UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADIEL DOS SANTOS NASCIMENTO
JAYME DA COSTA CABRAL JÚNIOR

AVALIAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A DETECÇÃO DE TRÁFEGO ANÔMALO EM AMBIENTE SEM FIO DOMÉSTICO

Belém
2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ADIEL DOS SANTOS NASCIMENTO
JAYME DA COSTA CABRAL JÚNIOR

AVALIAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A DETECÇÃO DE TRÁFEGO ANÔMALO EM AMBIENTE SEM FIO DOMÉSTICO

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Antônio Jorge Gomes Abelém

Coorientador: Msc.Igor Furtado Carvalho

Belém
2018

AVALIAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A DETECÇÃO DE TRÁFEGO ANÔMALO EM AMBIENTE SEM FIO DOMÉSTICO/ ADIEL DOS SANTOS NASCIMENTO
JAYME DA COSTA CABRAL JÚNIOR. – Belém, 2018.
43 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Antônio Jorge Gomes Abelém

Monografia – UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO, 2018.

1. Inteligência Artificial. 2. Aprendizagem de Máquina. 3. Mineração de Dados. 4. Redes de Computadores.

ADIEL DOS SANTOS NASCIMENTO
JAYME DA COSTA CABRAL JÚNIOR

AVALIAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA A DETECÇÃO DE TRÁFEGO ANÔMALO EM AMBIENTE SEM FIO DOMÉSTICO

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Ciência da Computação.

Data da Defesa: 18 de Dezembro de 2018
Conceito: Excelente

Banca Examinadora

Prof. Dr. Antônio Jorge Gomes Abelém
Faculdade de Computação - UFPA
Orientador

Prof.ª Dr.ª Fabiola Pantoja Oliveira Araújo
Faculdade de Computação - UFPA
Membro da Banca

Prof. Dr. Raimundo Viegas Junior
Faculdade de Computação - UFPA
Membro da Banca

Belém
2018

Dedicamos a realização deste trabalho primeiramente a Deus pelo dom da vida e do conhecimento, as nossas famílias pelo total e incondicional apoio durante essa ardua, porém vitoriosa caminhada, e aos professores pela partilha do conhecimento e pelos ensinamentos para a vida.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por ter me dado saúde e forças para superar os desafios oriundos da realização deste trabalho, e permitir que isso acontecesse não somente nos anos como universitário mas ao longo de minha vida. Agradeço também a Universidade Federal do Pará pelo seu corpo docente, direção e administração que oportunizaram um ambiente criativo e amigável, além da oportunidade de fazer o curso.

Não poderia me esquecer de todos os professores por me proporcionar não apenas o conhecimento não apenas racional mas na manifestação de caráter e afetividade, em especial aqueles que romperam os limites da hierarquia imposta pela universidade e se tornaram meus amigos os quais posso citar: Prof. Dr. Regiane Kawasaki, Prof. Dr. Fabiola Pantoja, Prof. Dr. Marcelle Mota e Prof. Josivaldo Araújo. Ao Prof. Dr. Antônio Abelém pela oportunidade de participação no projeto INSANE e apoio na elaboração deste trabalho, e ao meu co-orientador Msc. Igor Furtado pelo auxilio nas dúvidas e correções.

Agradeço em especial a minha família que apesar das dificuldades enfrentadas nesses 4 anos sempre me forneceram apoio, incentivo nas horas difíceis, de desânimo e cansaço. Aos meus amigos e companheiros de estudo e irmãos na amizade que desejo que continuem fazendo parte da minha vida, Muito Obrigado!

A todas as pessoas que contribuíram direta e indiretamente com a minha formação, meu muito obrigado!

- Adiel Nascimento

Em primeiro lugar agradeço a deus, por permitir que houvesse a possibilidade de eu alcançar este estágio da minha vida. A universidade por disponibilizar o espaço de ensino, a faculdade por prestar um ótimo serviço, tanto no atendimento e resolução de problemas, como no auxílio ao ensino. Aos meus colegas de laboratório e projeto por me ajudarem expandir os conhecimentos obtidos no curso. E aos professores por nos acompanharem de perto durante esta caminhada sempre ajudando no que era preciso, os quais não irei citar para não acabar esquecendo nenhum.

Por fim, agradeço todas as pessoas que me ajudaram a concluir esta etapa da minha vida, em especial a meus amigos e a meus familiares que desde o ensino básico, me incentivarem a buscar conhecimento, e por ajudarem sempre que foi preciso. Gostaria de dedicar a eles este trabalho.

- Jayme Cabral

"A verdadeira viagem de descobrimento
não consiste em procurar novas paisagens,
mas em ter novos olhos."
(Marcel Proust)

RESUMO

O advento tecnológico ocorrido nos últimos anos popularizou e democratizou o acesso à rede mundial de computadores, propiciando o surgimento de novas tecnologias multimídias hospedadas remotamente e na alta oferta de conteúdos e serviços disponibilizados na Internet. O principal impacto social consequente desses acontecimentos foi no tempo que as pessoas dedicam ao uso de seus aparelhos eletrônicos. Na contramão dos benefícios oferecidos por este avanço tecnológico o número de usuários mal-intencionados que utilizam métodos para driblar a segurança em roteadores de borda aumentou exponencialmente, eles utilizam artifícios que demandam de uma criatividade cada vez maior oferecendo riscos principalmente para usuários leigos. Por essa razão, a utilização de modelos que identificam anomalias no tráfego de roteadores de borda geradas por ataques apresenta-se com grande importância, visto que as técnicas de segurança estão aplicadas principalmente em servidores. Este trabalho promove uma avaliação dos principais algoritmos de aprendizado de máquina supervisionados, com a intenção verificar o comportamento apresentado por esses algoritmos na detecção de anomalias geradas por ataques de negação de serviço, em um dataset contendo o tráfego de uma rede doméstica simulada em laboratório. Os algoritmos KNN, Naive Bayes e Árvore de Decisões foram utilizados na realização dos experimentos que de maneira geral ambos os algoritmos obtiveram um desempenho acima dos 90% de precisão, no entanto o KNN se apresentou como melhor algoritmo mesmo considerando a necessidade de um processamento maior em comparação aos outros algoritmos.

Palavras-chave: Inteligência Artificial. Aprendizagem de Máquina. Mineração de Dados. Redes de Computadores.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgement
AM	Aprendizado de Máquina
APP	Aplicativo Móvel
CPD	Change Points Detection
CPU	Central Process Unit
GPL	General Public License
GB	Gigabyte
IoT	Internet of Things
ISP	Internet Service Provider
K-NN	k-nearest neighbors
LAN	Local Area Network
MB	Megabyte
RAM	Random Access Memory
SVM	Support Vector Machine
SYN	Synchronize
TCP	Transmission Control Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
Wi-Fi	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO

Nos dias atuais, é inimaginável o funcionamento de empresas e governos sem as facilidades oferecidas pelas redes de computadores em função do papel socioeconômico estratégico que desempenham na sociedade moderna. O consultor global da *We Are Social* (KEMP, 2018) divulgou um estudo realizado pela empresa que aponta que mais da metade da população mundial possui acesso à internet. Essa popularização é resultado do advento das tecnologias de comunicação móvel, em que a internet passou por um processo de popularização, atualmente elas são encontradas nos mais diversos e remotos lugares.

Como reflexo dessa popularização, a ação de cyber criminosos migraram para os meios digitais e os principais tipos de ataques são: ataques de negação de serviço (DDoS), infiltração cyber criminosa e o roubo de informações. Uma das principais dificuldades encontradas no combate dessas ações maliciosas é a facilidade em realizar os ataques dado a quantidade tutoriais na internet. Entretanto, esses ataques resultam em variações no tráfego de redes domésticas, principalmente o ataque de DDoS. Dessa forma, é possível explorar essa característica gerada pelos ataques no tráfego da rede, treinando algoritmos de aprendizado de máquina para detectar e classificar essas variações.

O aprendizado de máquina é um campo de estudo derivado da ciência da computação e consiste em dar para as máquinas a capacidade de aprender com base nas experiências passadas. O pesquisador da IBM Samuel (1959) conceitua o aprendizado de máquina como "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados". De forma geral o aprendizado de máquina se divide em duas principais abordagens, os métodos preditivos que trabalham com a classificação e regressão e os métodos prescritivos que trabalham principalmente com associações e agrupamentos. Existem várias outras maneiras de utilizar aprendizagem de máquina e uma delas é utilizando os erros como forma de ensinar o algoritmo, é o caso do aprendizado por reforço onde um agente é responsável por tratar o problema em um ambiente dinâmico com interações do tipo de tentativa e erro.

É evidente que as grandes empresas que disponibilizam seus serviços online possuem uma boa porcentagem da sua verba, destinada ao investimento na melhoria da infraestrutura de segurança de seus servidores. A analista da Gartner Pettey (2018) aponta o uso de aprendizado de máquina, como uma das tendências no gerenciamento de riscos de empresas. No entanto, os usuários domésticos não contam com essa proteção e acabam se tornando vítimas fáceis aos usuários maliciosos, que assumem o controle de dispositivos para a realização de ataques de negação de serviço, roubam informações sigilosas e expõem pessoas publicamente, sem o devido consentimento dos donos. Além disso, a utilização de Aprendizagem de Máquina (AM) na detecção de anomalias nas redes dos usuários pode beneficiar diretamente as provedoras de internet (ISP), reduzindo custos para a provedora e melhorando a qualidade na prestação de serviço aos clientes.

1.1 Motivação

A principal motivação para a realização do trabalho foi a oportunidade de trabalhar com as áreas de Rede de Computadores e Aprendizado de Máquina, que, em um mundo cada vez mais globalizado, ocupam um grande espaço no cotidiano das pessoas, seja nas conexões necessárias para a comunicação entre dispositivos computacionais na área de redes, ou na utilização de aprendizagem de máquina na avaliação de crédito de clientes de cartão de crédito por exemplo.

A utilização de Aprendizado de Máquina(AM) pode oferecer uma solução melhor que as tradicionais, sendo capaz de utilizar e atualizar sua base de dados, para se adaptar mediante a realização de treinos contínuos aos diversos métodos de ataques em redes domésticas. A característica adaptativa dos algoritmos de AM é a principal vantagem sobre os métodos tradicionais, pois ela garante que a prevenção de ataques se adéque aos ataques.

O principal benefício para as redes de computadores no uso de AM neste trabalho é a possibilidade de uma base de dados resultante em futuras implementações, capazes de identificar tráfego anômalo e classificá-lo. Partindo da classificação, iniciar ações para remediar esses ataques. Outros problemas existentes em redes podem ser solucionados com AM aplicada em roteadores, possibilitando a diminuição das filas de pacotes e consequentemente reduzindo os atrasos nas transmissões de pacotes melhorando o funcionamento da arquitetura da rede descrita por Kurose e Keith (2013).

Com o surgimento de novas tecnologias de hardware e software com menor custo e maior capacidade de processamento, é esperado para o futuro a popularização de aplicações utilizando AM, como é indicado no *hyper cicle* por (PETTEY, 2018). De forma geral, o AM pode ajudar em muitos problemas, sendo o único limitador a disponibilidade de mão de obra para realizar a modelagem dos dados, e o treinamento dos algoritmos.

1.2 Justificativa

Grandes mudanças sociais e econômicas decorreram do avanço tecnológico, no entanto, a compreensão dos riscos que a navegação oferece evoluem a passos curtos. Dessa forma, é de fundamental importância a realização de estudos que busquem o desenvolvimento de estratégias capazes de agir na prevenção de ataques em redes de computadores, garantindo mais segurança aos usuários conectados na internet.

Segundo o matéria *"5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018"* (GARTEN, 2018), da empresa que é responsável por elaborar e divulgar anualmente o *hype cicle*, o aprendizado de máquina é uma das tendências tecnológicas para o ano de 2018. Por conseguinte, utilizar um tema atual e relevante na comunidade científica, incentiva o desenvolvimento do trabalho dado a facilidade em encontrar referenciais práticos e teóricos.

Dessa forma, a utilização de novas tecnologias na solução de problemas já existentes são de muita importância, visto que os danos causados por ações maliciosas na internet afetam tanto as aplicações, quanto os usuários diretamente. Esses fatos são corroborados pelo relatório divulgado pela (NORTON, 2017), apontando o excesso de confiança dos usuários, como fator contribuinte para as ações dos hackers, gerando prejuízos que atingiram 172 bilhões de dólares, de 978 milhões de usuários em 20 países no ano de 2017.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo principal deste trabalho é a realização de uma avaliação de desempenho dos principais métodos de aprendizado de máquina supervisionados, utilizando os algoritmos Naive bayes, árvore de decisão J48 e o K-NN para os testes. Uma vez aplicados à uma base de tráfego de rede, coletado a partir de um ambiente experimental simulando um ambiente doméstico que é atacado por um usuário do Kali Linux utilizando os métodos *syn-flood*, *UDP-flood* e *ICMP-flood*.

1.3.2 Objetivos Específicos

- Utilizar uma ferramenta de código livre para realizar a captura dos pacotes correntes no roteador, possibilitando a formação da base de dados que fornecerá os dados necessários para a execução do treinamento dos algoritmos.
- Realizar experimentações utilizando os algoritmos de aprendizado de máquina abordados no trabalho, objetivando a obtenção dos dados a serem analisados posteriormente e a produção de resultados consistentes e confiáveis de cada um dos algoritmos abordados no trabalho.
- Analisar o comportamento dos algoritmos considerando a precisão e tempo de execução como principais fatores, realizando o levantamento dos dados estatísticos das simulações que servirão como métrica para as comparações futuras.
- Elaborar gráficos baseados nos dados estatísticos das simulações realizadas, comparando as características dos algoritmos abordados no trabalho, facilitando a realização de trabalhos futuros relacionados ao tema.

1.4 Metodologia

Este trabalho utiliza o método experimental, seguindo alguns passos importantes para a concretização do mesmo. Primeiramente foi realizada a definição do contexto ao qual o

trabalho se dispõe, em seguida o planejamento das tarefas que deveriam ser realizadas, após o planejamento a execução das tarefas planejadas. Após a execução das tarefas, iniciou-se a fase de análise e interpretação dos dados obtidos a partir dos testes, dos quais as informações foram retiradas e preparadas para a apresentação.

1.5 Estrutura do Trabalho

Partindo do capítulo introdutório, o capítulo 2 apresenta as referências teóricas, as quais atuaram como base para o norteamento da execução do trabalho. Em seguida, o capítulo 3 descreve a implementação do trabalho, tal como as características de hardware e software, utilizados na montagem do cenário de testes dos trabalho. O capítulo 4, apresenta os resultados quantitativos obtidos na execução dos testes, e as informações aferidas a partir desses dados. E por fim, no capítulo 5 estão, as considerações e conclusões obtidas a partir dos resultados, e também, sugestões de trabalhos futuros, sobre o tema deste trabalho.

2 REFERENCIAIS TEÓRICOS

2.1 Redes de Computadores

A realização do trabalho contempla a utilização de AM, entretanto conceitos básicos de redes de computadores são necessários para a formação da base de dados. As redes de computadores são complexas e possuem vários componentes diferentes, atuando simultaneamente para o funcionamento da mesma. Entretanto, para a formação da base de dados de entrada dos algoritmos avaliados, a fonte das informações necessárias foram abordadas por Kurose e Keith (2013) em seu livro, explanando a estrutura do cabeçalho da arquitetura TCP/IP, que contém as informações importantes para a realização dos experimentos, as quais podem ser citadas: endereço IP de origem e destino, tamanho do pacote e o protocolo utilizado.

Segundo (KUROSE; KEITH, 2013), às redes sem fio domésticas são classificadas como uma rede local LAN, usando um enlace multiponto que forma um barramento de múltiplo acesso, garantindo a comunicação entre os nós pertencentes àquela rede.. Essas redes apresentaram crescimento no número de ataques nos últimos anos, principalmente dos ataques de negação de serviço, que segundo Relatório Anual sobre Segurança da Infraestrutura Global de Redes da NETSCOUT, registaram um total de 264,9 mil ataques somente no ano de 2017.

2.1.1 Camada de Transporte

A camada de transporte é uma importante parte da arquitetura de redes na realização deste trabalho pois é nela que alguns tipos de ataques de negação de serviço estão localizados. Segundo Kurose e Keith (2013) a camada de transporte tem como principal objetivo fornecer uma conexão lógica entre processos de aplicações executando em diferentes máquinas, essa conexão lógica significa que a execução dos processos em diferentes hospedeiros ocorresse de forma direta mesmo que os processos estejam localizados em diferentes posições geográficas.

Como vemos na Figura 2, no lado remetente da conexão a camada de transporte atua convertendo as mensagens em pacotes, que são denominados na terminologia da internet de segmentos da camada de transporte. A partir desse momento a mensagem é encapsulada para as camadas inferiores até a camada de rede da arquitetura TCP/IP onde é encapsulada em um datagrama e enviada ao destinatário. Quando a mensagem é recebida pelo destinatário a camada de rede extrai o datagrama e repassa para a camada de transporte, que por sua vez disponibiliza o conteúdo da mensagem para a aplicação.

Para realizar a conexão lógica entre processos executando em diferentes hospedeiros a camada de transporte precisa utilizar os protocolos UDP e TCP, esses protocolos possuem suas peculiaridades e seus tipos de aplicações se dividido basicamente em duas formas de funcionamento desses protocolos.

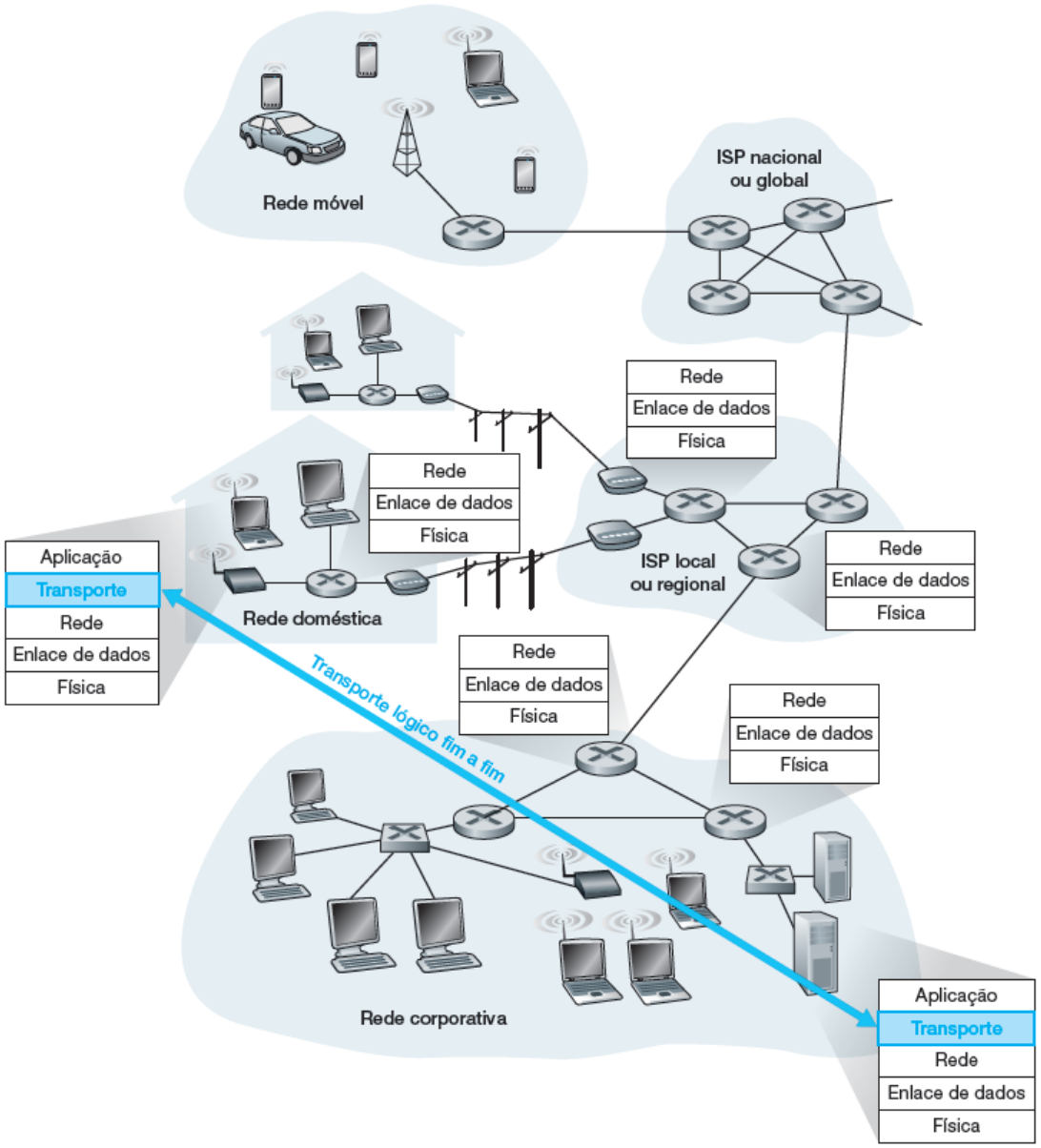

Figura 1 – Estrutura da Camada de Transporte, retirada de (KUROSE; KEITH, 2013)

O protocolo UDP é considerado um protocolo da camada de transporte não orientado para conexão pois mensagens sem nenhuma operação preliminar, essa característica permite que o protocolo UDP tenha uma complexidade menor em relação ao protocolo TCP sem oferecer garantias quanto a entrega dos pacotes. O custo da baixa complexidade do protocolo UDP é a falta de confiabilidade uma vez que não há garantia de entrega dos pacotes, entretanto é possível segurança na camada de aplicação. Por outro lado, o protocolo TCP é considerado um protocolo orientado para a conexão pois no processo de conexão entre emissor e receptor existe um pré-acordo chamado de *Three Way Handshake*, nesse conceito para estabelecer a conexão o cliente envia uma *flag Synchronize (SYN)*, o servidor responde com um pacote com as *flags* SYN + *Acknowledgement* (ACK), e na última etapa o cliente responde com um pacote ACK.

Como visto na figura 2, os protocolos da camada de transporte são usados em diversas

Aplicação	Protocolo da camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP	TCP
Acesso a terminal remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivo	FTP	TCP
Servidor de arquivo remoto	NFS	Tipicamente UDP
Recepção de multimídia	Tipicamente proprietário	UDP ou TCP
Telefonia por Internet	Tipicamente proprietário	UDP ou TCP
Gerenciamento de rede	SNMP	Tipicamente UDP
Protocolo de roteamento	RIP	Tipicamente UDP
Tradução de nome	DNS	Tipicamente UDP

Figura 2 – Principais Aplicações da Internet e Seus Respectivos Protocolos, retirado de (KUROSE; KEITH, 2013)

aplicações como correio eletrônico, web, telefonia por internet entre outros. Nas aplicações multimídia o uso do protocolo UDP é mais comum pois esse tipo de aplicação reage mal ao controle de congestionamento do TCP.

2.1.2 Rede Sem Fio

Os pontos de acesso *Wireless Fidelity* (Wi-Fi) são pontos onde os usuários podem acessar uma rede que utiliza como espectro padrão a frequência de 802.11, para a utilização de internet nessa arquitetura é necessário que a estação-base possua conexão cabeada com a mesma. A principal diferença de uma rede sem fio para uma cabeada convencional é na camada de enlace, que junto com os benefícios da utilização do meio de transmissão sem fio estão atreladas as desvantagens da utilização dessa tecnologia, e para contornar esses problemas existem algumas necessidades de tratamento dos mesmos. Entre as principais desvantagens estão a redução da força do sinal ocasionada por radiações eletromagnéticas que sofrem atenuação quando as ondas atravessam algum tipo de matéria, as interferências causadas por outras fontes de transmissão e a propagação multívias que ocorre quando parte das ondas eletromagnéticas refletem em objetos desviando do seu caminho original.

Vários elementos de *hardware* e *software* são necessários para o funcionamento da arquitetura de rede sem fio descrita na Figura 3, segundo Kurose e Keith (2013) os hospedeiros sem fio são os dispositivos finais executando as aplicações e podem ser notebooks, smartphones, desktops, tablets e etc. O enlace sem fio é a comunicação sem fio entre dois hospedeiros e pode variar seu desempenho de acordo com a tecnologia de transmissão utilizada. A estação base é fundamental na infraestrutura sendo responsável pelo envio e recebimento de dados entre vários hospedeiros os quais os dispositivos hospedeiros estão associados, e a infraestrutura de rede é a rede maior na qual o hospedeiro está conectado.

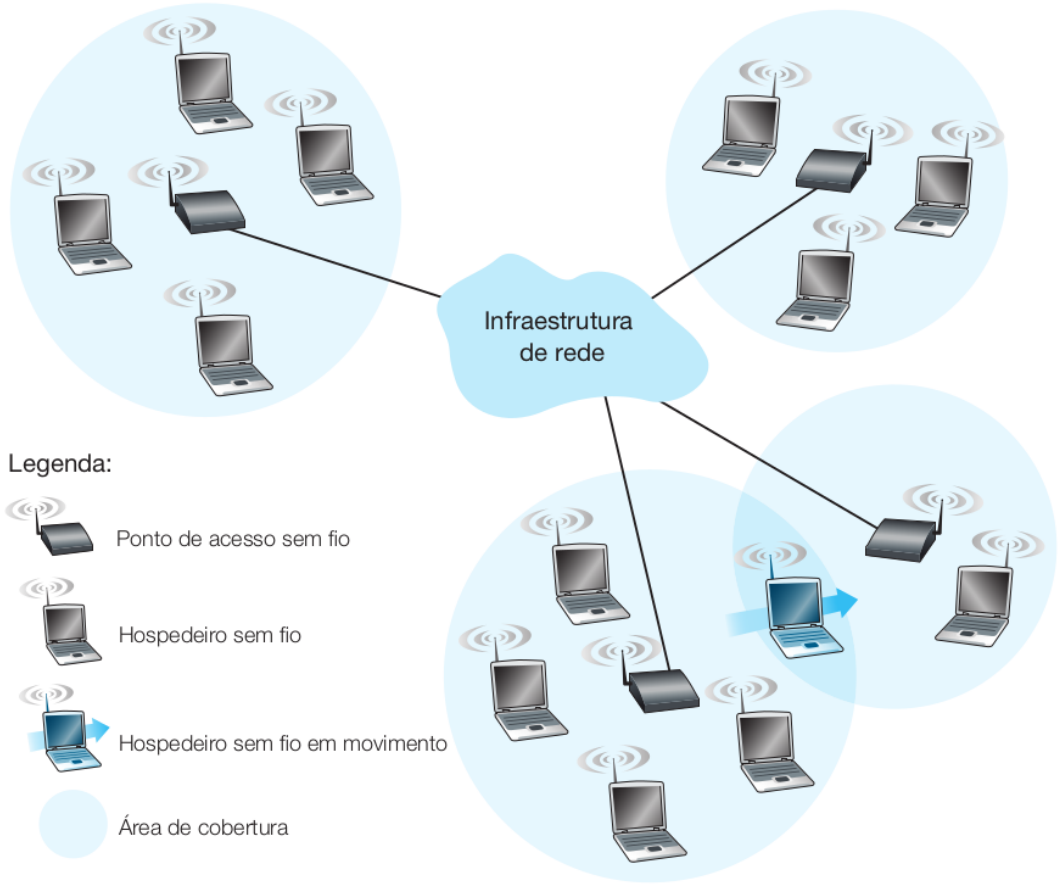

Figura 3 – Elementos de Uma Rede Sem Fio, retirado de (KUROSE; KEITH, 2013)

2.1.2.1 Arquitetura IEEE 802.11

As redes LAN estão presentes nos mais diversos lugares desde lanchonete a grandes empresas, e existem vários padrões 802.11 para a tecnologia LAN sem fio como pode ser visto na Figura 4 e todos os padrões descritos permitem a utilização do modo de infraestrutura e do modo *ad hoc*.

Padrão	Faixa de frequências (EUA)	Taxa de dados
802.11b	2,4–2,485 GHz	até 11 Mbits/s
802.11a	5,1–5,8 GHz	até 54 Mbits/s
802.11g	2,4–2,485 GHz	até 54 Mbits/s

Figura 4 – Padrões IEEE 802.11, retirado de (KUROSE; KEITH, 2013)

O conjunto básico de serviço ou BSS é formado por um conjunto de hospedeiros contendo uma estação-base central que é conhecida por todos os hospedeiros como está representado na Figura 5, estação-base essa que é conhecida com *acess point* (AP) de acordo com a terminologia definida pelo padrão 802.11 da IEEE. Segundo Kurose e Keith (2013) LANs sem fio que

disponibilizam APs recebem o nome de LANs sem fio de infraestrutura, isso significa que os APs junto com a infraestrutura ethernet cabeada responsável por interconectar os APs e um roteador.

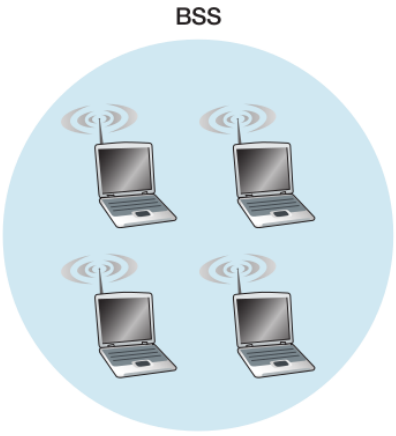

Figura 5 – Arquitetura de AD HOC IEEE 802.11, retirado de (KUROSE; KEITH, 2013)

A Figura 6 ilustra como as redes que utilizam o padrão 802.11 possuem a capacidade de agrupar e formar redes *ad hoc* que não possuem um controle central. Dessa forma, a formação das redes está diretamente relacionada a necessidade como dispositivos móveis por exemplo que quando próximos precisam se comunicar e não possuem uma infraestrutura adequada para tal. Uma rede *ad hoc* tem facilidade em sua formação, ela pode ser formada por pessoas que desejam jogar um jogo com suporte ao *multiplayer* e não possuem um roteador para fazer isso.

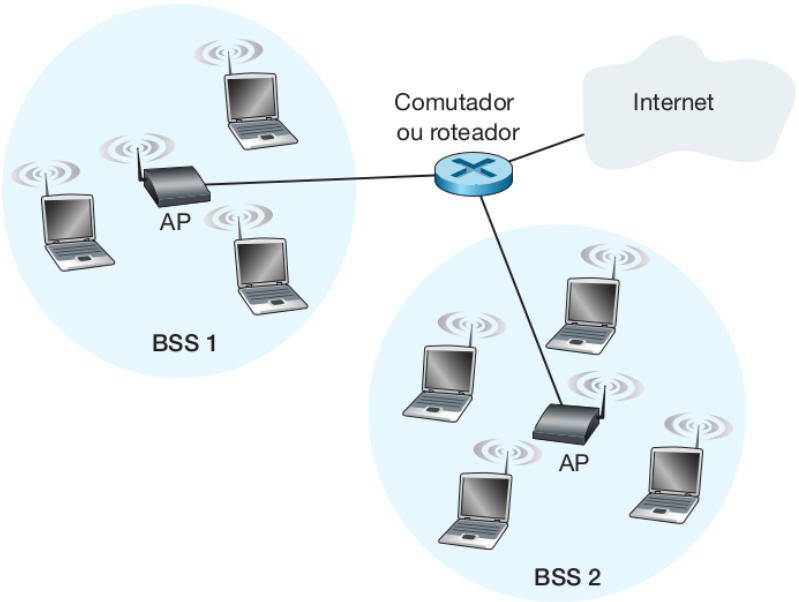

Figura 6 – Arquitetura de LAN IEEE 802.11, retirado de (KUROSE; KEITH, 2013)

2.1.2.2 Ataques em Redes de Computadores

Um ataque de negação de serviço tem o objetivo de explorar os protocolos utilizados por suas vítimas, gerando um alto tráfego de pacotes ip com a intenção de concessionar a banda ou a porta que os está recebendo, e assim deixar um recurso computacional do dispositivo atacado inacessível aos seus utilizadores. Dentre os principais métodos de prevenção desses ataques, um muito efetivo é monitorar o tráfego da rede para detectar o ataque identificando as anomalias causados por ele, classificar este tráfego e utilizar ferramentas capazes de bloquear-lo. Outra forma é utilizar *switchs* inteligentes para filtrar o tráfego. Porém, a utilização de AM pode ser uma forma de potencialização desses métodos tradicionais. O termo anomalias no tráfego refere-se ao ato da rede está se comportando de forma inesperada, sendo ou não por responsabilidade dos ataques, contudo quando a rede apresenta uma repetição de anomalias isto normalmente é resultado de um ataque.

Um exemplo de ataque de negação de serviço é o ataque *syn-flood* onde com o objetivo de gerar um alto consumo de memória, várias sessão TCP são iniciadas de forma incorreta e não são encerradas ocasionado vários ACKs responsáveis por este consumo elevado, outra forma de executar a negação de serviço é por meio do *UDP-flood* que produz uma alta quantidade de tráfego na rede, para que o atacante possa congestionar portas aleatórias do *host* que está sendo atacado, neste método os pacotes criados contém datagramas UDP que por causa da ausência do *handshake* não possuem proteção que limite sua taxa de *flood*.

Os ataques por inundação ainda podem ser melhorados quando o atacante utiliza um *cluster* que pode ser composto por vários computadores do atacante, ou como é mais comum, quando ele possui um *cluster botnet* que utiliza computadores sequestrados para auxiliar no ataque, podendo tanto ceder processamento como enviar as requisições.

2.2 Aprendizagem de máquina

O estudo das técnicas de aprendizagem de máquina é um campo derivado da ciência da computação, e consiste na capacidade de dar para as máquinas a capacidade de aprender com base nas experiências passadas, tendo como objetivo a criação de uma solução de problemas que possuem uma grande variedade de parâmetros de entrada e como consequência muitos resultados possíveis, assim necessitando que para o métodos consiga encontrar estes resultados além de ser necessário ele se adaptar a possibilidade constante de mudanças de cenários nesses problemas, ele consiga generalizar seu método para chegar a solução. Assim sendo as técnicas de aprendizagem de máquina são um dos meios para se resolver esse problema possibilitando a criação de um sistema que consegue adequar sua solução as possíveis mudanças na entrada de dados, e com o tempo aprender com os erros para aprimorar as repostas e diminuir as falhas.

Segundo a literatura (RUSSELL; NORVIG, 1995), os métodos de aprendizagem de máquina possuem uma divisão tradicional onde os algoritmos são classificados como: Aprendi-

zagem por reforço, onde por meio de ações que geram incentivos ou punições, tenta-se guiar a inteligência para a melhor resposta. Aprendizagem não-supervisionada que foca em encontrar semelhança entre os dados de entrada para poder dividi-los em grupos que com características parecidas, e por fim se tem a aprendizagem supervisionada onde os dados de entradas possuem um identificador ou rótulo que possibilita distingui-los em classes a partir dele e assim identificar as características mais frequentes de cada classe e por meio de aproximação prever a classe de dados novos que foram inseridos sem rótulo.

2.2.1 Tratamento da Base de Dados

Montar uma boa base de dados é tão importante quanto a escolha do algoritmo de aprendizagem de máquina ideal para a solução desejada, em uma base de dados de cartão de crédito por exemplo idades com campo negativos ou nulos podem gerar grandes problemas de inconsistência nos resultados dos algoritmos, por esse motivo antes de executar os algoritmos é necessário tratar os dados da base de dados retirando valores nulos, corrigindo valores errados e balanceado a base de dados.

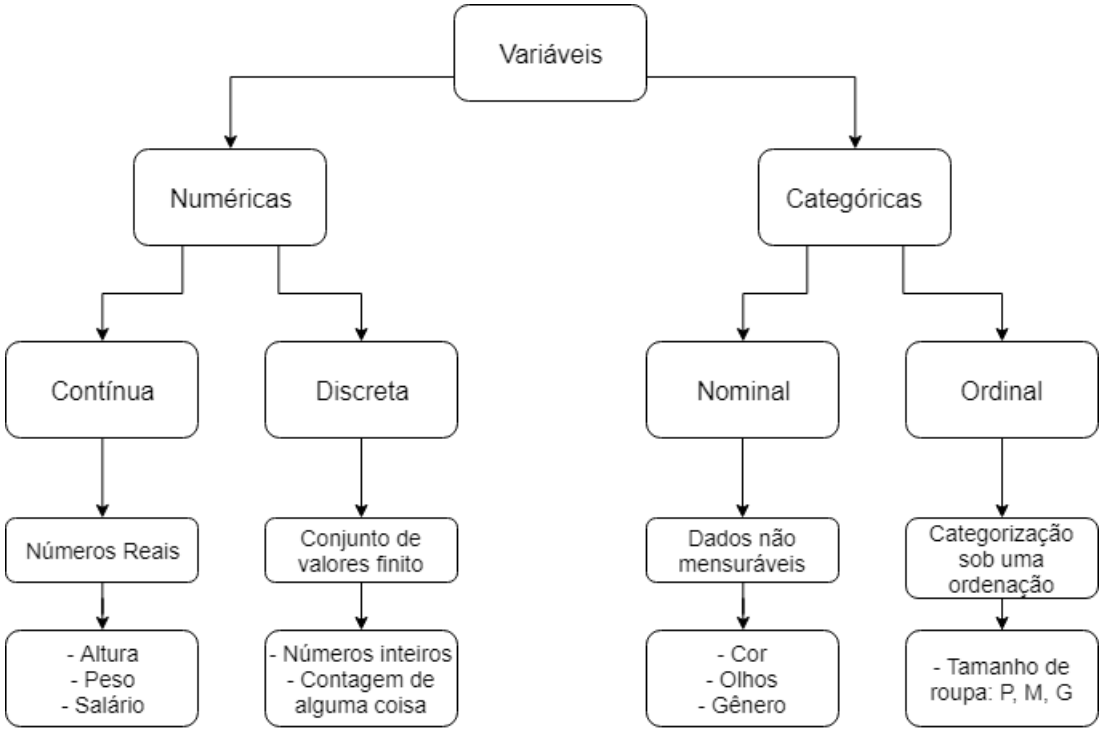

Figura 7 – Tipo das Variáveis em Uma Base de Dados, elaborada pelo autor

A Figura 7 ilustra uma árvore de informações sobre quais os tipos as variáveis de uma base de dados podem assumir para garantir uma boa aprendizagem, para cada tipo de solução desenvolvida à base de dados pode ser diferente a exemplo do supermercado onde os dados que irão alimentar o banco de dados são relacionadas as compras realizadas, em um *smartphone* o dados de uso do usuário serão utilizados para que o mesmo possa aprender com o seu dono e facilitar o seu uso.

No objeto de estudo deste trabalho os protocolos utilizados na transmissão dos pacotes são variáveis nominais pois não são mensuráveis, o tamanho e o atraso de cada pacote pode ser representado por uma variável contínua e os IPs de origem e destino podem ser classificados como variáveis nominais pois embora sejam números representam dados não mensuráveis apenas de identificação.

Após a formação da base de dados é necessário realizar o tratamento dos dados retirando valores nulos e inconsistentes e após esse dois processos o escalonamento dos dados, para tal existem duas fórmulas que podem ser utilizadas a primeira é a padronização descrita na Fórmula 2.1 e a segunda é a normalização descrita na Fórmula 2.2.

$$x = \frac{x - media(x)}{desviopadrao(x)} \tag{2.1}$$

$$x = \frac{x - minimo(x)}{maximo(x) - minimo(x)} \tag{2.2}$$

2.2.2 Métodos de Aprendizagem Supervisionada

Segundo (FACELI et al., 2011), "um algoritmo de AM preditivo é uma função que, dado um conjunto de exemplos rotulados, constrói um estimador". Esses assumem valores rótulos em um domínio conhecido, onde tem-se um problema de classificação se o domínio for um conjunto de valores nominais, ou tem-se um problema de regressão se o domínio for um conjunto infinito e ordenado de valores

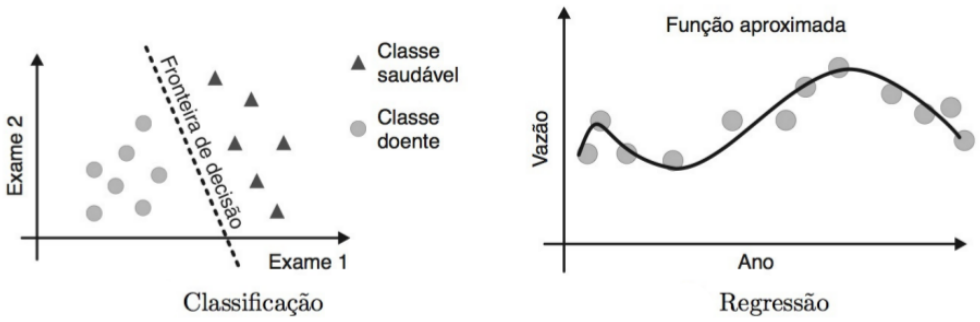

Figura 8 – Exemplo de problema abordado por algoritmos de aprendizagem de máquina, retirado de (FACELI et al., 2011)

A Figura 8 ilustra problemas de classificação e regressão. Na classificação, o objetivo é definição de uma fronteira de decisão, separando amostras da classe saudável das amostras da classe doente. Na regressão, o propósito é aprender uma função capaz de relacionar um ano a vazão de água de um dado rio, espera-se que a curva gerada a partir da função, se aproxime da curva verdadeira.

Em geral, os algoritmos supervisionados possuem uma maior precisão em comparação aos outros tipos de aprendizagem de máquina. Isto se deve em grande parte pelo fato de

eles executarem uma etapa de treinamento com dados similares aos que serão classificados, assim criando uma hipótese que os ajudará nesta classificação, como podemos ver na fala de (RUSSELL; NORVIG, 1995) "Aprendizagem e uma busca através do espaço de hipóteses possíveis por aquela que terá um bom desempenho, mesmo em novos exemplos além do conjunto de testes" os algoritmos utilizam o treinamento para gerar uma modelo dos dados, e com ele realizam a previsão do atributo identificador de cada dado da etapa de teste.

Normalmente os dados utilizados no teste não possuem identificador definido pois se está utilizando o algoritmo exatamente para descobri-lo. Porém, em um ambientes onde o objetivo é descobrir a eficiência do algoritmo, é necessário identificar a precisão da classificação gerada pela hipótese utilizada por ele. Então é fornecido uma base de dados contendo identificação para todas as suas instâncias sendo o algoritmo que a divide para treino e teste, na etapa de teste a identificação não é considerada na classificação, somente depois, quando os dados já foram classificados que a precisão da hipótese adotada pelo algoritmo é calculada, com a comparação do rótulo original de cada instância com o rótulo dado a elas.

Por causa dessa necessidade de conhecer os dados previamente, a aprendizagem de máquina supervisionada torna-se ideal para aplicações em que já se tenha informações de como os dados se comportam ou deve se comportar. A exemplo disto se tem o trabalho de (SILVA et al., 2012) que utiliza métodos supervisionados para a classificação de leveduras. Nas áreas da agricultura já se tem o modelo ideal ou aceitável para os dados da levedura, sendo assim uma boa aplicação para a aprendizagem supervisionada. Outra boa aplicação é a apresentada em (HENKE et al., 2015) onde é feita a detecção de Spam utilizando a métodos supervisionados, sendo efetiva por os spams seguirem quase sempre um modelo de mensagem para atrair a atenção da vítima.

2.2.2.1 Naive Bayes

O teorema de Bayes é uma forma de lidar com tarefas preditivas em AM, em especial quando as informações não estão bem definidas, calculando a probabilidade de um evento A ocorrer dado um a ocorrência de outro evento B. No Livro (FACELI et al., 2011) é afirmado que o Naive Bayes é mais restritivo, por não permitir a dependências entre atributos, e os modelos gráficos probabilístico são usados para os dados de entrada, desde previsões até o diagnóstico. A partir da fórmula da probabilidade condicional:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2.3}$$

obtém-se o teorema de bayes ao se considerada a independência entre os parâmetros e estender a fórmula para mais eventos se obtém a versão ingênua, ou naive, do teorema de bayes:

$$P(A|E1,...,En) = \frac{P(E1|A) * P(E2|A) * ... * P(En|A) * P(A)}{P(E1,...,En)} \tag{2.4}$$

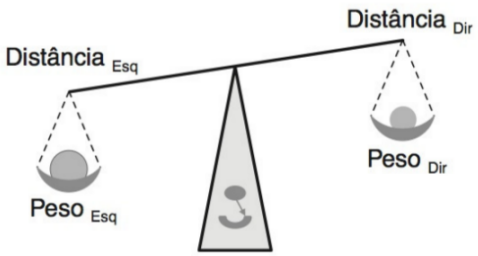

Figura 9 – O problema da balança, Imagem retirada de (FACELI et al., 2011)

A Figura 9 ilustra um exemplo de problema simples resolvido com o teorema de Bayes, no problema da balança a classe que será prevista é a posição da balança, podendo tender para direita, esquerda ou ficar no centro. Os atributos que ajudaram nesta identificação são os pesos dos objetos do lado esquerdo e do direito e as dimensões destes dois objetos, sendo utilizado a etapa de treinamento para identificar a relação destas informações com o estado da balança, assim gerando um modelo para a classificação.

A premissa do Naive Bayes é exatamente o que faz com que ele carrega o título de naive, porém o fato de ele ignorar as dependências entre suas variáveis o permite efetuar sua execução mais rapidamente que os outros algoritmos. Além de que para gerar a classificação ele não depende de um grande número de dados, assim o tornando uma boa escolha para implementações que necessitem resposta rápida com poucos dados como as aplicações de tempo real. Um exemplo de aplicação esta presente em (ALMEIDA; YAMAKAMI, 2011) onde é realizada a Redução de dimensionalidade aplicada na classificação de spams usando naive bayes.

2.2.2.2 K-NN

O algoritmo *k-nearest neighbors (K-NN)* ou K-Vizinhos mais Próximos trata-se de um dos métodos mais simples de aprendizagem de máquina supervisionada. Nele os dados de entrada são separados na base de dados de treino do algoritmo e base de dados de teste, assim sendo utilizados na fase de teste para identificar entre os dados de treino os pontos mais próximos do dado de entrada que está sendo testado e assim determinar a sua classe. Neste método, a classe se trata do rótulo que aquela instância possui, nos dados de treino ela fica visível para o algoritmo, e com isso é possível catalogar os atributos das instância de cada classe.

Já na etapa de teste, o algoritmo não possui os rótulos dos dados que ele está testando, para que se possa identificá-los, são avaliadas instância por instância de teste com todas as instâncias de treino calculando a distância entre elas utilizando seus atributos, sendo as mais aplicadas a distância euclidiana e a de Manhattan. Depois de calculada esta distância, é criada uma lista contendo as instâncias de treino próximas da instância de teste, para assim identificar a classe que é mais frequente entre estas instâncias escolhendo ela como a classe para a instância de teste. A quantidade de vizinhos que é considerada para encontrar a classe média é determinada pelo valor k do algoritmo que pode ser inserido pelo usuário, contudo é recomendável utilizar

números ímpares para diminuir a chance de empates.

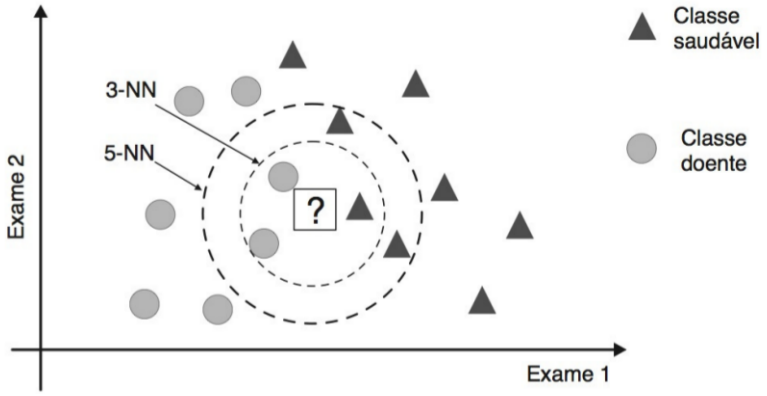

Figura 10 – Exemplo de classificação usando o algoritmo K-NN, retirado de (FACELI et al., 2011)

A Figura 10 descreve o funcionamento do algoritmo com um problema de duas classes, onde os indivíduos podem ser classificados como saudável ou doente. A classe saudável é representada por um triângulo, e o doente pelo círculo. A partir do espaço definido pelos atributos e usando a distância euclidiana, é calculado os objetos mais próximos e o objeto mais próximo do espaço é definido com doente. Também é possível identificar o funcionamento da classificação onde se o número de vizinhos for considerado 3 a instância é classificada como doente, porém se for considerado k igual a 5 ela e classificada como saudável.

2.2.3 Árvore de Decisão

Segundo (RUSSELL; NORVIG, 1995) “Uma árvore de decisão alcança sua decisão executando uma sequência de testes”, em resumo durante a execução deste método será utilizada a árvore onde os dados de entrada farão seu percurso. Em cada nó da arvoré são feitos testes com os dados, para avaliar a decisão a ser tomada e assim escolher o ramo seguinte conforme a resposta, repetindo isso desde a raiz até uma das folhas onde é retornado a classificação da entrada que está sendo avaliada.

Observando a Figura 11, pode-se ver um exemplo de árvore de decisão que representa a escolha de esperar ou não por uma mesa, para que se possa fazer os testes presentes nos nós e assim encontra a melhor decisão os dados de entrada precisam ter as informações de quantidade de pessoas no restaurante, lotação do restaurante, tempo de espera, clima entre outros, estas informações seriam os atributos que ajudaram a classificar aquela instância. Para a criação da árvore são utilizados os dados de treinamento, que a partir de seus atributos, geram as decisões que serão tomadas em cada um dos nós e assim fazem a montagem da árvore. Porém, para que o resultado seja satisfatório, também é necessário que a árvore gerada tenha boa acurácia e boa taxa de acerto. Como é ineficiente obter todas as árvores possíveis para testar suas acurácia muitos algoritmos utilizam outra maneira de geri-la. O algoritmo utilizado para teste foi o j48

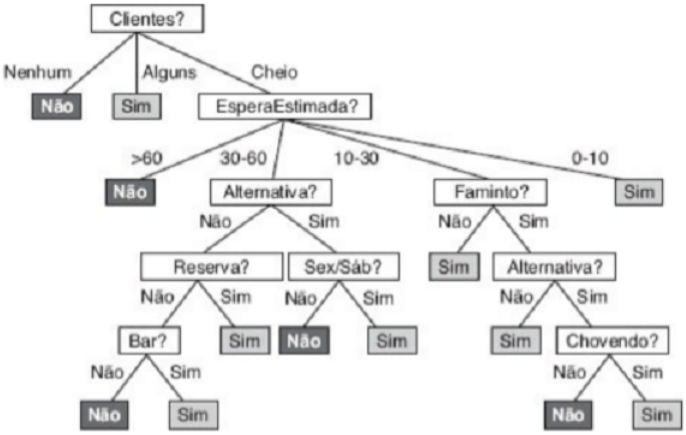

Figura 11 – Exemplo de classificação usando árvore de decisão retirada de (RUSSELL; NORVIG, 1995)

através da ferramenta (WEKA, 2018) que utiliza a heurística de procurar, para o estado em que a árvore se encontra, os testes que melhor dividem os dados.

2.2.4 Matriz de confusão

Segundo (FACELI et al., 2011) essa matriz ilustra o número de predições corretas e incorretas em cada classe. Para um determinado conjunto de dados, as linhas dessa matriz representa as classes verdadeiras e as colunas, as classes preditas pelo classificador. Sendo assim cada célula da matriz que possua linha e coluna com o mesmo identificador representa um acerto do algoritmo, já as outras representam falhas, assim sendo uma boa forma de representar o desempenho da classificação.

Tabela 1 – Exemplo de matriz de confusão

Classe original ↓	Exemplo	
Classe identificada →	Ataque	Normal
Ataque	Verdadeiro positivo	Falso negativo
Normal	Falso positivo	Verdadeiro negativo

Como vemos na Tabela 1, cada um desses casos possui uma nomenclatura específica, no primeiro caso uma instância que é originalmente da classe que queremos identificar foi classificada corretamente, elas são chamadas de verdadeiro positivo, e no caso de uma instância de outra classe ser classificada corretamente, elas são chamadas de verdadeiros negativos. Já um instância da classe que queremos identificar que não foi classificada corretamente é chamada de falso negativo, e em contrapartida instâncias de outras classes que foram classificada de forma errada são falso positivos.

2.3 Trabalhos Relacionados

Como trabalhos relacionados serão utilizados 4 artigos, dentre eles estão (DOSHI; APTHORPE; FEAMSTER, 2018), e (KAKIHATA et al., 2017). Onde são aplicados métodos de aprendizagem de máquina supervisionados para a detecção de possíveis ataques em redes de computadores por meio da previsão de classes de tráfego de ataque. Os outros artigos utilizados são (LIMA; PINTO, 2016) e (HAMADA; NETO, 2015) que também apresentam previsões com aprendizagem supervisionada, porém em aplicações diferentes diferentes, sendo usados para identificar *spam* em textos.

Tabela 2 – Trabalhos Relacionados

Artigos	Implementação	Algoritmos	Base de Dados
(DOSHI et al, 2018)	Scikit-learn Python library	5	491.855
(LIMA; PINTO, 2016)	WEKA	4	2.400 e 811
(Kakihata et al, 2017)	Java	4	4.414.541
(HAMADA; NETO, 2015)	WEKA	3	120 mil frases

Como se pode ver na Tabela 2, dois dos artigos utilizaram o aplicativo WEKA para implementar os algoritmos, sendo eles os artigos que avaliaram a aplicação em texto. Já os artigos de detecção de ataques utilizaram implementação próprias nas linguagens Python e Java. Sendo assim, em comparação a esses artigos, estes será o primeiro a avaliar a utilização do WEKA para implementar algoritmos supervisionados na detecção de ataques em redes.

Na Tabela 2 também é possível identificar o número de algoritmos avaliados em cada artigo, sendo a média de 4 algoritmos sendo que este artigo avalia apenas 3, porém foram utilizados o K-NN e a árvore de decisão que são os algoritmos com uso mais frequente. Em contrapartida não foi utilizados o Método SVM que também foi bastante utilizado no artigos avaliados.

Por fim ao se comparar a base de dados utilizadas pelos artigos, percebe-se que a base de dados deste artigo está na média delas tendo por volta de 500.000 instâncias quantidade similar a utilizada no artigo (DOSHI; APTHORPE; FEAMSTER, 2018), que também avalia a mesma aplicação de detecção de ataques, utilizando algoritmos similares, divergindo apenas no método de implementação que foi usado.

3 ALGORITMOS SUPERVISIONADOS PARA A DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO

3.1 Algoritmos Escolhidos

Para se realizar os experimentos foram escolhidos três algoritmos de aprendizagem de máquina supervisionada. O primeiro deles foi o Naive Bayes que conforme foi apresentado no capítulo 2 possui um tempo de execução bem pequeno e não necessita de uma base de dados muito grande para obter bons resultados dos teste, sendo assim uma boa escolha para aplicações em tempo real.

Outro algoritmo escolhido foi o de árvore de decisão, que acaba sendo uma boa forma de identificar como os dados se dividem e além de que também possui boa precisão, para a montagem da árvore foi escolhido o J48 pois segundo (LIMA; PINTO, 2016) seu método de procura a melhor divisão dos dados para o estado atual da árvore é um dos mais eficientes.

O último algoritmo escolhido foi o K-NN que além de ser um dos algoritmos de aprendizagem de máquina mais utilizados, também possui uma boa precisão e por ele considerar todos os campos da base de dados com a mesma importância durante o cálculo da distância no seus testes.

3.2 Descrição dos Experimentos

Diante das diversas dificuldades apresentadas na realização da coleta de dados em uma residência convencional, para a realização deste trabalho foi configurado um ambiente de rede local contendo acesso à internet, tendo ela seu uso exclusivo na realização da coleta de dados, e sendo caracterizada como uma rede isolada. O objetivo dessa configuração era simular as características de ambiente de rede sem fio doméstico, no que se refere aos dispositivos conectados e o tráfego analisado. A montagem do ambiente de teste foi baseada na utilização de softwares livres nas etapas de: coleta, preparação dos dados, treinamento dos algoritmos e elaboração gráfica dos resultados.

Para a realização dos ataques foi necessário controlar o ambiente de rede sem fio doméstico simulado mantendo o cliente e o servidor na mesma rede interna, essa restrição é consequência do comportamento dos ataques que foi responsável pela interrupção da coleta algumas vezes no decorrer dos experimentos, o roteador utilizado na coleta diversas vezes teve sua memória estourada necessitando reiniciar o roteador para restabelecer seu funcionamento normal, esse e outros possíveis problemas gerados a partir desses ataques o ambiente foi limitado a uma rede local oferecendo um nível menor de riscos ao normal funcionamento da rede interna da Universidade Federal do Pará.

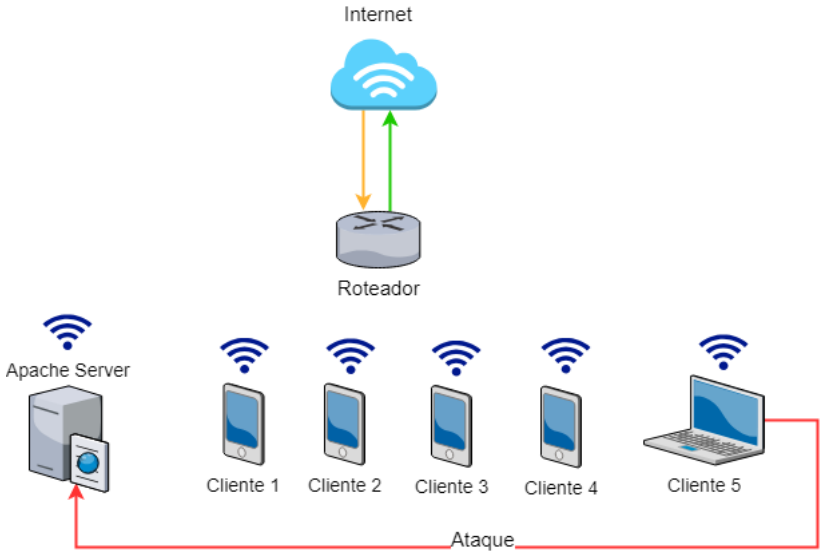

Figura 12 – Arquitetura de rede utilizada para a coleta de dados

A Figura 12 representa a arquitetura de rede utilizada na coleta de dados. O principal tráfego gerado a partir dos *smartphones* foi o *streaming* de vídeo em plataformas como famosas como Netflix e YouTube, que de acordo com o Mobility Report (ERICSSON, 2018), continua a crescer, impulsionado principalmente pelo aumento no tempo de visualização, popularização de serviços de *streaming*, além da migração para resoluções cada vez maiores. Além desses serviços, a utilização de redes sociais e o download de APPs foram considerados na execução dos testes. A Tabela 3, descreve a função e uma breve descrição dos dispositivos utilizados na geração e captura de tráfego, os ataques foram gerados a partir do cliente 5, como descrito na Figura 12. O Apache Server agiu como receptor dos ataques, e o roteador ficou responsável por gerenciar a rede local e realizar a captura dos dados.

Tabela 3 – Componentes da Arquitetura do Ambiente Experimental

Quantidade	Função	Descrição
1	Atacado	Notebook rodando o Apache Server
1	Atacante	Notebook rodando Kali Linux
4	Gerador de Tráfego	Smartphone Android
1	Gerenciar e Coletar Tráfego	Roteador TP link WR1043ND rodando OpenWRT

3.3 Coleta de Dados

Para efetuar a análise dos algoritmos de aprendizagem supervisionados, serão utilizados os dados coletados do roteador através do OpenWrt. Como dito as coletas se dividiram em três etapas começando com a coleta de tráfego normal, onde o objetivo era ter o modelo dos dados com a rede sendo normalmente usada sem nenhum ataque sendo feita a ela. A segunda etapa

foi a coleta de tráfego anômalo onde se teve o objetivo de identificar o comportamento da rede quando ela está sendo atacada, foram utilizados 3 tipos diferentes de ataques, sendo eles *ACK Flood, UDP Flood e TCP SYN Flood*, para aumentar a eficiência da coleta nesta etapa apenas o atacante estava conectado a rede. E na última etapa foi efetuada a coleta de tráfego geral da rede onde ao mesmo tempo os dispositivos do experimento estavam gerando seu tráfego de informações e esporadicamente a rede sofria ataques. Porém, dentre estes dados alguns não eram muito úteis considerando a aplicação de tentar identificar as possíveis anomalias no tráfego.

Para auxiliar na geração do tráfego anômalo foi utilizada a distribuição Kali Linux que segundo Kali (2018) é um projeto de código aberto que oferece ferramentas nativas de testes de penetração, entre as funções oferecidas pelo sistema podem ser citadas: *Pentest, SQL Inject, Exploits, Sniffers, Scanner, Cracking*, invasão de redes sem fio, sites e banco de dados, bem como quebra de senhas. Com a utilização dessa ferramenta é possível testar os mecanismos de segurança de Tecnologia da Informação (TI) e aperfeiçoar os mesmos.

Como podemos ver na Tabela 4 as principais métricas necessária para a identificação de tráfego malicioso que estão presentes na captura são Tamanho, Protocolo, IP Destino e IP Origem, e os outros dados que foram coletados necessitam ser avaliadas para que sejam mantidos ou eliminados.

Tabela 4 – Parte dos dados coletados com openwrt com o campo info retirado e com as bases de dados anormais e normais mescladas.

Número	Tempo	IP Origem	IP Destino	Protocolo	Tamanho	*Classe*
1	0.000000	192.168.1.1	192.168.1.100	HTTP	71	anormal
2	0.000252	192.168.1.100	192.168.1.1	TCP	66	anormal
3	0.021199	Tp-LinkT_a4 : 69 : *ba*	Broadcast	ARP	42	anormal
4	0.453152	192.168.1.100	192.168.1.1	HTTP	549	anormal
5	0.453304	192.168.1.1	192.168.1.100	TCP	66	anormal
6	0.475450	Tp-LinkT_a4 : 69 : *ba*	Broadcast	ARP	42	anormal
7	0.602305	192.168.1.1	192.168.1.100	TCP	135	anormal
8	0.602464	192.168.1.100	192.168.1.1	TCP	66	anormal
9	0.602584	192.168.1.1	192.168.1.100	TCP	188	anormal
10	0.602700	192.168.1.100	192.168.1.1	TCP	66	anormal

Fonte: Captura realizada diretamente no roteador por meio de OpenWrt (2018)

Como pode ser visto na Tabela 4 algumas informações coletadas podem não ser necessárias, por isso o primeiro passo do processamentos dos dados para que eles possam ser utilizados é a filtragem das informações que não ajudam nesta avaliação dos dados. A primeira informação a ser desconsideradas é a do campo info que consistem em geral por informações sobre as mensagem de ACK como “42186 > 443 [ACK] Seq=199 Ack=501 Win=1444 Len=0 TSval=5748881 TSecr=1748359812”, ou os comandos de controle como “Who has 192.168.1.188? Tell 192.168.1.1”, sendo assim um campo de texto grande e com conteúdo bem variado que pode

atrapalhar no funcionamento dos algoritmos. Outro campo capturado foi a numeração, pois ela tem como objetivo ser apenas um identificador dos pacotes conforme suas ordem de chegada, e o tempo de chegada de cada pacote em relação à coleta que está sendo realizada, inclusive isso foi um dos critérios que faz com que o campo tempo seja também um dos escolhidos para ser eliminado, pois por ser referente ao início da captura pode complicar o objetivo futuro do projeto, que é de fazer a identificação de tráfego em tempo real, pois como a coleta de dados será constante o tempo acumulado acaba crescendo demais, além de que esta informação nesse contexto não é muito importante para influenciar na decisão do algoritmo. Dessa forma, foi optado por utilizar o tempo de intervalo entre a chegada dos pacotes, que poder ser usado para identificar quando uma rajadas de pacotes está tendo comportamento anormal sendo aplicável em tempo real, e podia ser calculado com os dados que já tinham sidos obtidos. Além de que nesse estado ela é uma informação importante para auxiliar o algoritmo na sua análise.

Além da filtragem de campos foi necessário a criação deles para que fosse possível a aplicação da aprendizagem de máquina, visto que os dados ainda não possuem uma classe de rótulo que os identifique, o que é essencial na fase de treinamento dos métodos supervisionados, além de ser necessário no ambiente de experimento para se obter a acurácia e precisão dos algoritmos utilizados. Pensando nisso a coleta de dados foi dividida entre tráfego normal e tráfego anômalo, foi gerado duas base de dados sendo a primeira apenas com dados de tráfego normal, e a segunda com grande parte se seu tráfego sendo anormal, a partir delas foi possível gerar uma nova base de dados mesclando randomicamente as outras duas, esse novo conjunto possuía o campo classe responsável por categorizar as informações como normais ou anormais.

3.4 Métricas avaliadas

Como descreve (KUROSE; KEITH, 2013) em seu livro, existem vários campos no cabeçalho dos pacotes que trafegam na internet, entretanto, para a realização deste trabalho, alguns desses campos foram escolhidos como dados de entrada para os algoritmos de AM. Segundo (DOSHI; APTHORPE; FEAMSTER, 2018), os campos escolhidos para análise nesse trabalho, estão entre as 10 principais características de pacotes anômalos no tráfego de redes domésticas com dispositivos IoT.

Tabela 5 – Características dos Pacotes Escolhidas para Análise

Ordem	Características
1	Tamanho do Pacote
2	Protocolo
3	Intervalo de Chegada dos Pacotes
4	Endereço IP de Origem
5	Endereço IP de Destino

Baseado no artigo de (DOSHI; APTHORPE; FEAMSTER, 2018), e nas opções ofereci-

das pela ferramenta de coleta (OPENWRT, 2018), a Tabela 5 descreve os dados escolhidos para a formação da base de dados para treinamento, ordenados por seu devido grau de importância na realização do experimento.

3.5 Configuração de Hardware e Software dos Dispositivos Utilizados nos Testes

Considerando o grande volume de dados, foi utilizada uma máquina virtual contendo os softwares necessários para as execuções, dessa forma, o ambiente de teste concebido para analisar o comportamento dos algoritmos de AM, continha os vários componentes de hardware e software, onde os principais estão listados na Tabela 6:

Tabela 6 – Principais Componentes de Hardware e Software

Número	Tipo	Descrição
1	Hardware	16 GB de Memória RAM
2	Hardware	CPU 8 núcleos
3	Software	Sistema Operacional Debian versão 9 - 64 bits
4	Software	Weka Versão 3.6.14

A ferramenta WEKA, possui uma limitação pré programada de fábrica no uso de memória RAM, esse limite é de apenas 247.5 MB. Dessa forma, uma alteração no arquivo de configuração foi realizada, extendendo esse limite para 10 GB viabilizando os experimentos.

3.6 Experimento 1:

Como primeiro passo dos testes foi utilizado o WEKA para a retirada das métricas que foram coletadas mas que serviriam para a implementação avaliada. Em seguida foram aplicados filtros para adequar as variáveis da instancias aos requisitos dos algoritmos, como o K-NN que necessita de variáveis numéricas, para contornar isso nas instâncias de texto foram aplicados filtros de binarização chamado *NominalToBinary*.

O algoritmo Naive Bayes foi o primeiro a ser executado por se tratar de um algoritmo clássico e por que segundo (HUANG, 2003), tende a ter acurácia similar a outros algoritmos supervisionados como o de árvores de decisão, porém sua aplicação possui o objetivo apenas de estipular um resultado aceitável para os outros testes, pois como se sabe seu resultado tende a ser mais impreciso por causa de sua heurística de considerar todos os atributos independentes entre si que nem sempre pode ser verdade. Seguindo esta ideia o Naive Bayes foi o primeiro a ser testados para avaliar para quê nos seguintes testes esta comparação já possa ocorrer.

Neste primeiro experimento os algoritmos foram utilizados com a técnica de *Percentage Split* para dividir a base de dados entre treino e teste, nesta técnica certa porcentagem é separada de forma aleatória entre os dados de entrada para etapa de de treino do algoritmo, essa técnica tende a ser a mais rápida pois é necessário efetuar o treino somente uma vez. Como se pode ver na Figura 13, a base de dado possui 596916 instancias isso resulta em um total de aproximadamente 417841 instancias para treino e 179075 para teste.

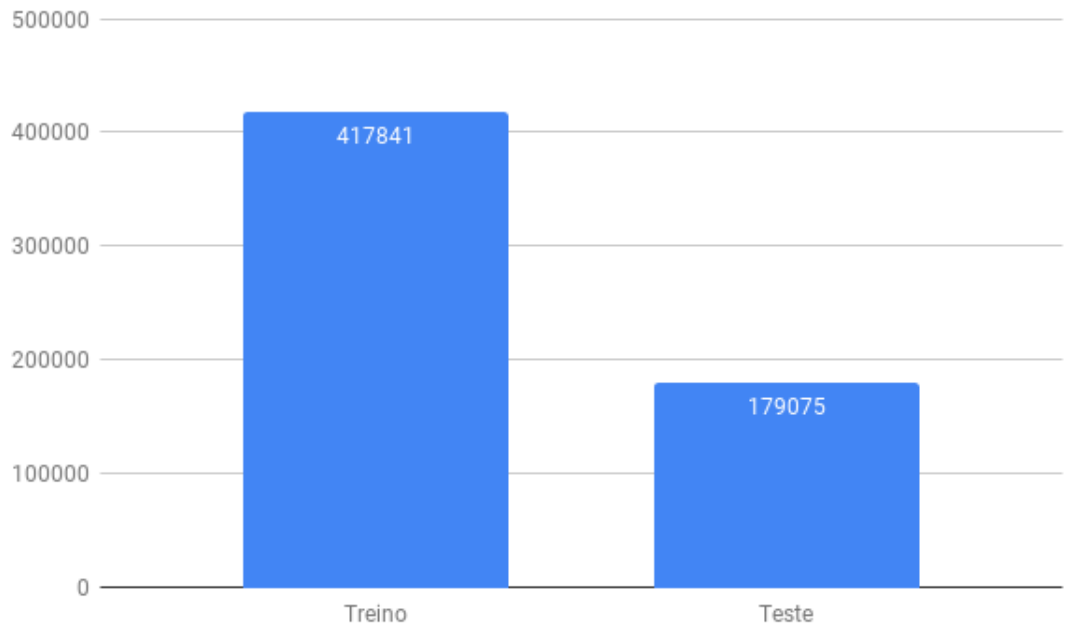

Figura 13 – Grafico com a divisão da base de dados para cada classe em número de instâncias.

Ao se analisar o campo classe foi nota que as instâncias são divididas entre 443645 de tráfego anormal e 153271 que são referentes ao tráfego normal, quando se compara estas informações se percebe que com a técnica de split há o risco de que em alguns teste os dados de teste que acabam sendo escolhidos sejam em grande parte de tráfego anômalo o que gera dificuldade em na identificação dos dados normais. Essa grande diferença entres as instância normais e anormais se deve ao fato de o ataque gerar um alto tráfego de pacotes na rede.

3.7 Experimento 2:

Após a implementação dos algoritmos utilizando Percentage Split para o treino, com o objetivo de contornar o possível problema gerado pela aplicação desta técnica os teste foram refeitos porém utilizando o método de validação cruzada(*cross-validation*) onde o conjunto dos dados é dividido em n partições(*folds*) de tamanho igual, sendo uma delas utilizada para o teste e as restantes para o treino. Porém o resultado encontrado nessa execução é considerado parcial, pois este método tem o objetivo de executar o algoritmo repetidas vezes considerando em cada uma delas um partição diferente como dados de testes, o resultado final vêm da média tirada

das acurácias dessa execuções. Por meio desta técnica todos os dados são usados tanto para treino como para teste o que permite um resultados mais apurado e preciso, porém ela tende a demandar mais tempo de execução por causa das suas várias iterações

Durante a implementação dos algoritmos deste experimento foram utilizadas 10 partições, que normalmente é o padrão utilizado na validação cruzada pelo weka, o que gerou uma divisão de aproximadamente 59692 instâncias para cada uma delas. No decorrer dos dois experimentos, no algoritmo K-NN, foi adotado K valendo 3 pois junto do 1 se trata dos valore padrões adotados pelo WEKA, porém foi optado pelo 3 com o objetivo de expandir as opções de comparações e assim identificar mais facilmente os ataques além de ele ser ímpar e diminuir a chance de empates, também foram utilizadas as distâncias euclidiana e de manhattan para que se possa também comparar seus resultados, contidos nos dois teste os resultados foram iguais.

4 RESULTADOS

Neste capítulo, estão detalhados os resultados obtidos a partir da elaboração e execução dos testes, de acordo com a metodologia definida no capítulo 1.

4.1 Análise de Precisão

Após a execução dos experimentos, Conforme ilustrado na Figura 14, os três métodos utilizados obtiveram resultados acima dos 99% de precisão nos dois experimentos, algo que rebate o possível problema apresentados no texto de que poderia haver classificação tendenciosa no experimento 1 pois ele utilizou a técnica *percentage split*, e os dados possuíam uma carga maior de instâncias de ataque do que as normais. Porém os resultados similares do experimento 2, que utilizou a validação, demonstram não houve problemas nestes testes. Também é importante salientar que alimentar a base de dados com uma grande quantidade de dados variados para treinamento tende a aumentar a precisão dos algoritmos, visto que eles iram possuir mais modelos para a classificação.

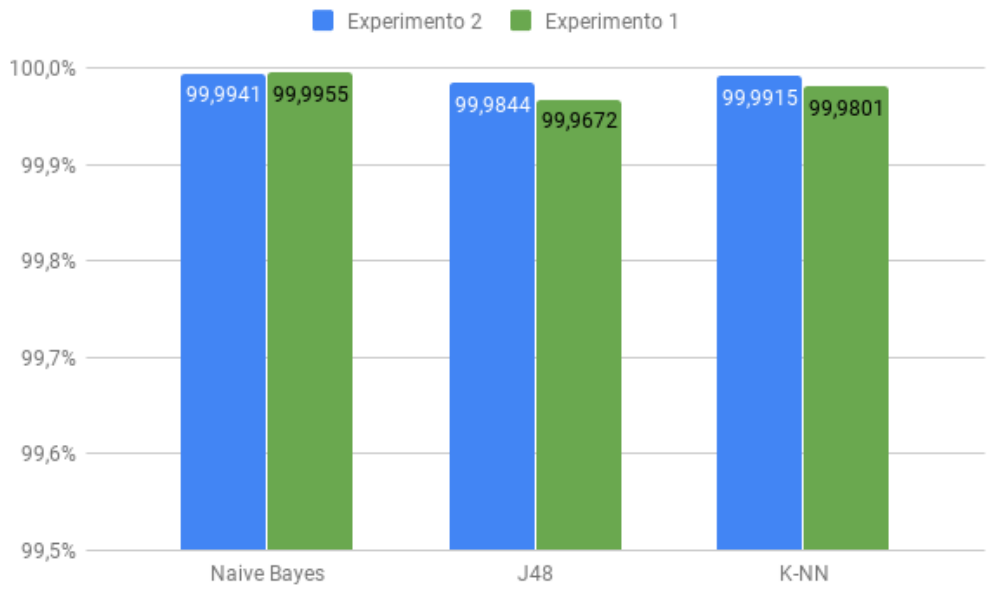

Figura 14 – Gráfico comparativo da precisão dos algoritmos testados

O naive bayes teve o melhor resultados em consequência de sua premissa de considerar os atributos independentes e porque o atacante gerava ips de origem aleatórios, que normalmente são ip fora da rede, como durante a coleta dos dados de tráfego normal todos os usuários estavam na conectados na rede local do roteador com OpenWrt isto foi uma forma de facilitar a identificação feita por ele. Porém se o atacante conseguir utilizar um ip da rede local isso acaba sendo mascarado, para que haja um ideia foi realizado um novo teste com o naive bayes sem se considerar o campo ip de origem, isso gerou uma avaliação com foco nos protocolos dos pacotes

e no tempo de intervalo da chegada deles, possuindo um resultado com precisão de 99.9338 inferior ao obtido anteriormente porém tendo uma avaliação mais segura.

Já a árvore de decisão usou como principal atributo para classificar o tráfego o ip de destino, sendo da mesmo forma que o naive bayes, uma escolha que pode acarretar em resultados não satisfatórios dependendo de como o ataque atuar. Por meio dos logs da execução foi identificado que o alvo dos ataques durante a coleta era somente um dos clientes do roteador, sendo ele o que estava executando o servidor apache. Isso acabou gerando assim um alto tráfego de pacotes com o mesmo destino, e o algoritmo percebendo isto, utilizou o campo ip destino para a classificação. Porém em ambientes não controlados é preciso estar preparado para as adversidades que podem surgir, com este objetivo os teste deste algoritmo foram refeitos sem que o ip de destino seja considerado, resultando em 99.9159 de precisão com validação cruzada e 99.9268 com percentage split.

Por fim, o Knn teve os atributos numéricos como seu diferencial, considerando o tamanho dos pacotes e o intervalo de tempo de chegada dos pacotes como seu diferencial na classificação, pois apesar de os ataque gerarem tamanhos de pacotes aleatórios eles também ocasionaram um espaço pequeno entre a chegada dos pacotes, por causa de seu objetivo de gerar um alto fluxo na rede para sobrecarregá-la. Por isso não foi necessário novos testes para este algoritmo.

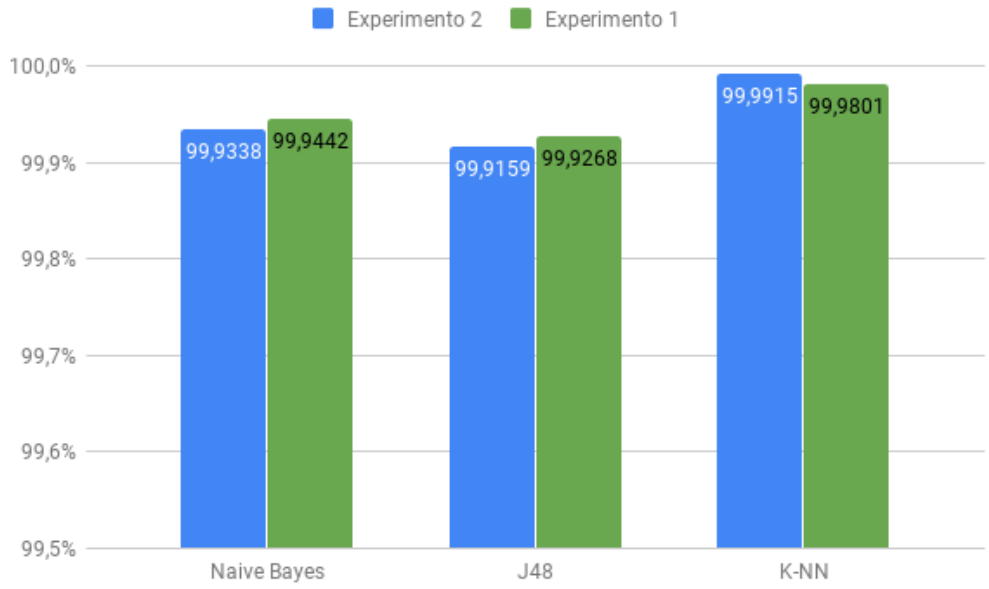

Figura 15 – Gráfico comparativo da precisão dos algoritmos depois que foram refeitos dos teste

Como se pode vê na Figura 15, assim que os teste foram refeitos para melhorar a resposta dos algoritmos em situações adversas. Os algoritmos que foram retestados tiveram uma leve diminuição nas suas precisões, o que destacou o K-NN, que continuou com seu resultado elevado e utiliza uma boa forma de identificar o tráfego anômalo, ao considerar todas os campos da base de dados, porém com foco no tempo de intervalo dos pacotes, que acaba revelando

uma característica dos ataque de negação de serviço. Como já foi citado, estes ataques geram intervalos pequenos de pacotes com o objetivo de sobrecarregar a rede.

4.2 Análise da Matriz Confusão

Nas Tabelas 7 e 8 pode-se ver a matriz de confusão gerada pelos algoritmos de pois de seus retestes, onde é apresentada a relação entre a classe original da instância avaliada com a classe que o algoritmo identificou para ela. Com as informações contidas nas Tabelas 7 e 8, é possível verificar os falsos positivos, tráfego de normal que foi classificado como ataque, sendo estes os de menor relevância, pois mesmo durante o ataque ainda são gerado pacotes de tráfego normal o que faz com que alguns deles tenham atributos similares aos pacotes dos ataques. Contudo também são apresentados os falsos negativos, onde uma instância é classificada como normal porém ela pertence ao ataque, estes erros são mais preocupantes pois aqui os algoritmos não foram capazes de identificar o ataque assim não sendo possível efetuar as contramedidas necessárias para evitá-los.

Tabela 7 – Matriz de confusão dos algoritmos no experimento 1

Classe original	Naive Bayes		J48		K-NN	
Classe identificada–>	Ataque	Normal	Ataque	Normal	Ataque	Normal
Ataque	74.197%	0.0039%	74.187%	0.014%	74.187%	0.0073%
Normal	0.0519%	25.747%	0.059%	25.74%	0.0123%	25.841%

Tabela 8 – Matriz de confusão dos algoritmos no experimento 2

Classe original	Naive Bayes		J48		K-NN	
Classe identificada–>	Ataque	Normal	Ataque	Normal	Ataque	Normal
Ataque	74.317%	0.0059%	74.311%	0.0119%	74.318%	0.0025%
Normal	0.0603%	25.617%	0.072%	25.605%	0.008%	25.674%

Como se pode ver, os algoritmos Naive Bayes e K-NN obtiveram uma taxa baixa de falsos negativos nos dois experimentos, tendo o J48 se destacado nos experimentos, errando mais que o dobrou em comparação aos outros, mesmo tendo um desempenho taxa de erro baixa seu desempenho neste quesito foi bastantes inferior aos outros. Em relação aos falsos positivos, o desempenhos dos algoritmos foi inferior, porém eles continuaram com um índice de erros baixo, desta vez quem se destaca é o K-NN obtendo resultados melhores que os outros dois algoritmos nos dois experimentos, principalmente no 2, onde houve uma melhora na identificação dos falsos positivos pelo K-NN e o Naive Bayes e o J48 pioraram seu desempenho quando comparado ao do experimento 1.

4.3 Análise do Tempo de Execução

Conforme se pode ver na Tabela 9 abaixo, o tempo de execução dos algoritmos tende a variar de acordo com qual método de dividir os dados foi escolhido, indo do mais sutil como o naive bayes, até o mais brusco onde no k-NN o tempo de execução do algoritmo com o percentage split de 70 por cento foi menos da metade do tempo da execução com validação cruzada. A discrepância do K-NN em comparação aos outros vem da forma como seu treinamento é executado, somado a alta quantidade de atributos de texto, sendo grande parte do tempo para treinamento. Isto acaba sendo um impedimento para o uso do K-NN, pois é necessário que a classificação do tráfego seja feita em tempo real para que se possa identificar os ataque o quanto antes para evitar suas consequências, e é nesse quesito que os outros algoritmos superam o K-NN, que apesar de possuir uma melhor precisão e melhor identificação dos ataques, perde aqui. Sendo o Naive Bayes o que se sai melhor para o objetivo classificação em tempo real por seu tempo de execução pequeno e por não haver necessidade de uma base de dados muito grande para classificação.

Tabela 9 – Tempo de execução dos algoritmos, em segundos

Algoritmos	Experimento 1	Experimento 2
Naive Bayes	0,45s	0,46s
J48	3,15s	4,16s
K-NN	80,55s	182,14s

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho de conclusão de curso teve como objetivo realizar uma avaliação de algoritmos de AM supervisionados, na identificação de tráfego anômalo em redes sem fio domésticas, utilizando-se da metodologia experimental para obter resultados coerentes. Foi utilizado software WEKA como ferramenta de estudo do comportamento desses métodos, dado uma base de dados de entrada coletada para esta avaliação. Observou-se uma alta precisão dos algoritmos testados, consequência dos algoritmos escolhidos e da base de dados utilizada que possuía muitos exemplos de instâncias de tráfego anômalos, que por sua vez foi o objeto de interesse na realização dos experimentos. As principais informações de resultados utilizadas para a comparação foram a precisão e o tempo de execução, pois esses dados, normalmente, são os mais usados por outros artigos as considerando muito importantes, pois além de possibilitar o ranqueamento dos concorrentes, pode influenciar na escolha de qual usar, de acordo com a relação de custo e poder de processamento.

A importância desta avaliação está na em futuras implementações em roteadores modificados com um hardware mais potente, servindo este trabalho como referencial na escolha do algoritmo capaz de atender demandas, cumprindo requisitos de custo e tempo de execução para cada tipo de implementação. Por exemplo, a implementação de um módulo automático de classificação em um roteador, possui limitações de hardware para execução de tarefas complexas, por outro lado, caso esse módulo esteja executando um um servidor, o mesmo será capaz de executar algoritmos mais complexos. Além disso, a realização de estudos em AM contribui com a popularização e facilitação de trabalhos futuros abordando o tema, resultando em soluções cada vez mais completas para a resolução de problemas da humanidade.

Para finalizar, partindo dos das informações produzidas para este trabalho, é possível o grande potencial de AM como ferramenta de segurança em redes domésticas, que possibilitam novas linhas de pesquisa sobre o mesmo tema, as quais podem ser citadas:

- A utilização de outros algoritmos de aprendizagem de máquina supervisionada como o de máquina de vetores avaliado em (LIMA; PINTO, 2016) e (HUANG, 2003), outra opção são os algoritmos de redes neurais também utilizado em (LIMA; PINTO, 2016). Para que possa ser realizada uma nova comparação mais completa e conclusiva.

- Utilizar Change Points Detection(CPD) para identificar mudanças no bruscas no tempo de chegada dos pacotes. Utilizando CPD, é possível identificar pontos de mudanças no intervalo de chegada dos pacotes, dessa forma a análise só seria feita nesses pacotes, característicos de ataques de negação de Serviço.

- Utilizar diferentes ambientes de redes na realização dos experimentos. Dessa forma, será possível analisar a existência ou não de mudanças no desempenho dos algoritmos e na diferença entre eles.

REFERÊNCIAS

ALMEIDA, T. A.; YAMAKAMI, A. Redução de dimensionalidade aplicada na classificação de spams usando filtros bayesianos. **Revista Brasileira de Computação Aplicada**, v. 3, n. 1, p. 16–29, 2011.

DOSHI, R.; APTHORPE, N.; FEAMSTER, N. Machine learning ddos detection for consumer internet of things devices. 2018. Disponível em: <https://arxiv.org/pdf/1804.04159.pdf>. Acesso em: 20 nov. 2018.

ERICSSON. **Ericsson Mobility Report November 2018**. 2018. Disponível em: <https://www.ericsson.com/en/mobility-report/reports/november-2018>. Acesso em: 25 nov. 2018.

FACELI, K. et al. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora Ltda, 2011. 378 p. Bibliografia: p. 343–373. ISBN 978-85-1880-5.

GARTEN. **5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018**. 2018. Disponível em: <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>. Acesso em: 06 nov. 2018.

HAMADA, L.; NETO, N. Desambiguação de homógrafos–heterófonos por aprendizado de máquina em português brasileiro. 2015. Disponível em: <http://www.aclweb.org/anthology/W15-5622>. Acesso em: 20 nov. 2018.

HENKE, M. et al. Deteção de spam baseada na evolução das características com presença de concept drift. Universidade Federal do Amazonas, 2015.

HUANG, J. e. a. Comparing naive bayes, decision trees, and svm with auc and accuracy. 2003. Disponível em: <https://pdfs.semanticscholar.org/8a73/74b98a9d94b8c01e996e72340f86a4327869.pdf>. Acesso em: 20 nov. 2003.

KAKIHATA, E. M. et al. Intrusion detection system based on flows using machine learning algorithms. **IEEE Latin America Transactions**, IEEE, v. 15, n. 10, p. 1988–1993, 2017.

KALI. **About Kali Linux**. 2018. Disponível em: <https://www.kali.org/about-us/>. Acesso em: 28 nov. 2018.

KEMP, S. **The State of the internet in Q4 2018**. 2018. Disponível em: <https://wearesocial.com/blog/2018/10/the-state-of-the-internet-in-q4-2018>. Acesso em: 13 nov. 2018.

KUROSE, F.; KEITH, W. **Redes de Computadores e a Internet**. São Paulo: Pearson Education do Brasil Ltda, 2013. 634 p. Bibliografia: p. 580–606. ISBN 978-85-430-1443-2.

LIMA, A.; PINTO, R. Estudo experimental de aprendizado de máquina para desenvolvimento de um classificador de texto de incidentes de grandes eventos. 2016. Disponível em: <http://bdm.unb.br/bitstream/10483/17162/1/2016_AndreLima_RenatoCarlosPinto_tcc.pdf>. Acesso em: 20 nov. 2018.

OPENWRT. **Open Wrt: Wireless Freedom**. 2018. Disponível em: <https://openwrt.org/>. Acesso em: 19 nov. 2018.

PETTEY, C. **Gartner Top 6 Security and Risk Management Trends For 2018**. 2018. Disponível em: <https://www.gartner.com/smarterwithgartner/gartner-top-5-security-and-risk-management-trends/>. Acesso em: 19 nov. 2018.

RUSSELL, S.; NORVIG, P. **Artificial intelligence : a modern approach**. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc, 1995. 932 p. Bibliografia: p. 859–932. ISBN 0-13-103805-2.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210 – 229, 1959. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5392560>. Acesso em: 10 nov. 2018.

SILVA, D. S. da et al. Classificação de leveduras utilizando transformada de hough e aprendizagem supervisionada. In: **Workshop de Visão Computacional**. [S.l.: s.n.], 2012.

WEKA. **Weka 3: Data Mining Software in Java**. 2018. Disponível em: <https://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: 20 nov. 2018.