.

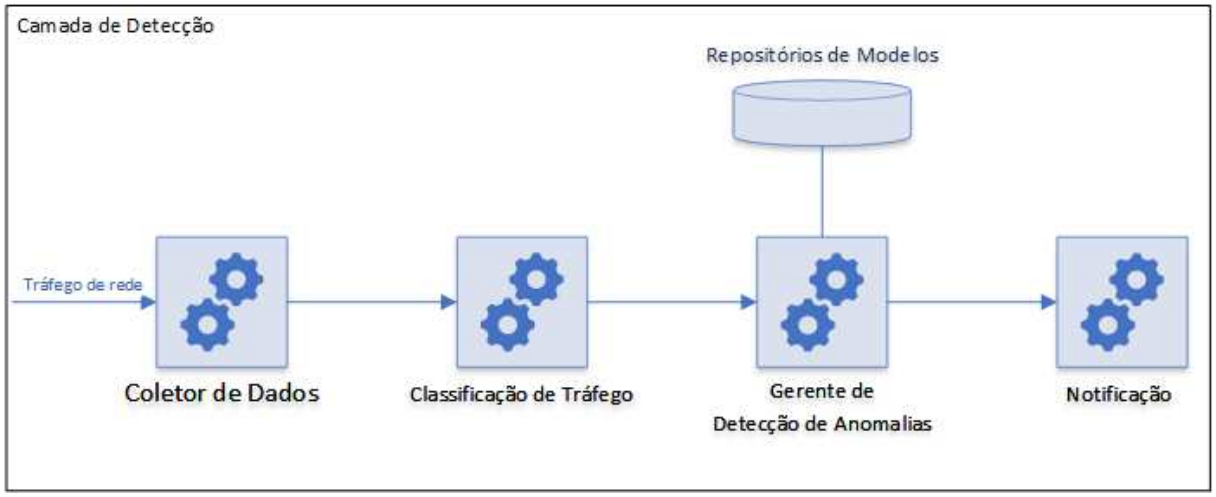


Figura 11 – Serviços presentes na Camada de Detecção

A seguir são apresentadas as particularidades de cada um dos serviços presentes nesta camada:

- Coletor de dados: este serviço possui três funcionalidades disponíveis: a primeira é receber os dados gerados pela Camada de Dispositivo através da Camada de Rede; a segunda é de buscar informações na rede, por exemplo, verificar se um sensor está ativo e qual o seu nível de bateria e gerar os dados; a terceira é tratar os dados recebidos/gerados realizando tarefas de limpeza, filtragem e pré-processamento, com o objetivo de gerar os dados que serão processados pelo serviço de Classificação de tráfego.

- Classificação de tráfego: possui o objetivo de classificar a informação recebida pelo Coletor. Essa classificação pode utilizar algorítimos de aprendizado de máquina, por exemplo, ao receber um fluxo de rede o serviço busca identificar se o fluxo pertence a dispositivos IoT ou dispositivos multifuncionais como *smartphones* e computadores (SIVANATHAN et al., 2017) (BAI et al., 2019). Após realizar o processo de classificação o serviço envia a informação e sua classificação para ser processada pelo Gerenciador de detecção de anomalias.

- Gerenciador de Detecção de Anomalias: no momento em que este serviço recebe o fluxo e sua classificação, a informação é encaminhada para o modelo adequado a fim

de que seja verificado se a informação é relativa a uma anomalia ou não. Em suma, esse serviço pode conter diversos modelos que são utilizados conforme as solicitações. Após averiguar a existência de uma anomalia, este serviço encaminha uma mensagem para o serviço de Notificação informando os dados sobre a informação analisada.

- Notificação: ao receber a mensagem do Gerenciador de Detecção de Anomalias, este serviço é responsável por encaminhar a mensagem para a camada de gerenciamento. Nessa fase será verificado qual comportamento será executado, com o propósito de que um ataque ou ameaça seja neutralizado.

4.3.4 Camada de Gerenciamento

Esta camada é responsável por monitorar e gerenciar as configurações do ambiente residencial, bem como colaborar para que as HSUs consigam exercer as suas funções. Os serviços são descritos a seguir:

- Gerente de Atualização: responsável por criar modelos de decisão que reflitam a realidade reportada pela HSU e forneça atualizações condizentes com a realidade do ambiente, ou seja, para que os modelos utilizados pela HSU não fiquem desatualizados caso ocorra uma mudança no ambiente por meio da inclusão ou remoção de algum dispositivo.

- *Backup*: este serviço oferece a HSUs a possibilidade de realizar backup das configurações locais, como políticas de segurança e regras de como o sistema deve se comportar diante de uma determinada situação. A título de exemplo, é possível citar uma situação em que a HSU identifica um novo dispositivo na rede; a decisão de manter ou realizar o bloqueio dependem de configurações do ambiente, dessa forma, todas essas configurações podem ser mantidas em segurança no CSA.

- Notificações: este serviço envia notificações e alertas com sugestões de correções para evitar que o ambiente se torne vulnerável. O envio dos alertas se dá mediante a identificação pelo CSA de uma configuração irregular ou comportamento anômalo na HSU.

A camada de Gerenciamento também engloba as aplicações que controlam os serviços locais do ambiente residencial. A Figura 12 apresenta a troca de mensagens entre as camadas ao detectar uma ameaça. O serviço Notificação consulta a base de dados em busca de uma regra que possa ser aplicada para conter a ameaça, e caso não encontre, é disparada uma notificação para um aplicativo de um membro da família. Ao informar a ação a ser tomada, a aplicação envia uma mensagem para o serviço Notificação, que por sua vez, comunica com a HSU para aplicar uma regra que impeça a ameaça de prejudicar

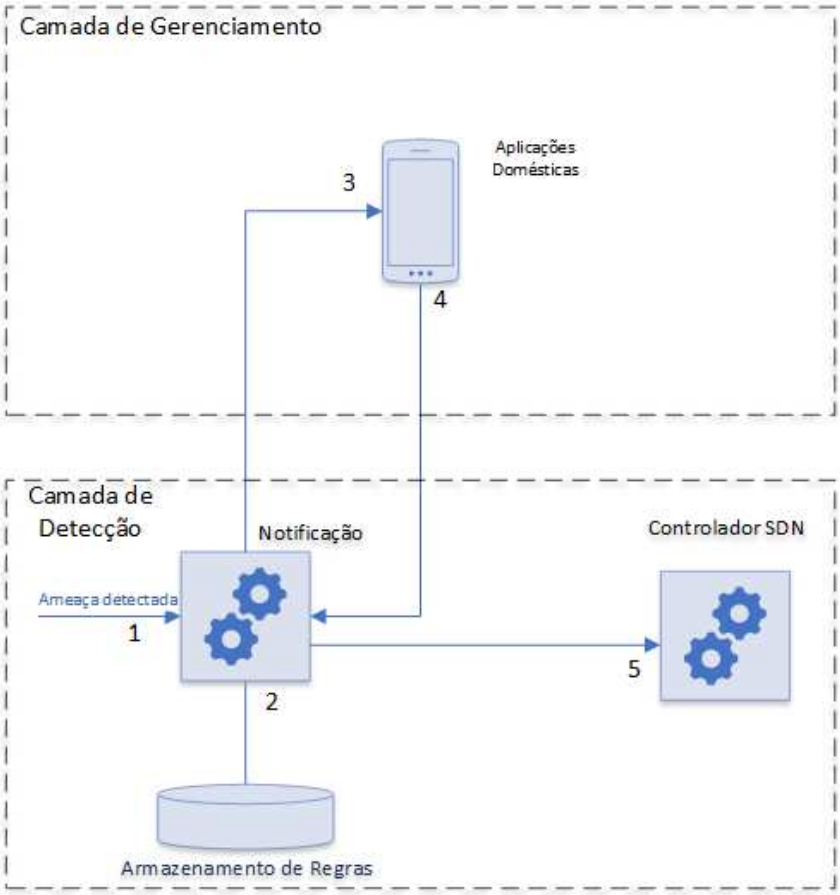


Figura 12 – Processo de detecção de anomalias

os ativos da rede. As trocas de mensagens podem ser armazenadas em logs e serem utilizadas para facilitar uma investigação forense.

Os componentes da arquitetura foram apresentados, no entanto, existe a necessidade de examinar a solução de forma detalhada para verificar se atende a necessidades do ambiente residencial. As soluções apresentadas na Seção 3.1 não exploram todo o fluxo da detecção de anomalias, ou seja, não descrevem a solução completa desde a captura dos pacotes de rede até a predição realizada pelos modelos de inteligência artificial. Baseado nisso, o próximo capítulo apresenta a validação da arquitetura a partir de uma série de experimentos.

Capítulo 5

Experimentos e Análise dos Resultados

Este capítulo apresenta os experimentos que objetivam validar e avaliar a aplicabilidade da FamilyGuard para fornecer proteção adicional ao ambiente residencial. Para os experimentos e as validações, foi utilizada uma versão da FamilyGuard implementada na linguagem de programação Python usando a versão 3.9, disponível no GitHub [1]. Os experimentos discorrem sobre como a proposta pode lidar com os problemas identificados nas soluções que compõem o estado da arte, sendo:

- ❏ Monitoramento de rede do ambiente residencial;
- ❏ Construção e avaliação dos modelos de detecção de anomalias; e
- ❏ Implantação da solução.

Com isso em mente, produziu-se uma investigação empírica e outra exploratória sobre o impacto da FamilyGuard em algumas operações críticas de segurança em residências inteligentes. Nesse contexto, avaliou-se empiricamente três aspectos da FamilyGard:

- ❏ Implementação de funcionalidades do HSU em um hardware de baixo custo e, nessa avaliação, o objetivo é responder perguntas como: A implantação da arquitetura nos ambientes residenciais é simples? O custo de implantação é baixo? Qual o hardware necessário para implantar os componentes de arquitetura?
- ❏ No quesito capacidade dos modelos de aprendizado de máquina para detectar ameaças potenciais na HAN - as seguintes perguntas orientam esta etapa: É possível usar modelos não supervisionados para detectar ameaças no ambiente doméstico? A mistura de tráfego entre IoT e Dispositivos não IoT adicionam complexidade adicional aos modelos? Os benefícios alcançados com os modelos para fornecer mecanismos de proteção adicionais para o residencial ambiente justificam a adoção da arquitetura?

[1] <https://github.com/pedrodamaso/FamilyGuard>

- ❏ Em termos do desempenho do HSU, durante o processo de detecção de anomalias, as seguintes questões direcionam essa investigação: Quanto tempo, em média, o HSU leva para processar um fluxo de rede e emitir uma decisão sobre isso? Este tempo é razoável para a tomada de decisão?

Na investigação exploratória foi analisado o impacto da FamilyGuard ao lidar com as seguintes situações: o risco dos modelos de aprendizado de máquina ficarem desatualizados e não fornecerem uma taxa de detecção de ameaças eficiente devido a mudanças no ambiente, como a adição de novos dispositivos e mudanças no comportamento do tráfego de rede ao longo do tempo. Assim, as seguintes questões são essenciais para mitigar esse risco: Como adicionar ou atualizar os modelos utilizados pela arquitetura? Quem fornecerá esses modelos? Como o modelo será disponibilizado para o HSU?

Por fim, foram analisados os pontos de ameaças existentes no ambiente residencial que podem afetar o funcionamento da arquitetura FamilyGuard. A seguir, serão detalhados cada um dos aspectos analisados na FamilyGuard.

5.1 Implantação do HSU em um hardware de baixo custo

Os componentes HSU, NFG, Controlador SDN, descritos na Seção 4.2, foram implantados em um Raspberry Pi3 Pi 3 Model B Quadcore 1.2ghz, com um cartão de memória de 32 Gigabytes e 1GB de RAM, executando o sistema operacional Raspberry Pi OS. Para usar o paradigma SDN em uma HAN, foi utilizado um roteador TP-LINK TL-WR1043ND v3 com o OpenWrt 18.06 e o *plugin* Open vSwitch 2.8.2. Dessa forma, quando o HSU identifica uma anomalia, é possível agir proativamente enviando uma mensagem ao controlador para bloquear o dispositivo infectado. A Figura 13 mostra a estrutura de implantação em uma HAN da arquitetura FamilyGuard.

Com esta implantação é possível que a arquitetura FamilyGuard seja utilizada em vários ambientes domésticos e com baixo custo; no entanto, não é suficiente determinar se a arquitetura é capaz de fornecer proteção adicional para o ambiente doméstico. Portanto, primeiro é necessário avaliar o desempenho do Raspberry Pi3 recebendo os componentes da arquitetura. Para isso, realizou-se o monitoramento de uso de memória, processador e temperatura enquanto 78.836 fluxos de tráfego de rede eram analisados através dos modelos de inteligência artificial com a ferramenta RPi-Monitor.

A Figura 14 mostra a carga média da CPU desde o início da execução dos componentes da arquitetura até sua finalização. A Figura 15 mostra a quantidade de memória disponível, memória livre e swap. É possível notar que o uso da CPU permanece constante ao longo da execução, e em relação à memória disponível, diminuiu consideravelmente ao carregar os modelos de aprendizado de máquina ao iniciar a HSU. Neste exemplo, cinco

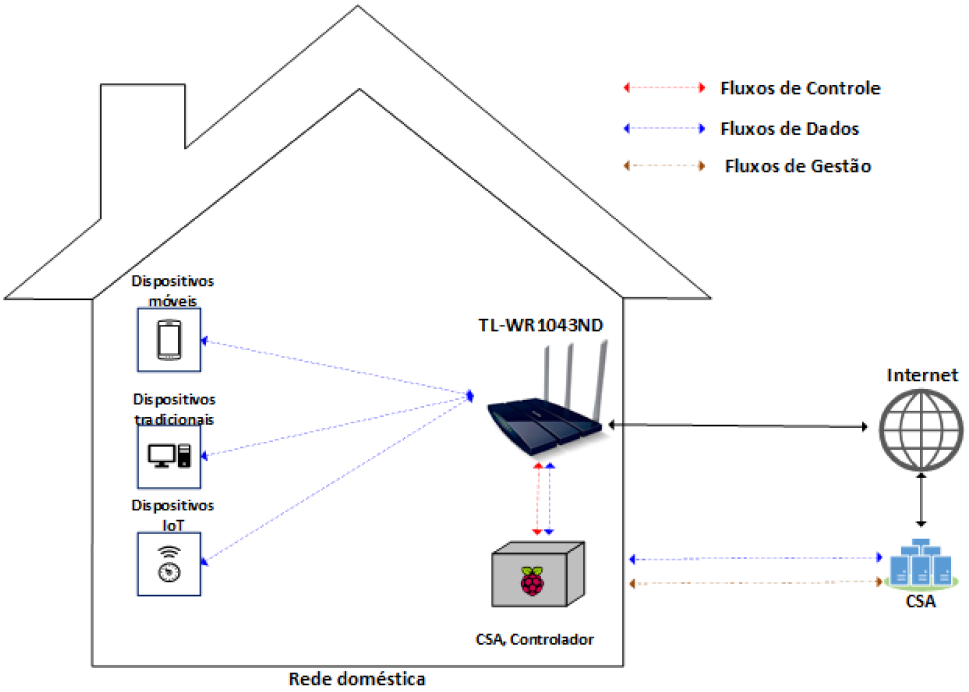


Figura 13 – Implantação em uma rede doméstica.

modelos foram carregados na memória, totalizando cerca de 245 MB. Por fim, a Figura 15 apresenta a temperatura do processador durante a execução do experimento, desse modo, observa-se que a temperatura atingiu 70° e ao concluir o experimento a temperatura volta para 50°.

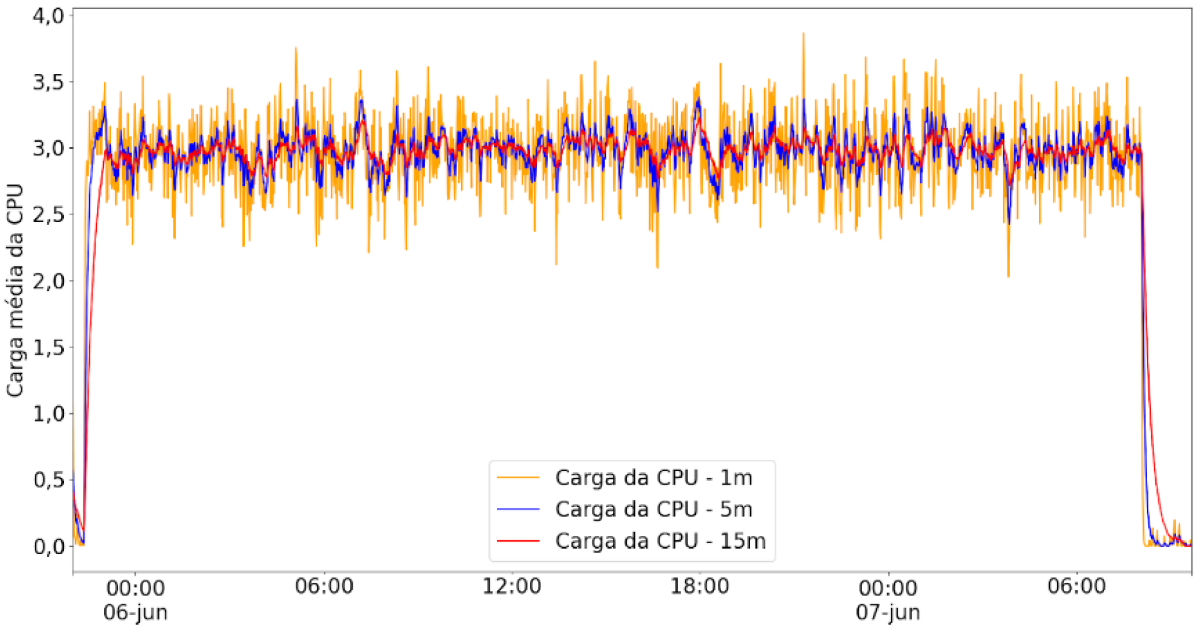


Figura 14 – Desempenho do processador durante a análise de fluxos de rede.

Para o exemplo analisado, o *Raspberry Pi3 Pi 3* Modelo B suportou todos os testes, no entanto, é possível notar que estava com pouca memória RAM disponível e, se fosse

necessário carregar novos modelos, enfrentaria uma limitação de desempenho que poderia comprometer sua eficácia. Portanto, é recomendável o uso do *Raspberry Pi 4* Modelo B com 4 GB de RAM como requisito mínimo para implantar a arquitetura FamilyGuard.

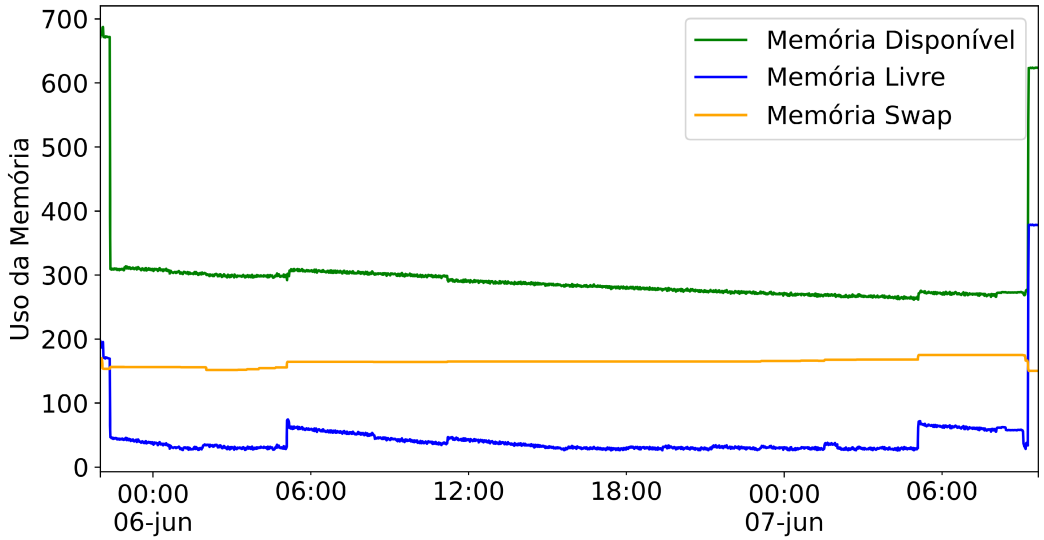


Figura 15 – Uso de memória durante a análise de fluxos de rede.

Foi analisado o desempenho de componentes arquitetônicos em um *Raspberry Pi3 Pi 3* Modelo B, porém, precisamos validar se os modelos de aprendizado de máquina conseguem responder em tempo hábil para detectar ameaças presentes no ambiente, e a partir disso, tomar uma decisão proativa. Assim, a subseção 5.2 fornece detalhes acerca do tempo gasto pelos modelos para identificar uma anomalia.

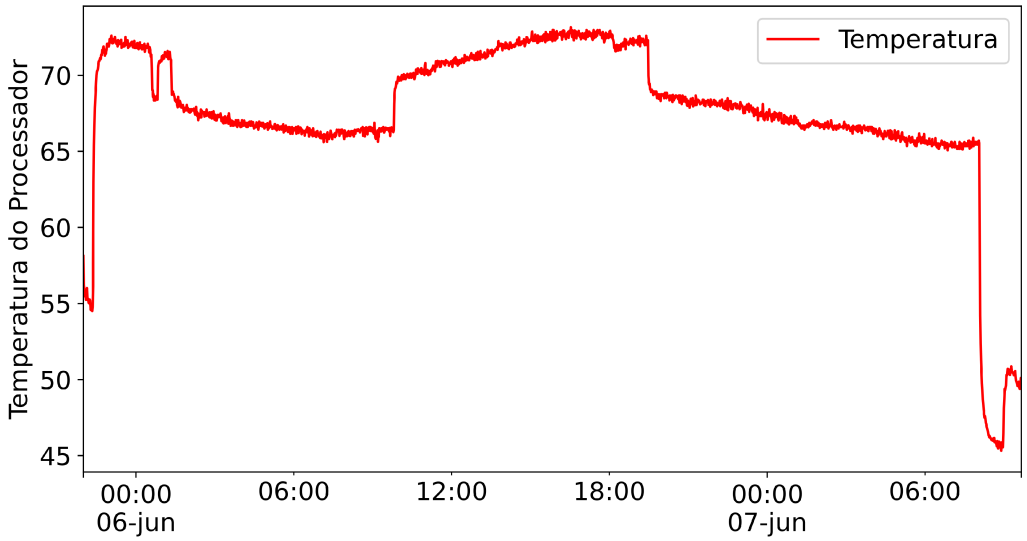


Figura 16 – Temperatura do processador durante a análise de fluxos de rede.

5.2 Usando aprendizado de máquina para detectar anomalias

O objetivo desta Seção é apresentar avaliações de modelos de aprendizado de máquina para detectar ameaças no ambiente residencial. Os modelos apresentados são usados pelos serviços fornecidos pela Camada de Detecção descrita na Subseção 4.3.3.

5.2.1 Bases de Dados

Está subseção descreve todos os procedimentos usados na criação das bases de dados utilizadas no processo e para criação e validação dos modelos de detecção de anomalias.

5.2.1.1 Preparação dos dados

Inicialmente, todas as informações enviadas para análise dos serviços disponibilizados pela Camada de Detecção passam pelo NFG. No decorrer do trabalho, a ferramenta *CICFlowMeter* (LASHKARI et al., 2017) (DRAPER-GIL et al., 2016) foi escolhida para fazer o papel do NFG, pois trata-se de uma ferramenta utilizada em diversas pesquisas e que possui a capacidade de gerar os fluxos do tráfego de rede com mais de oitenta características apresentadas no Anexo A.

A *CICFlowMeter* foi criada pelo *Canadian Institute for Cybersecurity (CIC)* e é capaz de gerar fluxos bidirecionais (Biflow), em que o primeiro pacote determina as direções de avanço (origem ao destino) e retrocesso (destino à origem). Os fluxos *Transmission Control Protocol (TCP)* geralmente são finalizados após o desmembramento da conexão (por pacote FIN) enquanto os fluxos *User Datagram Protocol (UDP)* são finalizados por um tempo limite de fluxo. O valor do tempo limite do fluxo pode ser atribuído arbitrariamente pelo esquema individual, por exemplo 600 segundos para TCP e UDP.

A ferramenta possui o código-fonte disponível e consegue trabalhar de duas formas: fazendo o monitoramento de uma interface de rede ou lendo um arquivo *Packet Capture Data File (PCAP)* para gerar os fluxos que serão analisados. Mediante o exposto, o HSU incorpora a CICFlowMeter, que extraí os fluxos de rede à medida que os pacotes de rede vão chegando na interface e após o fluxo ser encerrado ele é enviado para a análise nos serviços de classificação de tráfego e detecção de anomalias.

Para representar o tráfego de um residência foi necessário coletar dados de diversas fontes; os fluxos envolvem dispositivos IoT e dispositivos multifuncionais ou não-IoT, como *smartphones* e *laptops* (MAZHAR; SHAFIQ, 2020). Por conseguinte, os serviços de classificação de tráfego e detecção de anomalias precisam lidar com o tráfego de ambos os dispositivos. Por isso, foi necessário definir as bases de dados que representam o tráfego para os dois cenários que são apresentadas na Seção 5.2.1.2 e Seção 5.2.1.3.

5.2.1.2 Base de Dados IoT

Inicialmente, foi realizada uma pesquisa cujo objetivo era encontrar dados disponíveis que representassem dispositivos IoT utilizados no contexto residencial. No entanto, existe uma complexidade para encontrar esse dados com as características necessárias para os experimentos, que objetivam ter uma representação próxima do ambiente real.

Em vista disso, Sivanathan et al. (2019) apresenta um ambiente com vinte e oito dispositivos, contendo câmeras, lâmpadas, plugues, sensores de movimento, aparelhos e monitores de saúde. Utilizando essa infraestrutura, os autores coletaram o tráfego de rede gerado por tais dispositivos por um período de seis meses e parte dos dados coletados foram liberados para a comunidade.

O tráfego disponibilizado contém dispositivos IoT e não-IoT, sendo assim, foram removidos todos os dispositivos não-IoT para gerar uma base adequada ao cenário IoT. Os dispositivos removidos foram uma impressora *Hewlett-Packard (HP)*, smartphone android, laptop, MacBook, IPhone, Samsung Galaxy Tab e PIX-STAR Photo-frame, que não foram utilizados no cenário não-IoT, pois a quantidade de fluxos gerados e a diversidade do tráfego não era capaz de representar um ambiente residencial.

Por conseguinte, os dispositivos mantidos são representados na Tabela 4. Após filtrar os dispositivos nos arquivos *PCAPs*, utilizou-se a ferramenta CICFlowMeter para gerar os fluxos que representam os dispositivos IoT. Foram gerados 809.557 fluxos que representam os dispositivos inteligentes utilizados nos experimentos deste trabalho.

5.2.1.3 Base de Dados não-IoT

Para representar o tráfego de dispositivos multifuncionais, como *smartphones* e *laptops* e de outros dispositivos que não sejam IoT, foi utilizada uma base de dados fornecida pelo *CIC* (LASHKARI et al., 2017). A seguir é apresentada uma lista contendo os diferentes tipos de tráfego e aplicativos considerados pelo conjunto de dados:

- Navegação: representa tráfego *Hyper Text Transfer Protocol Secure (HTTPS)* gerado pelos usuários durante a navegação ou execução de qualquer tarefa que inclua o uso de um navegador. Por exemplo, chamadas de voz usando *hangouts*, nas quais, mesmo que a navegação não fosse a atividade principal, capturaram-se fluxos de navegação.

- E-mail: amostras de tráfego geradas usando um cliente *Thunderbird* e as contas do Gmail de Alice e Bob. Os clientes foram configurados para entregar e-mail por *Simple Mail Transfer Protocols (SMTPs)* e recebê-los usando *Post Office Protocol (POP3) / Secure Sockets Layer (SSL)* em um cliente e *Internet Message Access Protocol (IMAP) / SSL* em outro.

Tabela 4 – Dispositivos utilizados para gerar o tráfego IoT.

Dispositivos	Endereço MAC	Tipo de conexão
Smart Things	d0:52:a8:00:67:5e	Cabeada
Amazon Echo	44:65:0d:56:cc:d3	Sem fio
Netatmo Welcome	70:ee:50:18:34:43	Sem fio
TP-Link Day Night Cloud camera	f4:f2:6d:93:51:f1	Sem fio
Samsung SmartCam	00:16:6c:ab:6b:88	Sem fio
Dropcam	30:8c:fb:2f:e4:b2	Sem fio
Insteon Camera	00:62:6e:51:27:2e	Cabeada
Insteon Camera	e8:ab:fa:19:de:4f	Sem fio
Withings Smart Baby Monitor	00:24:e4:11:18:a8	Cabeada
Belkin Wemo switch	ec:1a:59:79:f4:89	Sem fio
TP-Link Smart plug	50:c7:bf:00:56:39	Sem fio
iHome	74:c6:3b:29:d7:1d	Sem fio
Belkin wemo motion sensor	ec:1a:59:83:28:11	Sem fio
NEST Protect smoke alarm	18:b4:30:25:be:e4	Sem fio
Netatmo weather station	70:ee:50:03:b8:ac	Sem fio
Withings Smart scale	00:24:e4:1b:6f:96	Sem fio
Blipcare Blood Pressure meter	74:6a:89:00:2e:25	Sem fio
Withings Aura smart sleep sensor	00:24:e4:20:28:c6	Sem fio
Light Bulbs LiFX Smart Bulb	d0:73:d5:01:83:08	Sem fio
Triby Speaker	18:b7:9e:02:20:44	Sem fio
Nest Dropcam	30:8c:fb:b6:ea:45	Sem fio
TPLink Router Bridge LAN (Gateway)	14:cc:20:51:33:ea	

Adaptação de Sivanathan et al. (2019)

- Bate-papo: aplicativos de mensagens instantâneas, como o Facebook e o Hangouts por meio de navegadores, Skype, IAM e ICQ usando um aplicativo chamado pidgin.
- *Streaming*: aplicativos de multimídia que exigem um fluxo contínuo e constante de dados, foi utilizado o tráfego do *YouTube* (versões HTML5 e flash) e dos serviços *Vimeo* usando *Chrome* e *Firefox*.
- Transferência de arquivos: aplicativos, cujo objetivo principal é enviar ou receber arquivos e documentos. Foram capturadas transferências de arquivos do Skype, sessões de tráfego *File Transfer Protocol (FTP)* sobre *Secure Shell (SSH)* (SFTP) e *FTP* sobre *SSL* (FTPS).
- *Voice over Internet Protocol (VoIP)*: tráfego gerado pelos aplicativos de voz. Foram capturadas chamadas de voz usando o *Facebook*, *Hangouts* e *Skype*.
- *Peer-to-peer (P2P)*: protocolos de compartilhamento de arquivos como o *Bittorrent*. Para gerar esse tráfego foram realizadas algumas tarefas, dentre elas o download de diferentes arquivos .torrent de um repositório público e o monitoramento das sessões de tráfego dos aplicativos *uTorrent* e *Transmission* ao realizar os *downloads*.

Novamente, a ferramenta *CICFlowMeter* foi utilizada para gerar os 110.394 fluxos a partir do conjunto de dados fornecidos pelo *CIC*. À vista disso, os fluxos gerados representam o tráfego de um ambiente residencial sem a presença de anomalias.

5.2.1.4 Base de Dados de Anomalias

Com a finalidade de representar as anomalias, este trabalho utiliza anomalias específicas para os dispositivos IoT e para os dispositivos não-IoT. A fim de representar o conjunto de dados não-IoT foram incluídas amostras de ataques fornecidos por (SHARAFALDIN et al., 2019), que incluem ataques DDoS usando NETBIOS, SYN Flood e UDP Flood. Com finalidade de representar as anomalias IoT, foram utilizados dados fornecidos pelo *Stratosphere Research Laboratory (SRL)* (GARCIA; PARMISANO; ERQUIAGA, 2020) contendo o tráfego das *botnets CoinMiner, Muhstik e Mirai*.

Diante da dificuldade de encontrar amostras de tráfego malicioso para gerar os fluxos, surgiu a necessidade de coletar dados de fontes diferentes, objetivando uma base diversificada para ter um cenário próximo do ambiente residencial.

Dos fluxos gerados para compor a base de dados de anomalias uma pequena parte é usada para criar os cenários experimentais. Dessa forma, a quantidade de fluxos e anomalias usadas são detalhadas em cada experimento, pois tratam-se de uma pequena parte dos fluxos que representam o comportamento normal da rede. Isso acontece para simular o comportamento da rede, em que uma pequena porcentagem do tráfego de rede representa um comportamento malicioso.

5.2.2 Experimentos

Esta subseção apresenta o método para a avaliação dos experimentos e mostra três experimentos: o primeiro utiliza algoritmos não-supervisionados para detecção de anomalias, o segundo opera com algoritmos supervisionados para classificação de fluxos anômalos e o terceiro analisa doze algoritmos não-supervisionados para detecção de fluxos anômalos.

5.2.2.1 Método para a Avaliação

Com o objetivo de avaliar os experimentos serão utilizadas métricas comumente empregadas na literatura para avaliar o desempenho de classificadores (Maloof et. al, 2006). A Tabela 5 apresenta a definição de classificação para cada classe dos modelos. Diante disso, tais definições são usadas para calcular o desempenho dos modelos com base nas seguintes definições:

- TP, ou verdadeiros positivos, representam o número de vezes em que o modelo prevê positivo quando o rótulo de exemplo é positivo.

Tabela 5 – Cálculos a partir de um conjunto de testes de duas classes

		Previsto	
		+	-
Real	+	TP	FN
	-	FP	TN

Fonte: Elaborado pelo autor.

- FN, ou falsos negativos, representam o número de vezes em que o modelo prevê negativo quando o rótulo de exemplo é positivo.
- FP, ou falsos positivos, representam o número de vezes em que o modelo prevê positivo quando o rótulo de exemplo é negativo.
- TN, ou negativos verdadeiros, representam o número de vezes em que o modelo prevê negativo quando o rótulo de exemplo é negativo.

As métricas são calculadas da seguinte forma:

- *Acurácia* é a parte dos exemplos de teste que o modelo prevê corretamente:

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \tag{1}$$

- Taxa de Erro (ER) é a parte dos exemplos no conjunto de testes que o modelo prevê incorretamente:

$$ER = \frac{FN + FP}{TP + FN + FP + TN} \tag{2}$$

- Taxa de Verdadeiro Positivo (TPR), sensibilidade, *recall* ou *revocação* é a parte dos exemplos positivos na qual o modelo prevê corretamente:

$$TPR = \frac{TP}{TP + FN} \tag{3}$$

- Taxa de Verdadeiros Negativos (TNR) é a parte dos exemplos de teste negativos em que o modelo prediz corretamente:

$$TNR = \frac{TN}{TN + FP} \tag{4}$$

- Área sob a curva (AUC) desempenho da abordagem usando todos os limiares de classificação:

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{5}$$

5.2.2.2 Cenários dos experimentos

Para validar as funcionalidades do componente AIWO apresentado na Seção 4.2.1, foi criado um fluxo de trabalho experimental com duas etapas. O primeiro para identificar se um fluxo de rede possui um comportamento normal ou uma anômalo e o segundo para classificar as anomalias que foram identificadas pelo primeiro modelo.

Os objetivos com esses experimentos incluem: avaliar a viabilidade do uso de algoritmos de aprendizado não supervisionado para detectar anomalias no ambiente residencial; o uso de algoritmos supervisionados para classificar os fluxos anômalos; verificar se o desempenho dos modelos podem ser prejudicados com a mistura de fluxos de rede IoT e não-IoT durante o processo de detecção de anomalias; e avaliar o desempenho do HSU durante o processo de detecção de anomalias.

A Figura 17 apresenta uma visão geral da estrutura utilizada para detectar e classificar as anomalias dentro do ambiente residencial. Inicialmente o tráfego da residência é direcionado para o NFG, que é responsável por gerar os fluxos com base nos pacotes de rede que chegam em sua interface, após gerar o fluxo ele é encaminhado para o primeiro modelo para identificar se o fluxo é normal ou se trata de uma anomalia. Caso seja normal, o fluxo é descartado, mas se for um fluxo anômalo, será encaminhado para um segundo modelo para classificar o tipo de anomalia.

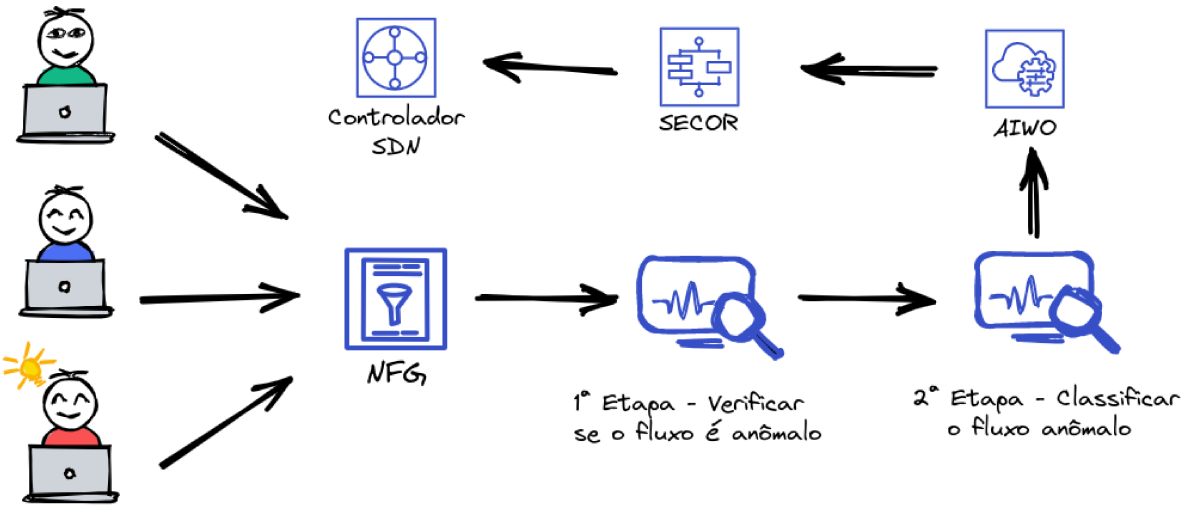


Figura 17 – Desempenho do processador durante a análise de fluxos de rede.

A Tabela 6 apresenta o resumo dos experimentos realizados, contendo a seção, descrição, objetivo e validação de cada experimento.

Para realizar os experimentos, foram criados três conjuntos de dados, ou Casos de Teste (CTs). O primeiro caso de teste (CT1) mistura o tráfego de dispositivos IoT e não-IoT. O segundo (CT2) é derivado do CT1 e contém apenas os exemplos referentes ao tráfego de dispositivos não-IoT. Por fim, no terceiro (CT3) contém apenas as amostras de dispositivos IoT, retiradas do CT1.

Tabela 6 – Resumo dos experimentos realizados.

Seção	Descrição	Objetivo	Validação
5.2.2.3	Detecção de anomalias usando algoritmos não-supervisionados	Verificar se o fluxo é anômalo	Validar um fluxo de trabalho através do AIWO - 1ª fase
5.2.2.4	Classificação de fluxos anômalos usando algoritmos supervisionados	Classificar um fluxo anômalo	Validar um fluxo de trabalho através do AIWO - 2ª fase
5.2.2.5	Explorando algoritmos não-supervisionados para detecção de fluxos anômalos	Comparar modelos de OCC	Validar se os modelos OCC são capazes fornecer proteção adicional para o ambiente residencial

Em todos os experimentos realizados foi aplicado o *Standard Scaler* para padronizar as *features*. Em seguida utilizou-se *PCA* com 35 componentes para reduzir a dimensionalidade. As próximas subseções mostram o processo de criação dos modelos usados na primeira e segunda etapa, que foram apresentadas na Figura 17. Também foram realizadas comparações entre os algoritmos OCSVM, LOF e IF na primeira etapa e KNN, NB e SVC na segunda.

5.2.2.3 Detecção de anomalias usando algoritmos não-supervisionados

Os dados utilizados no processo de geração dos casos de teste são detalhados na Seção 5.2.1. A Tabela 7 resume os casos de teste e os tipos de tráfego que constituem cada um deles, bem como o número de fluxos utilizados na fase de treinamento e teste. A quantidade total de amostras usadas no experimento trata-se da soma das colunas *T. Oversampling* e Teste.

Os algoritmos não-supervisionados OSVM, LOG e IF foram usados em cada um dos CTs presentes na Tabela 8, os parâmetros usados para cada algoritmo estão disponíveis no Anexo A. Os experimentos foram desenvolvidos usando a linguagem Python na versão 3.9 com a biblioteca PyOD 1.0.4 (ZHAO; NASRULLAH; LI, 2019).

Os resultados apresentados na Tabela 8 sugerem que algoritmos não supervisionados apresentam um desempenho satisfatório na detecção de ameaças com base na métrica AUC. O algoritmo LOF teve uma AUC superior a 0,88 para os três casos de teste, com esse valor tem-se indícios de que ele consegue colaborar para a identificação de ameaças para o ambiente residencial.

Por se tratar de algoritmos não supervisionados do tipo *one-class*, a base de dados utilizada na fase de treinamento foi somente de fluxos de rede que representam o tráfego normal de um ambiente residencial. Diante disso, neste experimento avaliou-se seis ameaças, mas espera-se que outras ameaças possam ser identificadas.

Com base na métrica TPR os resultados mostram que a combinação de tráfego de dispositivos IoT e não IoT adiciona complexidade na identificação de comportamentos

Tabela 7 – Casos de teste para criar os modelos não-supervisionados.

Caso	Fluxos	Tipo	Treinamento	T. Oversampling	Teste
CT1	não-IoT	Normal	161.579	272.539	35.907
	DDoS - NetBIOS	Anomalia	-	-	1.196
	DDoS - SYN flood	Anomalia	-	-	1.196
	DDoS - UDP flood	Anomalia	-	-	1.196
	IoT	Normal	290.347	272.539	35.764
	Muhstik	Anomalia	-	-	1193
	Mirai	Anomalia	-	-	1192
	Coinminer	Anomalia	-	-	1192
CT2	não-IoT	Normal	161.579	272.539	35.907
	DDoS - NetBIOS	Anomalia	-	-	1.196
	DDoS - SYN flood	Anomalia	-	-	1.196
	DDoS - UDP flood	Anomalia	-	-	1.196
CT3	IoT	Normal	290.347	272.539	35.764
	Muhstik	Anomalia	-	-	1.193
	Mirai	Anomalia	-	-	1.192
	Coinminer	Anomalia	-	-	1.192

Tabela 8 – Resumo dos resultados utilizando os classificadores OCSVM, LOF e IF

Classificador	Caso de Teste	TPR	TNR	ER	ACC	AUC
OCSVM	CT1	0,4527	0,9007	0,1399	0,8600	0,6767
	CT2	0,8667	0,8985	0,1043	0,8956	0,8826
	CT3	0,3958	0,8993	0,1464	0,8535	0,6476
LOF	CT1	0,9157	0,8684	0,1272	0,8727	0,8920
	CT2	0,9693	0,8561	0,1335	0,8664	0,9127
	CT3	0,8660	0,8980	0,1048	0,8951	0,8820
IF	CT1	0,4654	0,9004	0,1391	0,8608	0,6829
	CT2	0,9200	0,9005	0,0976	0,9023	0,9102
	CT3	0,3936	0,8979	0,1478	0,8521	0,6457

anômalos para todos os algoritmos analisados, obtendo um TPR menor no TC1. Acredita-se que isso acontece por se tratar de dispositivos com um padrão de tráfego diferente e isso pode adicionar uma complexidade nos modelos ao detectar ameaças. Com base nisso, é perceptível a importância de considerar todo o tráfego do ambiente residencial ao propor modelos detecção de anomalias, entretanto, os estudos anteriores mostram grande foco nos tráfego IoT ignorando o tráfego de dispositivos tradicionais ou não deixando claro como ele é tratado na solução.

5.2.2.4 Classificação de fluxos anômalos usando algoritmos supervisionados

Para complementar o cenário apresentado na Figura 17 serão apresentados os experimentos realizados para criar um modelo que seja capaz de identificar a anomalia de um determinado fluxo de rede, para isso foram utilizados três algoritmos: C-Support Vector Classification (SVC), Naive Bayes (NB) e K-ésimo Vizinho mais Próximo (KNN).

A Tabela 9 mostra os casos de teste que foram utilizados para criar os modelos, o tipo de fluxo e a quantidade de fluxos presentes na base de treinamento e teste. Todos os fluxos anômalos usados para criar os três casos de teste são detalhados Seção 5.2.1. O primeiro caso de teste (CT1) possui fluxos anômalos que pertencem a fluxos IoT e não-IoT, o segundo caso de teste (CT2) é derivado do CT1 contendo apenas os fluxos que pertencem aos dispositivos não-IoT, por fim, o terceiro caso de teste (CT3) também é oriundo do CT1 e possui apenas fluxos provenientes de tráfego IoT.

Tabela 9 – Casos de teste para criar os modelos supervisionados.

Caso de Teste	**Fluxos**	**Treinamento**	**T. Oversampling**	**Teste**
CT1	DDoS - NetBIOS	75.007	113.157	18.858
	DDoS - SYN flood	50.695	113.157	12.521
	DDoS - UDP flood	46.680	113.157	11.835
	Muhstik	70.504	113.157	17.523
	Mirai	113.157	113.157	28.316
	Coinminer	19.041	113.157	4.718
CT2	DDoS - NetBIOS	75.007	113.157	18.858
	DDoS - SYN flood	50.695	113.157	12.521
	DDoS - UDP flood	46.680	113.157	11.835
	Muhstik	70.504	113.157	17.523
CT3	Mirai	113.157	113.157	28.316
	Coinminer	19.041	113.157	4.718

O principal objetivo deste experimento é validar as funcionalidades do componente *AI Workflow Orchestrator (AIWO)* apresentado na Seção 4.2.1, que foi implementado em um protótipo usando a linguagem Python. O *AIWO* permite a definição de fluxos de trabalho usando um conjunto de modelos disponíveis. O fluxo definido para testar as funcionalidades da arquitetura proposta é apresentado na Figura 17, resumindo, existe um primeiro modelo que verifica se um fluxo é anômalo e em caso positivo ele é direcionado para outro modelo, com o propósito de que o fluxo seja classificado.

Para este experimento foi utilizado a linguagem Python na versão 3.9 e a biblioteca *scikit-learn*. Os algoritmos utilizados são supervisionados e multi-classe. Os parâmetros usados em cada um dos algoritmos são apresentados no Anexo A. A Tabela 10 apresenta os resultados para cada caso de teste.

Os algoritmos KNN e SVC obtiveram uma AUC superior a 98% para todos os fluxos anômalos, entretanto, para o NB é possível notar uma variação maior na AUC, mas quase todos os fluxos conseguiram atingir uma AUC superior a 90%. Com base na AUC os resultados mostram que os modelos são capazes de classificar os fluxos de tráfego de rede. No entanto, os algoritmos supervisionados precisam de amostras dos fluxos que possuem esse tráfego malicioso. Dessa forma, eles podem se tornar ineficazes ao longo do tempo, pois novas ameaças e variantes de *malware* surgem todos os dias. Nesse cenário, os algoritmos supervisionados ficam prejudicados, e de certa forma existe uma

Tabela 10 – Resumo dos resultados utilizando os classificadores SVC, NB, KNN

Class,	N. Caso	Fluxos	TPR	TNR	ER	ACC	AUC
KNN	CT1	DDoS - NetBIOS	0,99958	0,99987	0,00019	0,99981	0,99972
		DDoS - SYN flood	0,99928	0,99991	0,00017	0,99983	0,9996
		DDoS - UDP flood	0,99848	0,99979	0,00037	0,99963	0,99914
		Coinminer	0,99428	0,99923	0,00102	0,99898	0,99675
		Mirai	0,99778	0,99957	0,00097	0,99903	0,99867
		Muhstik	0,9976	0,99953	0,00083	0,99917	0,99857
	CT2	DDoS - NetBIOS	0,99952	0,99947	0,00051	0,99949	0,99949
		DDoS - SYN flood	0,9992	0,99974	0,00042	0,99958	0,99947
		DDoS - UDP flood	0,99831	0,99943	0,00088	0,99912	0,99887
	CT3	Coinminer	0,99491	0,99839	0,00194	0,99806	0,99665
		Mirai	0,99778	0,99879	0,00178	0,99822	0,99828
		Muhstik	0,99755	0,99912	0,00142	0,99858	0,99833
NB	CT1	DDoS - NetBIOS	0,99761	0,99999	0,00049	0,99951	0,9988
		DDoS - SYN flood	0,92333	0,99893	0,01117	0,98883	0,96113
		DDoS - UDP flood	0,99358	0,99957	0,00118	0,99882	0,99658
		Coinminer	0,4521	0,9747	0,05159	0,94841	0,7134
		Mirai	0,92969	0,99762	0,0229	0,9771	0,96365
		Muhstik	0,90361	0,93686	0,06935	0,93065	0,92024
	CT2	DDoS - NetBIOS	0,99836	0,99417	0,00400	0,99600	0,99626
		DDoS - SYN flood	0,9992	0,99778	0,0018	0,99820	0,99849
		DDoS - UDP flood	0,98276	0,99888	0,00553	0,99447	0,99082
	CT3	Coinminer	0,43027	0,95611	0,09296	0,90704	0,69319
		Mirai	0,61926	0,9991	0,21364	0,78636	0,80918
		Muhstik	0,89773	0,59953	0,29711	0,70289	0,74863
SVC	CT1	DDoS - NetBIOS	0,9991	0,99995	0,00022	0,99978	0,99952
		DDoS - SYN flood	0,99816	0,99986	0,00036	0,99964	0,99901
		DDoS - UDP flood	0,99789	0,9996	0,00062	0,99938	0,99874
		Coinminer	0,99703	0,99143	0,00829	0,99171	0,99423
		Mirai	0,9768	0,9984	0,00813	0,99187	0,9876
		Muhstik	0,9855	0,99903	0,0035	0,9965	0,99227
	CT2	DDoS - NetBIOS	0,99915	0,99975	0,00051	0,99949	0,99945
		DDoS - SYN flood	0,99896	0,99935	0,00076	0,99924	0,99916
		DDoS - UDP flood	0,99814	0,9992	0,00109	0,99891	0,99867
	CT3	Coinminer	0,99576	0,98438	0,01456	0,98544	0,99007
		Mirai	0,97853	0,99919	0,01238	0,98762	0,98886
		Muhstik	0,99121	0,99855	0,00400	0,99600	0,99488

grande complexidade em capturar novas amostras para re-treinar os modelos em um tempo significativamente curto, de modo que seja possível classificar essas ameaças. Para resolver este problema existem abordagens como o aprendizado incremental, aprendizado ativo e mineração de fluxos contínuos de dados (CHOU; JIANG, 2021) (MOLINA-CORONADO et al., 2020).

Apesar de potenciais limitações citadas anteriormente, a criação destes modelos contribuíram para a validação dos componentes da arquitetura e do fluxo de trabalho apre-

sentado na Figura 17. É importante destacar que para proteger o ambiente residencial o mais importante é identificar se um determinado fluxo é anômalo ou não. Com isso já é possível mitigar as ameaças identificadas, mas caso exista a possibilidade de classificá-las, trata-se de uma funcionalidade adicional que a arquitetura fornece através dos serviços fornecidos pelo AIWO.

Os algoritmos *KNN* e *SVC* apresentaram um ótimo resultado com base na métrica AUC. Já o NB teve maior dificuldade para classificar fluxos do *Coinminer* e *Muhstik*. Para este experimento a mistura de fluxos IoT e não-IoT não gerou um impacto no resultado. Dessa forma, conclui-se que para a classificação de tráfego, que trata-se de modelos supervisionados e multi-classe, existem indícios de que a mistura dos fluxos não gera impacto nos resultados dos modelos.

O resultados apresentados na Tabela 10 indicam um ótimo desempenho dos algoritmos supervisionados, mas pelo fato de sempre precisarem de novas amostras e um constante re-treino dos modelos, a próxima subseção explora o desempenho de doze algoritmos do tipo *one-class* em três casos de teste.

5.2.2.5 Explorando algoritmos não-supervisionados para detecção de fluxos anômalos

As contribuições deste experimento consistem em avaliar seis tipos de anomalias que podem prejudicar dispositivos e pessoas que vivem em um ambiente residencial. Em segundo lugar, avaliou-se o desempenho de doze algoritmos do tipo OCC para o problema de detecção de anomalias, que foram utilizados em redes domésticas a partir de três cenários diferentes: i) mesclando fluxos de tráfego de rede de dispositivos IoT e não-IoT, para avaliar se a detecção de anomalias se torna mais complexa quando tiver tráfego de diferentes classes de dispositivos; ii) procurou-se avaliar a detecção de anomalias usando apenas fluxos de dispositivos não-IoT; e iii) utilizou-se apenas fluxos de dispositivos IoT. Terceiro, com cenários i), ii) e iii) avaliou-se se há benefícios na detecção de anomalias ao criar modelos que separam o tráfego não IoT do tráfego IoT

Similar ao estudo apresentado na subseção 5.2.2.3, também foram utilizadas as mesmas ameaças, no entanto, houve alterações nas quantidades de fluxos e dos algoritmos usados nos testes. Pois, acredita-se que o uso de apenas três algoritmos seja insuficiente para afirmar que os algoritmos não-supervisionados do tipo OCC trazem benefícios para o ambiente residencial.

Os fluxos usados nos experimentos são apresentados na Tabela 11, que mostra a quantidade e os tipos de fluxos usados em cada teste. Para cada caso de teste avaliou-se o desempenho de doze algoritmos do tipo *one-class*, que estão disponíveis na biblioteca PyOD (ZHAO; NASRULLAH; LI, 2019) para a detecção de anomalias. Os parâmetros usados em cada um dos algoritmos estão descritos no Anexo A.

Os algoritmos analisados são: *Minimum Covariance Determinant (MCD), One-class SVM (OCSVM), Local Outlier Factor (LOF), Clustering Based Local Outlier Factor (CBLOF), Histogram-based Outlier Score (HBOS), K-Nearest Neighbours (KNN), Average K-Nearest Neighbours (AKNN), Median K-Nearest Neighbours (MKNN), Copula-Based Outlier Detection (COPOD), Isolation Forest (IF), Feature Bagging (FG) e Lightweight On-line Detector of Anomalies (LODA)*

Tabela 11 – Casos de teste usados para avaliar os algoritmos não-supervisionados

Caso	**Fluxos**	**Tipo**	**Treinamento**	**T. Oversampling**	Teste
CT1	não-IoT	Normal	161.579	290.347	17.954
	DDoS - NetBIOS DDoS - SYN flood DDoS - UDP flood	Anomalia	-	-	1.795
	IoT	Normal	290.347	290.347	17.956
	Muhstik Mirai Coinminer	Anomalia	-	-	1.796
CT2	não-IoT	Normal	161.579	290.347	17.954
	DDoS - NetBIOS DDoS - SYN flood DDoS - UDP flood	Anomalia	-	-	1795
CT3	IoT	Normal	290.347	290.347	17.956
	Muhstik Mirai Coinminer	Anomalia	-	-	1.796

O objetivo deste experimento é avaliar a usabilidade de algoritmos OCC para detectar anomalias de rede nesse ambiente. A Tabela 12 resume os resultados dos experimentos. Para avaliar a utilidade desses algoritmos no cenário residencial faz-se necessária a avaliação de cada caso de teste separadamente:

- Identificando anomalias no tráfego IoT e não IoT (CT1): considerando a métrica AUC, os classificadores tiveram mais dificuldade de detectar anomalias ao misturar os tráfegos dos dispositivos IoT e não-IoT. A Figura 18 esclarece este comportamento, mostrando a AUC para cada um dos algoritmos analisados. O MCD, LODA e HBOS obtiveram a pior taxa de detecção. Já os classificadores LOF, KNN e FB, por outro lado, tiveram uma taxa de detecção razoável, com base na AUC e na taxa de erro apresentada na Figura 19. Na Tabela 13 é possível notar o tamanho do modelo gerado por cada classificador, considerando isso é possível afirmar que para o CT1, o KNN foi o classificador mais adequado para detectar anomalias.

- Identificando anomalias no tráfego não IoT (CT2): para reconhecer anomalias no tráfego não-IoT (CT2), o desempenho dos classificadores melhoraram quando comparados ao CT1. Esse resultado sugere que as anomalias não-IoT escolhidas podem

Tabela 12 – Resumo dos resultados utilizando os algoritmos não-supervisionados

Class.	**Categoria**	**Nº Caso**	**TPR**	**TNR**	**ER**	**ACC**	**AUC**
MCD	Linear Model	CT1	0,1590	0,8983	0,1688	0,8311	0,5286
		CT2	0,2668	0,8987	0,1586	0,8413	0,5827
		CT3	0,6859	0,8968	0,1223	0,8776	0,7913
OCSVM	Linear Model	CT1	0,7287	0,8982	0,1171	0,8828	0,8135
		CT2	0,9013	0,8984	0,1013	0,8986	0,8998
		CT3	0,9626	0,9011	0,0932	0,9067	0,9319
LOF	Proximity-Based	CT1	0,9947	0,8936	0,0971	0,9028	0,9441
		CT2	0,9961	0,8840	0,1057	0,8942	0,9400
		CT3	0,8079	0,8989	0,1093	0,8906	0,8534
CBLOF	Proximity-Based	CT1	0,7365	0,8986	0,1160	0,8839	0,8176
		CT2	0,9114	0,8980	0,1007	0,8992	0,9047
		CT3	0,9610	0,8994	0,0949	0,9050	0,9302
HBOS	Proximity-Based	CT1	0,4230	0,8996	0,1436	0,8563	0,6613
		CT2	0,9348	0,8984	0,0982	0,9017	0,9166
		CT3	0,9482	0,9006	0,0950	0,9049	0,9244
KNN	Proximity-Based	CT1	0,9896	0,8843	0,1060	0,8939	0,9370
		CT2	0,9938	0,8782	0,1112	0,8887	0,9360
		CT3	0,9966	0,8983	0,0927	0,9072	0,9474
AVG-KNN	Proximity-Based	CT1	0,9935	0,8583	0,1293	0,8706	0,9259
		CT2	0,9966	0,8349	0,1503	0,8496	0,9158
		CT3	0,9966	0,9003	0,0908	0,9091	0,9485
MED-KNN	Proximity-Based	CT1	0,9913	0,8553	0,1322	0,8677	0,9233
		CT2	0,9961	0,8329	0,1522	0,8477	0,9145
		CT3	0,9966	0,9007	0,0905	0,9094	0,9487
COPOD	Probabilistic	CT1	0,7357	0,8998	0,1151	0,8848	0,8177
		CT2	0,9576	0,9011	0,0937	0,9062	0,9293
		CT3	0,9604	0,9191	0,0770	0,9229	0,9398
IF	Outlier Ensembles	CT1	0,7270	0,8991	0,1164	0,8835	0,8131
		CT2	0,9799	0,8990	0,0935	0,9064	0,9395
		CT3	0,9587	0,8986	0,0958	0,9041	0,9287
FB	Outlier Ensembles	CT1	0,9963	0,8935	0,0971	0,9028	0,9449
		CT2	0,9961	0,8839	0,1058	0,8941	0,9400
		CT3	0,8123	0,8984	0,1093	0,8906	0,8554
LODA	Outlier Ensembles	CT1	0,2915	0,9071	0,1488	0,8511	0,5993
		CT2	0,5988	0,9033	0,1243	0,8756	0,7510
		CT3	0,6909	0,8998	0,1191	0,8808	0,7953

ter um padrão de tráfego mais fácil de distinguir do tráfego normal. Os algoritmos MCD e LODA obtiveram a pior taxa de detecção, com base na AUC demonstrada na Figura 18 e na taxa de erro apresentada na Figura 19. Os classificadores LOF, KNN, IF e FB, por outro lado, tiveram uma boa taxa de detecção. Levando em consideração a AUC e o tamanho do modelo apresentado na Tabela 13, os resultados sugerem que o classificador mais adequado para o cenário não-IoT é o IF.

- Identificando anomalias no tráfego de IoT (CT3): ao distinguir as anomalias no tráfego de IoT (CT3), de forma geral, os algoritmos obtiveram melhores resultados em CT3 do que CT2 e CT1, ou seja, detectar anomalias de IoT quando há apenas tráfego normal de IoT beneficia o modelo de decisão, de acordo com a AUC exemplificada na Figura 18 e na taxa de erro apresentada na Figura 19. O MCD e LODA têm os piores desempenhos com AUC de $0,79$ e $0,80$, respectivamente. Os algoritmos KNN, AKNN e MKNN, por outro lado, tiveram uma boa taxa de detecção. Considerando a AUC e o tamanho do modelo apresentado na Tabela 13, os resultados sugerem que o classificador mais adequado para o cenário não-IoT é o MKNN.

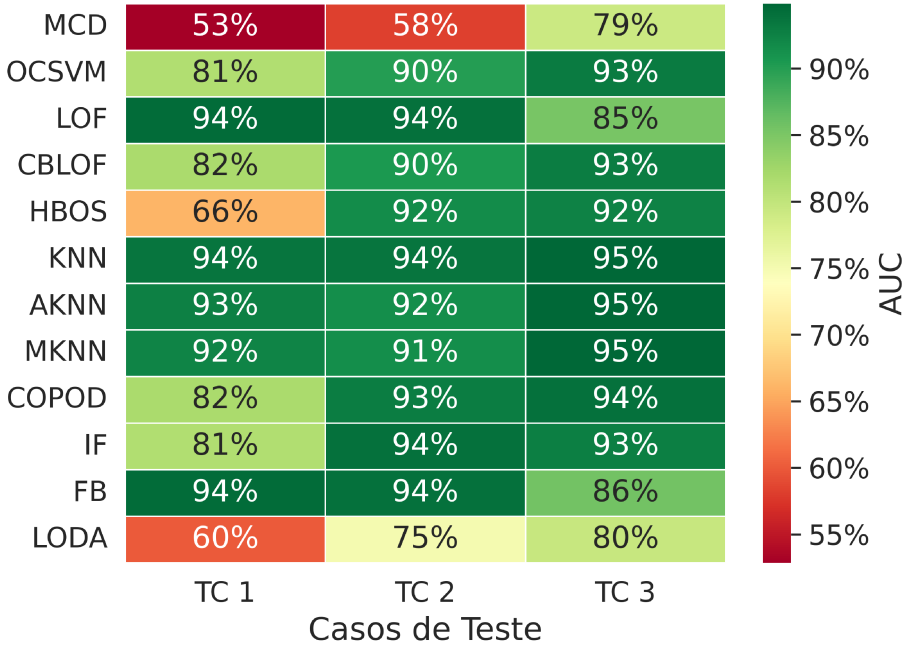


Figura 18 – AUC para cada caso de teste

Com base na métrica AUC, os classificadores MCD, OCSVM, CBLOF, HBOS, COPOD, IF, LODA obtiveram melhores taxas de detecção em CT2 e CT3 do que CT1. Ou seja, separar o tráfego IoT e não-IoT é benéfico para esses modelos de decisão. Os algoritmos KNN, AKNN, MKNN, por exemplo, obtiveram uma excelente taxa de detecção em CT3; porém, perdem desempenho em CT1 e CT2; assim, tais modelos podem ser adequados para cenários com apenas tráfego de rede IoT. Os classificadores LOF e FB, por outro lado, obtiveram bons resultados em CT2 e CT1, mas quando expostos apenas ao tráfego IoT, apresentaram queda em seu desempenho.

É importante notar que alguns classificadores têm melhor desempenho dependendo do tipo de tráfego disponível para treinamento. Por exemplo, LOF, CBLOF, IF, FB obtiveram o melhor desempenho para dispositivos não-IoT, enquanto MCD, OCSVM,

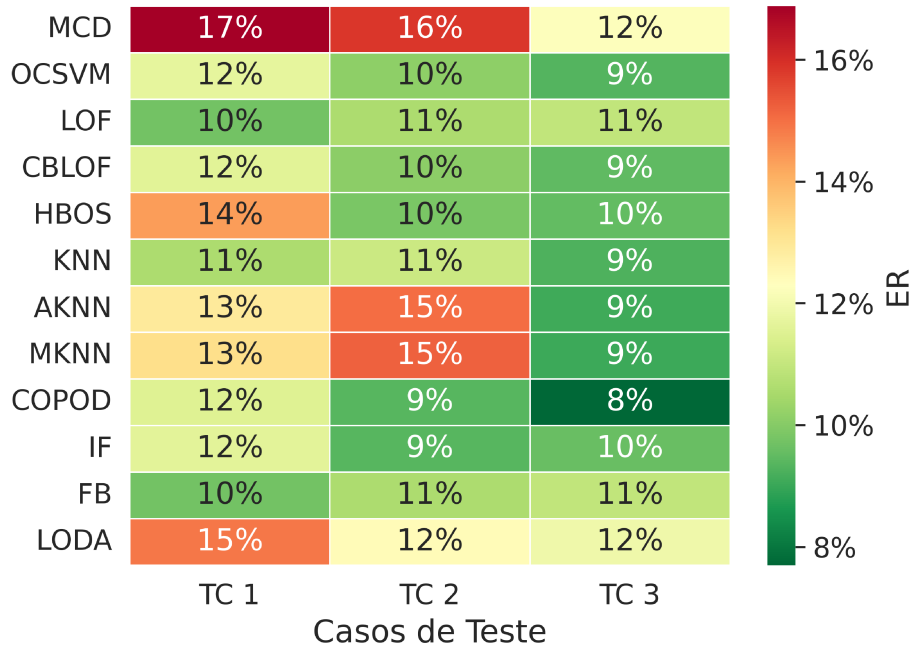


Figura 19 – ER para cada caso de teste

Tabela 13 – Tamanho dos modelos em megabytes (MB)

	Tamanho dos modelos (MB)		
Classificador	**CT1**	**CT2**	**CT3**
MCD	10,5	5,2	5,2
OCSVM	77,8	38,9	38,9
LOF	241,6	120,8	120,8
CBLOF	11,6	5,8	5,8
HBOS	9,3	4,7	4,7
KNN	152,4	76,2	76,2
AKNN	152,4	76,2	76,2
MKNN	152,4	76,2	76,2
COPOD	1228,8	590	590
IF	9,9	5,2	5,3
FB	2355,2	1126,4	1126,4
LODA	9,3	4,7	4,7

HBOS, KNN, AVG-KNN, MED-KNN, COPOD e LODA obtiveram um melhor resultado no tráfego IoT. Este resultado sugere que uma solução única pode não existir para a detecção de anomalias no ambiente residencial.

Com base na taxa de erro apresentada na Figura 19, é possível observar que os classificadores apresentam uma taxa de erro maior quando misturamos os fluxos de tráfego de rede dos dispositivos IoT e não IoT. Outro ponto importante é que os classificadores podem obter uma taxa de erro menor ao detectar anomalias usando apenas fluxos IoT do que cenários não-IoT.

Os experimentos mostram que a separação de tráfego é benéfica para dez dos doze classificadores analisados. Assim, eles conseguem obter uma melhor taxa de detecção de anomalias e uma taxa de erro menor quando treinados com um tipo específico de tráfego normal. Esse resultado indica que separar e analisar cada tipo de tráfego pode ser uma boa estratégia para construir sistemas de detecção de anomalias para ambientes de casas inteligentes. Adicionalmente, os resultado obtidos são coerentes com os experimentos realizados na Seção 5.2.2.3

Por fim, é importante destacar que testes estatísticos precisam ser realizados para comprovar efetivamente se existe diferença de desempenho entre os dois cenários. Também não é possível afirmar que o comportamento continuará sendo propagado à medida que novas anomalias surgirem no tráfego de rede. Ademais, a complexidade do tráfego aumenta à medida que novos dispositivos são adicionados no ambiente.

5.3 Tempo gasto durante a detecção de anomalias

Para avaliar se os modelos conseguem identificar uma ameaça em um tempo viável para que seja possível tomar alguma ação preventiva, analisou-se três casos de teste apresentados na Seção 5.2.2.3, monitorando o tempo que os modelos levam para fazer sua previsão; o tempo analisado foi desde o instante em que o fluxo de tráfego de rede chega ao HSU, até enviar uma resposta com o resultado. Na Figura 20 é possível observar que para estimar o tempo de analise do fluxo de rede coletou-se o tempo inicial, no momento em que fluxo chega ao HSU, depois de realizar o pré-processamento e analise do fluxo uma resposta é enviada ao SECOR, neste instante de tempo coletou-se o tempo final.

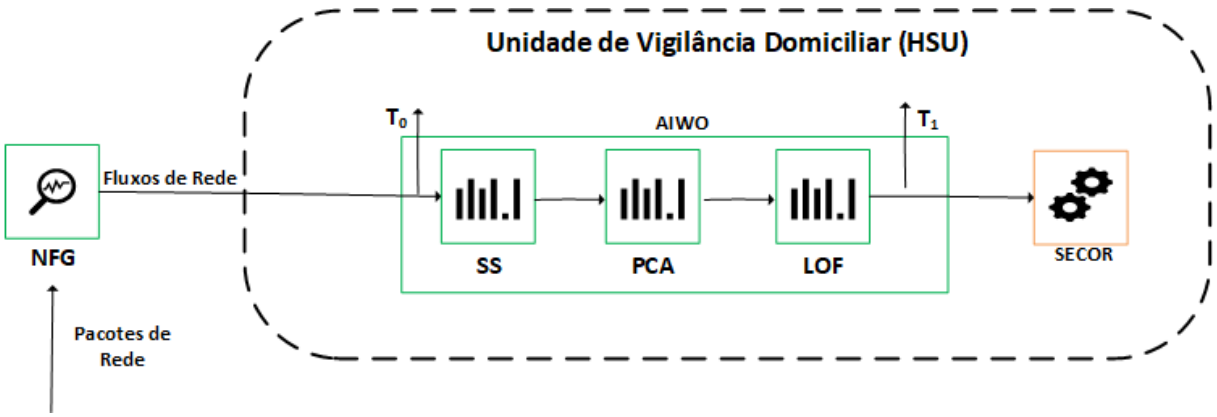


Figura 20 – Local onde foi estimado o tempo de analise dos fluxos de rede.

Por fim, utilizou-se um fluxo de trabalho contendo o modelo criado pelo *LOF* para identificar anomalias. A Figura 21 mostra o tempo médio desde a chegada do fluxo até a predição. Para CT1 foram analisados 78.836 fluxos, no CT2 39.495 e no CT3 39.341. Detalhes sobre cada um dos casos de teste são apresentados na Subseção 5.2.2.3.

É possível notar que o tempo médio para a predição de um fluxo está entre 0,37 e 0,55 segundos. Considerando o equipamento de baixo custo testado e suas limitações os tempos encontrados podem ser considerados satisfatórios

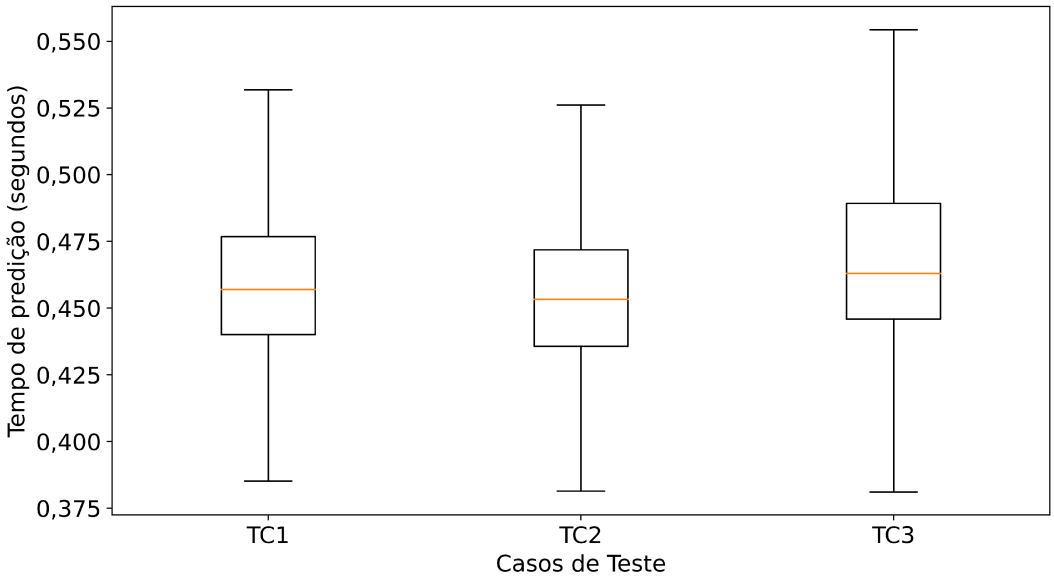


Figura 21 – Tempo médio desde a chegada do fluxo até a detecção de anomalias.

Adicionalmente, foi realizado o monitoramento dos casos de teste apresentados na Subseção 5.2.2.4 usando o modelo criado pelo KNN, ou seja, além de realizar o processo descrito na Figura 20 foi incluído um novo modelo para classificar uma anomalia que foi identificada. A Figura 22 mostra o tempo médio gasto para identificar e classificar uma anomalia. A predição de todo o fluxo de trabalho está entre 0,65 e 1,05 segundos, nota-se uma diferença média de 0,20 segundos do TC1 para os casos TC2 e TC3. Essa diferença no tempo de predição entre os casos de teste trata-se do cálculo da distância dos componentes realizado pelo KNN.

É importante avaliar quanto tempo é gasto para realizar a predição de um fluxo com o objetivo de identificar as anomalias e classificá-las. Com base nos dois cenários apresentados é possível definir dois fluxos de trabalho: O primeiro representado pela Figura 20 e o segundo adicionando o modelo KNN para classificar as ameaças identificadas. Dessa forma, a Figura 23 faz uma comparação entre o tempo dos fluxos de trabalho.

Percebe-se que do primeiro fluxo de trabalho para o segundo foi adicionado em média 0,20 segundos no tempo de processamento. Dessa forma, é importante definir fluxos de trabalho que são menores para não causar impacto no tempo de predição. Portanto, classificar as ameaças identificadas permite que uma mitigação mais assertiva, desse modo, é importante analisar o custo beneficio de cada cenário ao adotar os modelos de classificação de fluxos.

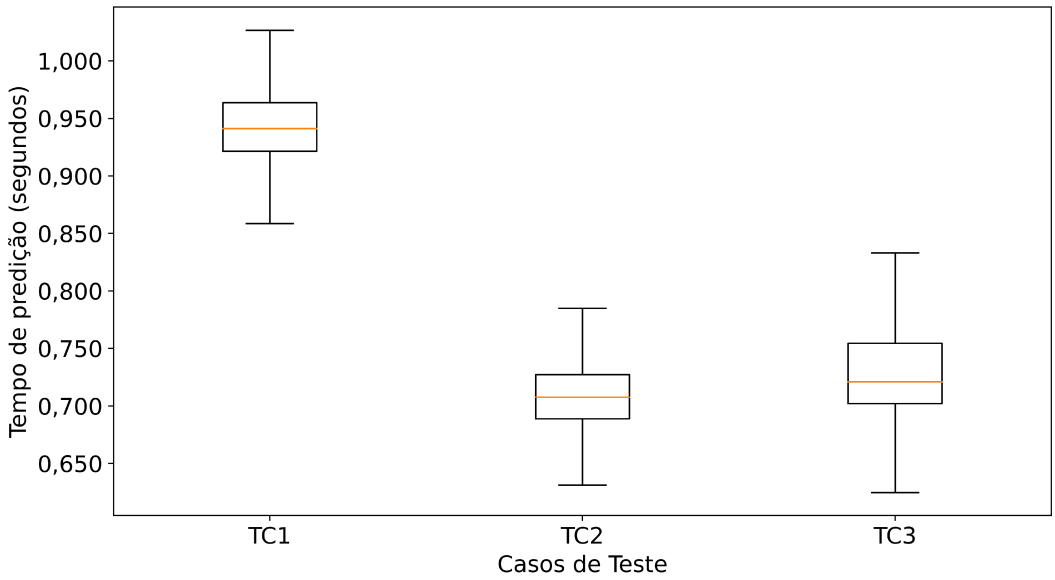


Figura 22 – Tempo médio desde a chegada do fluxo até a classificação de anomalias.

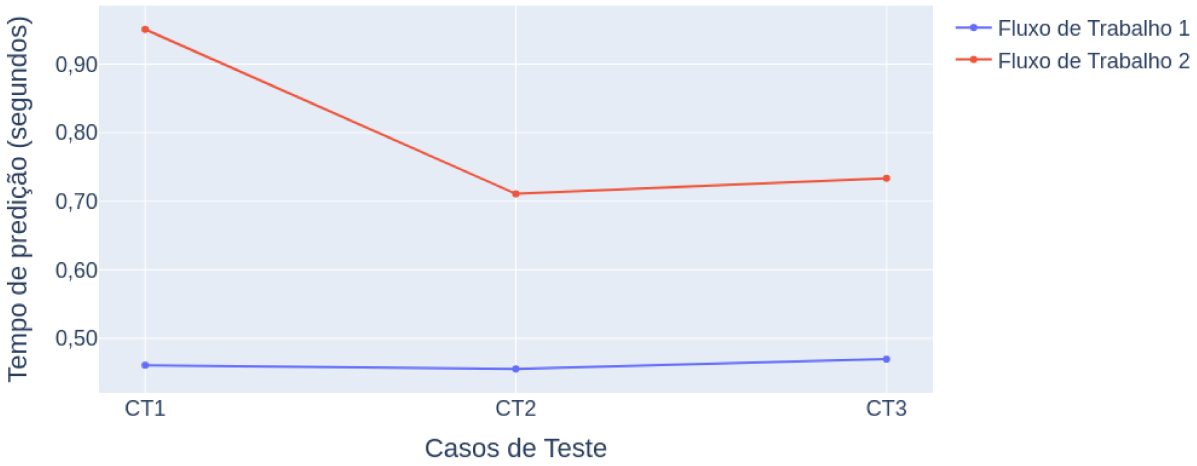


Figura 23 – Comparação do tempo médio de detecção e classificação das anomalias.

Foi avaliada a capacidade do Raspberry Pi3 Pi 3 Modelo B para suportar os componentes arquiteturais e o tempo de resposta que os modelos levam para realizar uma previsão. No entanto, outro fator de grande importância para proporcionar um ambiente residencial com mais segurança e privacidade trata-se da capacidade da arquitetura em lidar com as mudanças do ambiente. Por exemplo, adicionando um dispositivo conhecido à rede, desta forma, o caso de uso da subseção 5.4 busca descrever a solução proposta da arquitetura FamilyGuard.

5.4 Atualização dos modelos

Para lidar com a heterogeneidade do ambiente residencial e a inclusão de novos dispositivos e protocolos, a HSU conta com o auxílio do CSA descrito na subseção 4.2.2, que tem como objetivo auxiliar um grupo de HSUs na detecção de anomalias.

Os modelos utilizados pelo HSU não são atualizados com base no tráfego residencial, pois existe um grande risco que os modelos aprendam com algum tráfego malicioso existente. Outro motivo que contribui para terceirizar a criação de modelos de detecção é que eles demandam muito poder computacional, inviabilizando o uso de um dispositivo de baixo custo como o Raspberry Pi3. Desta forma, transferimos a responsabilidade dos modelos para o CSA.

O CSA é fornecido por provedores de serviços de segurança, que têm a função de criar os modelos de aprendizado de máquina e disponibilizá-los no HSU. Considere que a criação de modelos está sob responsabilidade de provedores de serviços de segurança especializados, pois precisa ser definida sem qualquer intervenção de anomalia.

Hipoteticamente, um ambiente residencial possui um modelo de detecção de anomalias para dispositivos IoT. Esse ambiente possui três dispositivos (*Amazon Echo, Netatmo Welcome e Belkin wemo motion sensor*), no entanto, com base na necessidade dos membros do ambiente residencial, um novo dispositivo IoT foi adicionado: um monitor de bebê inteligente. O modelo de detecção precisa ser atualizado para não confundir o tráfego normal do novo dispositivo coletado com a anomalia.

O usuário residencial pode entrar em contato com a operadora e solicitar um modelo adequado, incluindo todos os dispositivos que uma residência possui, ou as operadoras podem oferecer modelos baseados na identificação do tráfego residencial, tornando esse processo automático. Por exemplo, ao identificar um novo dispositivo na rede, o prestador de serviço de segurança já envia um modelo atualizado para o ambiente.

A Figura 24 mostra as conexões *websocket* entre o CSA e três HSUs para que ambas as partes possam enviar mensagens a qualquer momento. Quando um novo modelo está disponível para uso, o CSA pode enviar uma mensagem aos HSUs informando sobre a nova versão disponível. O HSU pode baixar o modelo e notificar os moradores de uma residência para reiniciar o serviço para que o novo modelo possa ser carregado e usado.

5.5 Considerações sobre os resultados

Embora o Capitulo 5 tenha mostrado a validação de várias funcionalidades e discutido como a arquitetura proposta avança o estado da arte, é importante destacar algumas limitações e ameaças à validade da proposta.

Na primeira validação, descrita na Seção 5.1, é apresentada a simplicidade da arquitetura implantando-a em hardware de baixo custo. Foi realizado o monitoramento do

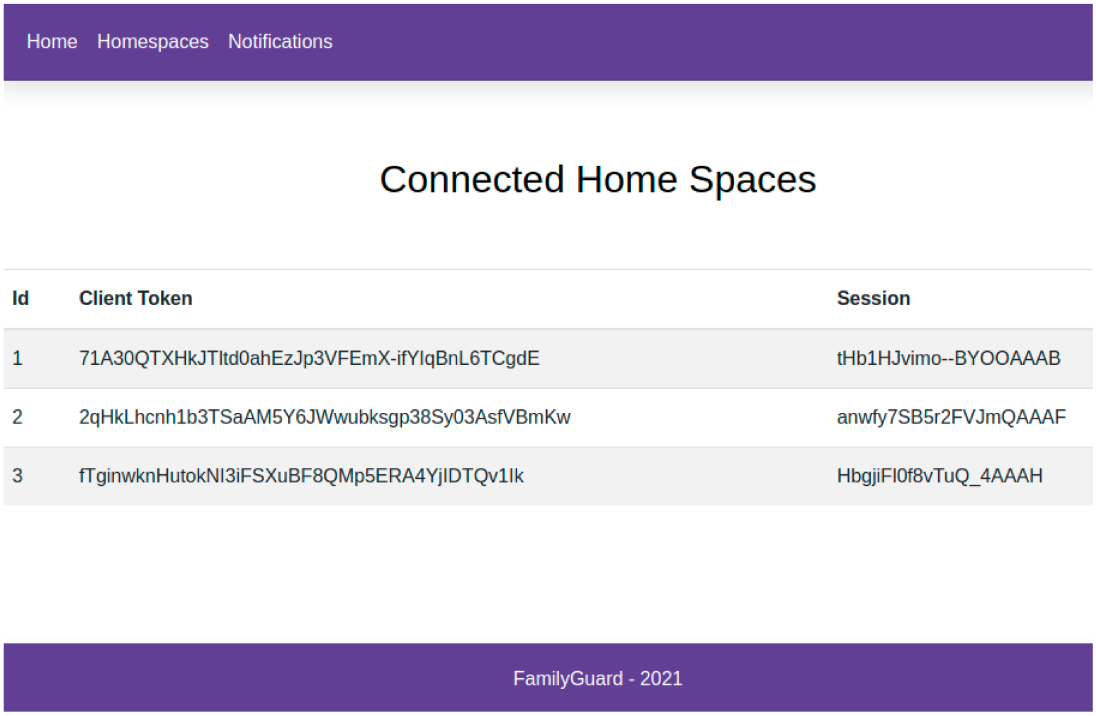


Figura 24 – HSUs conectados ao CSA

comportamento de um Raspberry Pi3 Modelo B durante a análise de 78.836 fluxos de tráfego de rede. Embora tenhamos demonstrado que é possível utilizar os componentes da arquitetura com um dispositivo de baixo custo, mas não é possível garantir que esse comportamento será o mesmo de outros modelos disponíveis no mercado.

Perante o exposto, realizou-se a implantação da arquitetura em um raspberry Pi3 Pi 3 Model B, Quadcore 1.2ghz com um cartão de memória de 32 Gigabytes e 1GB de RAM. Dessa forma, conclui-se que a implantação da arquitetura nos ambientes residenciais é simples e que pode ser feita com um hardware de baixo custo facilitando a sua adoção. A partir disso, destaca-se o custo/beneficio ao implantar a arquitetura em um *hardware* de baixo custo e o nível de proteção adquirido ao identificar e mitigar ameaças através dos modelos de detecção de anomalias.

Na segunda validação apresentada na Seção 5.2, foram descritos os experimentos usando modelos de aprendizado de máquina para detectar ameaças no ambiente residencial. Neste experimento, foi utilizado modelos não supervisionados para detectar ameaças. No entanto, o número de ameaças nos experimentos é muito pequeno para determinar se o modelo ainda é válido em um cenário com várias ameaças, desse modo, mais experimentos e estudos são necessários. Da mesma forma, é possível dizer que separar o tráfego IoT do não-IoT facilita a identificação de ameaças nos experimentos realizados. Ainda assim, o número de ameaças e a quantidade de dados utilizados no estudo não garantem o mesmo comportamento em um cenário mais complexo.

O ponto mais crítico desta validação foi utilizar um modelo para detectar ameaças em tempo real no ambiente residencial para validar a comunicação entre os componentes da

arquitetura. Uma limitação grave encontrada durante esta validação experimental foi a falta de conjuntos de dados públicos contendo amostras normais e de ataque de tráfego residencial. Contornou-se esse problema criando um conjunto de dados. No entanto, é importante avaliar a arquitetura FamilyGuard usando outros conjuntos de dados.

Em vista disso, apesar das limitações apresentadas, conclusões importantes devem ser destacadas por contribuírem com o estado da arte. A primeira delas é a possibilidade de utilizar modelos não-supervisionados para detectar ameaças no ambiente doméstico, além disso, vale a pena mencionar o uso inédito dos algoritmos do tipo *one-class* no ambiente residencial. A segunda, com base nos resultados, sugere que a mistura de tráfego IoT e não-IoT adiciona uma complexidade nos modelos de detecção de ameaças. A terceira, por sua vez, mostra que é possível classificar ameaças que já são conhecidas. Por fim, depois de experimentar diversos modelos, conclui-se que é possível fornecer proteção adicional para o ambiente residencial, o que justifica a adoção da arquitetura.

A terceira validação apresentada na Seção 5.3 tem como objetivo medir o tempo gasto para detectar e classificar as anomalias. O tempo médio gasto ficou entre 0,37 e 0,95 segundos. No entanto, foi utilizado um Raspberry Pi3 Modelo B. Dessa forma, não é possível garantir o mesmo desempenho do dispositivo em um cenário onde o tráfego de rede é muito intenso. Neste caso, novos testes podem ser realizados para testar os limites do dispositivo e quantidade de fluxos que consegue processar em um cenário adverso.

Posto isto, o experimento permite ter uma visão sobre o tempo necessário para emitir uma decisão sobre um determinado fluxo. Dessa forma, chega-se a conclusão de que esse tempo é factível com o uso de um dispositivo de baixo custo e que este tempo pode ser reduzido significativamente através um dispositivo com uma quantidade maior de memória e um poder de processamento melhor.

Portanto, conforme mencionado na última validação apresentada na Seção 5.4, o objetivo deste experimento foi demonstrar como os modelos de aprendizado de máquina são atualizados para lidar com mudanças no comportamento do tráfego de rede doméstica, que são causadas pela adição de novos dispositivos e protocolos. Atualizar os modelos usados pela arquitetura é simples. No entanto, a principal preocupação é com a qualidade dos modelos e se eles serão eficazes para o ambiente residencial. Não foi apresentado neste trabalho, mas os componentes propostos na arquitetura CSU e HSA permitem a implementação de modelos colaborativos de detecção de ameaças, solução que pode melhorar a qualidade dos modelos disponíveis para os ambientes residenciais.

Perante o exposto, com base em uma investigação exploratória, foi possível mostrar como a FamilyGuard é capaz de lidar com a atualização dos modelos e como podem ser disponibilizados ao HSU. Este é um passo importante, pois é apresentada uma forma de retro-alimentar os modelos de detecção de ameaças, concluindo um ciclo, que começa desde de a captura dos pacotes, geração dos fluxos, detecção de anomalias através dos modelos e suas atualizações, para que sejam capazes de identificar novas ameaças.

Capítulo 6

Conclusão

Foi uma longa jornada com inúmeros aprendizados, muitas realizações e com a expectativa de aprofundar as pequisas. Neste capítulo serão apresentadas as conclusões, contribuições e limitações pertinentes a este trabalho. Serão introduzidas algumas possibilidades de trabalhos futuros, sendo que o principal aprendizado foi estar aberto a novas possibilidades.

6.1 Contribuições

Este trabalho apresentou uma arquitetura de segurança para detecção de anomalias baseada na tipificação do comportamento normal de tráfego de redes domésticas. As principais novidades dessa arquitetura em relação às arquiteturas da literatura são:

- Definição de três componentes arquiteturais genéricos (CSA, HSU e NFG) para a segurança de redes domésticas.
- Fornece uma camada de proteção adicional para o ambiente residencial através dos modelos de detecção de ameaças, que são capazes de operar com os fluxos de rede de todos os dispositivos presentes do ambiente.
- Permite a combinação dos modelos de detecção de anomalias para tornar a descoberta de ameaças mais efetiva em cenários complexos.
- Possibilita a implantação em ambientes residenciais por intermédio de dispositivos com baixo custo.

Este trabalho também realizou experimentos relevantes que contribuem com o estado da arte no que tange à detecção de anomalias em redes domésticas, são eles:

- **Bases de dados:** Na concepção dos experimentos não foram encontrados conjuntos de dados públicos contendo amostras de tráfego residencial. Diante disso, este

trabalho também aprimorou a representação de amostras do tráfego de rede existente mediante a criação de bases de dados para representar o comportamento do ambiente com dispositivos IoT e não-IoT, também, foi incluído nas bases de dados seis tipos de ameaças, nas quais, três são para o cenário IoT e três para não-IoT.

- **Avaliação e comparação de algoritmos não-supervisionados:** Foi realizado um estudo com doze algoritmos para detecção de anomalias no ambiente residencial. Os resultados mostram que é possível identificar as ameaças utilizando algoritmos do tipo *one-class*, dessa forma, a arquitetura FamilyGuard consegue fornecer proteção adicional para o ambiente residencial através dos modelos de detecção de anomalias.

- **Avaliação e comparação de algoritmos supervisionados:** Depois de descobrir uma possível ameaça é importante identificá-la, para que seja possível gerar uma tratativa com o intuito de mitigá-la. Diante disso, no experimento foram utilizadas seis tipos de ameaças, os resultados demonstram que é possível identificar as ameaças através de modelos supervisionados com uma boa precisão.

- **Estimativa do tempo gasto para realizar as previsões:** Verificou-se a capacidade dos modelos em fornecer previsões em um prazo viável para mitigar as ameaças. É importante destacar que para estimar o tempo gasto foi usado uma implementação do modelo em um dispositivo real.

6.2 Limitações

Algumas limitações deste trabalho são descritas a seguir:

- **Ponto único de falha dentro do ambiente residencial:** Com a arquitetura apresentada é possível adquirir proteção adicional para o ambiente, mediante ao uso de modelos de detecção de anomalias para identificar possíveis ameaças, com o objetivo de mitigá-las. No entanto, com o uso do controlador SDN e o HSU em um único dispositivo trata-se de um ponto único de falha. Porém, o ambiente continuaria operando normalmente, mas a proteção adicional deixaria de funcionar até que os serviços fossem restabelecidos.

- **Quantidade de modelos de detecção de anomalias em uso:** A arquitetura proposta é capaz de operar em um *hardware* de baixo custo, no entanto, o número de modelos de detecção instalados no dispositivo depende da quantidade de memória disponível.

- **Comportamento dos modelos de detecção de anomalias:** Apesar de diversos experimentos realizados, tais amostras contaram com seis tipos de anomalias. Dessa forma, os resultados demonstram indícios de que é possível detectar essas ameaças.

No entanto, novos experimentos com diferentes tipos de ameaças devem ser feitos para melhorar a cobertura dos testes feitos ao longo deste trabalho.

6.3 Trabalhos Futuros

Alguns dos possíveis trabalhos futuros são descritos a seguir:

- **Novos experimentos com maior número de ameaças:** Avaliar o desempenho de algoritmos de uma classe para vários tipos de ameaças de segurança e identificar os melhores algoritmos de acordo com diferentes cenários de casas inteligentes. Dessa forma, também é possível explorar a ideia de construção de modelos colaborativos para melhorar a detecção de ameaças.

- **Explorar novos tipos de dados que podem ser coletados no ambiente residencial:** Este trabalho apresenta modelos para detecção de anomalias com base em fluxos de rede, entretanto, estes experimentos tiveram como objetivo a validação dos componentes e o uso da arquitetura em um ambiente real. No entanto, é possível explorar várias possibilidades que podem melhorar a segurança e o conforto das pessoas que vivem em um determinado ambiente, por exemplo, modelos para otimizar o consumo de energia, detectar ocupação dentro de um determinado cômodo da casa para verificar intrusos, reconhecer atividades ou eventos anormais para detectar perigos situações como incêndio ou inundação, ou monitorar as atividades dos idosos.

- **Investigar sobre a atualização dos modelos de detecção de anomalias:** Novas formas de aprendizado podem ser exploradas para atualizar os modelos, como o aprendizado incremental, aprendizado ativo e mineração de fluxos contínuos de dados. Nesse caso, acredita-se que os modelos consigam mitigar ameaças recém descobertas de maneira mais rápida.

- **Novas funcionalidades:** Explorar novas funcionalidades da arquitetura proposta com a enfase no usuário, por exemplo, o envio de notificação ao detectar ameaças e aplicação para controlar as políticas de segurança.

- **Testes de desempenho:** Analisar o desempenho dos componentes da arquitetura em diferentes dispositivos.

- **Avaliação de novos cenários:** Alterar posicionamento do CSA e HSU e realizar experimentos, para verificar se é possível implantá-los em *cloud, fog ou edge*.

- **Aprendizado federado**: trata-se de uma técnica de aprendizado de máquina que treina um algoritmo em vários dispositivos de borda descentralizados. Dessa forma,

os dados brutos de treinamento sempre permanecem nos dispositivos do usuário e nenhuma atualização individual é armazenada de forma identificável na nuvem. Baseado nisso, a descentralização torna-se interessante, pois permite implementação e testes mais rápidos, diminui a latência e o consumo de energia, garantindo a privacidade. Essas características contribuem para identificar ameaças de forma mais rápida e colaboram para que a privacidade dos dados dos usuários seja mantida.

6.4 Contribuições em Produção Bibliográfica

- P. H. A. D. de Melo, A. Araújo Martins de Resende, R. S. Miani and P. Frosi Rosa, "Evaluation of one-class algorithms for anomaly detection in home networks,"2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), 2021, pp. 682-689, doi: 10.1109/ICTAI52525.2021.00108.
- de Melo, P.H.A.D.; Miani, R.S.; Rosa, P.F. FamilyGuard: A Security Architecture for Anomaly Detection in Home Networks. Sensors 2022, doi: 10.3390/s22082895
- De Resende, Adriano AM, et al. "Traffic Classification of Home Network Devices using Supervised Learning."ICAART (3). 2022.

Referências

ABU-TAIR, M. et al. Towards Secure and Privacy-Preserving IoT Enabled Smart Home: Architecture and Experimental Study. **Sensors 20, no. 21 6131**, 2020. Disponível em: <www.mdpi.com/journal/sensors>.

ALI, W. et al. IoT based smart home: Security challenges, security requirements and solutions. In: **ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. ISBN 9780701702618.

ALRASHDI, I. et al. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In: **2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. p. 305–310. ISBN 9781728105543.

ALVES, A. R. et al. HomeNetRescue: An SDN service for troubleshooting home networks. In: **IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 1–7. ISBN 9781538634165.

AMEER, S.; BENSON, J.; SANDHU, R. The EGRBAC Model for Smart Home IoT. **Proceedings - 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science, IRI 2020**, Institute of Electrical and Electronics Engineers Inc., p. 457–462, aug 2020.

AMMI, M.; ALARABI, S.; BENKHELIFA, E. Customized blockchain-based architecture for secure smart home for lightweight IoT. **Information Processing & Management**, Pergamon, v. 58, n. 3, p. 102482, may 2021. ISSN 0306-4573.

ANDRESS, J. **The basics of information security: understanding the fundamentals of InfoSec in theory and practice**. [S.l.]: Syngress, 2014.

ANGIULLI, F.; PIZZUTI, C. Fast outlier detection in high dimensional spaces. In: ELOMAA, T.; MANNILA, H.; TOIVONEN, H. (Ed.). **Principles of Data Mining and Knowledge Discovery**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 15–27. ISBN 978-3-540-45681-0.

APTHORPE, N.; REISMAN, D.; FEAMSTER, N. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. **arXiv preprint arXiv:1705.06805**, 2017.

ASHFAQ, R. A. R. et al. Fuzziness based semi-supervised learning approach for intrusion detection system. **Information Sciences**, Elsevier Inc., v. 378, p. 484–497, feb 2017. ISSN 00200255.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, Elsevier, v. 54, n. 15, p. 2787–2805, oct 2010. ISSN 13891286.

BAI, L. et al. Automatic Device Classification from Network Traffic Streams of Internet of Things. In: **Proceedings - Conference on Local Computer Networks, LCN**. [S.l.]: IEEE Computer Society, 2019. v. 2018-October, p. 597–605. ISBN 9781538644133.

BAKAR, U. A. et al. Activity and anomaly detection in smart home: A survey. In: **Smart Sensors, Measurement and Instrumentation**. [S.l.]: Springer International Publishing, 2016. v. 16, p. 191–220. ISBN 9783319216713.

BALDWIN, D. A. The concept of security. **Review of International Studies**, Cambridge University Press, v. 23, n. 1, p. 5–26, 1997. ISSN 02602105.

BENGIO, Y.; GOODFELLOW, I. J.; COURVILLE, A. Deep learning, book in preparation for mit press (2015). **Disponıvel em http://www. iro. umontreal. ca/bengioy/dlbook**, 2015.

BEZERRA, V. H. et al. Iotds: A one-class classification approach to detect botnets in internet of things devices. **Sensors (Switzerland)**, MDPI AG, v. 19, 7 2019. ISSN 14248220.

Big Switch Networks. **Floodlight OpenFlow Controller**. 2014. Http://www.projectfloodlight.org/. Disponível em: <{http://www.projectfloodlight.org}.>

BOUTABA, R. et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. **Journal of Internet Services and Applications**, Springer London, v. 9, n. 1, p. 1–99, dec 2018. ISSN 18690238.

BREUNIG, M. M. et al. Lof: identifying density-based local outliers. In: **Proceedings of the 2000 ACM SIGMOD international conference on Management of data**. [S.l.: s.n.], 2000. p. 93–104.

BROWN, J.; ANWAR, M.; DOZIER, G. An evolutionary general regression neural network classifier for intrusion detection. In: **2016 25th International Conference on Computer Communications and Networks, ICCCN 2016**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509022793.

CHAKRABARTI, S.; CHAKRABORTY, M.; MUKHOPADHYAY, I. Study of snort-based IDS. In: **ICWET 2010 - International Conference and Workshop on Emerging Trends in Technology 2010, Conference Proceedings**. New York, New York, USA: ACM Press, 2010. p. 43–47. ISBN 9781605588124. Disponível em: <http://portal.acm.org/citation.cfm?doid=1741906.1741914>.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009.

CHOU, D.; JIANG, M. A survey on data-driven network intrusion detection. **ACM Computing Surveys (CSUR)**, ACM New York, NY, v. 54, n. 9, p. 1–36, 2021.

CLAISE, B. et al. Cisco systems netflow services export version 9. RFC 3954, October, 2004.

COLLEN, A. et al. GHOST - Safe-guarding home IoT environments with personalised real-time risk control. In: **Communications in Computer and Information Science**. [S.l.]: Springer Verlag, 2018. v. 821, p. 68–78. ISBN 9783319951881. ISSN 18650929.

DACIER, M. C. et al. Security Challenges and Opportunities of Software-Defined Networking. **IEEE Security & Privacy**, v. 15, n. 2, p. 96–100, mar 2017. ISSN 1540-7993. Disponível em: <http://ieeexplore.ieee.org/document/7891523/>.

DARBY, S. J. Smart technology in the home: time for more clarity. **https://doi.org/10.1080/09613218.2017.1301707**, Routledge, v. 46, n. 1, p. 140–147, jan 2017. ISSN 14664321. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/09613218.2017.1301707>.

DAS, S. K.; COOK, D. J. Designing Smart Environments: A Paradigm Based on Learning and Prediction. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, Springer, Berlin, Heidelberg, v. 3776 LNCS, p. 80–90, dec 2005. ISSN 03029743. Disponível em: <https://link.springer.com/chapter/10.1007/11590316_11>.

DAY, J. D.; ZIMMERMANN, H. The osi reference model. **Proceedings of the IEEE**, IEEE, v. 71, n. 12, p. 1334–1340, 1983.

DAYAL, N. et al. Research trends in security and ddos in sdn. **Security and Communication Networks**, Wiley Online Library, v. 9, n. 18, p. 6386–6411, 2016.

DEMERTZIS, K.; ILIADIS, L.; SPARTALIS, S. A spiking one-class anomaly detection framework for cyber-security on industrial control systems. In: **Communications in Computer and Information Science**. [S.l.]: Springer Verlag, 2017. v. 744, p. 122–134. ISBN 9783319651712. ISSN 18650929.

DEMETRIOU, S. et al. HanGuard: SDN-driven protection of smart home WiFi devices from malicious mobile apps. In: **Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017**. [S.l.]: Association for Computing Machinery, Inc, 2017. p. 122–133. ISBN 9781450350846.

DOSHI, R.; APTHORPE, N.; FEAMSTER, N. Machine learning DDoS detection for consumer internet of things devices. In: **Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 29–35. ISBN 9780769563497.

DRAPER-GIL, G. et al. Characterization of encrypted and vpn traffic using time-related. In: **Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)**. [S.l.: s.n.], 2016. p. 407–414.

FERNANDES, E.; JUNG, J.; PRAKASH, A. Security Analysis of Emerging Smart Home Applications. In: **2016 IEEE Symposium on Security and Privacy (SP)**. IEEE, 2016. p. 636–654. ISBN 978-1-5090-0824-7. ISSN 2375-1207. Disponível em: <http://ieeexplore.ieee.org/document/7546527/>.

FERNANDES, G. et al. **A comprehensive survey on network anomaly detection**. [S.l.]: Springer New York LLC, 2019. 447–489 p.

FILHO, G. P. et al. A fog-enabled smart home solution for decision-making using smart objects. **Future Generation Computer Systems**, Elsevier B.V., v. 103, p. 18–27, feb 2020. ISSN 0167739X.

FOUNDATION, O. N. **ONF Overview - Open Networking Foundation**. 2017. Disponível em: <https://www.opennetworking.org/about/onf-overview>. Acesso em: 01 jan. 2017.

GALEANO-BRAJONES, J. et al. Detection and Mitigation of DoS and DDoS Attacks in IoT-Based Stateful SDN: An Experimental Approach. **Sensors**, MDPI AG, v. 20, n. 3, p. 816, feb 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/3/816>.

GARCIA, S.; PARMISANO, A.; ERQUIAGA, M. J. **IoT-23: A labeled dataset with malicious and benign IoT network traffic**. Zenodo, 2020. More details here https://www.stratosphereips.org /datasets-iot23. Disponível em: <https://doi.org/10.5281/zenodo.4743746>.

GIOTIS, K. et al. Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments. **Computer Networks**, Elsevier, v. 62, p. 122–136, 2014.

GRAM-HANSSEN, K.; DARBY, S. J. "Home is where the smart is"? Evaluating smart home research and approaches against the concept of home. **Energy Research & Social Science**, Elsevier, v. 37, p. 94–101, mar 2018. ISSN 2214-6296.

HAFEEZ, I.; DING, A. Y.; TARKOMA, S. Securing Edge Networks with Securebox. dec 2017. Disponível em: <http://arxiv.org/abs/1712.07740>.

HARDIN, J.; ROCKE, D. M. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. **Computational Statistics Data Analysis**, v. 44, n. 4, p. 625–638, 2004. ISSN 0167-9473. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167947302002803>.

HASAN, M. et al. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. **Internet of Things**, Elsevier BV, v. 7, p. 100059, sep 2019. ISSN 25426605.

HE, G. Requirements for security in home environments. In: **Residential and Virtual Home Environments Seminar on Internetworking, Spring**. [S.l.: s.n.], 2002.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. **Pattern Recognition Letters**, v. 24, n. 9, p. 1641–1650, 2003. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167865503000035>.

HOFSTEDE, R. et al. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. **IEEE Communications Surveys & Tutorials**, IEEE, v. 16, n. 4, p. 2037–2064, 2014.

JAMAL, S. et al. Machine learning: What, why, and how? In: **Bioinformatics: Sequences, Structures, Phylogeny**. [S.l.]: Springer, 2018. p. 359–374.

KABIR, E. et al. A novel statistical technique for intrusion detection systems. **Future Generation Computer Systems**, Elsevier B.V., v. 79, p. 303–318, feb 2018. ISSN 0167739X.

KARIMI, K.; KRIT, S. Smart home-smartphone systems: Threats, security requirements and open research challenges. In: **Proceedings of 2019 International Conference of Computer Science and Renewable Energies, ICCSRE 2019**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. ISBN 9781728108278.

KARLOF, C.; WAGNER, D. Secure routing in wireless sensor networks: Attacks and countermeasures. In: **Ad Hoc Networks**. [S.l.]: Elsevier, 2003. v. 1, n. 2-3, p. 293–315. ISSN 15708705.

KDDCUP. Kdd cup 1999 data. In: . [s.n.], 1999. Disponível em: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

KHAN, S. S.; MADDEN, M. G. A survey of recent trends in one class classification. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [S.l.]: Springer, Berlin, Heidelberg, 2010. v. 6206 LNAI, p. 188–197. ISBN 364217079X. ISSN 03029743.

_____. One-Class Classification: Taxonomy of Study and Review of Techniques. nov 2013. Disponível em: <http://arxiv.org/abs/1312.0049http://dx.doi.org/10.1017/S026988891300043X>.

KOLIAS, C. et al. Ddos in the iot: Mirai and other botnets. **Computer**, IEEE, v. 50, n. 7, p. 80–84, 2017.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers Inc., v. 103, n. 1, p. 14–76, jan 2015. ISSN 15582256.

_____. Software-Defined Networking: A Comprehensive Survey. **Proceedings of the IEEE**, v. 103, n. 1, p. 14–76, jan. 2015. ISSN 0018-9219.

LABS, F. **Fortiguard Threat**. 2020. <https://www.fortiguardthreatinsider.com/pt/bulletin/Q4-2020>. Accessed: 2022-07-01.

LASHKARI, A. H. et al. Characterization of Tor Traffic using Time based Features. p. 253–262, mar 2017.

_____. Characterization of Tor Traffic using Time based Features. p. 253–262, jun 2020.

LAZAREVIC, A.; KUMAR, V. Feature bagging for outlier detection. In: **Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining**. [S.l.: s.n.], 2005. p. 157–166.

LEE, C. et al. Securing smart home: Technologies, security challenges, and security requirements. In: **2014 IEEE Conference on Communications and Network Security, CNS 2014**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 67–72. ISBN 9781479958900.

LEE, I.; LEE, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. **Business Horizons**, Elsevier Ltd, v. 58, n. 4, p. 431–440, jul 2015. ISSN 00076813.

LI, K. L. et al. Improving one-class SVM for anomaly detection. In: **International Conference on Machine Learning and Cybernetics**. [S.l.: s.n.], 2003. v. 5, p. 3077–3081. ISBN 0780378652.

LI, Z. et al. Copod: Copula-based outlier detection. In: **2020 IEEE International Conference on Data Mining (ICDM)**. [S.l.: s.n.], 2020. p. 1118–1123.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: **2008 Eighth IEEE International Conference on Data Mining**. [S.l.: s.n.], 2008. p. 413–422.

MAHDAVINEJAD, M. S. et al. Machine learning for internet of things data analysis: a survey. **Digital Communications and Networks**, Elsevier, v. 4, n. 3, p. 161–175, aug 2018. ISSN 2352-8648. Disponível em: <https://www.sciencedirect.com/science/article/pii/S235286481730247X>.

Maloof et. al. **Machine Learning and Data Mining for Computer Security**. [S.l.]: Springer-Verlag, 2006.

MASCARENHAS, C. et al. Project Urban Patrol: Building an Attack Resilient Smart Home Architecture. **2021 International Conference on Nascent Technologies in Engineering, ICNET 2021 - Proceedings**, Institute of Electrical and Electronics Engineers Inc., jan 2021.

MAZHAR, M. H.; SHAFIQ, Z. Characterizing Smart Home IoT Traffic in the Wild. jan 2020. Disponível em: <http://arxiv.org/abs/2001.08288>.

MCHUGH, J. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. **ACM Transactions on Information and System Security (TISSEC)**, ACM New York, NY, USA, v. 3, n. 4, p. 262–294, 2000.

MCKEOWN, N. Software-defined networking. **INFOCOM keynote talk**, v. 17, n. 2, p. 30–32, 2009.

MCKEOWN, N. et al. OpenFlow: Enabling Innovation in Campus Networks. **SIGCOMM Comput. Commun. Rev.**, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <http://doi.acm.org/10.1145/1355734.1355746>.

MOLINA-CORONADO, B. et al. Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. **IEEE Transactions on Network and Service Management**, IEEE, v. 17, n. 4, p. 2451–2479, 2020.

MORRIS, T. H.; THORNTON, Z.; TURNIPSEED, I. Industrial control system simulation and data logging for intrusion detection system research. **7th annual southeastern cyber security summit**, p. 3–4, 2015.

MOUSTAFA, N.; SLAY, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: **2015 Military Communications and Information Systems Conference (MilCIS)**. [S.l.: s.n.], 2015. p. 1–6.

NG, E.; CAI, Z.; COX, A. Maestro: A system for scalable openflow control. **Rice University, Houston, TX, USA, TSEN Maestro-Techn. Rep, TR10-08**, 2010.

NGUYEN-AN, H. et al. Detecting IoT Traffic Anomalies in Smart Home Environment. In: **2019 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2019**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. ISBN 9781728132792.

NILSSON, N. J.; NILSSON, N. J. **Artificial intelligence: a new synthesis**. [S.l.]: Morgan Kaufmann, 1998.

NOVÁK, M.; BIňAS, M.; JAKAB, F. Unobtrusive anomaly detection in presence of elderly in a smart-home environment. In: **Proceedings of 9th International Conference, ELEKTRO 2012**. [S.l.: s.n.], 2012. p. 341–344. ISBN 9781467311793.

NTHALA, N.; FLECHAIS, I. Rethinking Home Network Security. In: . [S.l.]: Internet Society, 2018.

PAHL, M.-O.; AUBET, F.-X. All eyes on you: Distributed multi-dimensional iot microservice anomaly detection. In: **2018 14th International Conference on Network and Service Management (CNSM)**. [S.l.: s.n.], 2018. p. 72–80.

PATCHA, A.; PARK, J. M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. **Computer Networks**, Elsevier, v. 51, n. 12, p. 3448–3470, aug 2007. ISSN 13891286.

PEVNỲ, T. Loda: Lightweight on-line detector of anomalies. **Machine Learning**, Springer, v. 102, n. 2, p. 275–304, 2016.

PLATT, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. **Advances in large margin classifiers**, Cambridge, MA, v. 10, n. 3, p. 61–74, 1999.

RAMAPATRUNI, S. et al. Anomaly Detection Models for Smart Home Security. **5th IEEE International Conference on Big Data Security on Cloud (BigDataSecurity 2019)**, apr 2019. Disponível em: <https://ebiquity.umbc.edu/paper/html/id/851/Anomaly-Detection-Models-for-Smart-Home-Security>.

RAMASWAMY, S.; RASTOGI, R.; SHIM, K. Efficient algorithms for mining outliers from large data sets. In: **Proceedings of the 2000 ACM SIGMOD international conference on Management of data**. [S.l.: s.n.], 2000. p. 427–438.

RAWAT, D. B.; REDDY, S. R. Software defined networking architecture, security and energy efficiency: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 19, n. 1, p. 325–346, 2017.

REBALA, G.; RAVI, A.; CHURIWALA, S. **An Introduction to Machine Learning**. [S.l.]: Springer, 2019.

ROSE, K. H. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Fifth Edition. **Project Management Journal**, SAGE Publications, v. 44, n. 3, p. e1–e1, jun 2013. ISSN 87569728. Disponível em: <http://doi.wiley.com/10.1002/pmj.21345>.

ROUSSEEUW, P. J.; DRIESSEN, K. V. A fast algorithm for the minimum covariance determinant estimator. **Technometrics**, Taylor & Francis, v. 41, n. 3, p. 212–223, 1999.

RUSSEL, S.; NORVIG, P. et al. **Artificial intelligence: a modern approach**. [S.l.]: Pearson Education Limited, 2013.

SAXENA, U.; SODHI, J. S.; SINGH, Y. Analysis of security attacks in a smart home networks. In: **Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 431–436. ISBN 9781509035182.

SCHöLKOPF, B. et al. Estimating the Support of a High-Dimensional Distribution. **Neural Computation**, v. 13, n. 7, p. 1443–1471, 07 2001. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/089976601750264965>.

SEN, J. A Survey on Wireless Sensor Network Security. **The Organization of Behavior**, p. 60–78, nov 2010. Disponível em: <http://arxiv.org/abs/1011.1529>.

SERROR, M. et al. Towards in-network security for smart homes. In: **ACM International Conference Proceeding Series**. New York, New York, USA: Association for Computing Machinery, 2018. p. 1–8. ISBN 9781450364485. Disponível em: <http://dl.acm.org/citation.cfm?doid=3230833.3232802>.

SHARAFALDIN, I. et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. **Proceedings - International Carnahan Conference on Security Technology**, Institute of Electrical and Electronics Engineers Inc., v. 2019-October, oct 2019. ISSN 10716572.

SHARMA, P. K. et al. SHSec: SDN based Secure Smart Home Network Architecture for Internet of Things. **Mobile Networks and Applications**, Springer New York LLC, v. 24, n. 3, p. 913–924, jun 2019. ISSN 15728153.

Sherlock. 2016. Disponível em: <http://bigdata.ise.bgu.ac.il/sherlock/>.

SHIRAVI, A. et al. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. **Computers Security**, v. 31, n. 3, p. 357–374, 2012. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404811001672>.

SIVANATHAN, A. et al. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. **IEEE Transactions on Mobile Computing**, Institute of Electrical and Electronics Engineers Inc., v. 18, n. 8, p. 1745–1759, aug 2019. ISSN 15580660.

_____. Characterizing and classifying IoT traffic in smart cities and campuses. In: **2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)**. IEEE, 2017. p. 559–564. ISBN 978-1-5386-2784-6. Disponível em: <http://ieeexplore.ieee.org/document/8116438/>.

SONI, V.; MODI, P.; CHAUDHRI, V. **Detecting Sinkhole Attack in Wireless Sensor Network**. 2013. 29–32 p.

SPINELLIS, D.; KOKOLAKIS, S.; GRITZALIS, S. Security requirements, risks and recommendations for small enterprise and home-office environments. **Information Management and Computer Security**, Emerald Group Publishing Ltd., v. 7, n. 3-4, p. 121–128, 1999. ISSN 09685227.

STEWART, C. E.; VASU, A. M.; KELLER, E. CommunityGuard: A crowdsourced home cyber-security system. In: **SDN-NFVSec 2017 - Proceedings of the ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, co-located with CODASPY 2017**. [S.l.]: Association for Computing Machinery, Inc, 2017. p. 1–6. ISBN 9781450349086.

SWARNKAR, M.; HUBBALLI, N. OCPAD: One class Naive Bayes classifier for payload based anomaly detection. **Expert Systems with Applications**, Elsevier Ltd, v. 64, p. 330–339, dec 2016. ISSN 09574174.

TAVALLAEE, M. et al. A detailed analysis of the kdd cup 99 data set. In: **2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications**. [S.l.: s.n.], 2009. p. 1–6.

TOOTOONCHIAN, A.; GANJALI, Y. Hyperflow: A distributed control plane for openflow. In: **Proceedings of the 2010 internet network management conference on Research on enterprise networking**. [S.l.: s.n.], 2010. p. 3–3.

TURING, A. M. Computing machinery and intelligence. In: **Parsing the turing test**. [S.l.]: Springer, 2009. p. 23–65.

Ul Alam, M. A. et al. Smart-energy group anomaly based behavioral abnormality detection. In: **2016 IEEE Wireless Health, WH 2016**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 38–45. ISBN 9781509030903.

UMER, M. F.; SHER, M.; BI, Y. Flow-based intrusion detection: Techniques and challenges. **Computers & Security**, Elsevier Advanced Technology, v. 70, p. 238–254, sep 2017. ISSN 0167-4048.

WANG, H.; GU, J.; WANG, S. An effective intrusion detection framework based on SVM with feature augmentation. **Knowledge-Based Systems**, Elsevier B.V., v. 136, p. 130–139, nov 2017. ISSN 09507051.

WOLFERS, A. "National Security"as an Ambiguous Symbol. **Political Science Quarterly**, v. 67, n. 4, p. 481, dec 1952. ISSN 00323195. Disponível em: <http://www.jstor.org/stable/2145138?origin=crossref>.

YAMAUCHI, M. et al. Anomaly Detection for Smart Home Based on User Behavior. In: **2019 IEEE International Conference on Consumer Electronics, ICCE 2019**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. ISBN 9781538679104.

ZAIDAN, A. A.; ZAIDAN, B. B. A review on intelligent process for smart home applications based on IoT: coherent taxonomy, motivation, open challenges, and recommendations. **Artificial Intelligence Review**, Springer, v. 53, n. 1, p. 141–165, jan 2020. ISSN 15737462.

ZARPELãO, B. B. et al. A survey of intrusion detection in internet of things. **Journal of Network and Computer Applications**, v. 84, p. 25–37, 2017. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804517300802>.

ZHANG, H. The optimality of naive bayes. **Aa**, v. 1, n. 2, p. 3, 2004.

ZHAO, Y.; NASRULLAH, Z.; LI, Z. Pyod: A python toolbox for scalable outlier detection. **arXiv preprint arXiv:1901.01588**, 2019.

ZHU, C.; SHENG, W.; LIU, M. Wearable Sensor-Based Behavioral Anomaly Detection in Smart Assisted Living Systems. **IEEE Transactions on Automation Science and Engineering**, Institute of Electrical and Electronics Engineers Inc., v. 12, n. 4, p. 1225–1234, oct 2015. ISSN 15455955.

Apêndices

APÊNDICE A

CICFlowMeter - Características

Tabela 14 – Características geradas pela ferramenta CICFlowMeter

Característica	Descrição
Flow duration	Duração do fluxo em microssegundos
total Fwd Packet	Total de pacotes na direção direta
total Bwd packets	Total de pacotes na direção inversa
total Length of Fwd Packet	Tamanho total do pacote na direção direta
total Length of Bwd Packet	Tamanho total do pacote na direção inversa
Fwd Packet Length Min	Tamanho mínimo do pacote na direção direta
Fwd Packet Length Max	Tamanho máximo do pacote na direção direta
Fwd Packet Length Mean	Tamanho médio do pacote na direção direta
Fwd Packet Length Std	Tamanho do desvio padrão do pacote na direção direta
Bwd Packet Length Min	Tamanho mínimo do pacote na direção inversa
Bwd Packet Length Max	Tamanho máximo do pacote na direção inversa
Bwd Packet Length Mean	Tamanho médio do pacote na direção inversa
Bwd Packet Length Std	Tamanho do desvio padrão do pacote na direção inversa
Flow Bytes/s	Número do fluxo de bytes por segundo
Flow Packets/s	Número do fluxo de pacotes por segundo
Flow IAT Mean	Tempo médio entre dois pacotes enviados no fluxo
Flow IAT Std	Tempo de desvio padrão entre dois pacotes
Flow IAT Max	Tempo máximo entre dois pacotes enviados em fluxo
Flow IAT Min	Tempo mínimo entre dois pacotes
Fwd IAT Min	Tempo mínimo entre dois pacotes
Fwd IAT Max	Tempo máximo entre dois pacotes
Fwd IAT Mean	Tempo médio entre dois pacotes
Fwd IAT Std	Tempo de desvio padrão entre dois pacotes

Continua na próxima página

Tabela 14 – continuação da página anterior

Característica	Descrição
Fwd IAT Total	Tempo total entre dois pacotes
Bwd IAT Min	Tempo mínimo entre dois pacotes
Bwd IAT Max	Tempo máximo entre dois pacotes
Bwd IAT Mean	Tempo médio entre dois pacotes
Bwd IAT Std	Tempo de desvio padrão entre dois pacotes
Bwd IAT Total	Tempo total entre dois pacotes enviados no sentido inverso
Fwd PSH flags	Número de vezes que o sinalizador PSH foi definido em pacotes
Bwd PSH Flags	Número de vezes que o sinalizador PSH foi definido em pacotes
Fwd URG Flags	Número de vezes que o sinalizador URG foi definido em pacotes
Bwd URG Flags	Número de vezes que o sinalizador URG foi definido em pacotes
Fwd Header Length	Total de bytes usados para cabeçalhos na direção direta
Bwd Header Length	Total de bytes usados para cabeçalhos na direção inversa
FWD Packets/s	Número de pacotes encaminhados por segundo
Bwd Packets/s	Número de pacotes de retorno por segundo
Packet Length Min	ML de um pacote
Packet Length Max	ML de um pacote
Packet Length Mean	ML de um pacote
Packet Length Std	Comprimento do desvio padrão de um pacote
Packet Length Variance	Comprimento de variação de um pacote
FIN Flag Count	Número de pacotes com FIN
SYN Flag Count	Número de pacotes com SYN
RST Flag Count	Número de pacotes com RST
PSH Flag Count	Número de pacotes com PUSH
ACK Flag Count	Número de pacotes com ACK
URG Flag Count	Número de pacotes com URG
CWE Flag Count	Número de pacotes com CWE
ECE Flag Count	Número de pacotes com ECE
down/Up Ratio	Taxa de download e upload
Average Packet Size	Tamanho médio do pacote
Fwd Segment Size Avg	Tamanho médio observado na direção para frente
Bwd Segment Size Avg	Número médio de taxa em massa de bytes na direção inversa
Fwd Bytes/Bulk Avg	Número médio de taxa em massa de bytes na direção direta
Fwd Packet/Bulk Avg	Número médio de taxa em massa de pacotes na direção direta
Fwd Bulk Rate Avg	Número médio de taxa em massa na direção direta
Bwd Bytes/Bulk Avg	Número médio de taxa em massa de bytes na direção inversa

Continua na próxima página

Tabela 14 – continuação da página anterior

Característica	Descrição
Bwd Packet/Bulk Avg	Taxa média do número de pacotes em massa na direção inversa
Bwd Bulk Rate Avg	Número médio de taxa em massa na direção inversa
Subflow Fwd Packets	O número médio de pacotes em um subfluxo
Subflow Fwd Bytes	O número médio de bytes em um subfluxo
Subflow Bwd Packets	O número médio de pacotes em um subfluxo
Subflow Bwd Bytes	O número médio de bytes em um subfluxo
Fwd Init Win bytes	O número total de bytes enviados na janela inicial
Bwd Init Win bytes	O número total de bytes enviados na janela inicial
Fwd Act Data Pkts	Contagem de pacotes com pelo menos 1 byte de TCP
Fwd Seg Size Min	Tamanho mínimo do segmento observado na direção direta
Active Min	Tempo mínimo em que um fluxo esteve ativo
Active Mean	Tempo médio em que um fluxo que estava ativo
Active Max	Tempo máximo em que um fluxo que ficou ativo
Active Std	Tempo de desvio padrão em que um fluxo que estava ativo
Idle Min	Tempo mínimo que um fluxo ficou ocioso antes de ba
Idle Mean	Tempo médio que um fluxo estava ocioso
Idle Max	Tempo máximo que um fluxo ficou ocioso
Idle Std	Tempo de desvio padrão em que um fluxo estava ocioso

Anexos

ANEXO A

Parâmetros dos Algoritmos

A.1 Algoritmos não-supervisionados

```
Local Outlier Factor = {'algorithm': 'auto', 'contamination': 0.1, 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': -1, 'n_neighbors': 5, 'p': 2}
```

```
Isolation Forest = {'behaviour': 'old', 'bootstrap': False, 'contamination': 0.1, 'max_features': 1.0, 'max_samples': 'auto', 'n_estimators': 100, 'n_jobs': -1, 'random_state': None, 'verbose': 0}
```

```
One-class SVM = {'cache_size': 200, 'coef0': 0.0, 'contamination': 0.1, 'degree': 3, 'gamma': 'auto', 'kernel': 'rbf', 'max_iter': -1, 'nu': 0.5, 'shrinking': True, 'tol': 0.001, 'verbose': False}
```

A.2 Algoritmos supervisionados

```
C-Support Vector Classification = {'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 'tol': 0.001, 'verbose': False}
```

```
Naive Bayes = {'priors': None, 'var_smoothing': 1e-09}
```

```
k Nearest Neighbors = {'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': -1, 'n_neighbors': 3, 'p': 2, 'weights': 'uniform'}
```

A.3 Algoritmos não-supervisionados do tipo *one-class*

```
Principal Component Analysis
{'contamination': 0.1, 'copy': True, 'iterated_power': 'auto', 'n_components': None, 'n_selected_components': None, 'random_state': None, 'standardization': True, 'svd_solver': 'auto', 'tol': 0.0, 'weighted': True, 'whiten': False}
```

```
Minimum Covariance Determinant = {'assume_centered': False, 'contamination': 0.1, 'random_state': None, 'store_precision': True, 'support_fraction': None}
```

```
One-class SVM = {'cache_size': 200, 'coef0': 0.0, 'contamination': 0.1, 'degree': 3, 'gamma': 'auto', 'kernel': 'rbf', 'max_iter': -1, 'nu': 0.5, 'shrinking': True, 'tol': 0.001, 'verbose': False}
```

```
Local Outlier Factor = {'algorithm': 'auto', 'contamination': 0.1, 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': -1, 'n_neighbors': 20, 'p': 2}
```

```
Clustering Based Local Outlier Factor = {'alpha': 0.9, 'beta': 5, 'check_estimator': False, 'clustering_estimator': None, 'contamination': 0.1, 'n_clusters': 8, 'n_jobs': -1, 'random_state': None, 'use_weights': False}
```

```
Histogram-based Outlier Score = {'alpha': 0.1, 'contamination': 0.1, 'n_bins': 10, 'tol': 0.5}
```

```
k Nearest Neighbors = {'algorithm': 'auto', 'contamination': 0.1, 'leaf_size': 30, 'method': 'largest', 'metric': 'minkowski', 'metric_params': None, 'n_jobs': -1, 'n_neighbors': 5, 'p': 2, 'radius': 1.0}
```

```
Average kNN = {'algorithm': 'auto', 'contamination': 0.1, '
    leaf_size': 30, 'method': 'mean', 'metric': 'minkowski', '
    metric_params': None, 'n_jobs': -1, 'n_neighbors': 5, 'p':
     2, 'radius': 1.0}
```

```
Median kNN = {'algorithm': 'auto', 'contamination': 0.1, '
    leaf_size': 30, 'method': 'median', 'metric': 'minkowski',
     'metric_params': None, 'n_jobs': -1, 'n_neighbors': 5, 'p
    ': 2, 'radius': 1.0}
```

```
COPOD: Copula-Based Outlier Detection = {'contamination': 0.1
    , 'n_jobs': -1}
```

```
Isolation Forest = {'behaviour': 'old', 'bootstrap': False, '
    contamination': 0.1, 'max_features': 1.0, 'max_samples': '
    auto', 'n_estimators': 100, 'n_jobs': -1, 'random_state':
    None, 'verbose': 0}
```

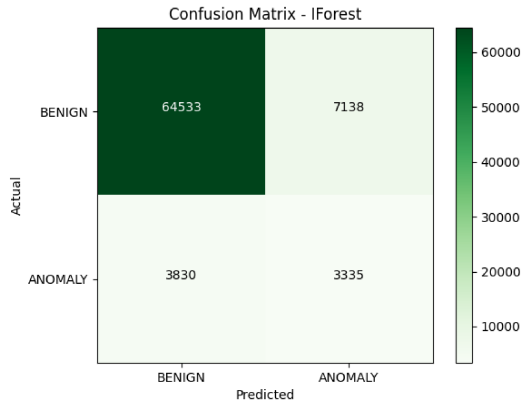
```
Feature Bagging = {'base_estimator': None, '
    bootstrap_features': False, 'check_detector': True, '
    check_estimator': False, 'combination': 'average', '
    contamination': 0.1, 'estimator_params': {}, 'max_features
    ': 1.0, 'n_estimators': 10, 'n_jobs': -1, 'random_state':
    None, 'verbose': 0}
```

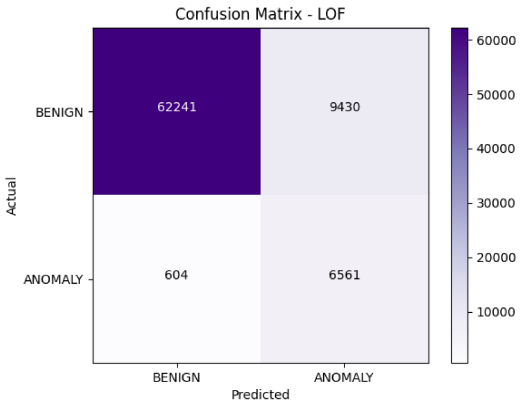
```
Lightweight On-line Detector of Anomalies = {'contamination':
     0.1, 'n_bins': 10, 'n_random_cuts': 100}
```

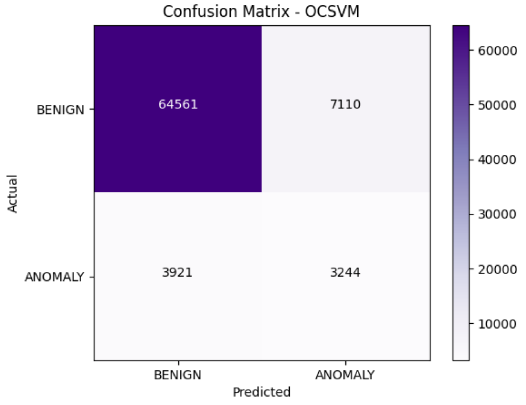
ANEXO B

Detecção de anomalias

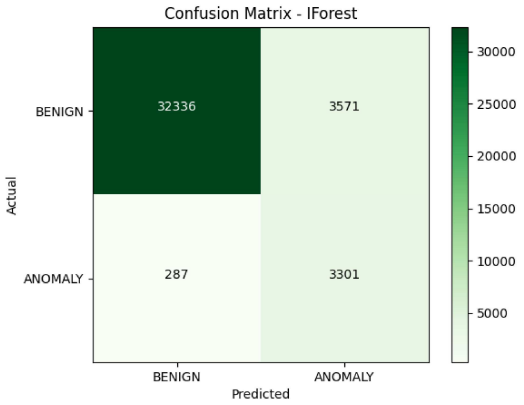
B.1 Matriz de Confusão - Caso de Teste 1

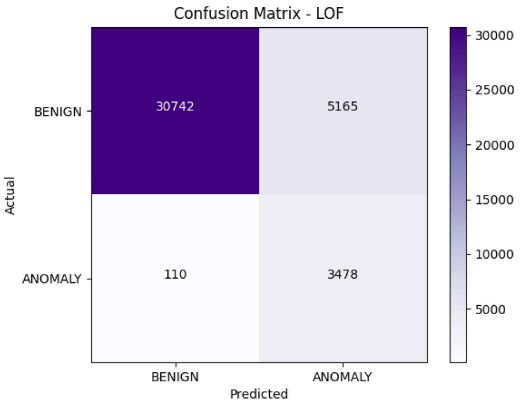


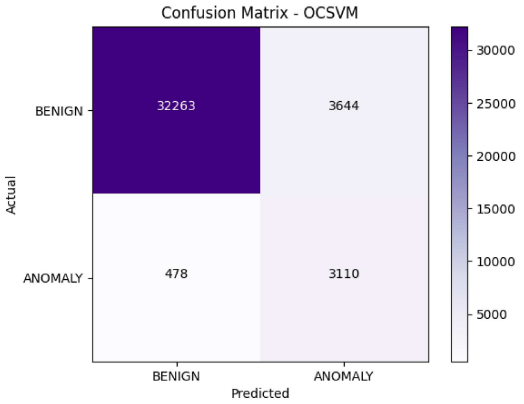




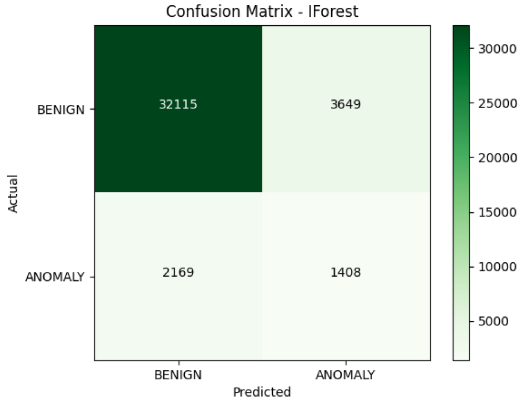
B.2 Matriz de Confusão - Caso de Teste 2

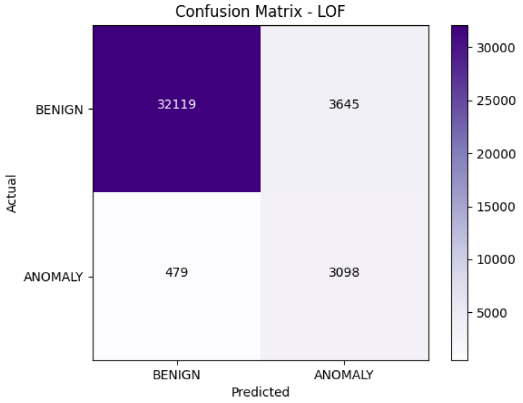


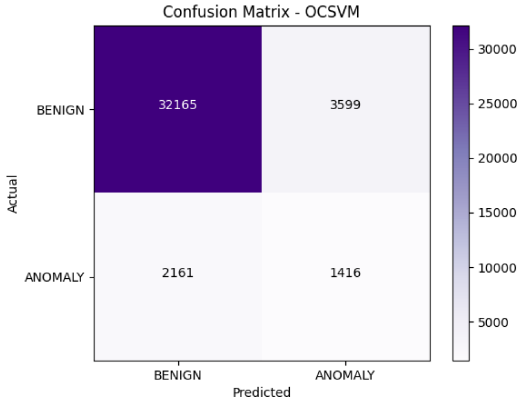




B.3 Matriz de Confusão - Caso de Teste 3



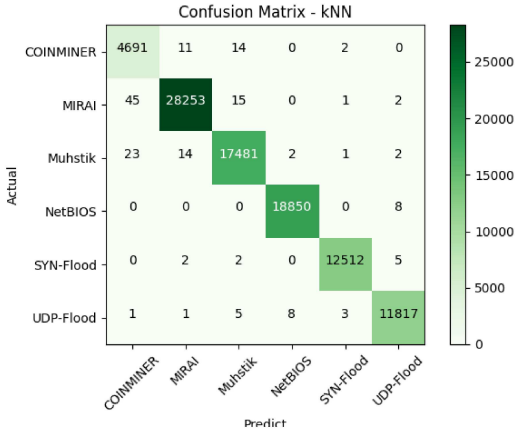


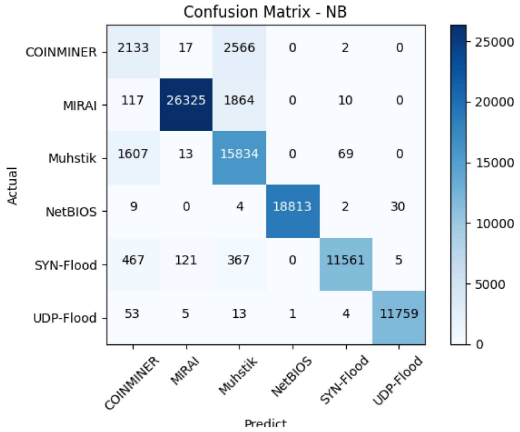


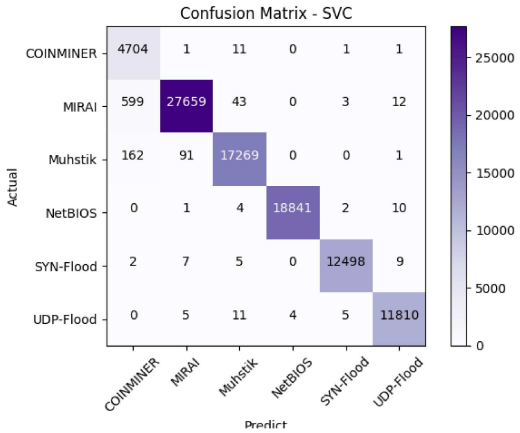
ANEXO C

Classificação de anomalias

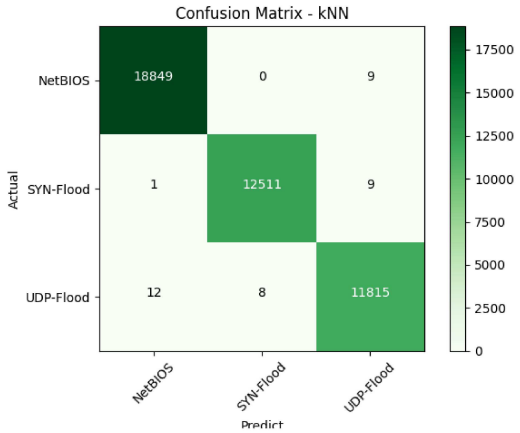
C.1 Matriz de Confusão - Caso de Teste 1

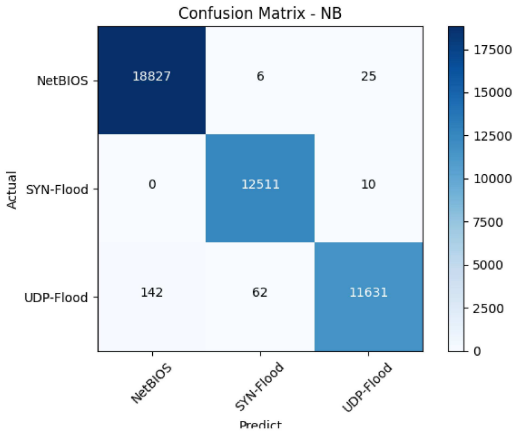


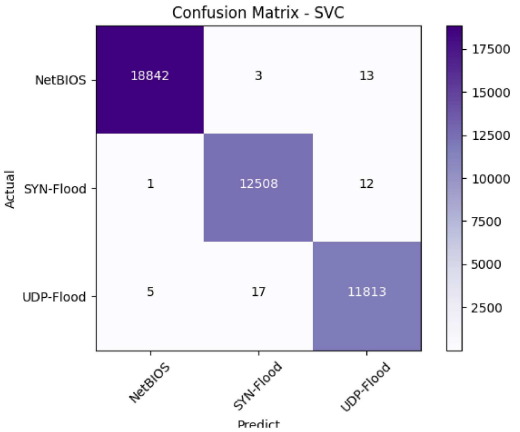




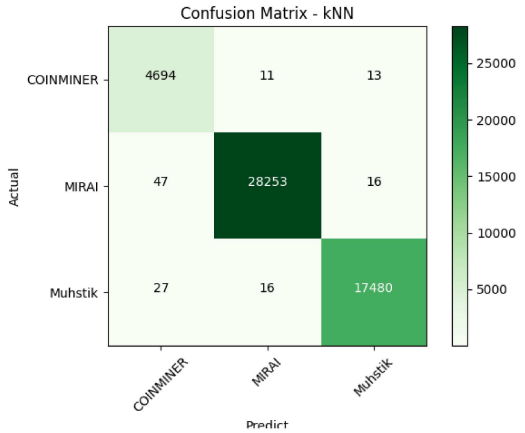
C.2 Matriz de Confusão - Caso de Teste 2

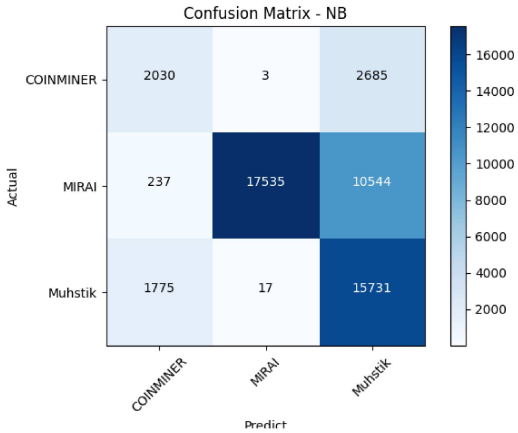


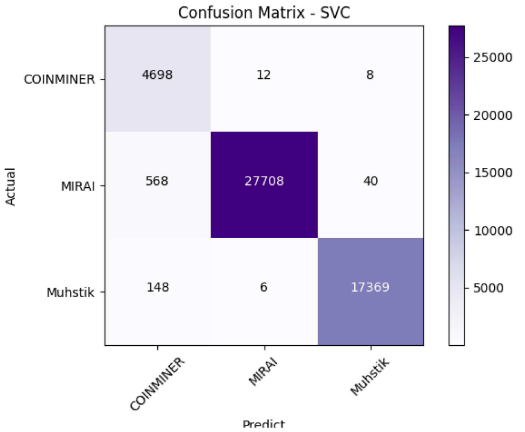




C.3 Matriz de Confusão - Caso de Teste 3



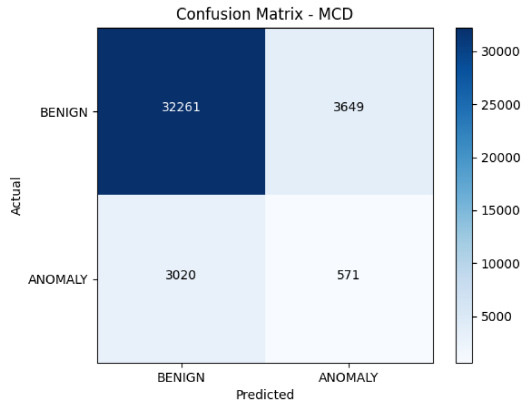


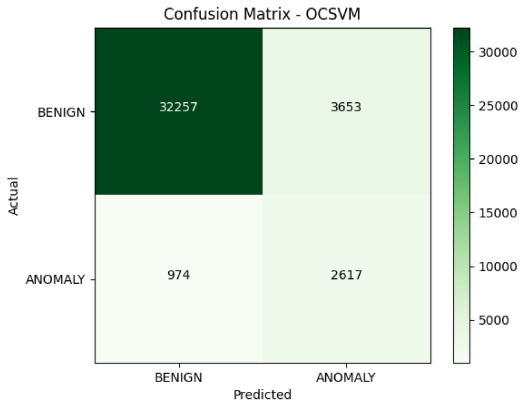


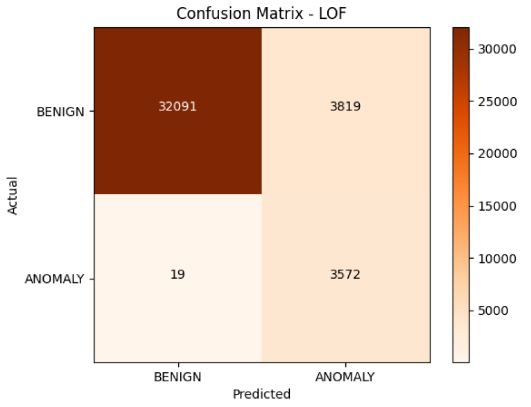
ANEXO D

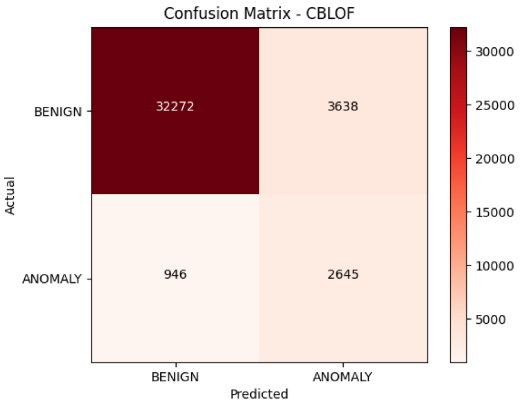
Algoritmos não-supervisionados

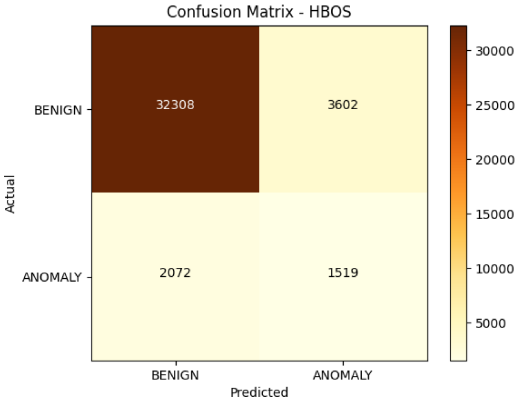
D.1 Matriz de Confusão - Caso de Teste 1

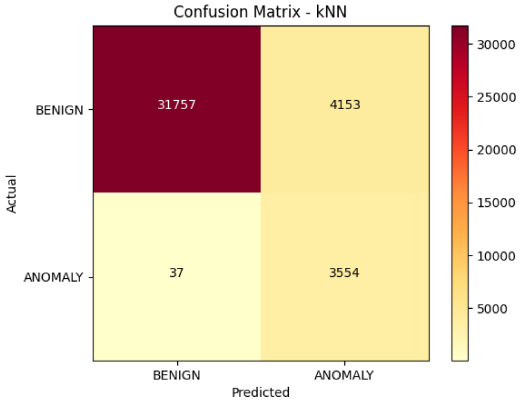


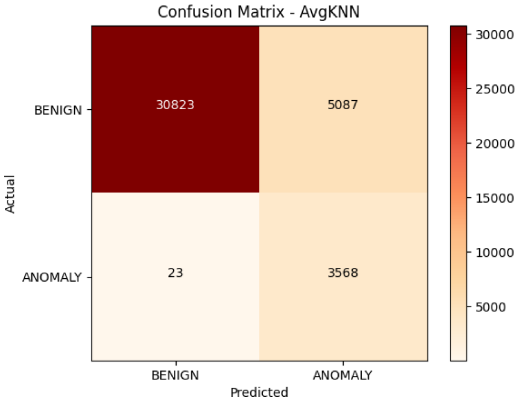


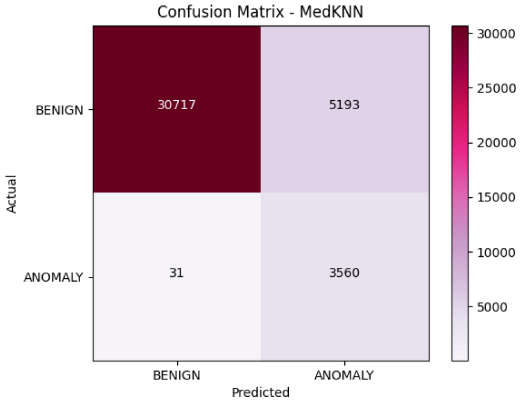


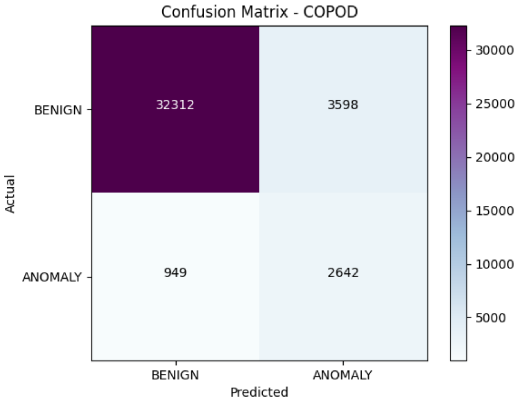


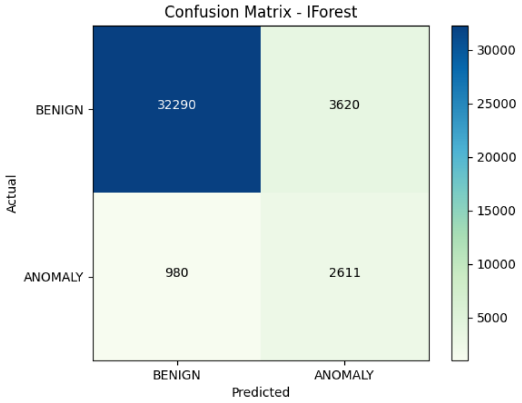

Confusion Matrix - HBOS
BENIGN
ANOMALY
Actual
32308
3602
2072
1519
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000

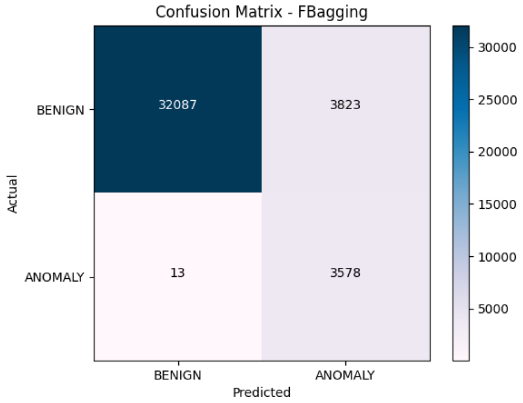

Confusion Matrix - kNN
BENIGN
ANOMALY
Actual
31757
4153
37
3554
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000

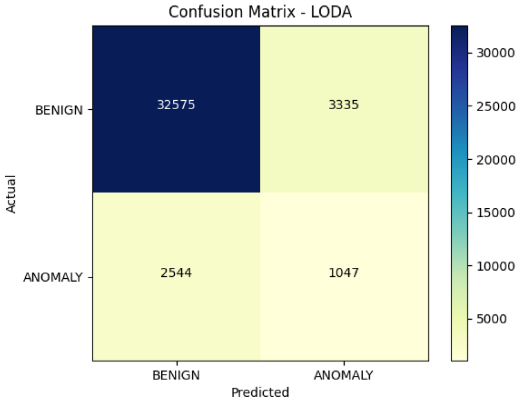

Confusion Matrix - AvgKNN
BENIGN
ANOMALY
Actual
30823
5087
23
3568
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000


Confusion Matrix - MedKNN
BENIGN
ANOMALY
Actual
30717
5193
31
3560
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000

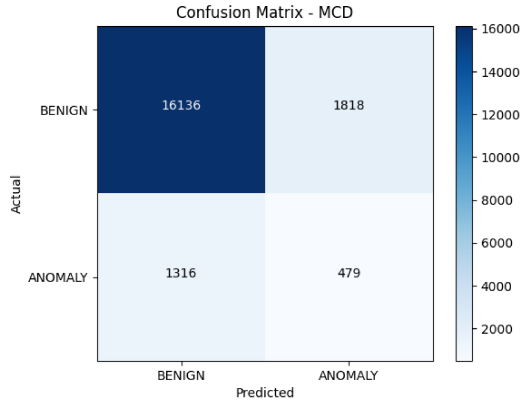

Confusion Matrix - COPOD
BENIGN
ANOMALY
Actual
32312
3598
949
2642
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000

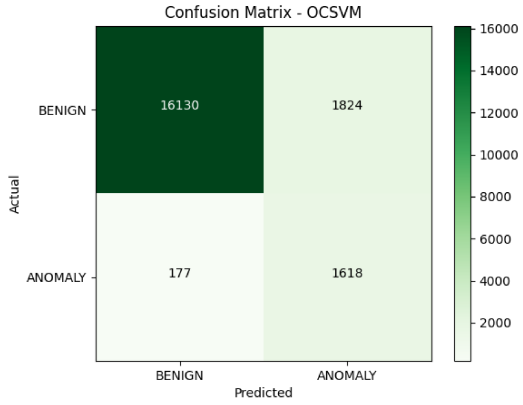

Confusion Matrix - IForest
BENIGN
ANOMALY
Actual
32290
3620
980
2611
BENIGN
ANOMALY
Predicted
30000
25000
20000
15000
10000
5000

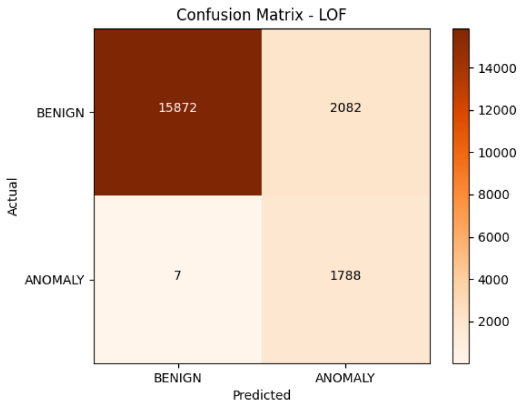


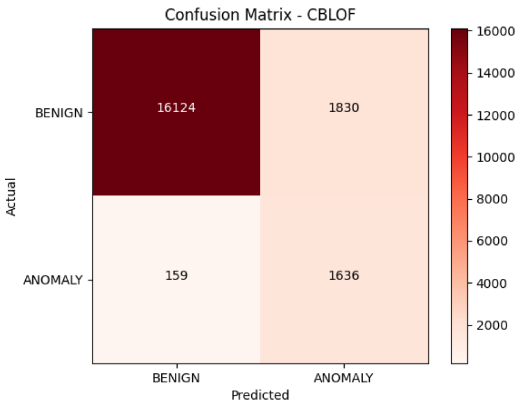


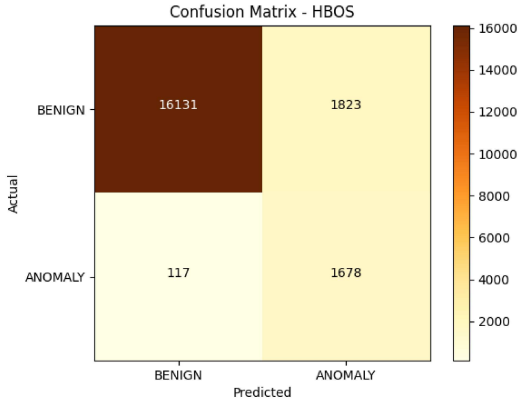
D.2 Matriz de Confusão - Caso de Teste 2

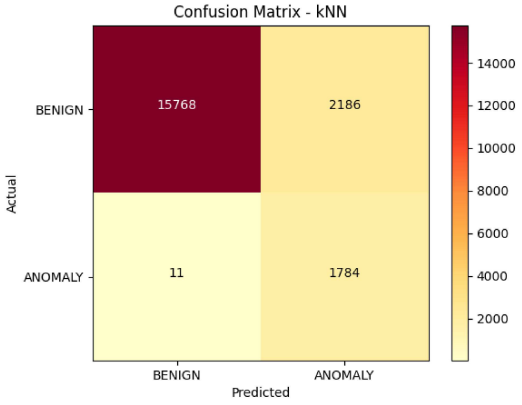


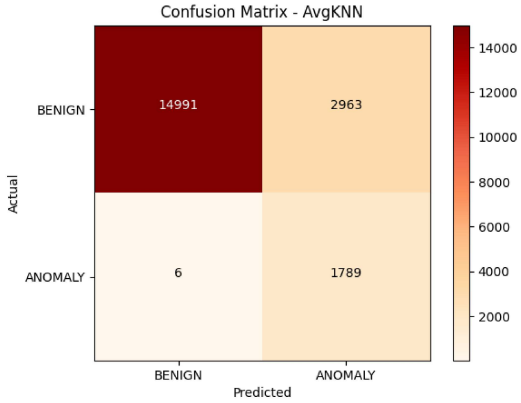


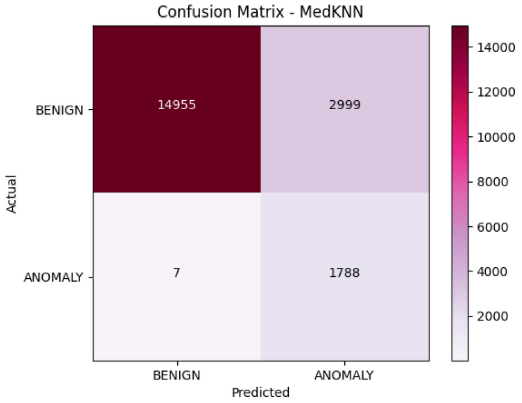


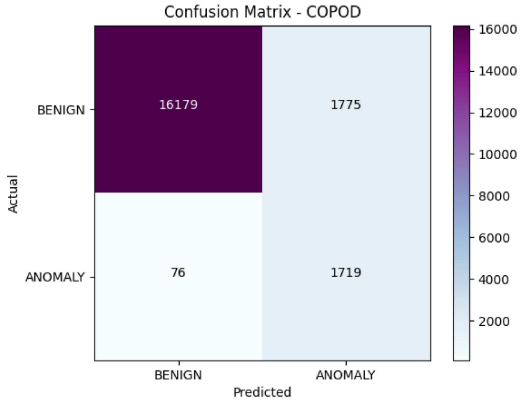


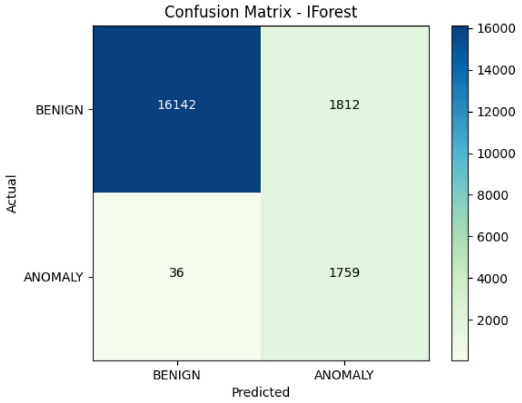

Confusion Matrix - HBOS
BENIGN
ANOMALY
Actual
16131
1823
117
1678
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000

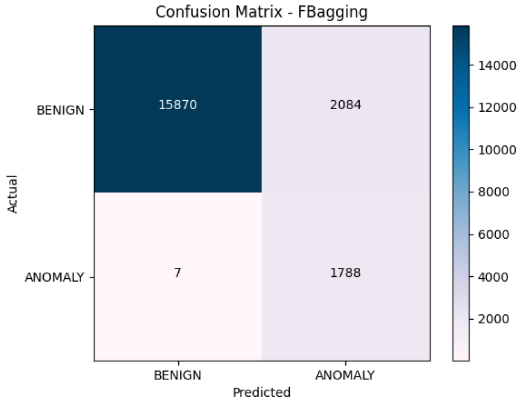

Confusion Matrix - kNN
BENIGN
ANOMALY
Actual
15768
2186
11
1784
BENIGN
ANOMALY
Predicted
14000
12000
10000
8000
6000
4000
2000

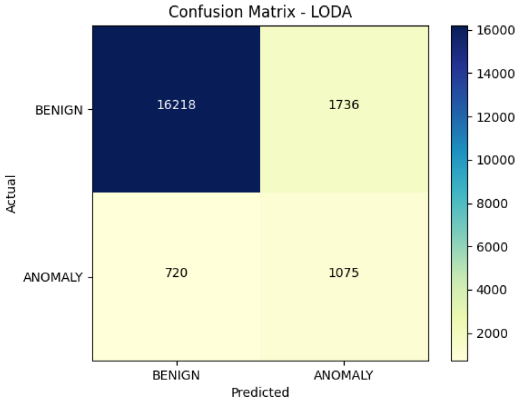

Confusion Matrix - AvgKNN
BENIGN
ANOMALY
Actual
14991
2963
6
1789
BENIGN
ANOMALY
Predicted
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - MedKNN
BENIGN
ANOMALY
Actual
14955
2999
7
1788
BENIGN
ANOMALY
Predicted
14000
12000
10000
8000
6000
4000
2000

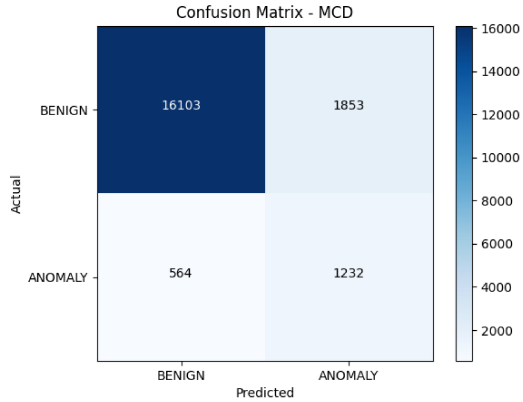

Confusion Matrix - COPOD
BENIGN
ANOMALY
Actual
16179
1775
76
1719
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000

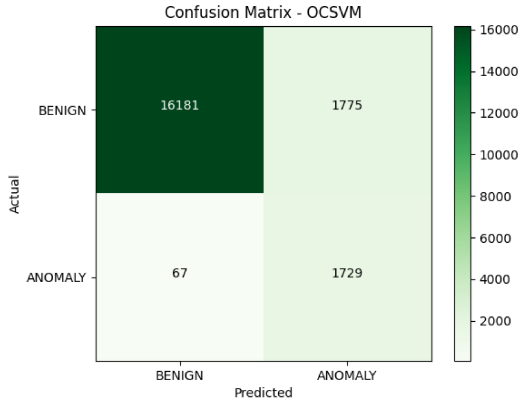

Confusion Matrix - IForest
BENIGN
ANOMALY
Actual
16142
1812
36
1759
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000

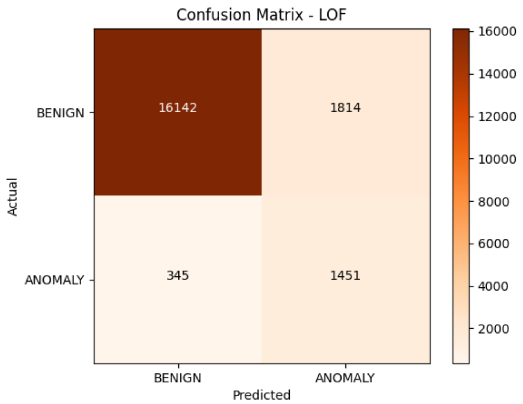


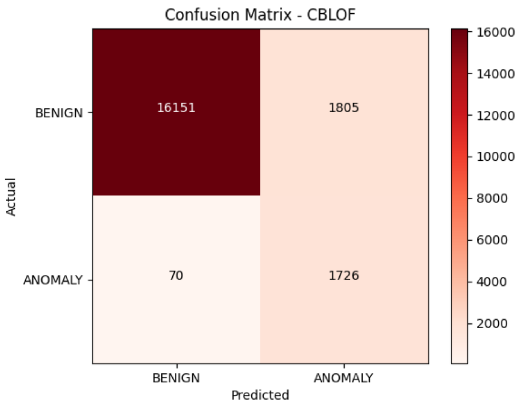


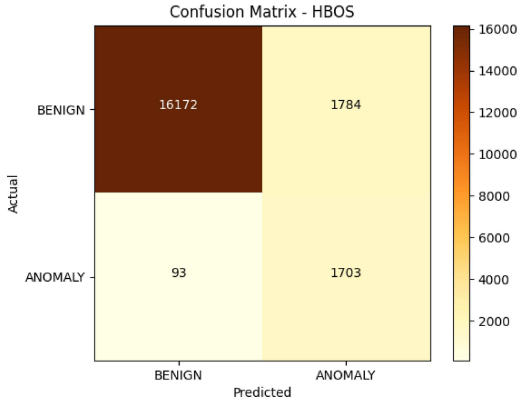
D.3 Matriz de Confusão - Caso de Teste 3

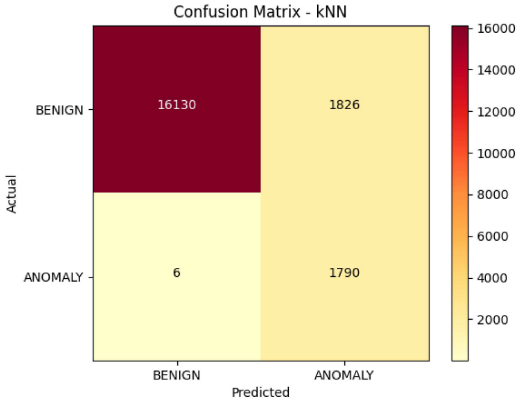


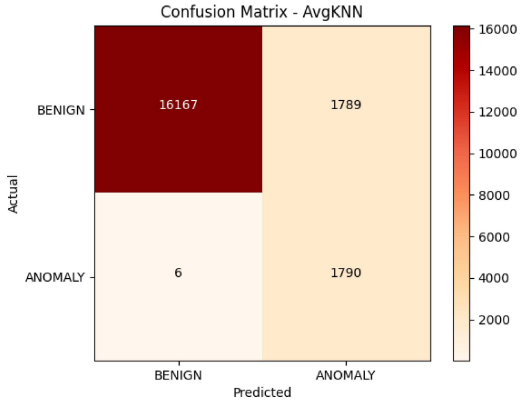


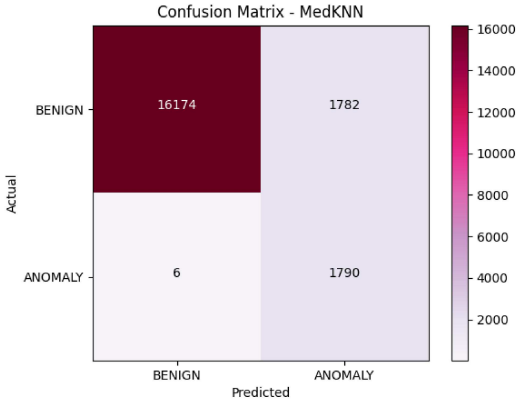


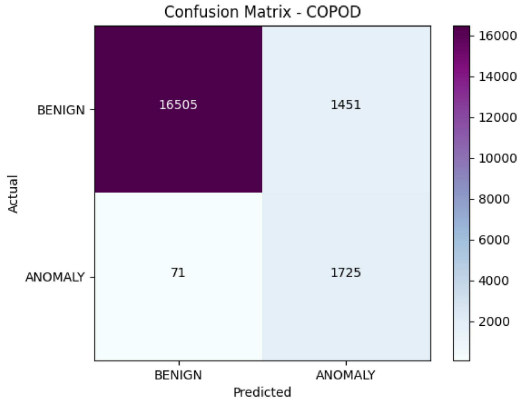


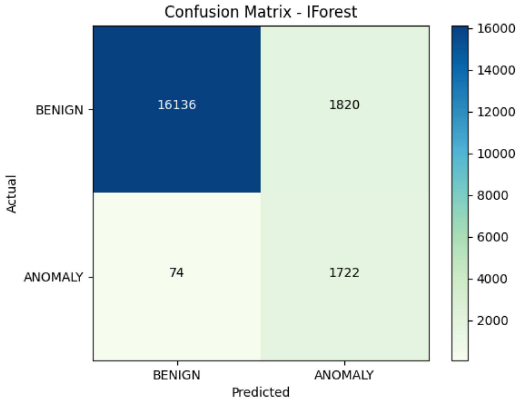

Confusion Matrix - HBOS
BENIGN
ANOMALY
Actual
16172
1784
93
1703
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000

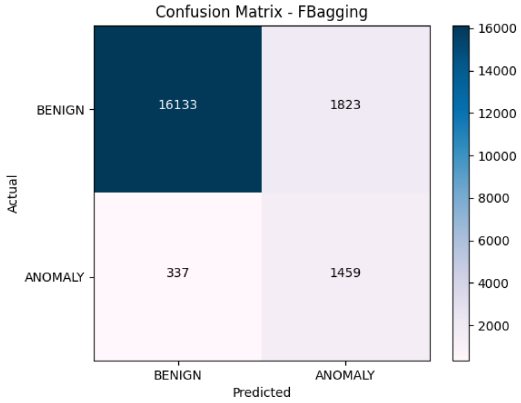

Confusion Matrix - kNN
BENIGN
ANOMALY
Actual
16130
1826
6
1790
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000

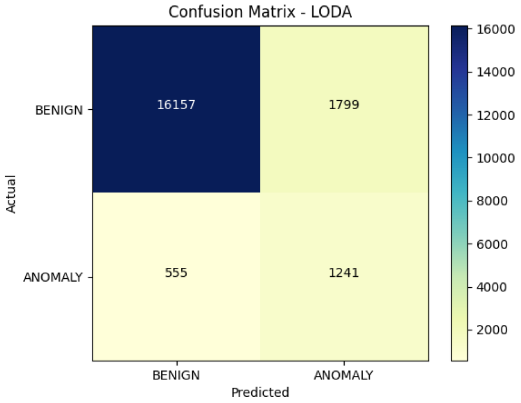

Confusion Matrix - AvgKNN
BENIGN
ANOMALY
Actual
16167
1789
6
1790
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - MedKNN
BENIGN
ANOMALY
Actual
16174
1782
6
1790
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - COPOD
BENIGN
ANOMALY
Actual
16505
1451
71
1725
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - IForest
BENIGN
ANOMALY
Actual
16136
1820
74
1722
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - FBagging
BENIGN
ANOMALY
Actual
16133
1823
337
1459
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000


Confusion Matrix - LODA
BENIGN
ANOMALY
Actual
16157
1799
555
1241
BENIGN
ANOMALY
Predicted
16000
14000
12000
10000
8000
6000
4000
2000