UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

VINÍCIUS EDUARDO OLIVEIRA

Abordagem baseada em clusterização e em Redes Neurais Artificiais para detecção de intrusão em dispositivos IoT

FLORIANÓPOLIS
2020

VINÍCIUS EDUARDO OLIVEIRA

Abordagem baseada em clusterização e em Redes Neurais Artificiais para detecção de intrusão em dispositivos IoT

Trabalho Conclusão do Curso de Graduação em Sistemas de Informação do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Sistemas de Informação
Orientador: Prof. Dr. Carlos Becker Westphall
Coorientador: Cristiano Antônio de Souza

Florianópolis
2020

Ficha de identificação da obra

Oliveira, Vinícius Eduardo
Abordagem baseada em clusterização e em Redes Neurais Artificiais para detecção de intrusão em dispositivos IoT / Vinícius Eduardo Oliveira; orientador, Carlos Becker Westphall, coorientador, Cristiano Antônio de Souza, 2020. 82 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Sistema de Informação, Florianópolis, 2020.

Inclui referências.
1. Sistema de Informação. 2. Internet of Things. 3. Intrusion Detection. 4. Clustering. 5. Artificial Neural Network. I. Westphall, Carlos Becker. II. de Souza, Cristiano Antônio. III. Universidade Federal de Santa Catarina. Graduação em Sistema de Informação. IV. Título.

VINÍCIUS EDUARDO OLIVEIRA

Abordagem baseada em clusterização e em Redes Neurais Artificiais para detecção de intrusão em dispositivos IoT

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel e aprovado em sua forma final pelo Curso de Sistemas de Informação.

Florianópolis, 01 de novembro de 2020

Prof. Dr. Cristian Koliver
Coordenador do Curso

Banca examinadora:

Prof. Dr. Carlos Becker Westphall
Orientador
Universidade Federal de Santa Catarina

Me. Cristiano Antônio de Souza
Coorientador
Universidade Federal de Santa Catarina

Prof.ª Dra. Carla Merkle Westphall
Universidade Federal de Santa Catarina

Prof. Dr. Jorge Werner
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço primeiramente a minha família, aos meus pais Eduardo e Tatiane que sempre se fizeram presentes, me apoiando e me incentivando na busca dos meus objetivos.

A Gabriela, minha noiva, que esteve comigo durante todo o período acadêmico, pelo apoio, carinho e por entender, por vezes, a minha ausência.

Aos meus colegas de curso, Cesar Bess, Lucas Rocha, Pedro Silveira e Pedro Steinheiser, pelos anos de convivência e companheirismo.

Estendo o agradecimento ao meu coorientador, Cristiano Antônio de Souza, pela paciência, atenção e pela contribuição intelectual para este trabalho.

"*O maior inimigo do conhecimento não é a ignorância, é a ilusão do conhecimento*".

(Stephen Hawking)

RESUMO

Computação em nuvem (*Cloud computing*) e a Internet das Coisas (*Internet of Things* – IoT) são tecnologias emergentes e inovadoras que levaram os sistemas de informações a um nível mais amplo com o rápido compartilhamento de vastos recursos da Web pela Internet. A IoT é um importante paradigma que permite que objetos possam se comunicar e interagir com o ambiente em que estão inseridos, além de realizar tarefas de maneira inteligente sem ser necessário intervenção humana. A IoT geralmente possui recursos restritos. Desse modo, as aplicações normalmente utilizam a computação em nuvem para processar e armazenar as informações capturadas pelos dispositivos IoT. Todavia, a separação dos dispositivos IoT e os datacenters de processamento podem gerar alta latência e instabilidades prejudiciais para dispositivos que trabalham com respostas em tempo real. Utilizando computação em névoa (*Fog Computing*), consegue-se resolver esses problemas, tendo em vista que o processamento e o armazenamento encontram-se mais próximos dos dispositivos IoT. Existem inúmeros desafios para garantir um ambiente de IoT e Fog Computing ideal, a segurança é um dos maiores. Considerando os aspectos de segurança, a detecção de intrusão é um ponto chave. Este trabalho de conclusão de curso propõe uma abordagem utilizando clusterização e redes neurais artificiais, que opera na Fog Computing, para detectar intrusão em ambiente IoT. Através dos experimentos foi possível verificar que a abordagem proposta com clusterização é capaz de melhorar a eficácia da rede neural na detecção binária e multiclasse.

Palavras-chave: Internet das Coisas. Detecção de Intrusão. Clusterização. Redes Neurais Artificiais.

ABSTRACT

Cloud computing and the Internet of Things (IoT) are emerging and innovative technologies that have taken information systems to a broader level with the rapid sharing of vast web resources over the Internet. IoT is an important paradigm that allows objects to communicate and interact with the environment in which they are inserted, in addition to performing tasks intelligently without the need for human intervention. IoT often has limited resources. In this way, applications typically use cloud computing to process and store information captured by IoT devices. However, the separation of IoT devices and processing data centers can lead to high latency and harmful instabilities for devices that work with real-time responses. Using fog computing, these problems can be solved, considering that processing and storage are closer to IoT devices. There are countless challenges to guarantee an ideal IoT and Fog Computing environment, security is one of the greatest. Considering security aspects, intrusion detection is a key point. This conclusion work proposes an approach using clustering and artificial neural networks, which operates at Fog Computing, to detect intrusion in an IoT environment. Through the experiments it was possible to verify that a proposed approach with clustering is able to improve the efficiency of the neural network in binary and multiclass detection.

Keywords: Internet of Things. Intrusion detection. Clustering. Artificial Neural Networks.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IoT – Internet of Things
ANN – Artificial Neural Network
RNA – Rede Neural Artificial
DDoS – Distributed Denial-of-Service
IDS – Intrusion Detection System
DECT ULE – Digital Enchanced Cordless Telecommunications Ultra-Low Energy
RFID – Radio Frequency Identification
BLE – Bluetooth Low Energy
NIDS – Network-based Intrusion Detection System
HIDS – Host-based Intrusion Detection System
DBSCAN – Density-Based Spatial Clustering of Applications with Noise
MLP – Multilayer Perceptron
PCA – Principal Component Analysis
SFC – Similarity-based Fuzzy Clustering
FC-IDS – Fog Computing Intrusion Detection System
DoS – Denial of Service
R2L – Root to Local
U2R – User to Root
FN – False Negative
FP – False Positive
TN – True Negative
TP – True Positive
ACC – Acurácia
ERR – Erro
PRE – Precisão
TPR – True Positive Rate
TNR – True Negative Rate
MCC – Matthews Correlation Coefficient
IEEE – Institute of Electrical and Electronics Enginners
UFSC – Universidade Federal de Santa Catarina

SUMÁRIO

1. INTRODUÇÃO

A chegada da Internet das Coisas (*Internet of Things* – IoT) levou à conexão universal de pessoas, objetos, sensores e serviços. O principal objetivo da IoT é fornecer uma infraestrutura de rede com protocolos de comunicação e software, permitindo a conexão e incorporação de sensores físicos, virtuais, computadores e dispositivos inteligentes, como carros, casas, geladeiras, etc. (ALABA et al., 2017).

A maioria dos dispositivos IoT possuem recursos limitados. Portanto, faz-se necessário que os dados desses dispositivos sejam transferidos através da Internet para um centro computacional onde será possível realizar o processamento e armazenamento. Assim, muitos dispositivos utilizam a computação em nuvem (*Cloud Computing*) para essa função. Porém, as grandes quantidades de dados gerados acabam resultando em congestionamento da rede na comunicação da IoT com a *Cloud*, então, para realizar o processamento e armazenamento dos temporários mais próximo aos dispositivos, surgiu a computação em névoa (*Fog Computing*), permitindo além de tudo, fazer com que o processamento em tempo real obtenha uma resposta mais rápida, uma vez que se encontra mais próximo do usuário, utilizando recursos locais como gateways e roteadores (ALABA et al., 2017) (BONOMI et al., 2012).

1.1. MOTIVAÇÃO

Os requisitos para a implantação em larga escala da IoT estão aumentando rapidamente, resultando em uma grande preocupação em segurança. Nos últimos anos vem ocorrendo um grande aumento no número de incidentes computacionais. Por exemplo, o incidente que ocorreu em 2016, um ataque DDoS massivo, sobrecarregando vários alvos em todo o mundo. Esses ataques estavam sob o controle de um novo botnet chamado Mirai. Botnets são dispositivos (computadores, celulares, dispositivos IoT) que são utilizados para realizar ataques em grande escala. No caso da Botnet Mirai, a grande maioria dos dispositivos utilizados eram dispositivos IoT. Estima-se que a Mirai controlava mais de 300 mil dispositivos, incluindo câmeras de segurança e roteadores, redirecionando o tráfego para

realizar ataques DDoS. Câmeras de segurança, por exemplo, eram invadidas minutos após se conectarem à rede. Esses ataques deixaram claro como os dispositivos IoT eram muito inseguros, pois através deles foi possível retirar do ar diversos serviços como o *Twitter, Spotify, Paypal, Playstation Network* entre outros, sendo o Brasil um dos países com mais infecções pelo *malware* da Mirai (ANTONAKAKIS et al., 2017). Questões como, privacidade, autorização, verificação, controle de acesso, armazenamento e gerenciamento de informações, são os principais desafios de um ambiente IoT (RADANLIEV et al., 2019). Portanto, faz-se necessário a utilização de sistemas de detecção de intrusão, tendo como objetivo reconhecer os comportamentos intrusivos em uma rede e alertar os administradores ou realizar ações de contramedidas automaticamente (MAPLE, 2017). A visibilidade oferecida por uma abordagem de detecção nos nós sensores e que reporte eventos para a fog é ideal e pode melhorar a detecção geral de ataques, no entanto, as capacidades restritas de recursos inviabilizam as ferramentas tradicionais de detecção (ZARPELÃO et al., 2017). Portanto, existe a necessidade de uma abordagem que realize a detecção de intrusões implantada na Fog monitorando o tráfego na rede de dispositivos IoT.

Neste trabalho é realizada uma revisão do atual estado da arte relacionado a detecção de intrusão em ambiente de *Fog Computing* e IoT, identificando os principais problemas e soluções existentes, além dos desafios futuros. Além disso, é apresentada uma proposta de uma abordagem utilizando clusterização e Redes Neurais Artificiais (*Artificial Neural Network* – ANN), que opera na *Fog Computing*, para detectar intrusão em ambiente IoT.

1.2. OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste TCC.

1.2.1. Objetivo Geral

Propor uma abordagem baseada em clusterização em Redes Neurais Artificiais, que opera na *Fog Computing* para detecção de intrusão em ambientes IoT e investigar a influência dos métodos de clusterização na eficácia da Rede Neural Artificial.

1.2.2. Objetivos Específicos

Como objetivos específicos foram listados os três principais:

1. Realizar uma contextualização sobre o atual estado da arte em relação à segurança em *Fog Computing* e IoT, de modo a identificar quais as questões em aberto;
2. Propor uma abordagem baseada em clusterização e Redes Neurais Artificiais, que opera na *Fog Computing*, para detecção de intrusão em ambientes IoT.
3. Investigar se a clusterização pode influenciar positivamente na tarefa de detecção com RNA.
4. Realizar experimentos com uma base de dados de intrusões para avaliação da viabilidade de aplicação em ambiente real.

1.3. ORGANIZAÇÃO DO TRABALHO

No texto que segue, o Capítulo 2 trata dos conceitos fundamentais sobre IoT*, Cloud Computing, Fog Computing*. Além disso, são apresentados os conceitos de sistemas de detecção de intrusão e de técnicas de aprendizado de máquina, como clusterização e Redes Neurais Artificiais. O Capítulo 3 trata da contextualização do estado atual das pesquisas e estudos do tema do trabalho, contendo um resumo de quatro artigos sobre o tema. No Capítulo 4, são apresentados detalhes a respeito da abordagem proposta, onde é justificado a escolha desse tema do trabalho e explicado como o que será desenvolvido. No Capítulo 5 é apresentada a metodologia para a avaliação da proposta, os resultados obtidos com os

experimentos e uma discussão sobre os mesmos. No Capítulo 6, conclusão e trabalhos futuros.

2. CONCEITOS FUNDAMENTAIS

Neste Capítulo são apresentados alguns conceitos básicos envolvidos na temática deste trabalho. É descrito o conceito de IoT e as principais tecnologias de comunicação relacionadas; a definição de *Fog Computing* e a integração dos dispositivos IoT com a mesma. Com foco na detecção de intrusão, é apresentado a definição de um sistema de detecção de intrusão (*Intrusion Detection System* – IDS), bem como os tipos de detecção existentes, os tipos de arquiteturas de capturas de eventos, o comportamento pós detecção e a frequência de uso. Por fim, é descrito o conceito de clusterização e os métodos que serão utilizados neste trabalho, além da definição de Redes Neurais Artificiais, com suas características e seu funcionamento.

2.1. IOT

A Internet das Coisas é uma das tecnologias mais inovadoras da atualidade. O termo se refere ao conceito de conectividade entre dispositivos, e entre dispositivos e sistemas. Esses dispositivos podem ser utilizados em diversas áreas, como casas e carros inteligentes, agricultura, medicina, entre outros. Essa troca de informações é possível graças a sensores e atuadores que estão embarcados nesses dispositivos. A Internet das Coisas busca eliminar a necessidade de intervenção humana em diversos aspectos, facilitando ao mesmo tempo a vida de quem a utiliza (ALABA et al., 2017).

2.1.1. Tecnologias de comunicação

Existem diversas tecnologias de comunicação que são utilizadas em IoT. Uma delas é conhecida como “*Digital Enchanced Cordless Telecommunications Ultra-Low Energy*” (DECT ULE), que fornece transmissão de dados de baixa largura de banda e baixa potência. Pode ser usado para transmitir mensagens de status

para um profissional da saúde onde se usa muito pouca energia da bateria (BELLO; ZEADALLY; BADRA, 2017).

Outro exemplo é o *IEEE 1901.2 standard*, que permite comunicação através de linhas elétricas alternadas e contínuas para dispositivos com frequência inferiores a 500KHz. Pode ser utilizado na rede doméstica, em eletrodomésticos, onde é possível que esses dispositivos interajam com a rede elétrica para relatar a energia utilizada na sua utilização e permitir aos usuários ajustar o seu consumo de energia (BELLO; ZEADALLY; BADRA, 2017).

Na Identificação de Frequência de Rádio (*Radio Frequency Identification* - RFID), é possível utilizar sinais de radiofrequência para identificar e monitorar objetos ou pessoas em tempo real. Através de uma tag (microchip), há comunicação via links sem fio em frequências de rádio entre 125 KHz e 915Mhz. Essas tags são dispositivos somente leitura que não possui capacidade de processamento. Um exemplo clássico são as etiquetas utilizadas em roupas em lojas. Se essas roupas saírem da loja sem que essas tags sejam desativadas, um alarme é acionado (BELLO; ZEADALLY; BADRA, 2017).

O *Bluetooth Low Energy* (BLE), que é um aprimoramento do bluetooth clássico, onde permite a conectividade sem fio para dispositivos de baixo custo e com baixa potência. Pode ser utilizado para dispositivos que desejam transferir pequenas quantidades de dados em intervalos relativamente curtos. No BLE existem dois tipos de canais, que são chamados de piconet e publicidade. Os dispositivos não podem utilizar os dois canais simultaneamente. Os canais piconet são utilizados para a comunicação dos dispositivos conectados e os de publicidade para transmitir informações para os dispositivos desconectados ou configurar conexões (BELLO; ZEADALLY; BADRA, 2017).

Zigbee consegue fornecer comunicações sem fio com baixo custo e baixo consumo de energia. Os dispositivos Zigbee trocam informações em pacotes de dados. Um pacote de dado é a menor unidade de informação transmitida pelo ar. Esse pacote pode ser dividido em três partes:

- Cabeçalho: que contém informações sobre o conteúdo no corpo da mensagem;
- Corpo da mensagem: é onde estão os dados;

- Rodapé: que contém informações que possam identificar erros no pacote recebido (BRONZATTI, 2013).

A função dos dispositivos Zigbee podem ser classificadas em três modos: coordenador (que gerencia a rede, sendo responsável pela maioria das atividades na rede), o roteador (que repassa as mensagens de um dispositivo para o outro) e os *end-devices* (apenas recebem/transmitem mensagens e não as reencaminha para outros dispositivos) (BRONZATTI, 2013).

2.2. CLOUD COMPUTING

A computação em nuvem é um modelo que permite acesso onipresente, conveniente e sob demanda da rede a um conjunto de recursos de computação configuráveis (redes, servidores, aplicativos, armazenamento, serviços) que pode ser rapidamente provisionado e liberado com o mínimo esforço de gerenciamento ou interação do provedor de serviços. Na nuvem, não há necessidade de se preocupar com o tamanho e não há necessidade de dar manutenção para aplicativos. Além disso, como a maior parte do processamento fica na nuvem, é possível obter um maior aproveitamento dos investimentos em hardware, já que é possível acessar aplicações e serviços de qualquer lugar, basta estar conectado à internet (MELL; GRANCE, 2011).

A segurança em nuvem é responsável pela proteção de informações essenciais contra roubo, vazamento de dados e exclusão. Ela é parecida como a segurança que gerenciamos, mas, com outras maneiras e soluções. É mais fácil, em nuvem, fazer o gerenciamento da segurança, pois é mais ágil efetuar ações de detecção e correção (MELL; GRANCE, 2011).

Os dados estão armazenados em *datacenters*, e em alguns países que exigem que os dados sejam armazenados localmente, a escolha de um provedor que tenha datacenters espalhados por todo mundo pode ser uma ótima escolha (MELL; GRANCE, 2011).

Geralmente, o armazenamento de dados inclui determinados requisitos de conformidade, especialmente durante o armazenamento de números de cartão de crédito ou informações de saúde. Muitos provedores de nuvem oferecem relatórios de auditoria independentes elaborados por terceiros para atestar que seu processo

interno existe e é eficaz no gerenciamento da segurança dentro de suas instalações durante o armazenamento de dados (MELL; GRANCE, 2011).

As características básicas supracitadas da computação em nuvem, a tornam um importante recurso de processamento para aplicações IoT. Principalmente, aquelas compostas por uma grande quantidade de sensores, que lidam com grandes extrações de informações dos ambientes em que estão inseridas.

2.3. IOT E CLOUD COMPUTING

Hoje, as pessoas estão familiarizadas com a computação em nuvem e a expansão da Internet das Coisas (IoT). O momento atual exige integração da IoT e computação em nuvem. Com isso acontecendo, num futuro próximo, o número de dispositivos conectados seria centenas de vezes maior que o número de pessoas conectadas (AAZAM et al., 2014).

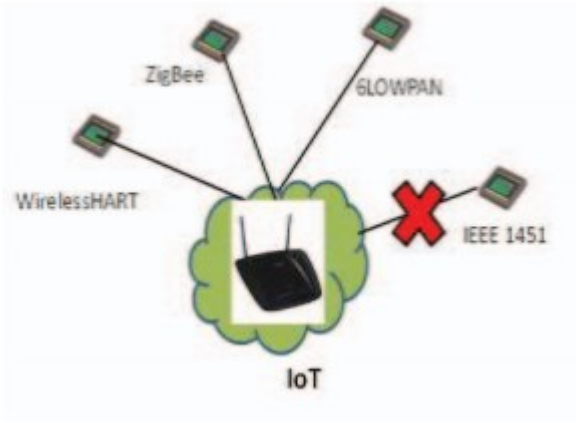
Não é tão simples permitir que tudo se torne parte da IoT e, em seguida, ter todos os recursos disponíveis através computação em nuvem. Existem algumas questões que devem ser cuidadas para permitir que a *"Cloud of Things"* prevaleça para a melhoria do mundo em geral. Além de dados e recursos, a nuvem tem que lidar com o ponto de vista do negócio. A nuvem das coisas criará mais oportunidades de negócios, e consequentemente aumentando a chance de ameaças ou ataques (AAZAM et al., 2014).

A proteção de identidades torna-se algo muito importante em nuvens híbridas utilizadas por empresas. A rede deve ter flexibilidade para suportar todos os tipos de dados, de acordo com as necessidades. Algum dos principais problemas são discutidos a seguir:

- Suporte ao protocolo: para que diferentes coisas sejam conectadas à Internet, diferentes protocolos estarão sendo utilizados. Mesmo se houver entidades homogêneas, como por exemplo um sensor IoT, então existe a possibilidade desses sensores estarem trabalhando em diferentes protocolos. Alguns dos protocolos serão suportados pelo gateway, outros podem não ser. Na perspectiva do usuário, um sensor mais barato teria preferência, mas, não pode ser garantido que um sensor recém adicionado

será configurado com sucesso. Para corrigir esse problema, seria necessário mapear os protocolos padronizados no gateway. A Figura 1 apresenta a ilustração de alguns tipos de protocolos existentes.

Figura 1 - Tipos de protocolos.



Fonte: AAZAM et al., 2014.

- Gerenciamento de identidade: os nós de comunicação na Internet são identificados unicamente. Quando objetos estão se tornando parte da IoT, eles também precisam de uma identificação única. Além disso, como no caso de celulares, por exemplo, sensores móveis em veículos e outros objetos, precisam ter mapeamento de identidade na nova rede que acabaram de entrar. Como o espaço de endereços IPv6 é o suficiente para apoiar até mesmo esse tipo de rede, atribuição de endereços IPv6 pode ser uma maneira mais do que razoável a para esse problema;
- Implantação IPv6: se o IPv6 for utilizado para a identificação de comunicações de objetos, a implantação formal do IPv6 também seria um problema. Se não houver um mecanismo adequado, padronizado e eficiente, o IPv6 não seria de grande ajuda;
- Segurança e privacidade: é um grande problema com o tipo de computação que temos hoje. A segurança de dados pode ser um problema junto à IoT, e a privacidade ao lado da nuvem. Portanto, dados confidenciais ou privados devem ser armazenados em um servidor de armazenamento virtual localizado dentro do país ou num domínio geográfico confiável (AAZAM et al., 2014).

A *Cloud Computing* pode ter alta latência por causa da grande quantidade de dispositivos e de dados que podem ser gerados, e, da grande distância entre as aplicações e os data centers das nuvens. Diante disso, surge a *Fog*, que fica mais próxima da borda da rede, diminuindo a latência e consequentemente melhorando o desempenho.

2.4. FOG COMPUTING E IOT

Fog Computing é uma plataforma altamente virtualizada, podendo ser composta por roteadores e gateways, que fornece serviços de computação, armazenamento e rede entre dispositivos finais e data centers de computação em nuvem, normalmente, mas não exclusivamente, na borda da rede. Em 2012, a Cisco introduziu o conceito de *Fog Computing* com o intuito de solucionar os obstáculos de latência enfrentados pelas aplicações IoT que utilizam a computação em nuvem para acesso, processamento e armazenamento de dados. A *Fog Computing* é um paradigma de computação distribuída que atua em uma camada intermediária, entre a nuvem e os dispositivos de IoT (BONOMI et al., 2012). Em outras palavras, na *Fog* os dados podem ser analisados e processados por aplicativos dentro da rede e não diretamente na nuvem. A *Fog* surgiu para trazer esse processamento para mais próximos dos dispositivos IoT de modo a reduzir a latência, visto que esses dispositivos possuem pouca capacidade de processamento e armazenamento. Portanto, a *Fog* e a *Cloud* interagem com o objetivo de se beneficiar uma da outra, tendo em vista que somente com uma combinação inteligente de comunicações é possível atender aos requisitos da IoT (YANNUZZI et al., 2014) (ALABA et al., 2017).

2.5. IDS

Nas últimas décadas, devido ao grande aumento do uso de redes, começaram a aparecer diversos problemas de segurança relacionado a Internet e os sistemas. Qualquer invasão ou ataque as vulnerabilidades da rede, sistemas ou computadores podem violar as políticas de segurança e até causar catástrofes. Para sanar esses problemas, uma solução são os Sistemas de Detecção de Intrusão

(*Intrusion Detection System* - IDS), que fortalecem a segurança dos sistemas de informação e automatizam o processo de detecção de intrusão. O IDS monitora e coleta dados de um sistema que deve ser protegido, correlaciona as informações coletadas e inicia as respostas quando detecta uma intrusão (LIN; ZHANG; OU, 2010) (LIAO et al., 2013).

2.5.1. Tipos de detecção

A classificação mais tradicional dos sistemas de detecção de intrusão é quanto as suas metodologias de detecção, que podem ser baseadas por regras ou anomalia. Na detecção por regras, assinaturas são geradas para representar o comportamento de ataques já conhecidos. Essas assinaturas são relacionadas e os eventos do sistema são comparados com as mesmas, com o objetivo de identificar um padrão de comportamento que se encaixe nas especificações da assinatura (FERREIRA, 2016).

Alguns ataques são desenvolvidos através de evoluções de ataques que já são conhecidos. Portanto, essas assinaturas contribuem com a localização das tentativas de quebrar a segurança, sendo que através de uma confirmação parcial é possível indicar uma tentativa de intrusão (FERREIRA, 2016).

Um dos principais problemas nesse tipo de detecção é por ser ineficiente contra comportamentos intrusivos que ainda não foram descobertos ou divulgados, uma vez que para ser eficiente, esse método precisa de atualizações frequentes das assinaturas de intrusão do sistema (FERREIRA, 2016).

A detecção por anomalia é um importante problema e tem sido estudado em diversas áreas de pesquisa e domínios de aplicação. Esse tipo de detecção, busca identificar dados que não estão em conformidade com o comportamento esperado. Esses desvios do comportamento esperado são chamados de anomalias (MUSSOI DE LIMA, 2005).

A detecção por anomalia é utilizada em diversos aplicativos como detecção de fraude em cartões de crédito, seguro, assistência médica, etc. O motivo da detecção por anomalias ser tão importante se dá pelo fato que os dados com as anomalias se traduzem em informações significativas. Por exemplo, um tráfego anômalo em uma rede de computadores pode significar que um computador

hackeado está enviando dados confidenciais à um destino que não foi autorizado (MUSSOI DE LIMA, 2005).

Nesse tipo de detecção, qualquer comportamento anômalo é considerado intrusivo. Porém, algumas atividades anômalas podem não ser intrusivas. Existem quatro estados de detecção para essas atividades:

- Intrusivo e anômalo: a atividade é intrusiva e apontada como tal por ser também anômala; são conhecidos como falsos positivos;
- Não intrusivo e não anômalo: esses são os verdadeiros negativos, onde a atividade não é anômala e não é apontada como intrusiva.
- Intrusivo, mas não anômalo: são os falsos negativos, onde é intrusiva, mas não anômala, gerando uma falha na detecção.
- Não intrusivo, mas anômalo: falsos positivos (como são chamados), não é intrusivo, mas por ser anômalo, o sistema entende como intrusivo, reportando de forma incorreta (FERREIRA, 2016).

Para um sistema de detecção de intrusão eficiente, é preciso definir parâmetros que apontam os comportamentos anômalos, pois um nível alto de detecções falso-positivas pode comprometer a eficiência, gerando grande quantidade de alertas nas atividades normais dos usuários, informando que são intrusivas. Todavia, é preciso ajustar também de forma que não ocorram detecções falsas negativas (GARCÍA-TEODORO et al., 2009) (MUSSOI DE LIMA, 2005).

2.5.2. Tipos de arquiteturas de capturas de eventos

Existem dois tipos diferentes de detecção de intrusão que podem ser empregados: os baseados em rede e os baseados na estação (host). Os sistemas de detecção de intrusão em redes (*Network-based Intrusion Detection System* - NIDS) são sistemas responsáveis por monitorar ambientes de redes e, como objetivo, detectar ações maliciosas sobre as informações e serviços em execução nesses ambientes (FERREIRA, 2016).

Utiliza-se dois componentes principais nesse tipo de sistema de detecção de intrusão, que são os sensores e estação de gerenciamento. Esses sensores são componentes distribuídos estrategicamente em determinados segmentos de rede. Eles monitoram, além da máquina onde estão instalados, todo o tráfego do

seguimento, pois a interface é configurada para capturar os pacotes de todo o tráfego da rede e não somente da estação onde foi instalado. Os dados capturados são selecionados para depois serem analisados de acordo com o mecanismo de detecção adotado. Feito isso, esse mecanismo ficará responsável por categorizar os eventos e identificar se é ou não intrusivo. Os que forem, serão reportados para a estação de gerenciamento, que tem como responsabilidade realizar o tratamento adequado para cada situação (MUSSOI DE LIMA, 2005).

Ao contrário dos NIDS, os sistemas de intrusão baseados em host (*Host-based Intrusion Detection System* - HIDS) são sistemas de detecção de intrusão que verificam os sinais de intrusão nas próprias máquinas onde estão instalados, utilizando-se de registros de logs e dos registros do sistema operacional (LIN; ZHANG; OU, 2010).

Uma das dificuldades dos HIDS é a análise frequente de um sistema específico, acarretando na perda de desempenho, e, podendo ser atacado, comprometendo o controle de logs e afetando a confiabilidade da detecção (LIN; ZHANG; OU, 2010).

É possível, através desse tipo de IDS, verificar a integridade dos arquivos, checando possíveis alterações desde a última verificação. Para esse processo, é gerado um *hash* de cada arquivo. Esses *hashs* serão armazenados e serão utilizados no futuro para verificar se houve ou não alterações nos arquivos em questão (LIN; ZHANG; OU, 2010).

2.5.3. Arquitetura segundo o local

Os IDS também podem ser classificados de acordo com o modo pelo qual os seus componentes estão dispostos. Desse modo, os IDS podem ser classificados em centralizados, parcialmente distribuídos e distribuídos (CAMPELLO; WEBER, 2001).

Os IDS centralizados possuem todos seus componentes funcionais em apenas um único ponto, desde a captura dos eventos até a configuração e gerência. Como vantagens, esse tipo de IDS apresenta facilidade no desenvolvimento, instalação e configuração. Como desvantagem, uma solução totalmente

centralizada pode ser inviável em sistemas complexos (PORRAS; NEUMANN, 1997).

Os IDS que possuem seus componentes dispostos em pontos diversificados e se comunicam através de mensagens são chamados de IDS distribuídos. Esse tipo de IDS pode ter grupos de detectores e analisadores em locais distintos o que fornece maior abrangência de detecção, com módulos espalhados por diferentes pontos do sistema (ASAKA et al., 1999).

2.5.4. Comportamento pós detecção

O comportamento pós-detecção corresponde a geração de respostas após a detecção de um evento intrusivo. Algumas das respostas comumente geradas por IDS são confecções de relatórios de resultados, coleta de informações e reconfigurações de serviços (BACE; MELL, 2001).

Abordagens passivas pós-detecção consistem na emissão de alarmes e notificações para informar aos usuários quando um ataque é detectado. A outra classe de abordagens pós detecção é a ativa, onde uma das principais respostas utilizadas é a coleta de informações adicionais sobre os ataques. Além disso, outra resposta viável é interromper um ataque em andamento e em seguida bloquear o acesso do atacante (BACE; MELL, 2001).

2.5.5. Frequência de uso

A frequência de uso do IDS refere-se ao tempo decorrido entre os eventos que são monitorados e a análise desses eventos.

Os IDS baseados em intervalos, também conhecidos como IDS de análise periódica, não possuem fluxo contínuo entre os pontos de monitoramento e os mecanismos de análise. Os dados coletados pelos pontos de monitoramento são analisados em períodos predefinidos. Desse modo, o uso de processamento pode ser reduzido e melhor gerenciado (ILGUN, 1993). Importante ressaltar que IDS baseados em intervalos não conseguem realizar respostas ativas (ILGUN, 1993).

Diferentemente dos IDS baseados em intervalo, os IDS de monitoramento contínuo, também conhecidos como IDS de tempo real, realizam uma análise

imediata nos eventos coletados pelos pontos de monitoramento. Esse tipo de IDS é geralmente encontrado em rede, pois os dados analisados são provenientes de tráfego constante de pacotes na rede, exigindo uma análise imediata (BACE; MELL, 2001).

2.6. CLUSTERIZAÇÃO

Clusterização é a classificação não supervisionada de padrões (observações, itens, dados, vetores, etc.) em grupos (*cluster*). Diferente do conceito de classificação, a clusterização é uma técnica mais “primitiva”, onde não existem suposições acerca dos grupos (*cluster*). A ideia básica é que os elementos que compõem um determinado cluster devem apresentar alta similaridade, mas devem ser muito dissimilares de objetos de outros clusters. A grande vantagem de utilizar clusterização é que ao agrupar os dados similares, é possível descrever de forma mais eficiente e eficaz as características de cada grupo, fornecendo maior entendimento do conjunto de dados original. Existem diversos algoritmos que podem ser usados para realizar essas classificações, como podemos ver a seguir.

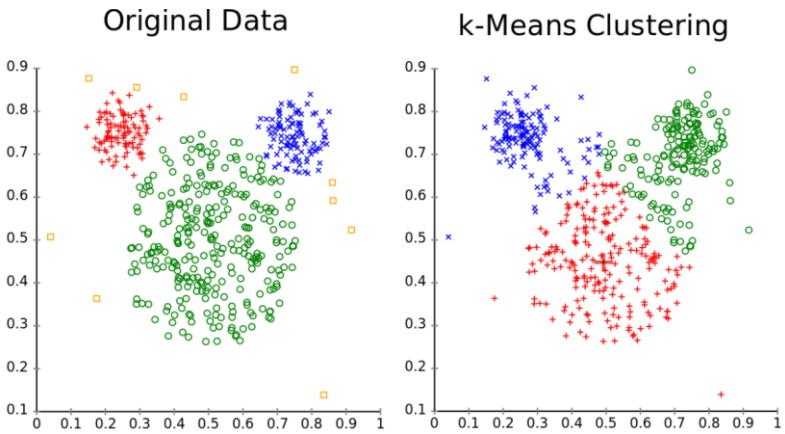
2.6.1. Métodos de clusterização

Os métodos de clusterização utilizam como entrada um conjunto de exemplos no formato atributo-valor. O objetivo de um método de clusterização é dividir um conjunto de objetos em clusters. Uma divisão do conjunto de dados é chamada de partição.

O método *K-means* é um dos principais métodos de clusterização. Para utilização desse método é necessário que seja passado como parâmetro de entrada o número de clusters (K) que deve ser construído, e a métrica de distância que deve ser utilizada (comumente é utilizada a distância euclidiana). Então, é escolhido um conjunto C de K pontos distintos para serem definidos como centros dos grupos, chamados de centroides. Cada cluster (K) está associado a um centroide. Os centroides são recalculados como sendo os pontos médios de todos os pontos pertencentes ao cluster (K). Esse processo se repete até que não haja mais troca de exemplos entre os clusters (BARCELLOS et. al, 2016). Na Figura 2 podemos

observar a diferença da classificação dos dados após a utilizando do método de clusterização *K-means*.

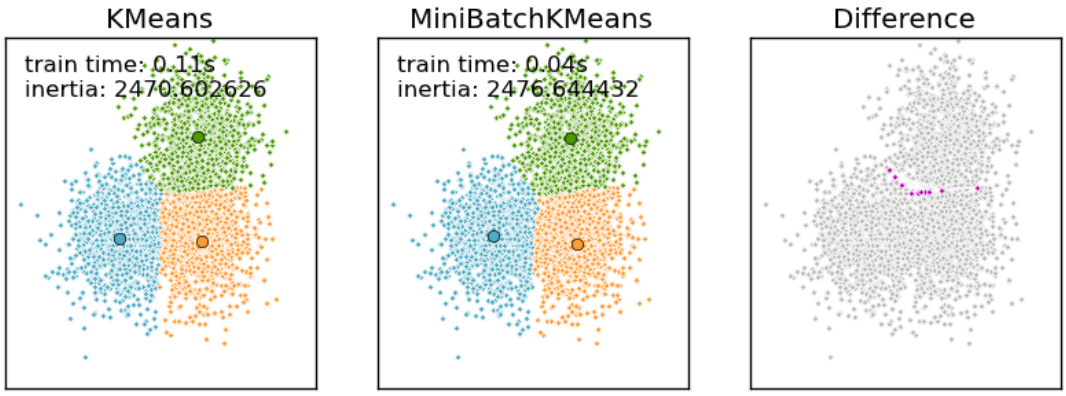
Figura 2 - Método de clusterização K-means.



Fonte: K-means clustering | Wikiwand, 2020.

Para esse trabalho foi utilizado uma variação desse método denominado *Mini Batch K-means*. O *Mini Batch K-means* possui um melhor tempo de processamento em relação ao *K-means*. Nesse método são utilizados *mini-batchs* (mini lotes), que são subconjuntos dos dados de entrada amostrados aleatoriamente em cada iteração de treinamento. Esses mini lotes diminuem a quantidade de computação necessária para conseguir a solução. Funciona em duas etapas principais: na primeira etapa as amostras são retiradas aleatoriamente do conjunto de dados, formando um mini lote. Esses mini lotes são atribuídos ao centroide mais próximo. Na segunda etapa esses centroides são atualizados. Para cada amostra no mini lote, o centroide atribuído é atualizado tomando a média de fluxo da amostra e todas as amostras atribuídas anteriormente ao mesmo. Ao longo do tempo, essas atualizações conseguem diminuir a taxa de alteração de um centroide. A Figura 3 representa a diferença entre o método *K-means* e o *Mini Batch K-means*. Nessa figura é possível observar que o tempo de processamento no segundo método foi inferior, além da diferença na classificação da utilização dos dois métodos.

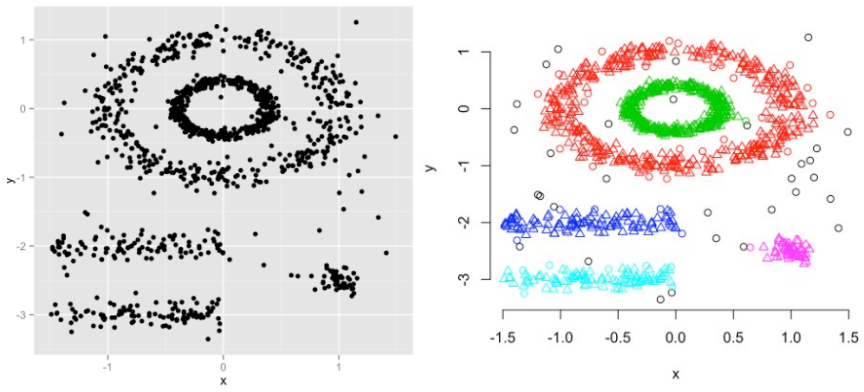
Figura 3 - Comparação entre o método de clusterização *K-means* e *Mini Batch K-means.*



Fonte: A demo of the K-Means clustering algorithm - scikit-learn 0.11-git documentation, 2020.

Outro tipo de método de clusterização são os métodos baseados em densidade. Neste caso, os clusters são definidos como regiões densas, separadas por regiões menos densas. Esse método se difere pela forma com que crescem os clusters: através da densidade da vizinhança dos objetos, ou, de acordo com alguma função de densidade (MENEZES, 2013). Para desenvolvimento do trabalho o método baseado em densidade escolhido foi o Agrupamento Espacial Baseado em Densidade de Aplicações com Ruído (*Density-Based Spatial Clustering of Applications with Noise* – DBSCAN). Esse método consegue separar os clusters em áreas de alta densidade e de baixa densidade. Aqui, os clusters podem ter qualquer formato, ao contrário do *K-means* que têm formato convexo. Um cluster, no DBSCAN é um conjunto de amostras de núcleo, que são amostras que estão em áreas de alta densidade. No cluster também existe as amostras não essenciais, que são as amostras vizinhas de uma amostra principal (que estão a margem de um cluster). Na Figura 4 é possível perceber as áreas com maior densidade separadas pelo método DBSCAN.

Figura 4 - Ilustração do método DBSCAN.



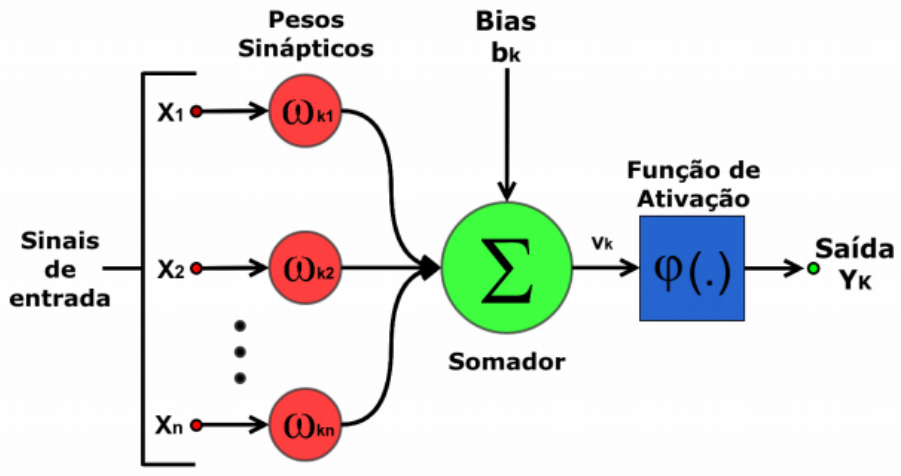
Fonte: Unsupervised Machine Learning – Easy Guides – Wiki – STHDA, 2020.

2.7. REDES NEURAIS ARTIFICIAIS

Por muito tempo cientistas tentaram emular o sistema neural real, acreditando que o processo humano de aprendizagem pudesse ser reproduzido por um algoritmo. Na tentativa de simular o ambiente do sistema nervoso biológico surgiu as redes neurais artificiais (*Artificial Neural Network* – ANN), combinando vários elementos de computação simples (neurônios) em um sistema interconectado, que, através da sua auto-organização e aprendizagem, esperava-se surgir fenômenos complexos, como a “inteligência”. Uma rede neural artificial é capaz de processar grande quantidade de dados e fazer previsões que podem ser surpreendentemente precisas (SARLE, 1994).

Como citado anteriormente, uma das principais características de uma RNA é poder aprender e melhorar seu desempenho através do processo de treinamento, pois após cada iteração do processo de treinamento ela se torna mais conhecedora do seu próprio ambiente. No neurônio artificial, os dendritos são substituídos por entradas e as ligações dessas entradas com o corpo celular artificial são conhecidas como pesos, que simulam as sinapses do nosso sistema nervoso. Os estímulos recebidos pelas entradas são processados pela função de soma, demonstrado na Figura 5 e o limiar de disparo do neurônio biológico é simulado pela função de ativação do neurônio artificial (CHU; YANG, 1988). A Figura 5 ilustra como funciona um Neurônio Artificial.

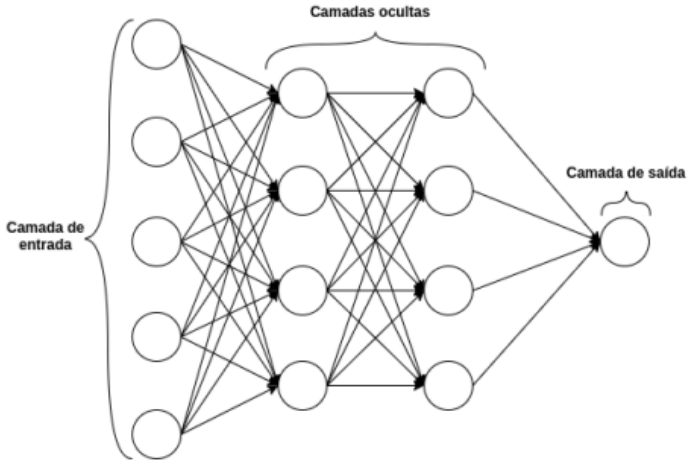
Figura 5 - Modelo simplificado de um Neurônio Artificial.



Fonte: Haykin, 2001.

Existem diversos modelos de rede neural artificial. A Figura 6 demonstra uma estrutura básica das redes neurais *Multilayer Perceptron* (MLP) *feedforward*. Nesse modelo o fluxo do processo sináptico ocorre da camada de entrada em direção a saída. Na MLP pode haver diversas variações no design da sua camada oculta pois não há uma quantidade específica de camadas e de neurônios que cada uma dessas camadas pode ter.

Figura 6 - Arquitetura simplificada de uma rede neural *feedforward*.



Fonte: O autor.

O treinamento desse tipo de rede geralmente é realizado através do algoritmo *backpropagation*. Utilizando-se da propagação dos pesos sinápticos da

camada de entrada para a camada de saída, passando por todas as camadas ocultas, sem haver alteração nos pesos, é realizado o processo de treinamento. O erro é calculado através do resultado esperado e o valor de saída da última camada. Assim, há um ajuste nos pesos baseado no erro calculado e é realizado uma nova iteração do treinamento. O treinamento é concluído quando o erro for suficiente pequeno. Então, a rede passa a operar apenas no sentido *forward* para classificar novos exemplos.

3. ESTADO DA ARTE

Este Capítulo possui como objetivo contextualizar o estado atual das pesquisas e estudos referentes ao tema detecção de intrusão em dispositivos IoT. Foram realizados pequenos resumos, apresentando a ideia de alguns trabalhos relacionados ao tema, demonstrando o panorama atual da detecção de intrusão em IoT.

Os autores Bhushan e Sahoo (2019) propuseram um sistema de detecção de intrusão integrado, utilizando o conceito de cluster junto com a assinatura digital. Propuseram o método de auto ajuste de frequência, onde o algoritmo PCA (*Principal Component Analysis*) pode reduzir o número de variáveis e eliminar recursos com baixas discriminações. A dimensão reduzida dos dados foi dividida pelo algoritmo SFC (*Similarity-based Fuzzy Clustering*) como dados de alto e baixo risco, que são detectados usando frequências diferentes alcançando maior eficiência e precisão de detecção.

Os autores An et al. (2018) propuseram uma análise e modelo dos ataques DDoS (*Distributed Denial of Service)* sob uma estrutura da FC-IDS *(Fog Computing Intrusion Detection System*). Propuseram um cluster de hipergráficos baseado no algoritmo Apriori. Este modelo pode descrever a associação entre os nós da fog que sofrem com as ameaças de DDoS. Por fim, por meio de uma simulação, os autores verificaram que a taxa de utilização de recursos do sistema pode ser efetivamente promovida através dessa análise.

Liang et. al (2020), com o objetivo de abordar a questão da baixa taxa de precisão de detecção, alta taxa de falsos positivos e baixo desempenho que um sistema de detecção de intrusão pode ter, propuseram um algoritmo de detecção de intrusão de rede industrial no modelo de otimização de clustering de dados, onde “as distâncias ponderadas e coeficientes de segurança dos dados são classificados com base no limite de prioridade do recurso de atributo de dados para cada nó na rede”. Os resultados obtidos mostraram que o algoritmo proposto foi superior em termos de taxa de detecção e tempo em comparação com outros algoritmos. A precisão de detecção de dados anormais atingiu 97,8% e o falso positivo diminuiu para 8,8%.

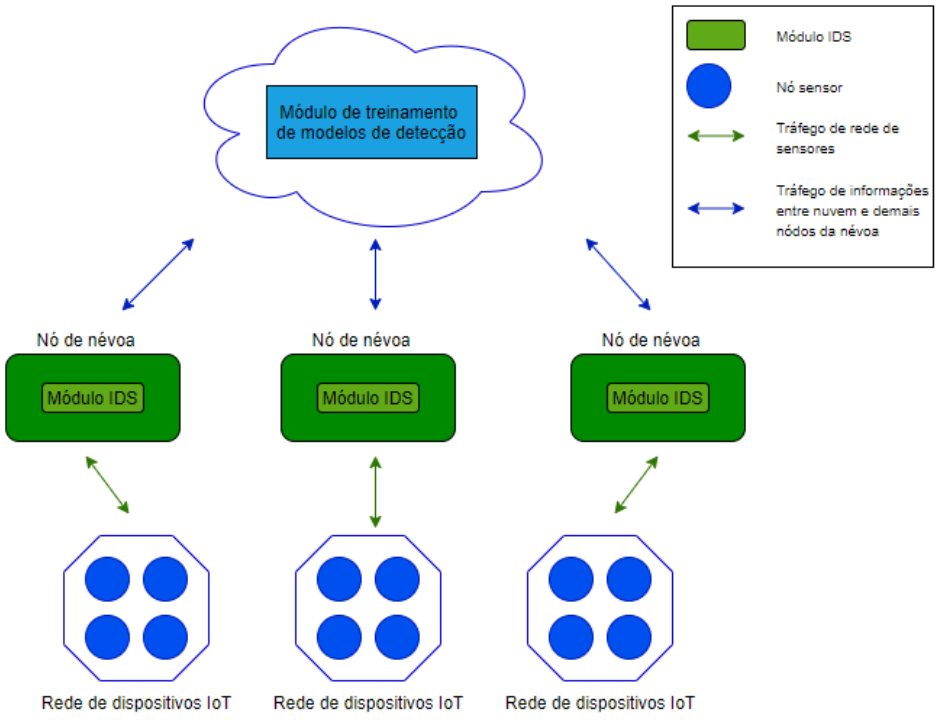
Os autores Shojafar et. al (2019) propuseram um algoritmo de cluster automático como parte de uma arquitetura de um IDS (*Intrusion Detection System*). Esse algoritmo foi baseado em conceitos de coerência e separação. O algoritmo encontra clusters com a maior semelhança entre os elementos de cluster propostos e a menor semelhança com outros clusters. Nesse caso o cluster proposto é otimizado ainda mais, considerando dois tipos de funções de índice objetivo e artificial: Colônia de Abelhas (ABC – *Artificial Bee Colony*), Otimização de enxame de partículas (PSO – *Particle Swarm Optimization*) e Métodos de evolução diferencial (DE – *Differential Evolution*). Os resultados obtidos mostraram melhorias em termos do baixo número médio de funções de avaliação, alta precisão e baixo custo de computação.

Após a análise de alguns trabalhos relacionados ao tema foi possível observar que a maioria dos trabalhos existentes focam em detecção binária, e, os que possuem detecção multiclasse possuem uma baixa eficácia na detecção. Existe uma certa importância para se ter uma detecção multiclasse eficaz, principalmente na execução das contramedidas necessárias. Outro ponto analisado foi que os sistemas de detecção de intrusão geralmente operam na nuvem, e não na Fog.

4. PROPOSTA E DESENVOLVIMENTO

A realização da análise de detecção de intrusões na névoa é interessante porque fornece uma visão geral da rede de dispositivos IoT. Os dispositivos IoT são restritos de recursos, desse modo é inviável a utilização de métodos e abordagens complexas de detecção nos próprios nós sensores, já que os mesmos não possuem capacidade de executar métodos robustos de aprendizado de máquina como redes neurais devido as suas restrições. Sendo assim, uma solução ideal, possuiria uma abordagem de detecção na névoa. A Figura 7 apresenta a ilustração do local de atuação do mecanismo proposto em uma arquitetura de um ambiente IoT seguro, baseado em computação em névoa. Nota-se que a abordagem de detecção proposta encontra-se na névoa, capturando o tráfego da rede de dispositivos IoT que está vinculado ao nó da Fog, analisando os dados da rede.

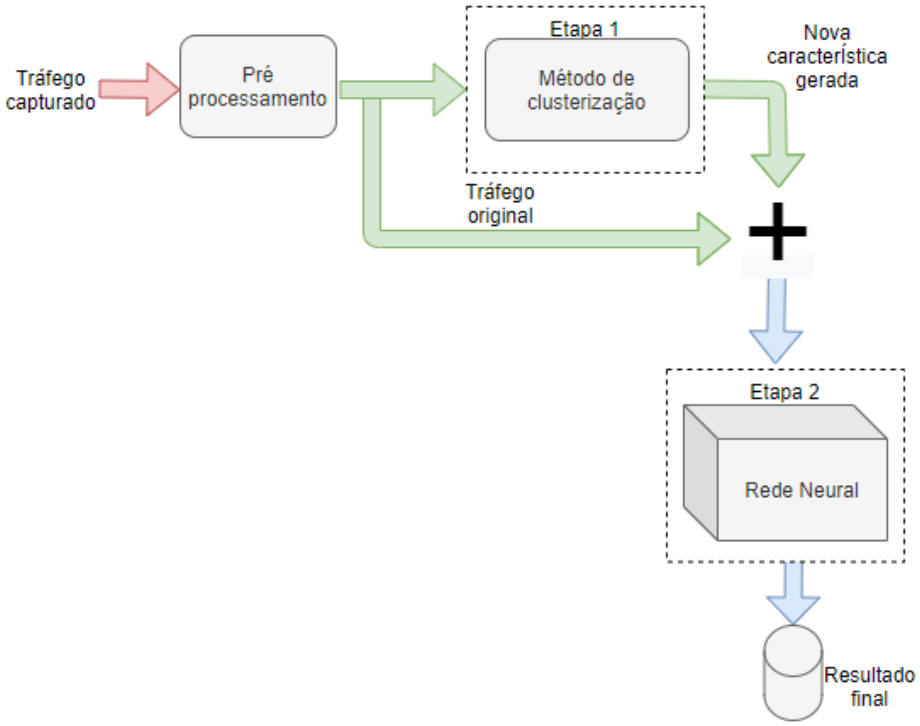
Figura 7 - Arquitetura do IDS proposto



Fonte: O autor.

A abordagem proposta possui duas etapas principais. A etapa 1 é baseada em clusterização. Essa etapa é responsável por gerar, a partir das informações capturadas da rede, uma nova informação baseada na clusterização das já existentes. Todas essas informações são então utilizadas na segunda etapa da abordagem, onde o modelo neural realiza a classificação. Na Figura 8 é apresentada uma ilustração da abordagem proposta.

Figura 8 - Abordagem proposta.



Fonte: O autor.

4.1. ETAPA 1 - CLUSTERIZAÇÃO

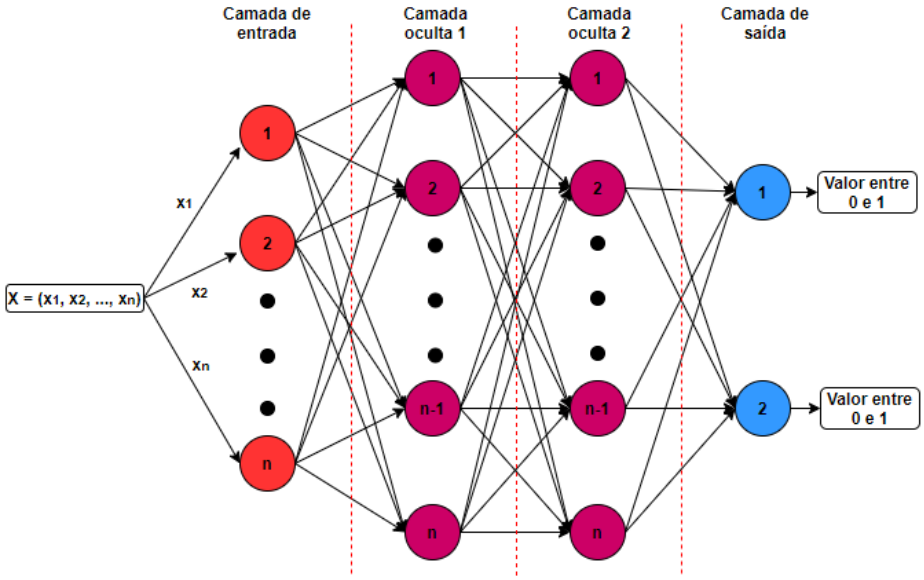
Conforme ilustrado na Figura 8 o tráfego capturado da rede de dispositivos IoT passa por uma fase de pré-processamento. Nessa etapa o tráfego capturado é convertido em atributos que podem ser submetidos a rede neural. No entanto, antes dos atributos de um determinado tráfego serem submetidos ao modelo neural, eles são enviados a um método de clusterização. Para esse trabalho será utilizado para avaliação os métodos de clusterização *Mini Batch K-means*, que é uma variação do método *K-means* e o método por densidade DBSCAN. Esses métodos geram um novo atributo que corresponde a identificação do cluster em que o tráfego foi

agrupado. Os métodos de clusterização já possuem clusters criados durante a fase de treinamento. Esses métodos geram dois clusters para a saída binária e cinco para a multiclasse, pois é o número de classes que a rede neural vai classificar. Desse modo, somente calcula a qual cluster o atual tráfego tem maior associação. Após a clusterização, o novo atributo é combinado com os demais originais do tráfego, e então submetido ao modelo neural.

4.2. ETAPA 2 - CLASSIFICAÇÃO

Na segunda etapa da abordagem proposta ocorre a classificação com o modelo neural. A saída da rede neural vai depender da sua arquitetura de saída (binária ou multiclasse), dois ou cinco neurônios. Esses neurônios utilizam a função *softmax*, gerando um valor entre 0 e 1, resultando na probabilidade de a entrada estar em uma determinada classe (NWANKPA et al., 2018). Essas redes neurais possuem uma camada de entrada, duas camadas ocultas e uma camada de saída. Como função de ativação dos neurônios da camada oculta foi utilizada a ReLu. Essa função é responsável por transformar a entrada de um nó para a saída desse mesmo nó. Ela é uma função de ativação padrão para muitos tipos de redes neurais por atingir um melhor desempenho e ter uma maior facilidade de treinamento. A Figura 9 apresenta uma ilustração da arquitetura binária. Nessa arquitetura a camada de saída apresenta 2 neurônios, onde um corresponde a categoria normal e o outro a categoria ataque.

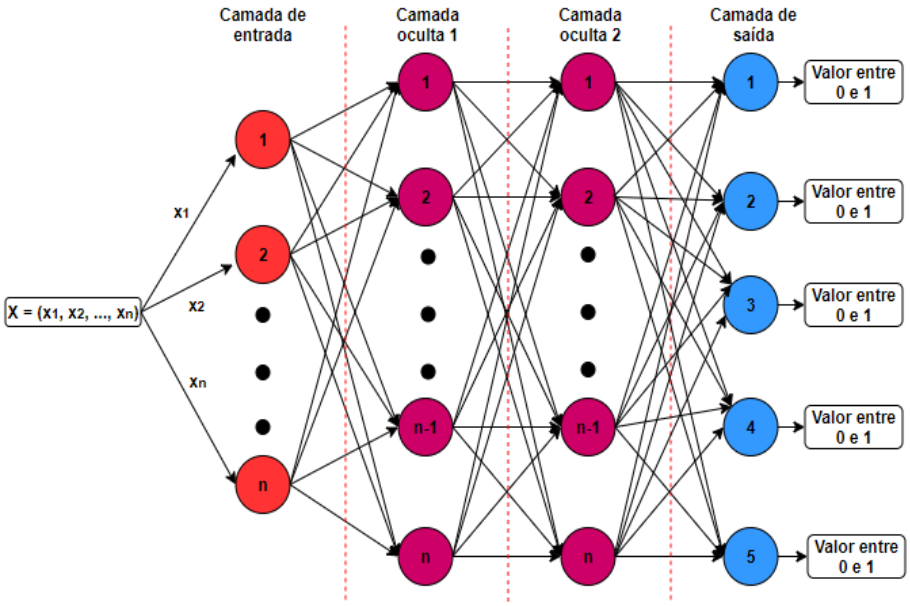
Figura 9 - Arquitetura RNA com saída binária.



Fonte: O autor.

Na Figura 10 é apresentada uma ilustração da arquitetura com saída multiclasse. Nessa arquitetura, a camada de saída apresenta 5 neurônios: quatro representam as categorias de ataques (DoS, Probe, R2L e U2L) e um a categoria normal.

Figura 10 - Arquitetura RNA com saída multiclasse.



Fonte: O autor.

5. AVALIAÇÃO

Neste capítulo é apresentada a metodologia definida para a avaliação da proposta. Além disso, são apresentados os resultados obtidos com os experimentos e também uma discussão sobre os mesmos.

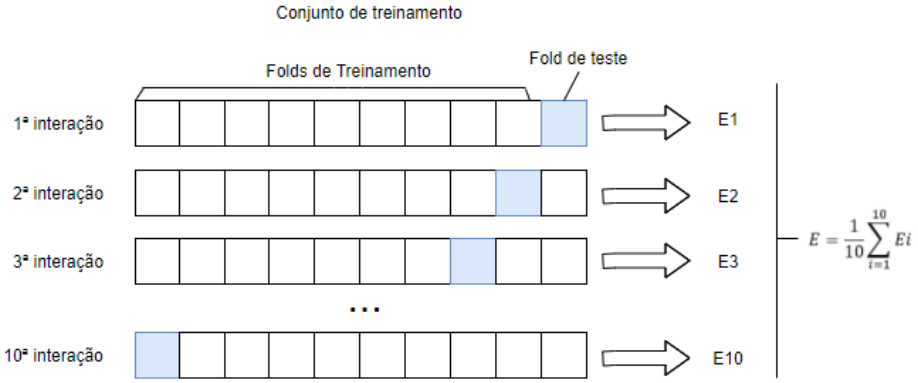
5.1. EXPERIMENTOS

Nesta seção é apresentada a metodologia experimental aplicada na avaliação da abordagem proposta. A técnica de cross-validation com 10 folds foi utilizada para estimar o quão preciso o modelo é na prática.

5.1.1. Cross-validation

Na maioria dos aplicativos reais, apenas uma quantidade limitada de dados está disponível. Isso dificulta a avaliação dos modelos pois para se ter uma avaliação justa é necessário que os dados de teste sejam inéditos para o modelo, ou seja, não sejam os mesmos dados usados no treinamento. O *cross-validation* é uma estratégia interessante para avaliar o quão preciso é o modelo na prática e para contornar a limitação de dados (ARLOT; CELISSE, 2010). A principal ideia desse algoritmo é calcular a taxa de erro em um certo subconjunto de dados de teste que foi separado previamente, ou seja, que não participa do aprendizado do algoritmo, sendo possível obter uma estimativa da precisão das abordagens. No *cross-validation* os dados são divididos em vários *folds* (BROWNE, 2000). Em um *cross-validation* de 10 *folds*, são realizadas 10 iterações, onde em cada iteração 9 *folds* são utilizados para treinar o modelo e o *fold* restante é utilizado para testar. Em cada iteração um *fold* diferente é deixado para o teste. Após a execução do *cross-validation* de 10 *folds* se tem 10 modelos treinados. Além disso, o resultado final gerado é a média dos resultados dos 10 *folds*. A Figura 11 demonstra o funcionamento do processo do *cross-validation*.

Figura 11 - Processo do *cross-validation* 10 *folds*.



Fonte: O autor.

5.1.2. Métricas de avaliação

A partir dos resultados obtidos é possível categorizar as classificações dos eventos da base de dados da seguinte forma:

- Falso negativo (*False negative* - FN): eventos classificados como normais pelo método, mas são intrusões;
- Falso positivo (*False positive* - FP): eventos não intrusivos e classificados como intrusivos pela técnica de detecção de intrusão;
- Verdadeiro negativo (*True negative* - TN): eventos não intrusivos que foram classificados corretamente pelo método;
- Verdadeiro positivo (*True positive* - TP): eventos intrusivos que foram classificados corretamente pelo método.

Através desses termos é possível realizar o cálculo de diferentes métricas, auxiliando na avaliação dos experimentos. As métricas utilizadas para avaliar os experimentos deste trabalho foram as seguintes (ALMIANI; ABUGHAZLEH; AL-RAHAYFEH; RAZAQUE, 2019):

- Acurácia (ACC): corresponde a proporção de instâncias classificadas corretamente em relação ao total de instâncias. Pode ser calculada através da seguinte forma:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Erro (ERR): corresponde a proporção de instâncias classificadas incorretamente em relação ao total de instâncias. Pode ser calculado através da seguinte forma:

$$ERR = \frac{FP+FN}{TP+TN+FP+FN} \quad (2)$$

- Precisão (PRE): corresponde às instâncias intrusivas corretamente classificadas. Pode ser calculada da seguinte forma:

$$PRE = \frac{TP}{TP+FP} \quad (3)$$

- *Recall* ou *True Positive Rate* (TPR): corresponde ao número de instâncias classificadas como intrusivas, dentre todas que realmente são intrusivas. Pode ser calculada da seguinte forma:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

- *True Negative Rate* (TNR): corresponde ao número de instâncias classificadas como não intrusivas, entre todas que realmente são não intrusivas. Pode ser calculada da seguinte forma:

$$TNR = \frac{TN}{TN+FP} \quad (5)$$

- F1-Score: corresponde a precisão de um teste. É a média harmônica de precisão e recuperação, onde seu melhor valor é 1 e o pior 0. Pode ser calculada da seguinte forma:

$$F1 - Score = 2 * \frac{PRE*Recall}{PRE+Recall} \quad (6)$$

- *Matthews Correlation Coefficient* (MCC): corresponde a uma medida da qualidade das classificações binárias no aprendizado de máquina. Possui um intervalo de -1 a 1, onde -1 indica um classificador binário completamente errado, enquanto 1 indica um classificador binário completamente correto. Pode ser calculado da seguinte forma:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(FN+TN)*(FP+TN)*(TP+FN)}} \quad (7)$$

5.1.3. Base de dados

Para a avaliação são utilizadas base de dados públicas de ataques. Para a utilização desse trabalho foi utilizada a base de dados NSL-KDD. Essa base de dados cobre quatro categorias principais de ataques:

- Ataque de negação de serviço (*Denial of Service* - DoS): bloqueia solicitações legítimas para um recurso da rede, consumindo banda ou sobrecarregando recursos computacionais;

- Ataque por sondagem (*Probe Attack*): revela as informações da rede, podendo ser explorada por outro tipo de ataque. Coleta informações antes de iniciar um ataque, quebrando seus controles de segurança;

- Ataque de usuários remotos (*Remote to Local* - R2L): nesse tipo de ataque um invasor que não possui uma conta em uma máquina remota envia um pacote para a máquina através da rede, explorando vulnerabilidades para obter acesso local;

- Ataque de usuário Root (*User to Root* - U2R): nesse tipo de ataque um invasor com um usuário normal no sistema explora vulnerabilidades a fim de obter acesso root.

Essa base de dados possui 22 tipos de ataque no conjunto de treinamento, com 17 adicionais no conjunto de teste e 125973 registros. Essa base não inclui registros redundantes no conjunto de treino, bem como não há registros duplicados no conjunto de teste (para não ser influenciado pelos métodos que possuem melhores taxas de detecção de registros frequentes). Cada instância da base possui contém 41 recursos (números, nominais e binários), conforme a Tabela 1, que são rotulados como normal ou como um ataque com um tipo específico:

Tabela 1 - Classificação de características da base de dados.

Tipo	**Características**
Nominal	*Protocol_type(2), Service(3), Flag(4)*
Binário	*Land(7), logged_in(12), root_shell(14), su_attempted(15), is_host_login(21), is_guest_login(22)*
Numérico	*Duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)*

Fonte: O autor

Na Tabela 2 é apresentada a descrição de cada um dos 41 recursos que foram apresentados na Tabela 1:

Tabela 2 - Descrição das características da base de dados.

Número	Nome	Tipo	Descrição
1	*Duration*	C	Comprimento (número de segundos) da conexão
2	*protocol_type*	D	Tipo de protocolo (TCP, UDP, etc)
3	*Service*	D	Serviço de rede (HTTP, telnet, etc.)
4	*Flag*	D	Status da conexão (normal ou erro)
5	*src_bytes*	C	Número de bytes de dados da fonte para o destino
6	*dst_bytes*	C	Número de bytes de dados do destino para a fonte
7	*Land*	D	1 se a conexão for de/para o mesmo host/porta; 0 outra forma
8	*wrong_fragment*	C	Número de fragmentos errados
9	*Urgent*	C	Número de pacotes urgentes
10	*Hot*	C	Número de indicadores quentes
11	*num_failed_logins*	C	Número de tentativas de logins falhos
12	*Logged_in*	D	1 se logado com sucesso; 0 se não
13	*num_compromised*	C	Número de condições comprometidas
14	*root_shell*	C	1 se o shell root for obtido, 0 se não

15	*su_attempted*	C	1 se o comando su root for tentado; 0 se não
16	*num_root*	C	Número de acessos root
17	*num_file_creations*	C	Número de criações de arquivos de operações
18	*num_shells*	C	Número de prompts de shell
19	*num_access_files*	C	Número de operações em arquivos de controle de acesso
20	*num_outbound_cmds*	C	Número de comandos de saída em uma sessão FTP
21	*is_host_login*	D	1 se o login pertencer a lista quente; 0 caso não
22	*is_guest_login*	D	1 se o login for um convidado; 0 caso não
23	*Count*	C	Número de conexões para o mesmo host com a conexão atual nos últimos dois segundos
24	*srv_count*	C	Número de conexões para o mesmo serviço com a conexão atual nos últimos dois segundos
25	*serror_rate*	C	Número de conexões para o mesmo host com a conexão atual nos últimos dois segundos
26	*srv_serror_rate*	C	% de conexões com erros SYN (verificar)
27	*rerror_rate*	C	% de conexões que tem erros REJ (verificar)
28	*srv_rerror_rate*	C	% de conexões que tem erros REJ (verificar)

29	*same_srv_rate*	C	% de conexões do mesmo serviço
30	*diff_srv_rate*	C	% de conexões de serviços diferentes
31	*srv_diff_host_rate*	C	% de conexões de diferentes hosts
32	*dst_host_count*	C	Número de conexões do mesmo host para o host de destino com a conexão atual nos últimos dois segundos
33	*dst_host_srv_count*	C	Número de conexões do mesmo serviço para o host de destino com a conexão atual nos últimos dois segundos
34	*dst_host_same_srv_rate*	C	% de conexões do mesmo serviço para o host de destino
35	*dst_host_diff_srv_rate*	C	% de conexões de diferentes serviços para o host de destino.
36	*dst_host_same_src_port_rate*	C	% de conexões das mesmas portas de serviço para o host de destino
37	*dst_host_srv_diff_host_rate*	C	% de conexões de diferentes hosts para o mesmo serviço no host de destino
38	*dst_host_serror_rate*	C	% de conexões que possuem SYN erros do mesmo host para o host de destino
39	*dst_host_srv_serror_rate*	C	% de conexões que possuem SYN erros do mesmo serviço para o host de destino

40	*dst_host_rerror_rate*	C	% de conexões que possuem REJ erros do mesmo host para o host destino
41	*dst_host_srv_rerror_rate*	C	% de conexões que possuem REJ erros do mesmo serviço para o host destino.

C – Contínuo D – Discreto

Fonte: O autor.

Na Tabela 3 é apresentada a distribuição da instância da base de dados NSL-KDD.

Tabela 3 - Distribuição da instância no conjunto de dados de treinamento.

Tipo de ataque	**Número de registros**
Normal	67343
DoS	45927
Probe	11656
R2L	995
U2R	52
Total	125973

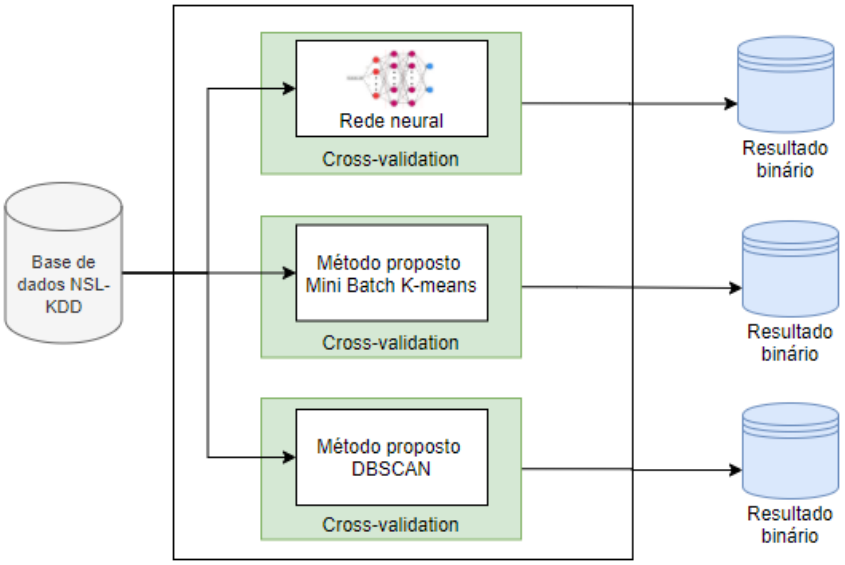
Fonte: O autor.

5.1.4. Descrição dos experimentos

Foram utilizados apenas os dados referentes aos conjuntos de treino com o auxílio do método *cross-validation* com 10 folds, onde as 125973 amostras foram divididas em 10 subconjuntos, e cada um desses subconjuntos foi utilizado uma vez para teste enquanto os demais foram utilizados na etapa de treinamento.

Para a realização do experimento binário, houve um pré-processamento dos dados, transformando todos os dados considerados como ataques em 1 e os normais em 0. Na Figura 12 é apresentada uma ilustração do experimento com saída binária.

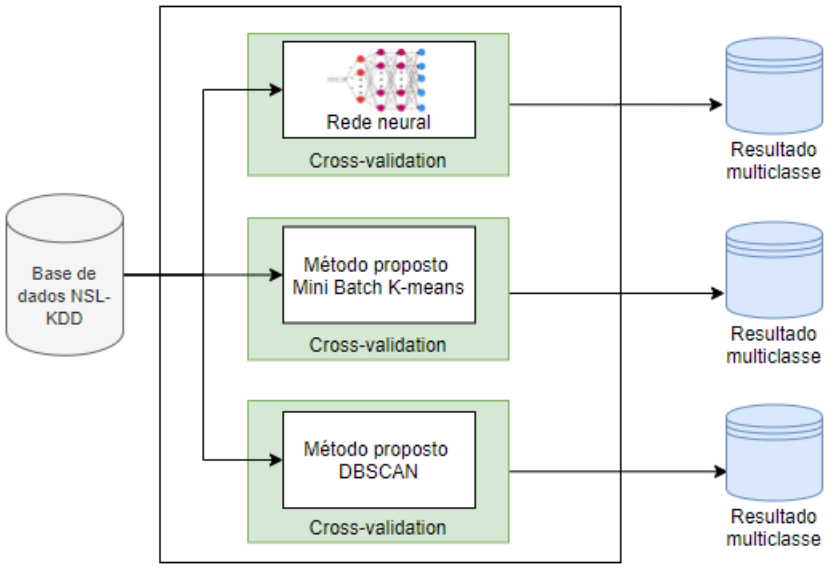
Figura 12 - Experimentos utilizando a base de dados KDD com saída binária.



Fonte: O autor.

Na Figura 13 é apresentada uma ilustração do experimento com saída multiclasse. A mesma base de dados é utilizada na RNA pura, além dos dois métodos de clusterização, cada um resultando na sua respectiva saída.

Figura 13 - Experimentos utilizando a base de dados KDD com resultados multiclasse



Fonte: O autor.

Para a realização desses experimentos foi utilizada a ferramenta *Google Colaboratory* do Google, que fornece ambiente para a implementação dos experimentos e também execução. As Figuras 14 e 15 mostram informações da máquina que foi utilizada para os experimentos.

Figura 14 - Informações de processamento da máquina utilizada nos experimentos.

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 63
model name      : Intel(R) Xeon(R) CPU @ 2.30GHz
stepping        : 0
microcode       : 0x1
cpu MHz         : 2300.000
cache size      : 46080 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf mds swapgs
bogomips        : 4600.00
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:
```

Fonte: O autor.

Figura 15 - Informações de memória da máquina utilizada nos experimentos.

```
MemTotal:       13333596 kB
MemFree:        10878048 kB
MemAvailable:   12545192 kB
Buffers:           72916 kB
Cached:          1750992 kB
SwapCached:            0 kB
Active:           634708 kB
Inactive:        1561536 kB
Active(anon):     348484 kB
Inactive(anon):      324 kB
Active(file):     286224 kB
Inactive(file):  1561212 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:             0 kB
SwapFree:              0 kB
Dirty:               676 kB
Writeback:             0 kB
AnonPages:        372704 kB
Mapped:           184804 kB
Shmem:               928 kB
Slab:             157244 kB
SReclaimable:     121100 kB
SUnreclaim:        36144 kB
KernelStack:        3700 kB
PageTables:         4940 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     6666796 kB
Committed_AS:    2461260 kB
VmallocTotal:   34359738367 k
VmallocUsed:           0 kB
VmallocChunk:          0 kB
Percpu:              928 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:       83176 kB
DirectMap2M:     6207488 kB
DirectMap1G:     9437184 kB
```

Fonte: O autor.

Para a clusterização foi utilizada a biblioteca a *sklearn*. Essa biblioteca integra algoritmos clássicos de aprendizagem de máquina. Para a RNA foi utilizada a biblioteca *Keras*, que é um *front-end* para a biblioteca *TensorFlow*. A *Tensorflow* implementa a parte de RNA em baixo nível e a *Keras*, utilizando o *TensorFlow*, fornece uma implementação em alto nível. Para o carregamento e utilização da base de dados foi utilizada a biblioteca *Pandas*. Essa biblioteca foi criada para a linguagem *Python* e serve para manipulação e análise de dados, permitindo manipular tabelas numéricas e séries temporais.

5.2. RESULTADOS

Nesta seção são apresentados os resultados obtidos através de experimentos com o método proposto. Foram realizados experimentos utilizando a RNA pura, RNA juntamente com o método de clusterização *Mini Batch K-means* e RNA com o método de clusterização DBSCAN, a fim de obter diferentes resultados para verificar se os resultados utilizando clusterização são mais eficazes do que sem.

Inicialmente a abordagem proposta é avaliada em relação a detecção binária. O objetivo é verificar a influência dos métodos de clusterização para a tarefa de classificação binária realizada pela rede neural. Os resultados obtidos pela abordagem com RNA, pela abordagem proposta com DBSCAN (RNA + DBSCAN) e pela abordagem proposta com *Mini Batch K-Means* (RNA + *Mini Batch K-means*), são apresentados na Tabela 4.

Tabela 4 - Resultados dos experimentos na classificação binária.

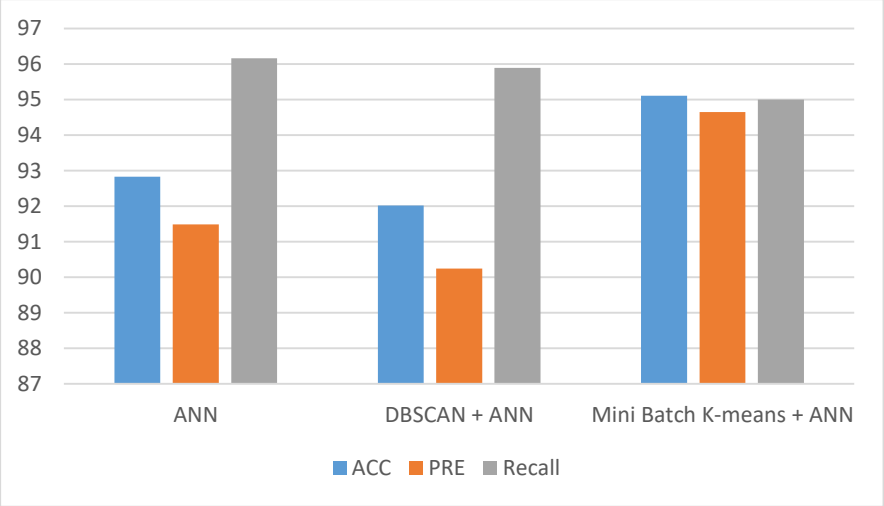
	ACC	**ERR**	**PRE**	**Recall**	**TNR**	**F1-SCORE**	**MCC**
RNA	92,83	7,17	91,49	96,16	90,01	93,21	0,87
RNA + DBSCAN	92,02	7,98	90,24	95,89	88,65	92,49	0,85
RNA + MINI BATCH K-MEANS	95,11	4,89	94,65	95,00	95,23	94,75	0,90

Fonte: O autor.

Como pode ser observado, a RNA pura obteve uma acurácia de 92,83%. Utilizando o método de clusterização por densidade DBSCAN, nesse caso, a acurácia foi um pouco inferior (92,02%). Já utilizando o método *Mini Batch K-Means*,

houve uma melhora na acurácia (95,11%) em relação às duas análises anteriores. Além disso, o método *Mini Batch K-means* foi o que teve uma melhor precisão na classificação dos dados, comprovando que foi o método com menos falsos positivos, e, um maior MCC. O MCC é a confirmação que esse método possui uma melhor qualidade na detecção binária. Nessa métrica, valores mais próximos a 1 significam uma detecção mais correta. O gráfico a seguir é uma ilustração da acurácia, recall e precisão dos resultados dos experimentos. Podemos verificar através do gráfico que o *Mini Batch K-means* foi o método que obteve um maior balanceamento entre acurácia, recall e precisão.

Gráfico 1 - Comparação dos resultados binários entre os métodos propostos considerando a acurácia, precisão e recall.



Fonte: O autor.

Na segunda etapa foram realizados experimentos para verificar o desempenho da abordagem proposta em relação a detecção multiclasse. Nessa etapa, os resultados obtidos são classificados como normal ou nas quatro categorias de ataques da base. Os resultados obtidos pela abordagem com RNA sem nenhum método de clusterização são demonstrados na Tabela 5.

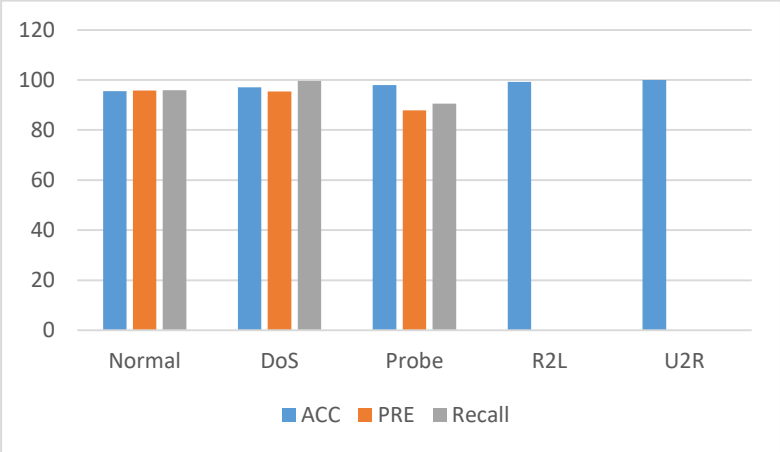
Tabela 5 - Resultados multiclasse de experimentos utilizando RNA

CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	95,57	4,43	95,84	95,86	95,25	95,85
DoS	97,06	2,94	95,35	99,67	97,29	96,00
Probe	97,94	2,06	87,84	90,58	98,70	89,18
R2L	99,24	0,76	0	0	100	0
U2R	99,95	0,05	0	0	100	0

Fonte: O autor.

Analisando os dados é possível observar uma acurácia consideravelmente alta em todas as classificações. Na categoria R2L e U2R, apesar de RNA apresentar uma acurácia de quase 100%, obteve um recall de 0 significando que não houve ataques detectados. Além disso, a precisão baixa também caracteriza a classificação de muitos falsos positivos. O gráfico a seguir é uma ilustração da acurácia, recall e precisão dos resultados dos experimentos. É possível verificar através do gráfico que as categorias R2L e U2R possuem uma maior acurácia, mesmo com precisão e recall 0. Isso se dá pelo fato de haver poucas instâncias dessas duas categorias, que acabam não interferindo na acurácia. Nesse caso houve uma melhor classificação nos outros três tipos de categorias, com destaque para DoS e Normal, que tiveram um melhor recall e melhor precisão.

Gráfico 2 – Comparação dos resultados multiclasse utilizando somente a RNA considerando a acurácia, precisão e recall.



Fonte: O autor.

Na Tabela 6 são demonstrados os resultados obtidos pela abordagem utilizando o método de clusterização DBSCAN juntamente com a RNA. Nessa tabela os resultados também são classificados como normal ou nas quatro categorias de ataques da base.

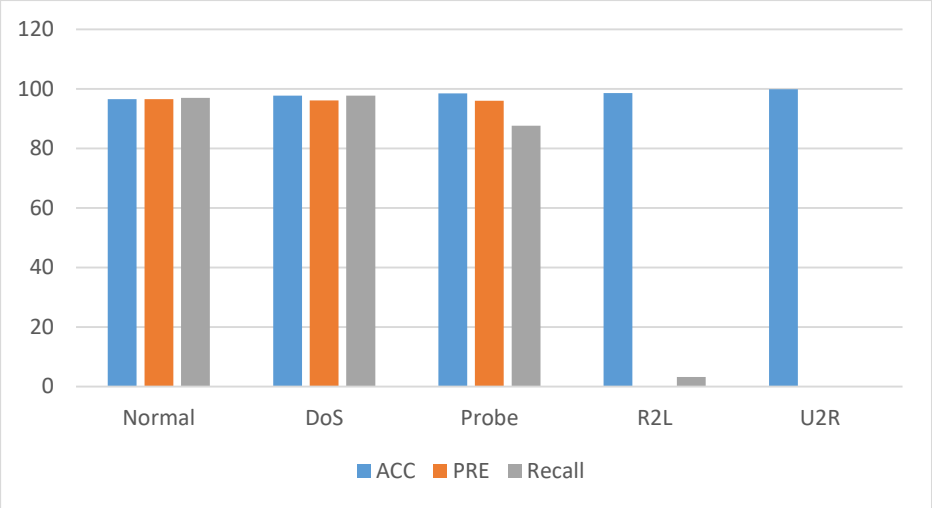
Tabela 6 - Resultados multiclasse de experimentos utilizando RNA juntamente com DBSCAN

CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	96,55	3,45	96,57	97,01	96,01	96,79
DoS	97,78	2,22	96,18	97,79	97,78	96,97
Probe	98,50	1,50	96,02	87,64	99,62	91,39
R2L	98,56	1,44	0	3,19	99,32	0
U2R	99,90	0,10	0	0	99,92	0

Fonte: O autor.

Através dos resultados obtidos pode-se observar também uma acurácia alta em todas as classificações. Em relação aos dados obtidos no experimento sem nenhum método de clusterização, aqui verifica-se uma melhora, mesmo que muito pouca, no recall da categoria R2L. Além disso, uma leve melhora na precisão como um todo, com destaque para a classificação da categoria Probe que alcançou uma melhora de mais de 8% em relação aos resultados anteriores. Através do gráfico a seguir, pode-se observar um maior balanceamento entre as categorias Normal, DoS e Probe.

Gráfico 3 – Comparação dos resultados multiclasse utilizando RNA juntamente com DBSCAN considerando a acurácia, precisão e recall.



Fonte: O autor.

Por fim, na Tabela 7 são demonstrados os resultados obtidos pela abordagem utilizando o método de clusterização *Mini Batch K-means* juntamente com a RNA. Neste experimento a base de dados também foi classificada de acordo com as cinco categorias existentes.

Tabela 7 - Resultados multiclasse de experimentos utilizando RNA juntamente com Mini Batch K-means

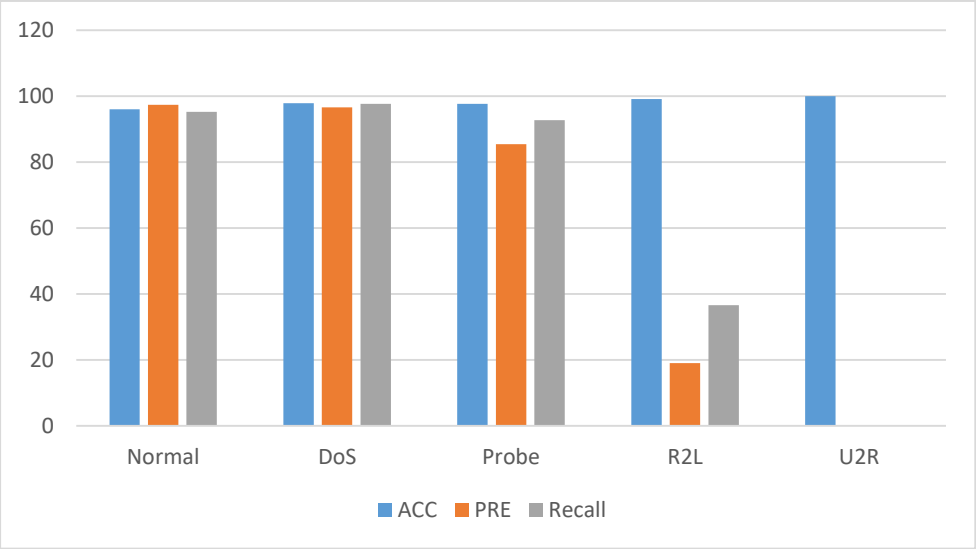
CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	96,02	3,98	97,35	95,20	96,96	96,25
DoS	97,89	2,11	96,64	97,67	98,02	97,14
Probe	97,92	2,08	85,44	92,69	98,43	88,92
R2L	99,15	0,85	18,98	36,63	99,56	0
U2R	99,96	0,04	0	0	99,99	0

Fonte: O autor.

Através dos resultados obtidos pode-se observar também que esta abordagem alcançou uma acurácia alta em todas as classificações. Em relação aos dados obtidos no experimento sem nenhum método de clusterização e com DBSCAN, aqui verifica-se pela primeira vez uma melhora na precisão. Além disso, esta abordagem obteve uma melhora de 36,63% no recall em relação a RNA pura

e de 33,44% em relação ao DBSCAN da categoria R2L. Através do gráfico a seguir, pode-se observar um maior balanceamento entre as categorias Normal e DoS e Probe, com destaque para leve melhora da categoria R2L.

Gráfico 4 – Comparação dos resultados multiclasse utilizando RNA juntamente com Mini Batch K-means considerando a acurácia, precisão e recall.



Fonte: O autor.

5.3. DISCUSSÃO

Em relação aos resultados obtidos, comprova-se que a abordagem utilizando *Mini Batch K-means* foi mais eficaz em relação as outras duas abordagens. Na classificação binária, houve um aumento da acurácia e precisão, resultando na diminuição de falsos positivos. Observando o MCC, que corresponde a uma medida da qualidade das classificações binárias, o Mini Batch *K-means* obteve uma melhor classificação, pois seu valor, dentre as três abordagens, foi o que mais se aproximou de 1.

Em relação a análise multiclasse, a abordagem utilizando *Mini Batch K-means* foi a que apresentou maior melhora na detecção. Na categoria R2L houve um aumento da acurácia e recall, indicando uma maior classificação de verdadeiros positivos, provando que o *Mini Batch K-means* contribui com a melhora na detecção.

Nenhuma das técnicas, nem a RNA pura, conseguiu detectar U2R. Nesse tipo de ataque, qualquer usuário normal do sistema obtém acesso ilegal aos

privilégios de superusuário, por isso, há uma importância em maiores estudos para melhorar a taxa de detecção desse tipo de ataque.

A abordagem utilizando clusterização apresentou melhoras tanto na classificação binária como na classificação multiclasse, portanto pode ser considerada interessante para a detecção de intrusão. Todavia, existe a necessidade de maiores estudos, avaliando outras estruturas de redes neurais. Acredita-se que estruturas de redes neurais mais robustas (com mais camadas e mais neurônios por camadas) possam ser mais úteis para a detecção principalmente das categorias R2L e U2R.

6. CONCLUSÃO

Com a expansão da internet das coisas (*Internet of Things* – IoT) e a informatização de dados, surgiram dificuldades para manter a segurança dos mesmos. Privacidade, controle de acesso, armazenamento e gerenciamento de informações são exemplos de desafios de um ambiente IoT. Como solução, os sistemas de detecção de intrusão (*Intrusion Detection System* – IDS) auxiliam a prevenir o acesso não autorizado à rede, analisando o tráfego de rede e classificando os registros como normais ou anômalos.

Nos experimentos realizados percebeu-se que a utilização de métodos de clusterização juntamente com RNA auxiliam na detecção de intrusão. Dos métodos de clusterização utilizados, o *Mini Batch K-means* foi o que apresentou um melhor resultado, com uma melhor taxa de acurácia e precisão. Além disso, esse método foi o que conseguiu um melhor recall na categoria R2L e o único que conseguiu uma melhora na precisão, resultando numa melhor detecção de verdadeiros positivos e diminuição de falsos positivos.

A partir dos resultados obtidos, observa-se que a abordagem híbrida com clusterização e RNA se mostra promissora. Como trabalhos futuros, são necessários estudos considerando a utilização de outros tipos de métodos de clusterização juntamente com redes neurais mais robustas (com maior quantidade de camadas ocultas e mais neurônios por camadas) para obter uma melhora ainda mais significativa na detecção.

7. REFERÊNCIAS

A demo of the K Means clustering algorithm — scikit-learn 0.11-git documentation. Disponível em <https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/cluster/plot_mini_batch_kmeans.html>. Acesso em: 30 de outubro de 2020.

AAZAM, Mohammad et al. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. **Proceedings Of 2014 11th International Bhurban Conference On Applied Sciences & Technology (ibcast) Islamabad, Pakistan, 14th - 18th January, 2014**, [s.l.], jan. 2014. IEEE. http://dx.doi.org/10.1109/ibcast.2014.6778179.

ALABA, Fadele Ayotunde et al. Internet of Things security: A survey. **Journal Of Network And Computer Applications**, [s.l.], v. 88, jun. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2017.04.002.

ALMIANI, Muder; ABUGHAZLEH, Alia; AL-RAHAYFEH, Amer; RAZAQUE, Abdul. Cascaded hybrid intrusion detection model based on SOM and RBF neural networks. **Concurrency And Computation**: Practice and Experience, [S.L.], v. 32, n. 21, p. 0-1, 7 mar. 2019. Wiley. http://dx.doi.org/10.1002/cpe.5233.

AN, Xingshuo; SU, Jingtao; LÜ, Xing; LIN, Fuhong. Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system. **Eurasip Journal On Wireless Communications And Networking**, [s.l.], v. 2018, n. 1, 22 out. 2018. Springer Science and Business Media LLC.

ANTONAKAKIS, Manos; APRIL, Tim; BAILEY, Michael; BERNHARD, Matthew; BURSZTEIN, Elie; COCHRAN, Jaime; DURUMERIC, Zakir; INVERNIZZI, Luca; HALDERMAN, J. Alex; KALLITSIS, Michalis; KUMAR, Deepak; LEVER, Chaz; MA, Zane; MASON, Joshua; MENSCHER, Damian; SEAMAN, Chad; SULLIVAN, Nick; THOMAS, Kurt; ZHOU, Yi. Understanding the Mirai Botnet. **Usenix The Advanced Computing Systems Association**, Vancouver, 16 ago. 2017.

ARLOT, Sylvain; CELISSE, Alain. A survey of cross-validation procedures for model selection. **Statistics Surveys**, [S.L.], 2010. Institute of Mathematical Statistics. http://dx.doi.org/10.1214/09-ss054.

Asaka, M., Okazawa, S., Taguchi, A. & Goto, S. (1999). A method of tracing intruders by use of mobile agents, INET'99 Proceedings.
Bace, R. & Mell, P. (2001). Nist special publication on intrusion detection systems, Technical report, BOOZ-ALLEN AND HAMILTON INC MCLEAN VA.

BARCELLOS, Raissa; VITERBO, José; BERNARDINI, Flavia C. Uso de algoritmos de clusterização para a identificação de padrões de consumo de energia elétrica. Universidade Federal Fluminense. 2016

BELLO, Oladayo; ZEADALLY, Sherali; BADRA, Mohamad. Network layer inter-operation of Device-to-Device communication technologies in Internet of Things

(IoT). **Ad Hoc Networks**, [s.l.], v. 57, p.52-62, mar. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.adhoc.2016.06.010.

BHUSHAN, Bharat; SAHOO, G.. A Hybrid Secure and Energy Efficient Cluster Based Intrusion Detection system for Wireless Sensing Environment. **2019 2nd International Conference On Signal Processing And Communication (icspc)**, [s.l.], mar. 2019. IEEE.

BONOMI, Flavio et al. Fog computing and its role in the internet of things. **Proceedings Of The First Edition Of The Mcc Workshop On Mobile Cloud Computing - Mcc '12**, [s.l.], 2012. ACM Press. http://dx.doi.org/10.1145/2342509.2342513.

BRONZATTI, Luiz Fernando Casarin. Análise sobre a tecnologia de rede sem fio Zigbee / IEEE 802.15.4. **Universidade de São Paulo**, jun. 2013. http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-08112013-101735/.

BROWNE, Michael W. Cross-Validation Methods. **Journal Of Mathematical Psychology**, [S.L.], mar. 2000. Elsevier BV. http://dx.doi.org/10.1006/jmps.1999.1279.

BUYYA, Rajkumar et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Generation Computer Systems**, [s.l.], v. 25, n. 6, jun. 2009. Elsevier BV. http://dx.doi.org/10.1016/j.future.2008.12.001.

Campello, R. S. & Weber, R. F. (2001). Sistemas de detecçao de intrusao, Minicurso procedente do 19o Simpósio Brasileiro de Redes de Computadores.

CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection. **Acm Computing Surveys**, [s.l.], v. 41, n. 3, 1 jul. 2009. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/1541880.1541882.

CHIANG, Mung; ZHANG, Tao. Fog and IoT: An Overview of Research Opportunities. **Ieee Internet Of Things Journal**, [s.l.], v. 3, n. 6, dez. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/jiot.2016.2584538.

DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning - Easy Guides - Wiki – STHDA. Disponível em <http://www.sthda.com/english/wiki/wiki.php?id_contents=7940>. Acesso em: 30 de outubro de 2020.

DEPREN, Ozgur et al. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. **Expert Systems With Applications**, [s.l.], v. 29, n. 4, p.713-722, nov. 2005. Elsevier BV. http://dx.doi.org/10.1016/j.eswa.2005.05.002.

FERREIRA, Vinicius Oliveira. Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando

métodos de agrupamento. **Programa de Pós-Graduação em Ciência da Computação – UNESP**, abr. 2016. http://hdl.handle.net/11449/138755.

GARCÍA-TEODORO, P. et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. **Computers & Security**, [s.l.], v. 28, n. 1-2, fev. 2009. Elsevier BV. http://dx.doi.org/10.1016/j.cose.2008.08.003.

Ilgun, K. (1993). Ustat: A real-time intrusion detection system for unix, Research in Security and Privacy, 1993. Proceedings., 1993 IEEE Computer Society Symposium on, IEEE.

k-means clustering | Wikiwand. Disponível em <https://www.wikiwand.com/en/K-means_clustering>. Acesso em: 30 de outubro de 2020.

LIAO, Hung-jen et al. Intrusion detection system: A comprehensive review. **Journal Of Network And Computer Applications**, [s.l.], v. 36, n. 1, jan. 2013. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2012.09.004.

LIN, Ying; ZHANG, Yan; OU, Yang-jia. The Design and Implementation of Host-Based Intrusion Detection System. **2010 Third International Symposium On Intelligent Information Technology And Security Informatics**, [s.l.], abr. 2010. IEEE. http://dx.doi.org/10.1109/iitsi.2010.127.

MAPLE, Carsten. Security and privacy in the internet of things. **Journal Of Cyber Policy**, [s.l.], v. 2, n. 2, 4 maio 2017. Informa UK Limited. http://dx.doi.org/10.1080/23738871.2017.1366536.

MELL, P M; GRANCE, T. The NIST definition of cloud computing. **NIST**, [s.l.], 2011. National Institute of Standards and Technology. http://dx.doi.org/10.6028/nist.sp.800-145.

MENEZES, Rodrigo Cezar. CLUSTERIZAÇÃO DE DADOS UTILIZANDO O ALGORITMO K-MEANS. [s.l.], 2013. Faculdades Catolicas.

MUSSOI DE LIMA, Igor Vinícius. Uma abordagem simplificada de detecção de intrusão baseada em redes neurais artificiais. **Programa de Pós-Graduação em Ciência da Computação – UFSC**, 2005. Disponível em: http://repositorio.ufsc.br/handle/123456789/103038.

Porras, P. A. & Neumann, P. G. (1997). Emerald: Event monitoring enabling response to anomalous live disturbances, Proceedings of the 20th national information systems security conference.
RADANLIEV, Petar et al. Cyber Risk in IoT Systems. [s.l.], 8 mar. 2019. MDPI AG. http://dx.doi.org/10.20944/preprints201903.0104.v1.

SARLE, Warren S.. Neural Networks and Statistical Models. **Proceedings Of The Nineteenth Annual Sas Users Group International Conference**, Sas Institute Inc., Cary, Nc, Usa, abr. 1994.

SHOJAFAR, Mohammad; TAHERI, Rahim; POORANIAN, Zahra; JAVIDAN, Reza; MIRI, Ali; JARARWEH, Yaser. Automatic Clustering of Attacks in Intrusion Detection Systems. **2019 Ieee/acs 16th International Conference On Computer Systems And Applications (aiccsa)**, [s.l.], nov. 2019. IEEE.

W. Liang, K. Li, J. Long, X. Kui and A. Y. Zomaya, "An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063-2071, March 2020.

WILLIS, M.J.; MONTAGUE, G.A.; MASSIMO, C. di; THAM, M.T.; MORRIS, A.J.. Artificial neural networks in process estimation and control. **Automatica**, [S.L.], nov. 1992. Elsevier BV. http://dx.doi.org/10.1016/0005-1098(92)90059-o.

YANNUZZI, M. et al. Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. **2014 Ieee 19th International Workshop On Computer Aided Modeling And Design Of Communication Links And Networks (camad)**, [s.l.], dez. 2014. IEEE. http://dx.doi.org/10.1109/camad.2014.7033259.

ZARPELÃO, Bruno Bogaz et al. A survey of intrusion detection in Internet of Things. **Journal Of Network And Computer Applications**, [s.l.], v. 84, abr. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2017.02.009.

8. APÊNDICES

8.2. APÊNDICE A – ARTIGO NO FORMATO SBC

Abordagem baseada em clusterização e em Redes Neurais Artificiais para detecção de intrusão em ambientes IoT

Vinícius Eduardo Oliveira

Centro Tecnológico – Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

vinicius.eduardo.oliveira@grad.ufsc.br

Abstract. *Internet of Things (IoT) allows objects to communicate and interact with the environment in which they are inserted, in addition to performing tasks intelligently without the need for human intervention. Security is one of the biggest challenges in ensuring an ideal IoT and Fog Computing environment. Considering security aspects, intrusion detection is a key point. This article proposes an approach using clustering and artificial neural networks, which operates at Fog, to detect intrusion in an IoT environment. Through the experiments it was possible to verify that a proposed approach with clustering is able to improve the efficiency of the neural network in binary and multiclass detection.*

Resumo. *A Internet das Coisas (IoT) permite que objetos possam se comunicar e interagir com o ambiente em que estão inseridos, além de realizar tarefas de maneira inteligente sem ser necessário intervenção humana. A segurança é um dos maiores desafios para garantir um ambiente de IoT e Fog Computing ideal. Considerando os aspectos de segurança, a detecção de intrusão é um ponto chave. Este artigo propõe uma abordagem utilizando clusterização e redes neurais artificiais, que opera na Fog, para detectar intrusão em ambiente IoT. Através dos experimentos foi possível verificar que a abordagem proposta com clusterização é capaz de melhorar a eficácia da rede neural na detecção binária e multiclasse.*

1. Introdução

A chegada da Internet das Coisas (*Internet of Things* – IoT) levou à conexão universal de pessoas, objetos, sensores e serviços. O principal objetivo da IoT é fornecer uma infraestrutura de rede com protocolos de comunicação e software, permitindo a conexão e incorporação de sensores físicos, virtuais, computadores e dispositivos inteligentes, como carros, casas, geladeiras, etc. (ALABA et al., 2017).

A maioria dos dispositivos IoT possuem recursos limitados. Portanto, faz-se necessário que os dados desses dispositivos sejam transferidos através da Internet para um centro computacional onde será possível realizar o processamento e armazenamento. Assim, muitos dispositivos utilizam a computação em nuvem (*Cloud Computing*) para essa função. Porém, as grandes quantidades de dados gerados acabam resultando em congestionamento da rede na comunicação da IoT com a *Cloud*, então, para realizar o processamento e armazenamento dos temporários mais próximo aos dispositivos, surgiu a computação em névoa (*Fog Computing*), permitindo além de tudo, fazer com que o processamento em tempo real obtenha uma resposta mais rápida, uma vez que se encontra mais próximo do usuário, utilizando recursos locais como gateways e roteadores (ALABA et al., 2017) (BONOMI et al., 2012).

1.1. Motivação

Os requisitos para a implantação em larga escala da IoT estão aumentando rapidamente, resultando em uma grande preocupação em segurança. Nos últimos anos vem ocorrendo um grande aumento no número de incidentes computacionais. Por exemplo, o incidente que ocorreu em 2016, um ataque DDoS massivo, sobrecarregando vários alvos em todo o mundo. Esses ataques estavam sob o controle de um novo botnet chamado Mirai. No caso da Botnet Mirai, a grande maioria dos dispositivos utilizados eram dispositivos IoT. Estima-se que a Mirai controlava mais de 300 mil dispositivos, incluindo câmeras de segurança e roteadores, redirecionando o tráfego para realizar ataques DDoS. Esses ataques deixaram claro como os dispositivos IoT eram muito inseguros, pois através deles foi possível retirar do ar diversos serviços como o *Twitter, Spotify, Paypal, Playstation Network* entre outros (ANTONAKAKIS et al., 2017). Questões como, privacidade, autorização, verificação, controle de acesso, armazenamento e gerenciamento de informações, são os principais desafios de um ambiente IoT (RADANLIEV et al., 2019). Portanto, faz-se necessário a utilização de sistemas de detecção de intrusão, tendo como objetivo reconhecer os comportamentos intrusivos em uma rede e alertar os administradores ou realizar ações de contramedidas automaticamente (MAPLE, 2017).

Neste trabalho é realizada uma revisão do atual estado da arte relacionado a detecção de intrusão em ambiente de *Fog Computing* e IoT, identificando os principais

problemas e soluções existentes, além dos desafios futuros. Além disso, é apresentada uma proposta de uma abordagem utilizando clusterização e Redes Neurais Artificiais (*Artificial Neural Network* – ANN), que opera na *Fog Computing*, para detectar intrusão em ambiente IoT.

1.2. Contribuições

As principais contribuições desse trabalho são:

1. Contextualização sobre o atual estado da arte em relação à segurança em Fog Computing e IoT, de modo a identificar quais as questões em aberto;

2. A proposta de uma abordagem baseada em clusterização e Redes Neurais Artificiais, que opera na Fog Computing, para detecção de intrusão em ambientes IoT.

3. Investigação se a clusterização poderia influenciar positivamente na tarefa de detecção com RNA.

4. Realização de experimentos com uma base de dados de intrusões para avaliação da viabilidade de aplicação em ambiente real.

2. Estado da arte

Esta seção possui como objetivo contextualizar o estado atual das pesquisas e estudos referentes ao tema detecção de intrusão em dispositivos IoT. Foram realizados pequenos resumos, apresentando a ideia de alguns trabalhos relacionados ao tema, demonstrando o panorama atual da detecção de intrusão em IoT.

Os autores Bhushan e Sahoo (2019) propuseram um sistema de detecção de intrusão integrado, utilizando o conceito de cluster junto com a assinatura digital, onde foi possível alcançar maior eficiência e precisão de detecção de intrusão.

Os autores An et al. (2018) propuseram uma análise e modelo dos ataques DDoS (*Distributed Denial of Service)* sob uma estrutura da FC-IDS *(Fog Computing Intrusion Detection System*). Propuseram um cluster de hipergráficos baseado no algoritmo Apriori. Por meio de uma simulação, os autores verificaram que a taxa de utilização de recursos do sistema pode ser efetivamente promovida através dessa análise.

Liang et. al (2020), com o objetivo de abordar a questão da baixa taxa de precisão de detecção, alta taxa de falsos positivos e baixo desempenho que um sistema de detecção de intrusão pode ter, propuseram um algoritmo de detecção de intrusão de rede industrial no modelo de otimização de clustering de dados, onde "as distâncias ponderadas e coeficientes de segurança dos dados são classificados com base no limite de prioridade do recurso de atributo de dados para cada nó na rede". Os resultados obtidos mostraram que o algoritmo proposto foi superior em termos de taxa de detecção e tempo em comparação com outros algoritmos.

Os autores Shojafar et. al (2019) propuseram um algoritmo de cluster automático como parte de uma arquitetura de um IDS (*Intrusion Detection System*). Esse algoritmo foi baseado em conceitos de coerência e separação. O algoritmo encontra clusters com a maior semelhança entre os elementos de cluster propostos e a menor semelhança com outros clusters. Os resultados obtidos mostraram melhorias em termos do baixo número médio de funções de avaliação, alta precisão e baixo custo de computação.

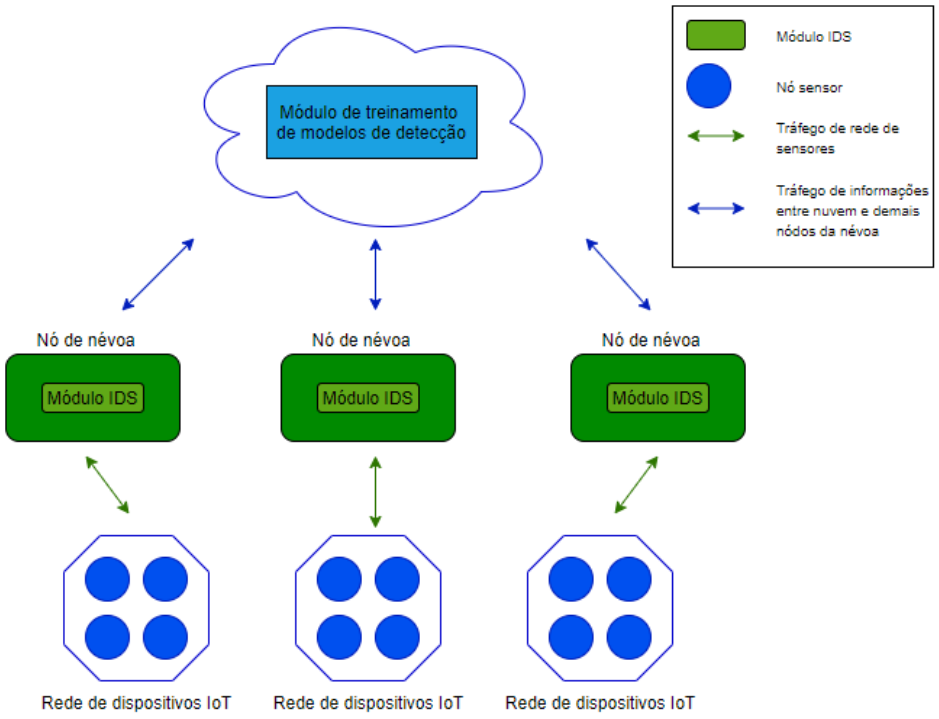
Após a análise de alguns trabalhos relacionados ao tema foi possível observar que a maioria dos trabalhos existentes focam em detecção binária, e, os que possuem detecção multiclasse possuem uma baixa eficácia na detecção. Existe uma certa importância para se ter uma detecção multiclasse eficaz, principalmente na execução das contramedidas necessárias. Outro ponto analisado foi que os sistemas de detecção de intrusão geralmente operam na nuvem, e não na Fog.

3. Proposta e desenvolvimento

A realização da análise de detecção de intrusões na névoa é interessante porque fornece uma visão geral da rede de dispositivos IoT. Os dispositivos IoT são restritos de recursos, desse modo é inviável a utilização de métodos e abordagens complexas de detecção nos próprios nós sensores, já que os mesmos não possuem capacidade de executar métodos robustos de aprendizado de máquina como redes neurais devido as suas restrições. Sendo assim, uma solução ideal, possuiria uma abordagem de detecção na névoa. A Figura 1 apresenta a ilustração do local de atuação do mecanismo proposto em uma arquitetura de um ambiente IoT seguro, baseado em computação em névoa. Nota-se que a abordagem de detecção proposta encontra-se na névoa, capturando o

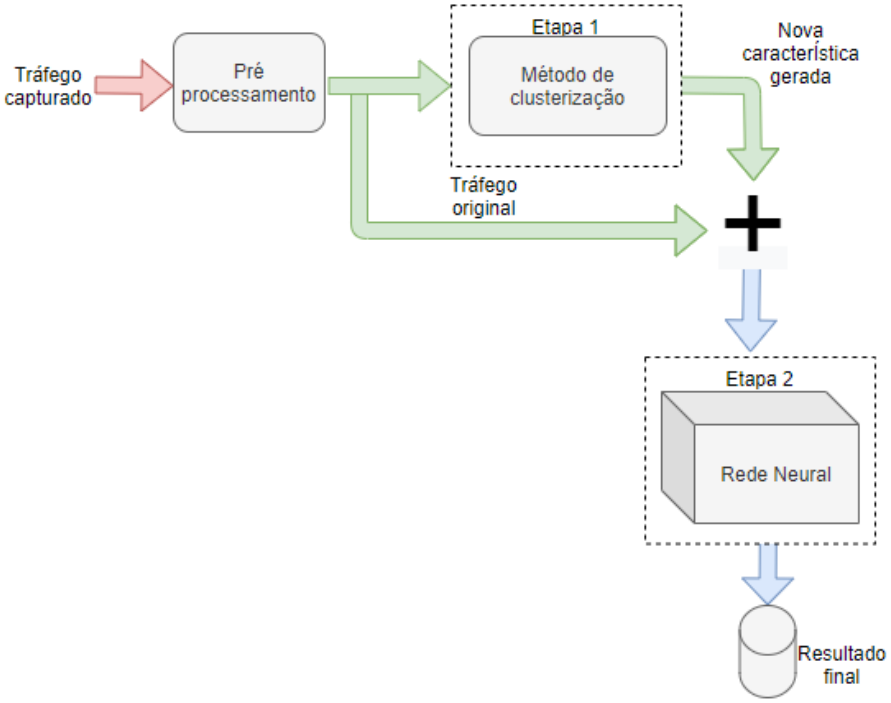
tráfego da rede de dispositivos IoT que está vinculado ao nó da Fog, analisando os dados da rede.

Figura 1 – Arquitetura do IDS proposto.



A abordagem proposta possui duas etapas principais. A etapa 1 é baseada em clusterização. Essa etapa é responsável por gerar, a partir das informações capturadas da rede, uma nova informação baseada na clusterização das já existentes. Todas essas informações são então utilizadas na segunda etapa da abordagem, onde o modelo neural realiza a classificação. Na Figura 2 é apresentada uma ilustração da abordagem proposta.

Figura 2 – Abordagem proposta.



3.1. Etapa 1 – Clusterização

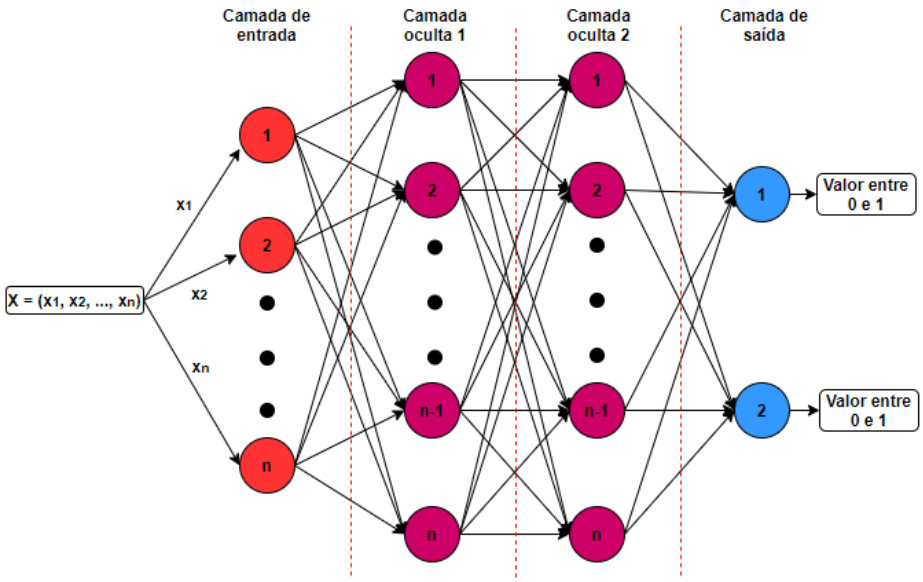
Conforme ilustrado na Figura 2 o tráfego capturado da rede de dispositivos IoT passa por uma fase de pré-processamento. Nessa etapa o tráfego capturado é convertido em atributos que podem ser submetidos a rede neural. No entanto, antes dos atributos de um determinado tráfego serem submetidos ao modelo neural, eles são enviados a um método de clusterização. Para esse trabalho será utilizado para avaliação os métodos de clusterização *Mini Batch K-means*, que é uma variação do método *K-means* e o método por densidade DBSCAN. Esses métodos geram um novo atributo que corresponde a identificação do cluster em que o tráfego foi agrupado. Os métodos de clusterização já possuem clusters criados durante a fase de treinamento. Esses métodos geram dois clusters para a saída binária e cinco para a multiclasse, pois é o número de classes que a rede neural vai classificar. Desse modo, somente calcula a qual cluster o atual tráfego tem maior associação. Após a clusterização, o novo atributo é combinado com os demais originais do tráfego, e então submetido ao modelo neural.

3.2. Etapa 2 – Classificação

Na segunda etapa da abordagem proposta ocorre a classificação com o modelo neural. A saída da rede neural vai depender da sua arquitetura de saída (binária ou multiclasse), dois ou cinco neurônios. Esses neurônios utilizam a função *softmax*, gerando um valor

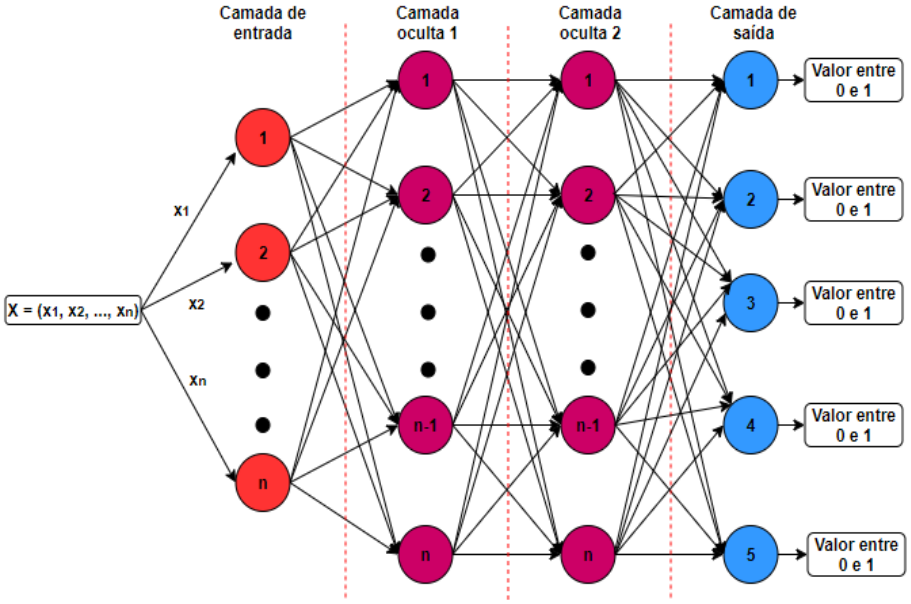
entre 0 e 1, resultando na probabilidade de a entrada estar em uma determinada classe (NWANKPA et al., 2018). Essas redes neurais possuem uma camada de entrada, duas camadas ocultas e uma camada de saída. Como função de ativação dos neurônios da camada oculta foi utilizada a ReLu. Essa função é responsável por transformar a entrada de um nó para a saída desse mesmo nó. Ela é uma função de ativação padrão para muitos tipos de redes neurais por atingir um melhor desempenho e ter uma maior facilidade de treinamento. A Figura 3 apresenta uma ilustração da arquitetura binária. Nessa arquitetura a camada de saída apresenta 2 neurônios, onde um corresponde a categoria normal e o outro a categoria ataque.

Figura 3 – Arquitetura RNA com saída binária.



Na Figura 4 é apresentada uma ilustração da arquitetura com saída multiclasse. Nessa arquitetura, a camada de saída apresenta 5 neurônios: quatro representam as categorias de ataques (DoS, Probe, R2L e U2L) e um a categoria normal.

Figura 4 – Arquitetura RNA com saída multiclasse.



4. Avaliação

Nesta seção é apresentada a metodologia definida para a avaliação da proposta. Além disso, são apresentados os resultados obtidos com os experimentos e também uma discussão sobre os mesmos.

4.1. Experimentos

Nesta seção é apresentada a metodologia experimental aplicada na avaliação da abordagem proposta. A técnica de cross-validation com 10 folds foi utilizada para estimar o quão preciso o modelo é na prática.

4.1.1. Métricas de avaliação

A partir dos resultados obtidos é possível categorizar as classificações dos eventos da base de dados da seguinte forma:

- Falso negativo (*False negative* - FN): eventos classificados como normais pelo método, mas são intrusões;
- Falso positivo (*False positive* - FP): eventos não intrusivos e classificados como intrusivos pela técnica de detecção de intrusão;
- Verdadeiro negativo (*True negative* - TN): eventos não intrusivos que foram classificados corretamente pelo método;

- Verdadeiro positivo (*True positive* - TP): eventos intrusivos que foram classificados corretamente pelo método.

Através desses termos é possível realizar o cálculo de diferentes métricas, auxiliando na avaliação dos experimentos. As métricas utilizadas para avaliar os experimentos deste trabalho foram as seguintes (ALMIANI; ABUGHAZLEH; AL-RAHAYFEH; RAZAQUE, 2019):

- Acurácia (ACC): corresponde a proporção de instâncias classificadas corretamente em relação ao total de instâncias. Pode ser calculada através da seguinte forma:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Erro (ERR): corresponde a proporção de instâncias classificadas incorretamente em relação ao total de instâncias. Pode ser calculado através da seguinte forma:

$$ERR = \frac{FP+FN}{TP+TN+FP+FN} \quad (2)$$

- Precisão (PRE): corresponde às instâncias intrusivas corretamente classificadas. Pode ser calculada da seguinte forma:

$$PRE = \frac{TP}{TP+FP} \quad (3)$$

- *Recall* ou *True Positive Rate* (TPR): corresponde ao número de instâncias classificadas como intrusivas, dentre todas que realmente são intrusivas. Pode ser calculada da seguinte forma:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

- *True Negative Rate* (TNR): corresponde ao número de instâncias classificadas como não intrusivas, entre todas que realmente são não intrusivas. Pode ser calculada da seguinte forma:

$$TNR = \frac{TN}{TN+FP} \quad (5)$$

- F1-Score: corresponde a precisão de um teste. É a média harmônica de precisão e recuperação, onde seu melhor valor é 1 e o pior 0. Pode ser calculada da seguinte forma:

$$F1 - Score = 2 * \frac{PRE*Recall}{PRE+Recall} \quad (6)$$

- *Matthews Correlation Coefficient* (MCC): corresponde a uma medida da qualidade das classificações binárias no aprendizado de máquina. Possui um intervalo de -1 a 1, onde -1 indica um classificador binário completamente errado, enquanto 1 indica um classificador binário completamente correto. Pode ser calculado da seguinte forma:

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(FN+TN)*(FP+TN)*(TP+FN)}} \quad (7)$$

4.1.2. Base de dados

Para a avaliação são utilizadas base de dados públicas de ataques. Para a utilização desse trabalho foi utilizada a base de dados NSL-KDD. Essa base de dados cobre quatro categorias principais de ataques:

- Ataque de negação de serviço (*Denial of Service* - DoS): bloqueia solicitações legítimas para um recurso da rede, consumindo banda ou sobrecarregando recursos computacionais;
- Ataque por sondagem (*Probe Attack*): revela as informações da rede, podendo ser explorada por outro tipo de ataque. Coleta informações antes de iniciar um ataque, quebrando seus controles de segurança;
- Ataque de usuários remotos (*Remote to Local* - R2L): nesse tipo de ataque um invasor que não possui uma conta em uma máquina remota envia um pacote para a máquina através da rede, explorando vulnerabilidades para obter acesso local;
- Ataque de usuário Root (*User to Root* - U2R): nesse tipo de ataque um invasor com um usuário normal no sistema explora vulnerabilidades a fim de obter acesso root.

Essa base de dados possui 22 tipos de ataque no conjunto de treinamento, com 17 adicionais no conjunto de teste e 125973 registros. Essa base não inclui registros redundantes no conjunto de treino, bem como não há registros duplicados no conjunto de teste (para não

ser influenciado pelos métodos que possuem melhores taxas de detecção de registros frequentes). Cada instância da base possui contém 41 recursos (números, nominais e binários). Na Tabela 1 é apresentada a distribuição da instância da base de dados NSL-KDD.

Tabela 1 – Distribuição da instância no conjunto de dados de treinamento

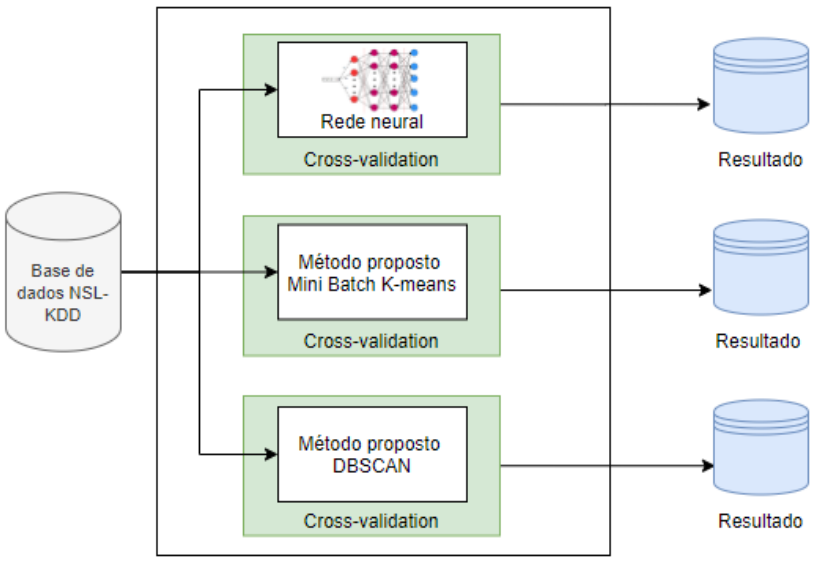
Tipo de ataque	**Número de registros**
Normal	67343
DoS	45927
Probe	11656
R2L	995
U2R	52
Total	125973

4.1.3. Descrição dos experimentos

Foram utilizados apenas os dados referentes aos conjuntos de treino com o auxílio do método *cross-validation* com 10 *folds*, onde as 125973 amostras foram divididas em 10 subconjuntos, e cada um desses subconjuntos foi utilizado uma vez para teste enquanto os demais foram utilizados na etapa de treinamento.

Para a realização do experimento binário, houve um pré-processamento dos dados, transformando todos os dados considerados como ataques em 1 e os normais em 0. Na Figura 5 é apresentada uma ilustração do experimento, onde a mesma base de dados é utilizada tanto para a binária quanto para a multiclasse, gerando um resultado binário e multiclasse para a RNA pura, para a RNA juntamente com o Mini Batch K-means e para a RNA com o método DBSCAN.

Figura 5 – Experimentos utilizando a base de dados NSL-KDD.



4.2. Resultados

Nesta seção são apresentados os resultados obtidos através de experimentos com o método proposto. Foram realizados experimentos utilizando a RNA pura, RNA juntamente com o método de clusterização *Mini Batch K-means* e RNA com o método de clusterização DBSCAN, a fim de obter diferentes resultados para verificar se os resultados utilizando clusterização são mais eficazes do que sem.

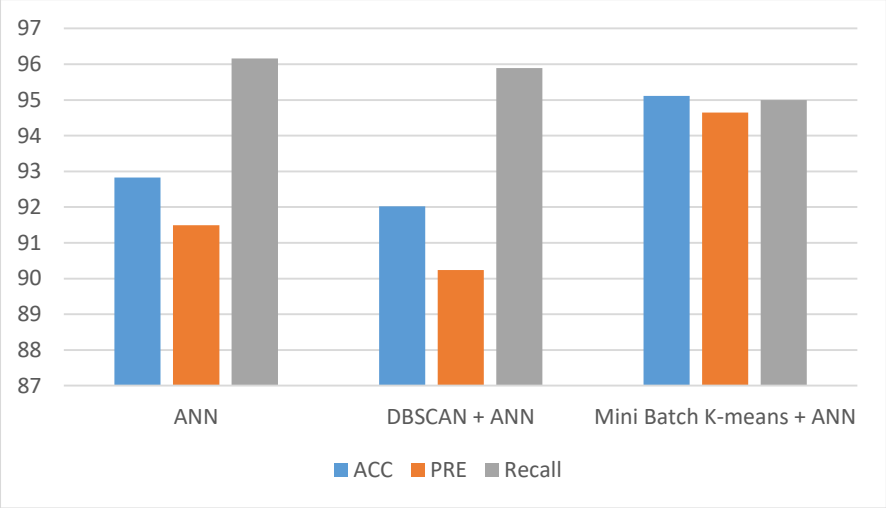
Inicialmente a abordagem proposta é avaliada em relação a detecção binária. O objetivo é verificar a influência dos métodos de clusterização para a tarefa de classificação binária realizada pela rede neural. Os resultados obtidos pela abordagem com RNA, pela abordagem proposta com DBSCAN (RNA + DBSCAN) e pela abordagem proposta com *Mini Batch K-Means* (RNA + *Mini Batch K-means*), são apresentados na Tabela 2.

Tabela 2 – Resultados dos experimentos na classificação binária

	ACC	**ERR**	**PRE**	**Recall**	**TNR**	**F1 - SCORE**	**MCC**
RNA	92,83	7,17	91,49	96,16	90,01	93,21	0,87
RNA + DBSCAN	92,02	7,98	90,24	95,89	88,65	92,49	0,85
RNA + MINI BATCH K-MEANS	95,11	4,89	94,65	95,00	95,23	94,75	0,90

Como pode ser observado, a RNA pura obteve uma acurácia de 92,83%. Utilizando o método de clusterização por densidade DBSCAN, nesse caso, a acurácia foi um pouco inferior (92,02%). Já utilizando o método *Mini Batch K-Means*, houve uma melhora na acurácia (95,11%) em relação às duas análises anteriores. Além disso, o método *Mini Batch K-means* foi o que teve uma melhor precisão na classificação dos dados, comprovando que foi o método com menos falsos positivos, e, um maior MCC. O MCC é a confirmação que esse método possui uma melhor qualidade na detecção binária. Nessa métrica, valores mais próximos a 1 significam uma detecção mais correta. O gráfico a seguir é uma ilustração da acurácia, recall e precisão dos resultados dos experimentos. Podemos verificar através do gráfico que o *Mini Batch K-means* foi o método que obteve um maior balanceamento entre acurácia, recall e precisão.

Gráfico 1 - Comparação dos resultados binários entre os métodos propostos considerando a acurácia, precisão e recall.



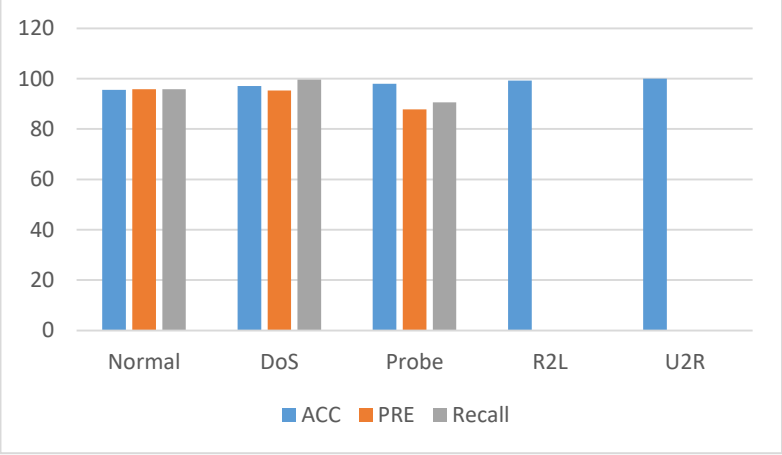
Na segunda etapa foram realizados experimentos para verificar o desempenho da abordagem proposta em relação a detecção multiclasse. Nessa etapa, os resultados obtidos são classificados como normal ou nas quatro categorias de ataques da base. Os resultados obtidos pela abordagem com RNA sem nenhum método de clusterização são demonstrados na Tabela 3.

Tabela 3 - Resultados multiclasse de experimentos utilizando RNA.

CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	95,57	4,43	95,84	95,86	95,25	95,85
DoS	97,06	2,94	95,35	99,67	97,29	96,00
Probe	97,94	2,06	87,84	90,58	98,70	89,18
R2L	99,24	0,76	0	0	100	0
U2R	99,95	0,05	0	0	100	0

Analisando os dados é possível observar uma acurácia consideravelmente alta em todas as classificações. Na categoria R2L e U2R, apesar de RNA apresentar uma acurácia de quase 100%, obteve um recall de 0 significando que não houve ataques detectados. Além disso, a precisão baixa também caracteriza a classificação de muitos falsos positivos. O gráfico a seguir é uma ilustração da acurácia, recall e precisão dos resultados dos experimentos. É possível verificar através do gráfico que as categorias R2L e U2R possuem uma maior acurácia, mesmo com precisão e recall 0. Isso se dá pelo fato de haver poucas instâncias dessas duas categorias, que acabam não interferindo na acurácia. Nesse caso houve uma melhor classificação nos outros três tipos de categorias, com destaque para DoS e Normal, que tiveram um melhor recall e melhor precisão.

Gráfico 2 – Comparação dos resultados multiclasse utilizando somente a RNA considerando a acurácia, precisão e recall.



Na Tabela 4 são demonstrados os resultados obtidos pela abordagem utilizando o método de clusterização DBSCAN juntamente com a RNA. Nessa tabela os

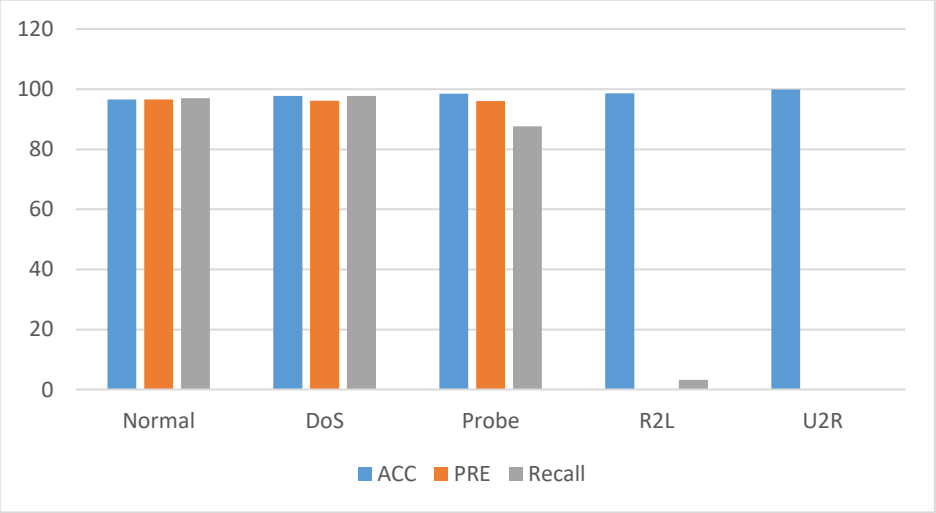
resultados também são classificados como normal ou nas quatro categorias de ataques da base.

Tabela 4 - Resultados multiclasse de experimentos utilizando RNA juntamente com DBSCAN.

CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	96,55	3,45	96,57	97,01	96,01	96,79
DoS	97,78	2,22	96,18	97,79	97,78	96,97
Probe	98,50	1,50	96,02	87,64	99,62	91,39
R2L	98,56	1,44	0	3,19	99,32	0
U2R	99,90	0,10	0	0	99,92	0

Através dos resultados obtidos pode-se observar também uma acurácia alta em todas as classificações. Em relação aos dados obtidos no experimento sem nenhum método de clusterização, aqui verifica-se uma melhora, mesmo que muito pouca, no recall da categoria R2L. Além disso, uma leve melhora na precisão como um todo, com destaque para a classificação da categoria Probe que alcançou uma melhora de mais de 8% em relação aos resultados anteriores. Através do gráfico a seguir, pode-se observar um maior balanceamento entre as categorias Normal, DoS e Probe.

Gráfico 3 – Comparação dos resultados multiclasse utilizando RNA juntamente com DBSCAN considerando a acurácia, precisão e recall.



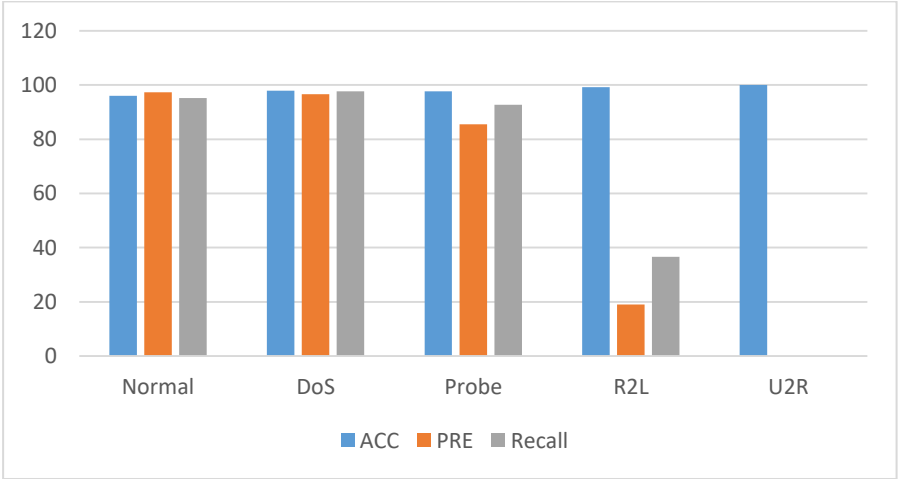
Por fim, na Tabela 5 são demonstrados os resultados obtidos pela abordagem utilizando o método de clusterização *Mini Batch K-means* juntamente com a RNA. Neste experimento a base de dados também foi classificada de acordo com as cinco categorias existentes.

Tabela 5 - Resultados multiclasse de experimentos utilizando RNA juntamente com Mini Batch K-means.

CLASSE	ACC	ERR	PRE	Recall	TNR	F1-SCORE
Normal	96,02	3,98	97,35	95,20	96,96	96,25
DoS	97,89	2,11	96,64	97,67	98,02	97,14
Probe	97,92	2,08	85,44	92,69	98,43	88,92
R2L	99,15	0,85	18,98	36,63	99,56	0
U2R	99,96	0,04	0	0	99,99	0

Através dos resultados obtidos pode-se observar também que esta abordagem alcançou uma acurácia alta em todas as classificações. Em relação aos dados obtidos no experimento sem nenhum método de clusterização e com DBSCAN, aqui verifica-se pela primeira vez uma melhora na precisão. Além disso, esta abordagem obteve uma melhora de 36,63% no recall em relação a RNA pura e de 33,44% em relação ao DBSCAN da categoria R2L. Através do gráfico a seguir, pode-se observar um maior balanceamento entre as categorias Normal e DoS e Probe, com destaque para leve melhora da categoria R2L.

Gráfico 4 – Comparação dos resultados multiclasse utilizando RNA juntamente com Mini Batch K-means considerando a acurácia, precisão e recall.



4.3.Discussão

Em relação aos resultados obtidos, comprova-se que a abordagem utilizando *Mini Batch K-means* foi mais eficaz em relação as outras duas abordagens. Na classificação binária, houve um aumento da acurácia e precisão, resultando na diminuição de falsos positivos. Observando o MCC, que corresponde a uma medida da qualidade das classificações binárias, o Mini Batch *K-means* obteve uma melhor classificação, pois seu valor, dentre as três abordagens, foi o que mais se aproximou de 1.

Em relação a análise multiclasse, a abordagem utilizando *Mini Batch K-means* foi a que apresentou maior melhora na detecção. Na categoria R2L houve um aumento da acurácia e recall, indicando uma maior classificação de verdadeiros positivos, provando que o *Mini Batch K-means* contribui com a melhora na detecção.

Nenhuma das técnicas, nem a RNA pura, conseguiu detectar U2R. Nesse tipo de ataque, qualquer usuário normal do sistema obtém acesso ilegal aos privilégios de superusuário, por isso, há uma importância em maiores estudos para melhorar a taxa de detecção desse tipo de ataque.

A abordagem utilizando clusterização apresentou melhoras tanto na classificação binária como na classificação multiclasse, portanto pode ser considerada interessante para a detecção de intrusão. Todavia, existe a necessidade de maiores estudos, avaliando outras estruturas de redes neurais. Acredita-se que estruturas de redes neurais mais robustas (com mais camadas e mais neurônios por camadas) possam ser mais úteis para a detecção principalmente das categorias R2L e U2R.

5. Conclusão e trabalhos futuros

Com a expansão da internet das coisas (*Internet of Things* – IoT) e a informatização de dados, surgiram dificuldades para manter a segurança dos mesmos. Privacidade, controle de acesso, armazenamento e gerenciamento de informações são exemplos de desafios de um ambiente IoT. Como solução, os sistemas de detecção de intrusão (*Intrusion Detection System* – IDS) auxiliam a prevenir o acesso não autorizado à rede, analisando o tráfego de rede e classificando os registros como normais ou anômalos.

Nos experimentos realizados percebeu-se que a utilização de métodos de clusterização juntamente com RNA auxiliam na detecção de intrusão. Dos métodos de clusterização utilizados, o *Mini Batch K-means* foi o que apresentou um melhor resultado, com uma melhor taxa de acurácia e precisão. Além disso, esse método foi o

que conseguiu um melhor recall na categoria R2L e o único que conseguiu uma melhora na precisão, resultando numa melhor detecção de verdadeiros positivos e diminuição de falsos positivos.

A partir dos resultados obtidos, observa-se que a abordagem híbrida com clusterização e RNA se mostra promissora. Como trabalhos futuros, são necessários estudos considerando a utilização de outros tipos de métodos de clusterização juntamente com redes neurais mais robustas (com maior quantidade de camadas ocultas e mais neurônios por camadas) para obter uma melhora ainda mais significativa na detecção.

6. Referências

A demo of the K Means clustering algorithm — scikit-learn 0.11-git documentation. Disponível em <https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/cluster/plot_mini_batch_kmeans.html>. Acesso em: 30 de outubro de 2020.

AAZAM, Mohammad et al. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. **Proceedings Of 2014 11th International Bhurban Conference On Applied Sciences & Technology (ibcast) Islamabad, Pakistan, 14th - 18th January, 2014**, [s.l.], jan. 2014. IEEE. http://dx.doi.org/10.1109/ibcast.2014.6778179.

ALABA, Fadele Ayotunde et al. Internet of Things security: A survey. **Journal Of Network And Computer Applications**, [s.l.], v. 88, jun. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2017.04.002.

ALMIANI, Muder; ABUGHAZLEH, Alia; AL-RAHAYFEH, Amer; RAZAQUE, Abdul. Cascaded hybrid intrusion detection model based on SOM and RBF neural networks. **Concurrency And Computation**: Practice and Experience, [S.L.], v. 32, n. 21, p. 0-1, 7 mar. 2019. Wiley. http://dx.doi.org/10.1002/cpe.5233.

AN, Xingshuo; SU, Jingtao; LÜ, Xing; LIN, Fuhong. Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system. **Eurasip Journal On Wireless Communications And Networking**, [s.l.], v. 2018, n. 1, 22 out. 2018. Springer Science and Business Media LLC.

ANTONAKAKIS, Manos; APRIL, Tim; BAILEY, Michael; BERNHARD, Matthew; BURSZTEIN, Elie; COCHRAN, Jaime; DURUMERIC, Zakir; INVERNIZZI, Luca; HALDERMAN, J. Alex; KALLITSIS, Michalis; KUMAR, Deepak; LEVER, Chaz; MA, Zane; MASON, Joshua; MENSCHER, Damian; SEAMAN, Chad; SULLIVAN, Nick; THOMAS, Kurt; ZHOU, Yi. Understanding the Mirai Botnet. **Usenix The Advanced Computing Systems Association**, Vancouver, 16 ago. 2017.

ARLOT, Sylvain; CELISSE, Alain. A survey of cross-validation procedures for model selection. **Statistics Surveys**, [S.L.], 2010. Institute of Mathematical Statistics. http://dx.doi.org/10.1214/09-ss054.

Asaka, M., Okazawa, S., Taguchi, A. & Goto, S. (1999). A method of tracing intruders by use of mobile agents, INET'99 Proceedings.

Bace, R. & Mell, P. (2001). Nist special publication on intrusion detection systems, Technical report, BOOZ-ALLEN AND HAMILTON INC MCLEAN VA.

BARCELLOS, Raissa; VITERBO, José; BERNARDINI, Flavia C. Uso de algoritmos de clusterização para a identificação de padrões de consumo de energia elétrica. Universidade Federal Fluminense. 2016

BELLO, Oladayo; ZEADALLY, Sherali; BADRA, Mohamad. Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). **Ad Hoc Networks**, [s.l.], v. 57, p.52-62, mar. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.adhoc.2016.06.010.

BHUSHAN, Bharat; SAHOO, G.. A Hybrid Secure and Energy Efficient Cluster Based Intrusion Detection system for Wireless Sensing Environment. **2019 2nd International Conference On Signal Processing And Communication (icspc)**, [s.l.], mar. 2019. IEEE.

BONOMI, Flavio et al. Fog computing and its role in the internet of things. **Proceedings Of The First Edition Of The Mcc Workshop On Mobile Cloud Computing - Mcc '12**, [s.l.], 2012. ACM Press. http://dx.doi.org/10.1145/2342509.2342513.

BRONZATTI, Luiz Fernando Casarin. Análise sobre a tecnologia de rede sem fio Zigbee / IEEE 802.15.4. **Universidade de São Paulo**, jun. 2013. http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-08112013-101735/.

BROWNE, Michael W. Cross-Validation Methods. **Journal Of Mathematical Psychology**, [S.L.], mar. 2000. Elsevier BV. http://dx.doi.org/10.1006/jmps.1999.1279.

BUYYA, Rajkumar et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Generation Computer Systems**, [s.l.], v. 25, n. 6, jun. 2009. Elsevier BV. http://dx.doi.org/10.1016/j.future.2008.12.001.

Campello, R. S. & Weber, R. F. (2001). Sistemas de detecçao de intrusao, Minicurso procedente do 19o Simpósio Brasileiro de Redes de Computadores.

CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection. **Acm Computing Surveys**, [s.l.], v. 41, n. 3, 1 jul. 2009. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/1541880.1541882.

CHIANG, Mung; ZHANG, Tao. Fog and IoT: An Overview of Research Opportunities. **Ieee Internet Of Things Journal**, [s.l.], v. 3, n. 6, dez. 2016. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/jiot.2016.2584538.

DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning - Easy Guides - Wiki – STHDA. Disponível em <http://www.sthda.com/english/wiki/wiki.php?id_contents=7940>. Acesso em: 30 de outubro de 2020.

DEPREN, Ozgur et al. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. **Expert Systems With Applications**, [s.l.], v. 29, n. 4, p.713-722, nov. 2005. Elsevier BV. http://dx.doi.org/10.1016/j.eswa.2005.05.002.

FERREIRA, Vinicius Oliveira. Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando métodos de agrupamento. **Programa de Pós-Graduação em Ciência da Computação – UNESP**, abr. 2016. http://hdl.handle.net/11449/138755.

GARCÍA-TEODORO, P. et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. **Computers & Security**, [s.l.], v. 28, n. 1-2, fev. 2009. Elsevier BV. http://dx.doi.org/10.1016/j.cose.2008.08.003.

Ilgun, K. (1993). Ustat: A real-time intrusion detection system for unix, Research in Security and Privacy, 1993. Proceedings., 1993 IEEE Computer Society Symposium on, IEEE.

k-means clustering | Wikiwand. Disponível em <https://www.wikiwand.com/en/K-means_clustering>. Acesso em: 30 de outubro de 2020.

LIAO, Hung-jen et al. Intrusion detection system: A comprehensive review. **Journal Of Network And Computer Applications**, [s.l.], v. 36, n. 1, jan. 2013. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2012.09.004.

LIN, Ying; ZHANG, Yan; OU, Yang-jia. The Design and Implementation of Host-Based Intrusion Detection System. **2010 Third International Symposium On Intelligent Information Technology And Security Informatics**, [s.l.], abr. 2010. IEEE. http://dx.doi.org/10.1109/iitsi.2010.127.

MAPLE, Carsten. Security and privacy in the internet of things. **Journal Of Cyber Policy**, [s.l.], v. 2, n. 2, 4 maio 2017. Informa UK Limited. http://dx.doi.org/10.1080/23738871.2017.1366536.

MELL, P M; GRANCE, T. The NIST definition of cloud computing. **NIST**, [s.l.], 2011. National Institute of Standards and Technology. http://dx.doi.org/10.6028/nist.sp.800-145.

MENEZES, Rodrigo Cezar. CLUSTERIZAÇÃO DE DADOS UTILIZANDO O ALGORITMO K-MEANS. [s.l.], 2013. Faculdades Catolicas.

MUSSOI DE LIMA, Igor Vinícius. Uma abordagem simplificada de detecção de intrusão baseada em redes neurais artificiais. **Programa de Pós-Graduação em Ciência da Computação – UFSC**, 2005. Disponível em: http://repositorio.ufsc.br/handle/123456789/103038.

Porras, P. A. & Neumann, P. G. (1997). Emerald: Event monitoring enabling response to anomalous live disturbances, Proceedings of the 20th national information systems security conference.

RADANLIEV, Petar et al. Cyber Risk in IoT Systems. [s.l.], 8 mar. 2019. MDPI AG. http://dx.doi.org/10.20944/preprints201903.0104.v1.

SARLE, Warren S.. Neural Networks and Statistical Models. **Proceedings Of The Nineteenth Annual Sas Users Group International Conference**, Sas Institute Inc., Cary, Nc, Usa, abr. 1994.

SHOJAFAR, Mohammad; TAHERI, Rahim; POORANIAN, Zahra; JAVIDAN, Reza; MIRI, Ali; JARARWEH, Yaser. Automatic Clustering of Attacks in Intrusion Detection

Systems. **2019 Ieee/acs 16th International Conference On Computer Systems And Applications (aiccsa)**, [s.l.], nov. 2019. IEEE.

W. Liang, K. Li, J. Long, X. Kui and A. Y. Zomaya, "An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063-2071, March 2020.

WILLIS, M.J.; MONTAGUE, G.A.; MASSIMO, C. di; THAM, M.T.; MORRIS, A.J.. Artificial neural networks in process estimation and control. **Automatica**, [S.L.], nov. 1992. Elsevier BV. http://dx.doi.org/10.1016/0005-1098(92)90059-o.

YANNUZZI, M. et al. Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. **2014 Ieee 19th International Workshop On Computer Aided Modeling And Design Of Communication Links And Networks (camad)**, [s.l.], dez. 2014. IEEE. http://dx.doi.org/10.1109/camad.2014.7033259.

ZARPELÃO, Bruno Bogaz et al. A survey of intrusion detection in Internet of Things. **Journal Of Network And Computer Applications**, [s.l.], v. 84, abr. 2017. Elsevier BV. http://dx.doi.org/10.1016/j.jnca.2017.02.009.