UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JULIANO SILVA DO NASCIMENTO

COMPARAÇÃO DOS ALGORITMOS RANDOM FOREST, RANDOM TREE E J48 PARA DETECTAR ATAQUES DDoS

PONTA GROSSA

2022

JULIANO SILVA DO NASCIMENTO

COMPARAÇÃO DOS ALGORITMOS RANDOM FOREST, RANDOM TREE E J48 PARA DETECTAR ATAQUES DDoS

Comparison of random forest, random tree and j48 algorithms to detect ddos attacks

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Augusto Foronda.

PONTA GROSSA

2022

JULIANO SILVA DO NASCIMENTO

COMPARAÇÃO DOS ALGORITMOS RANDOM FOREST, RANDOM TREE E J48 PARA DETECTAR ATAQUES DDoS

Trabalho de Conclusão de Curso apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 31/outubro/2022

_____

Augusto Foronda
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Luiz Rafael Schmitke
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Vinícius Camargo Andrade
Mestrado
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2022

“Dedico este trabalho, primeiramente, a Deus, pois sem ele não teria forças para me manter firme. A toda minha família e amigos pelos momentos em que estive ausente. A minha querida e tão amada avó Ana Maria Leite da Silva (in memoriam), exemplo de amor e bondade, cujo sua presença foi de extrema importância em minha vida, pois tudo que é eterno e verdadeiro se eterniza em nossos corações. Também dedico ao meu pai Francisco Ribeiro do Nascimento (in memoriam), meu exemplo de vida por todo amor para com nossa família.”

AGRADECIMENTOS

Certamente Palavras não serão suficientes para expressar minha eterna gratidão primeiramente a Deus, e a todos os familiares, amigos e professores que através de palavras e presença me incentivaram a não desistir e acreditar que mesmo após uma longa tempestade sempre há um novo dia repleto de alegrias e realizações, pois sem o apoio de vocês não teria chegado até aqui. Estendo esses agradecimentos a família Lahoud e a família Schwab que de amigos se tornaram minha segunda família e esteve sempre se preocupando comigo.

Agradeço em especial a minha família que esteve sempre ao meu lado dando forças para seguir mesmo quando eu acreditava que nada mais fazia sentido, pois quem esteve ao meu lado sabe o quanto o ano de 2019 foi difícil para mim por 2 percas que sofri, primeiramente com o falecimento de meu pai no dia 12/02/2019 e depois quando estava se recuperando dessa perca tão sofrida e dolorosa veio mais uma perca que foi o falecimento de minha avó ocorrido no dia 10 de Outubro do mesmo ano, o que deixou meu coração totalmente em pedaços. Somente quem esteve por perto pode perceber o quanto tudo isso deixou-me e ainda assim deixa-me abalado, pois foram percas de pessoas mais que importante em minha vida.

Mas acredito que mesmo após as cicatrizes deixadas á sempre a esperança de que um dia nos reencontraremos, e tenho certeza que sempre estiveram e estão ao meu lado me segurando e intercedendo junto de Deus por mim e por toda minha família. E que daí de cima vó a senhora possa ver que todo o esforço que fez para me manter aqui em Ponta Grossa não foi em vão, pois não sei o que seria de mim sem a senhora ao meu lado sempre me aconselhando e puxando minha orelha quando precisava.

Obrigado família e em especial a minha mãe Célia Cristina Silva do Nascimento por sempre acreditar em mim e por todo o respaldo financeiro que me deu nessa longa jornada, que Deus possa recompensá-los infinitamente.

Ao meu orientador Prof. Dr. Augusto Foronda meu muito obrigado por toda a dedicação na orientação para elaboração deste trabalho, e por não desistir de mim mesmo quando pelos motivos citados acima estive ausente, mas que ainda assim estava sempre disposto a me ajudar e a colaborar com as minhas ideias dando sugestões para enriquecer ainda mais o desenvolvimento deste trabalho.

Termino os agradecimentos de forma geral com uma releitura do padre Fabio de Melo, uma de minhas maiores inspirações para a vida.

Não há aprendizado na vida, que não passe pela experiência dos erros. Que possamos vivenciar a metáfora da vida de nosso 1º caderno, pois quando os erros cometidos eram demais, nossa professora nos sugeria a virar a página e recomeçar.

Recomece, mas de forma diferente, pois a partir desses erros seguimos mais maduros. Erros não precisam ser fontes de castigo, erros podem ser fontes de virtudes.

Deus é semelhante ao caderno, ele nos permite os erros para que aprendamos a fazer do jeito certo.

“Mesmo quando tudo parece desabar cabe a mim
decidir entre rir ou chorar, ir ou ficar, desistir ou
lutar, porque descobri no caminho incerto da vida,
que o mais importante é o decidir.”
(Cora Carolina).

RESUMO

Ataque distribuído de negação de serviço DDoS *(Distributed Denial of Service)* é um tipo de ataque no qual um cyber criminoso gera uma grande quantidade de solicitações de requisições para um servidor, até que o servidor fique sobrecarregado por conta de tantas solicitações, fazendo com que seus recursos diminuam, trave ou cause lentidão na rede. Existem algumas técnicas e ferramentas para identificar este tipo de ataque, como a análise baseada em assinatura da ferramenta de detecção de intrusão SNORT. Mas o problema desta técnica é que não identifica novos ataques, somente ataques conhecidos. Para resolver este problema, tem surgido novas técnicas baseadas em inteligência artificial que utiliza o processo de aprendizagem para conseguir identificar e classificar por meio de padrões conhecidos aqueles tráfegos que são propícios a serem maliciosos, comparados a um tráfego normal. Este trabalho analisa os algoritmos de classificação Random Forest, Random Tree e J48 para verificar sua eficácia para identificar novos ataques do tipo DDoS. São feitas simulações com diferentes porcentagens de tamanho para teste e treino para verificar qual apresenta melhor resultado.

Palavras-chave: Inteligência Artificial; DDoS; Redes de Computadores.

ABSTRACT

Distributed Denial of Service (DDoS) attack is a type of attack in which a cyber criminal generates a large number of requests for requests to a server, until the server is overwhelmed by so many requests, causing your resources slow down, crash, or slow down your network. There are some techniques and tools to identify this type of attack, such as the signature-based analysis of the SNORT intrusion detection tool. But the problem with this technique is that it does not identify new attacks, only known attacks. To solve this problem, new techniques based on artificial intelligence have emerged that use the learning process to be able to identify and classify through known patterns those traffics that are prone to be malicious, compared to normal traffic. This work analyzes the Random Forest, Random Tree and J48 classification algorithms to verify their effectiveness to identify new DDoS attacks. Simulations are made with different percentages of size for testing and training to verify which one presents the best result.

Keywords: Artificial Intelligence; DDoS; Computer Network.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
IA	Inteligência Artificial
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
RNA	Rede Neural Artificial
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
VPN	*Virtual Private Network*

LISTA DE ACRÔNIMOS

SERPRO	Serviço Federal de Processamento de Dados
ISO	*International Organization for Standardization*
OSI	*Open Systems de Interconnection*
DOD	*Department of Defense*
HTTP	*Hypertext Transfer Protocol*

SUMÁRIO

1 INTRODUÇÃO

Redes de Computadores é uma das áreas da computação mais dinâmicas em função do crescimento nas últimas décadas, tanto em tamanho, quanto em complexidade e que apresenta uma série de problemas de segurança, como o ataque DDoS (*Distributed Denial of Service*). É um tipo de ataque em que o cyber criminoso gera uma quantidade gigantesca de requisições a um servidor, de forma com que o mesmo não consiga suportar a sobrecarga e fique inacessível, sem poder atender a novas requisições de visitantes reais (MEDINA, 2004).

Uma das maneiras de detectar este tipo de ataque é por meio de um IDS *(Intrusion Detection System)* e uma das ferramentas de IDS é o SNORT. É um software livre capaz de realizar análise de tráfego em tempo real e registro de pacotes. Este mapeamento ocorre por meio da camada de transporte coletando os dados TCP *(Transmission Control Protocol)*, UDP *(User Datagram Protocol)* e ICMP *(Internet Control Message Protocol)*. Atualmente é um dos softwares mais utilizados para analisar possíveis ataques de intrusão na rede. A maior empresa pública de prestação de serviços em tecnologia da informação do Brasil SERPRO (Serviço Federal de Processamento de Dados) utiliza essa ferramenta desde o ano de 2003 (SERPRO, 2008).

É sabido que a quantidade de dados decorrentes da atividade de monitoração de uma rede é muito alta. O tipo de processamento clássico de um IDS, baseado em assinatura, pode tirar a visibilidade de alterações de estado de algum tráfego que, se analisadas em um contexto mais "inteligente" teriam condições de fornecer indícios de tendências de comportamento, indicando um ataque DDoS. Ou seja, uma ferramenta de IDS não detecta novos ataques, somente ataques conhecidos (TAROUCO, 1990).

A análise mais "inteligente" pode ser potencializada com a aplicação da Inteligência Artificial (IA). Inteligência Artificial é o estudo dos sistemas que agem de um modo que um observador qualquer pareça ser inteligente, onde utiliza métodos baseados no comportamento intelectual de humanos e outros animais para solucionar problemas complexos (COPPIN, 2010). Um algoritmo de IA permite a qualquer computador digital duplicar algumas funções do cérebro humano de uma forma limitada (TAROUCO, 1990).

O algoritmo Random Tree é baseado na extração de regras dos dados que serão analisados para que com isso possa gerar a árvore de decisão daquele modelo treinado e testado. O algoritmo Random Forest é o sucessor do Random Tree, pois considera diversas arvores de decisão geradas para poder fazer uma parametrização e com isso obter um modelo que melhor se adeque em relação as árvores geradas tornando a Random Forest mais eficiente em relação a sua antecessora. O algoritmo J48 é baseado em árvore de decisão, mas com um conjunto de regras menor se comparado com o algoritmo random tree.

1.1 Objetivos

Esta seção descreve o objetivo geral e os específicos, respectivamente nas seções 1.1.1 e 1.1.2.

1.1.1 Objetivo geral

Comparar os algoritmos Random Forest, Random Tree e J48 para detector ataques DDoS.

1.1.2 Objetivos específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Analisar os dados de um ataque DDoS coletados do dataset CICIDS2017;
- Analisar os Algoritmos Random Forest, Random Tree e J48;
- Simular os algoritmos para detectar novos ataques DDoS;
- Comparar os resultados obtidos com os três algoritmos.

2 REFERENCIAL TEÓRICO

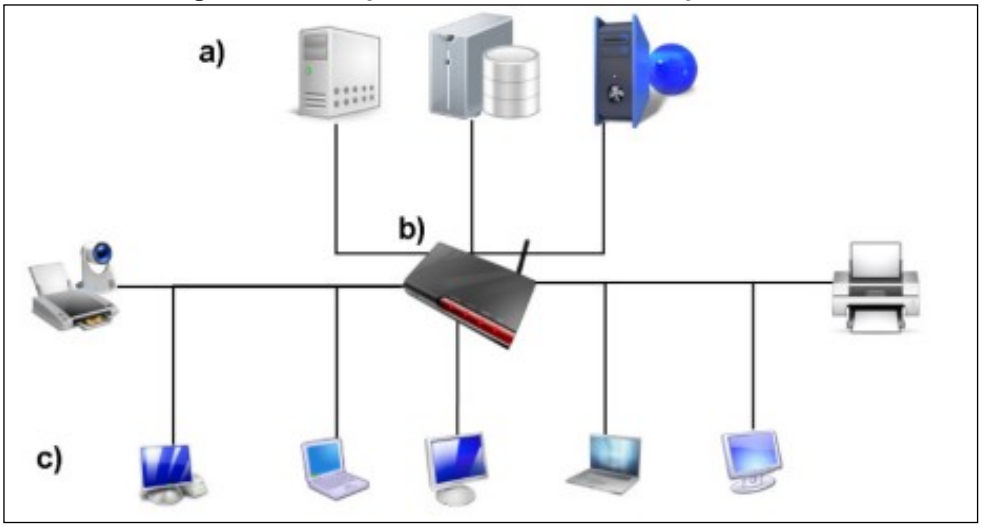
Neste Capítulo serão apresentados conceitos de teoria de redes de computadores, modelo TCP/IP e conceitos fundamentais de segurança, DDoS, IA e os algoritmos Random Forest, Random Tree e J48.

2.1 Redes de computadores

Rede de Computadores é um conjunto de dispositivos finais, como computadores, celulares, servidores e impressoras que são conectados por intermédio de equipamentos como switches e roteadores, para que possam trocar ou até mesmo compartilharem dados entre si, como mostrado na Figura 1. Para que essa comunicação ocorra são utilizados protocolos de rede, estes por sua vez possuem as regras para fazer esta comunicação entre os dispositivos interligados (MACEDO et al. 2018).

Na Figura 1 é possível ver a ilustração de uma rede de computadores.

Figura 1 - Exemplo de uma Rede de Computadores



Fonte: MACEDO et al (2018)

Essa rede é composta por três principais dispositivos sendo eles, A, B e C. Os dispositivos localizados na parte superior com a letra A representam os servidores, por sua vez esses servidores possuem grande capacidade de armazenamento em relação aos demais, pois são responsáveis pelo fornecimento de serviços para essa rede. O dispositivo central representado pela letra B diz respeito ao roteador, onde o mesmo realiza a interconexão desses dispositivos com a rede baseados nas regras para interligá-los. Na parte inferior da figura

representado pela letra C estão os dispositivos denominados como clientes ou usuários, estes por sua vez tem capacidade de armazenamento inferior ao servidor.

Com isso está montada uma rede de computadores, em que os usuários solicitam o recebimento ou envio de informações no qual essa solicitação é controlada pelo roteador, pois ele é a ponte de conexão entre os clientes e servidores.

2.2 Modelo TCP/IP

Os modelos de referência para comunicação surgiram por conta de incompatibilidade de equipamentos. Um exemplo a ser citado é a placa de redes que só poderia estar conectada a uma outra placa do mesmo fabricante, que por sua vez é conectado através de um meio físico que fosse compatível com o seu respectivo modelo de fabricante (MENDES, 2007).

Por volta de 1970 a ISO *(International Standards Organization)* criou o modelo de referência OSI *(Open Systems Interconnection)* para resolver a incompatibilidade entre os fabricantes, garantindo com isso que os dispositivos pudessem se comunicar independentemente de seus fabricantes, pois com isso era possível aos fabricantes criarem protocolos e componentes a partir desse modelo.

O modelo OSI é composto por sete camadas com a intenção de garantir a padronização de comunicação entre os dispositivos e equipamentos, como mostrado na Figura 2.

Figura 2 - Modelo referência arquitetura OSI

Fonte: TANENBAUM (2011)

A camada física é responsável pela transmissão de bits, garantindo integridade nos recebimentos de dados e está diretamente ligada ao hardware.

Também é responsável pelos sinais elétricos que serão utilizados para transmissão de bits 1 e 0, tempo de duração de cada bit e especificação o tipo da transmissão se ela será half-duplex (envia e recebe os dados alternadamente nos dois sentidos) ou full-duplex (com ambos os lados enviando e recebendo simultaneamente) (TANENBAUM, 2011).

A camada de enlace de dados faz a interface confiável entre o meio físico e os dados do computador, detectando erros e fazendo o controle de fluxo. Pode-se dizer que esta camada é responsável por realizar a transformação de um canal de transmissão normal para uma linha que seja livre de erros de transmissão (TANENBAUM, 2011).

A camada de rede realiza o direcionamento de pacotes entre diferentes redes e é responsável por determinar o modo em que os pacotes são roteados da origem ao destino, sendo mapeados por meio de rotas estáticas ou dinâmicas (TANENBAUM, 2011).

A camada de transporte faz o controle de fluxo dos dados entre o emissor e o receptor. Basicamente essa camada aceita os dados da camada de sessão e faz a fragmentação desses dados e os repassa a camada de rede garantindo assim que todos os dados fragmentados cheguem corretamente a outra extremidade.

A camada de sessão é responsável pelo gerenciamento de comunicação, pois permite que os usuários em diferentes máquinas se comuniquem e com isso estabelece um controle deste trafego garantindo que apenas um usuário por vez transmita ou execute determinada operação para evitar o conflito. Caso ocorra uma falha nesta comunicação a conversa pode ser reestabelecida do ponto em que parou (TANENBAUM, 2011).

A camada de apresentação realiza a conversão dos dados para um formato padrão, está relacionada á sintaxe e semântica das informações transmitidas. Esse processo garante que duas redes distintas se comuniquem e suas estruturas de dados que serão trocadas poderão ser definidas de maneira abstrata (TANENBAUM, 2011).

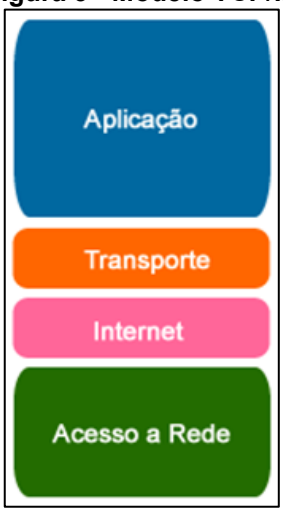
A camada de aplicação converte os dados de uma mensagem de e-mail em bits, anexando um cabeçalho para poder identificar o computador emissor e o receptor, sua principal função é: (determinar como acontecerá o diálogo, fazer a identificação de endereços e nomes e controlar o acesso e a integridade dos dados). O protocolo de aplicação HTTP *(HyperText Transfer Protocol)* é utilizado na maioria

das vezes, pois constitui a base da World Wide Web, quando um navegador deseja uma página web, ele envia o nome da página desejada para o servidor que hospeda a página utilizando o HTTP, com isso o servidor transmite a página de volta. Existem outros protocolos de aplicação que são utilizados para transferência de arquivos, correio eletrônico e transmissão de notícias pela rede (TANENBAUM, 2011).

Por volta da década de 1980 o Departamento de Defesa dos Estados Unidos DOD *(Department of Defense)* desenvolveu o modelo de referência TCP/IP. A ARPANET que era uma rede de pesquisa no qual foi patrocinada pelo DOD teve um papel importante nessa criação, pois elaborou esta rede que na época era experimental com o intuito de conectar computadores de longa distância para trocarem informações entre sí, sejam por compartilhamento de dados ou até mesmo por trocas de mensagens via e-mail (TANENBAUM, 2011).

Este modelo é composto por 4 camadas como pode ser observado na Figura 3.

Figura 3 - Modelo TCP/IP



Fonte: TANENBAUM (2011)

O modelo TCP/IP não tem as camadas de Apresentação e Sessão em relação ao modelo OSI e as camadas de enlace e física se unificaram gerando uma única camada denominada camada de Acesso a Rede. A camada de internet apresenta as mesmas características apresentadas no modelo OSI, assim como a camada de transporte. Por fim tem-se a camada de Aplicação que contém todos os protocolos de nível mais alto.

- TELNET: protocolo de terminal virtual utilizado para conexão remota entre duas máquinas;

- FTP: protocolo de transferência de arquivos que são um conjunto de regras de dispositivos para realizar a transferência de arquivos entre esses dispositivos;
- SMTP: protocolo de correio eletrônico utilizado para envio e recebimento de e-mail;
- DNS: (Domain Name Service) serviço que faz o mapeamento dos nomes para seus respectivos endereços de rede;
- HTTP: protocolo utilizado para busca de páginas;
- RTP: protocolo que realiza a entrega da mídia em tempo real (voz ou vídeo) (TANENBAUM, 2011).

2.3 Conceitos de segurança

Segurança é um termo utilizado em diversas áreas, onde cada área específica tem um significado próprio, mas em geral é um termo utilizado para conjunto de regras e medidas que visam à proteção de riscos, perigos ou perdas a pessoas ou coisas. Em outras palavras pode-se dizer que é um conjunto de ações ou recursos utilizados para proteção (PRIBERAM, 2022).

Na área de redes de computadores segurança são quaisquer mecanismos de defesas que são projetados para impedir o acesso indevido aos dados, garantindo a integridade de informações e fazendo o gerenciamento da rede para evitar que as ameaças entrem ou se espalhem na rede. Por sua vez, a segurança de rede é composta por um conjunto de camadas de defesa, onde cada camada é responsável por determinadas políticas e controles para que apenas usuários autorizados possam ter acesso aos dados e impedindo que os agentes possam acessar esses dados ou atacarem a rede (CISCO, 2022).

Existem diversos tipos de segurança como *firewalls*, segurança de e-mails, softwares de antivírus, controles de acesso, análise de comportamento, sistemas de prevenção contra invasões, prevenções contra perda de dados, *VPNs*, segurança web e sem fio dentre outros.

2.3.1 SNORT

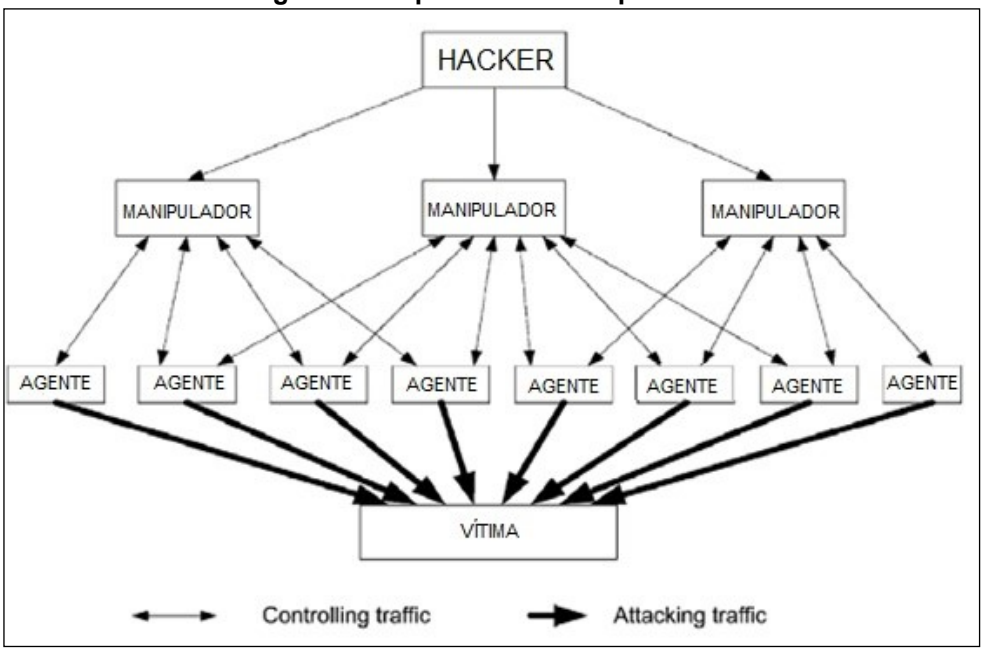
A ferramenta SNORT é um IDS *(Intrusion Detection System)*, capaz de identificar ataques na rede onde procura identificar anomalias na rede por meio de assinatura. O SNORT é um software livre que foi desenvolvido por Martin Roesch, realizando análises de tráfegos e registros de pacotes em tempo real.

2.4 Conceitos de DDoS

DDoS é um ataque distribuído de negação de serviço que se comparado a um ataque DoS é mais complexo de ser identificado devido ao grande número de falsos alertas que podem ser emitidos confundindo-os como ataques maliciosos. Por sua vez este ataque sobrecarrega o sistema que fica inoperante por conta do congestionamento a rede, essa sobrecarga geralmente ocorre por conta de tantas solicitações que são enviadas por *cyber* criminosos e a partir disso conseguem obter dados pessoais, senhas de acesso para realizar transações bancárias dentre outras atividades.

Apesar de ser semelhante a um ataque DoS (Denial of Service), é mais complexo, pela dificuldade em detectá-lo. É um ataque distribuído, onde o mesmo pode vir de milhares de computadores que podem ter sido infectados sob um vírus que está sobre controle de um cyber criminoso, podendo tomar proporções gigantescas (RIGHI & NUNES, 2015). É importante ressaltar que as técnicas utilizadas para detecção de ataques DDoS ainda possuem muitas limitações e sua eficácia pode ser comprometida devido ao excesso de falsos alertas que são emitidos (GANAME et al. 2008). A Figura 4 ilustra a arquitetura de como acontece um ataque DDoS.

Figura 4 - Arquitetura de Ataque DDoS



Fonte: OO, T. T.; PHYU, T. A, p. 1767 (1990)

Antes de o hacker iniciar o ataque DDoS, ele procura hosts na rede que são vulneráveis. Os hackers conseguem manipular estes hosts e recebem autoridade de administrador para instalar e controlar programas. Com isso dão instruções/comando de controle aos manipuladores, para fazer pedidos aos agentes e geralmente os agentes são controlados por um ou mais manipuladores. Isso faz com que esses agentes enviem continuamente enormes números de pacotes inúteis para a vítima. Quando essa vítima recebe os pacotes, acaba não respondendo por todas essas solicitações de entradas, e fica congestionada. O sistema da vítima acaba travando por não conseguir atender ao número elevado de solicitações e se torna muito lento, com isso ocorre o então conhecido “acesso negado”, onde por conta dessas solicitações os recursos do sistema se tornam indisponíveis para a vítima.

Os números de ameaças deste ataque continuam a aumentar. Cerca de 50 milhões de ataques ocorrem todos os anos, o que equivale a um ou dois ataques por segundo todos os dias. Estes ataques ocorrem de maneira simultânea, cerca de dois terços são de um gigabit por segundo (Gbps) ou mais e criminosos estão utilizando estes ataques como distrações para encobrir atividades ilegais como fraude e roubo. Os custos associados a esses ataques estão aumentando, e o setor de serviços financeiros teve um prejuízo estimado de US$ 17 milhões por ataque DDos em 2012 (VERISIGN, 2018).

Com o passar do tempo os prejuízos estão aumentando cada vez mais, o que gera total preocupação na área de segurança. Dados apontados pelo site de investimento de valores econômicos (Valorinveste) mostram que dentre as principais organizações que sofrem estes ataques todos os anos estão os órgãos governamentais, indústrias e organizações de setor financeiro. Só no Brasil em 2021 foram registrados uma alta de 140% de ataques em relação ao ano anterior, cujos alvos foram dados bancários de clientes (VALORINVESTE, 2022).

De acordo com a empresa voltada para área de soluções em cibersegurança e combate a DDoS (Huge Networks) só em 2018 numa escala global o setor financeiro teve um prejuízo estimado em US$ 45 bilhões (R$ 180 bilhões) por conta desses ataques ilegais. O Brasil é alvo de 54% de todos os ataques ocorridos na América Latina, é o 5º território no mundo com maiores indices de casos, sem contar que está em segundo lugar com o *downtime* (Tempo de Inatividade) mais caro do mundo, em média US$ 306 (R$ 1224) de prejuízo por cada hora para uma operação fora do ar (CANALTECH, 2019).

2.5 Conceitos de IA

A inteligência artificial (IA) é uma área da ciência computacional que surgiu por volta da década de 50 após a segunda guerra mundial. Seu conceito fundamental é baseado em armazenar informações e executa-los de maneira correta, pode-se dizer que seu aprendizado é baseado em racionalidade onde aprende de acordo com os dados que são armazenados. Possui uma vasta ramificação em diversas áreas baseados em aprendizagem e percepções como área das engenharias para provar determinados teoremas matemáticos através de algoritmos utilizados na área da computação, ou até mesmo na área da medicina para identificar doenças de maneira mais eficaz e eficiente, dentre diversas outras áreas que podem utilizar a IA para resoluções de problemas, por conta de ampla aplicação pode-se dizer que é considerada uma área universal, pois é importante em quaisquer área envolvendo tarefas intelectuais (NORVIG, 2021).

Um de seus grandes feitos foi uma máquina desenvolvida pelo matemático, cientista e filósofo Allan Turing, no qual utilizou conceitos matemáticos e computacionais para decifrar os códigos que eram utilizados pelos Alemães nazistas

na segunda guerra mundial. Turing conseguiu por meio de sua máquina decifrar os códigos que eram criptografados e que até então eram considerados inquebráveis na máquina Enigma por possuir infinitas possibilidades de leituras e combinações no qual levaria aproximadamente 20 milhões de anos para checar todas as combinações. O filme jogo da imitação lançado em 2014 retrata essa história considerada uma das mais importantes da ciência da computação.

Com o passar do tempo foram surgindo diversos outros métodos e algoritmos baseados em IA, como redes neurais artificiais, aprendizagem de máquinas que utiliza árvores de decisão para classificação e agrupamento de dados, teoremas probabilísticos baseados em entropia.

A RNA (Rede Neural Artificial) conhecida como “Redes Neurais Artificiais”, pode ser utilizada para detectar DDoS, pois compara resultados de detecção com árvore de decisão, RNA, entropia e Bayesiana (JIE-HAO et al. 2012). O funcionamento de uma RNA é baseado no processamento de informações semelhante ao comportamento do cérebro humano (CHEN et al. 2009).

O processo de aprendizagem de uma RNA é contínuo e adaptativo, podem ser utilizadas para classificação de dados e reconhecimento de padrões (BULUSU et al. 2009). Para o bom funcionamento de uma rede neural é necessário estabelecer os processos corretos com os dados que deseja trabalhar, é possível calibrar uma RNA de 2 formas, por meio de sua entrada verificar se o dado está correto caso não fazer a análise reversa onde irá recalibrar o processo de maneira recursiva, ajustando aquilo que for necessário até que o dado esteja correto aplicando os pesos que forem necessários para seu ajuste.

2.6 Conceitos dos algoritmos

Nesta seção serão apresentados conceitos dos algoritmos Random Tree, Random Forest e J48.

2.6.1 Conceitos do Random Tree

O modelo Random Tree é um tipo de aprendizado supervisionado no qual permite a utilização para classificação e regressão, este aprendizado por sua vez é

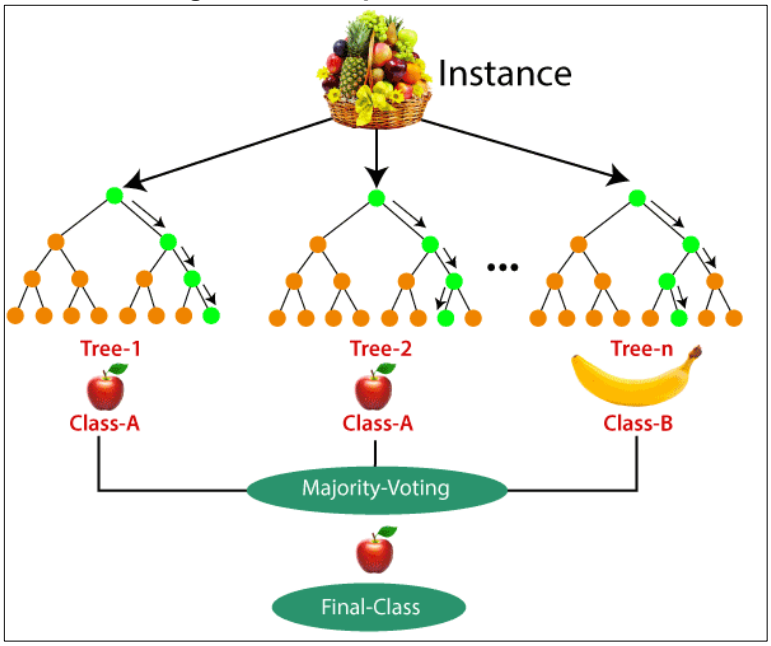
composto de regras de decisão simples que são extraídas dos dados para poder realizar a predição e com isso gerar a árvore de decisão (MARIN et al. 2021).

2.6.2 Conceitos do Random Forest

Já o Random Forest é um modelo composto por várias árvores de decisão, basicamente utiliza as árvores de decisão geradas como classificadores fracos com a intenção de gerar um classificador forte obtendo com isso um melhor desempenho, pois seus testes ao invés de gerar uma única árvore irão testar várias e com isso gerar uma que melhor se adeque ao conjunto dos modelos treinados (MARIN et al. 2021).

Na Figura 5 pode-se observar como funciona o algoritmo Random Forest.

Figura 5 - Exemplo Random Forest



Fonte: ANALYTICS (2022)

Considerando a cesta de frutas como as instâncias com as amostras, tira-se n amostras desta cesta de frutas e para cada instância é construída uma árvore de decisão. Por sua vez para cada árvore será gerada uma saída neste exemplo da Figura 5 definido como classe. O resultado final será composto por uma votação majoritária, esse processo de votação considera as n amostras geradas e escolherá aquela que for obtida na saída das maiorias das árvores. Como observado a votação majoritária para este exemplo resultou numa saída como uma maçã, como foi a que

mais apareceu nas saídas de cada árvore então a classe final resultante será a maçã.

2.6.3 Conceitos do J48

O algoritmo J48 é uma implementação em java do algoritmo c4.5, geralmente utilizado para fazer classificação de dados, utiliza entropia para fazer a construção da árvore de decisão. Basicamente o algoritmo irá escolher aquele atributo que mais particionará seu conjunto em subconjuntos de dados, por sua vez este particionamento é definido como ganho de informação normalizado (diferença em entropia), após ter definido um atributo com maior ganho então este atributo é escolhido para tomar a decisão e assim repete este processo nos subconjuntos até que seja criada a árvore de decisão, logo após a criação desta árvore o algoritmo fará a poda da árvore onde irá percorrer os ramos anteriores para remoção daqueles que não ajudaram no processo de divisões em subconjuntos e substitui-os por nós folha.

A árvore de decisão para ser construída utiliza as fórmulas de entropia e ganho de informação no qual a entropia é calculada a partir das somas das probabilidades de determinado evento ocorrer vezes o log de cada probabilidade de maneira geral utilizando a seguinte fórmula: $Entropia(S) = -\sum p_i . log_2 p_i$, depois multiplica-se pelo ganho de informação de cada atributo (variável) utilizando a seguinte fórmula: $Ganho\ de\ informação(S, A) = Entropia\ de(S) - \sum v \in valores(A)\ p(A_v) * Entropia\ (A_v)$, o ganho de informação é utilizado para definir qual será o melhor atributo para ser o primeiro nó da árvore.

A seguir no Quadro 1 é representado como funciona este cálculo da entropia e ganhos de informação para ser escolhido o melhor nó da árvore.

Quadro 1 - Exemplo para construção da árvore

Sexo	Altura	Cabelo	Olhos Claros
Masculino	Baixo(a)	Claro	Sim
Feminino	Alto(a)	Escuro	Sim
Masculino	Baixo(a)	Escuro	Sim
Masculino	Alto(a)	Escuro	Não
Feminino	Alto(a)	Claro	Não
Feminino	Baixo(a)	Claro	Não

Fonte: Autoria própria (2022)

O total de instâncias são 6 logo calcula-se a entropia para cada um dos 3 atributos para saber se determinado indivíduo tem ou não olhos claros.

Calculando a entropia de a pessoa ter ou não olhos claros tem-se:

$$Entropia\ olhos\ claros = -\left(\left(\frac{1}{2}\right) * log_2\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) * log_2\left(\frac{1}{2}\right)\right) = 1$$

Agora calcula-se o ganho de informação para um atributo, neste caso será calculado o ganho de informação para o atributo “Sexo”. Como o atributo sexo possui 2 instâncias sendo “Masculino” e “Feminino” é preciso calcular antes a entropia das 2 instâncias que resultará em:

$$Entropia\ (Sexo\ masculino) = -\left(\left(\frac{2}{3}\right) * log_2\left(\frac{2}{3}\right) + \left(\frac{1}{3}\right) * log_2\left(\frac{1}{3}\right)\right) = 0{,}92$$

$$Entropia\ (Sexo\ feminino) = -\left(\left(\frac{1}{3}\right) * log_2\left(\frac{1}{3}\right) + \left(\frac{2}{3}\right) * log_2\left(\frac{2}{3}\right)\right) = 0{,}92$$

Então, aplica-se na fórmula geral da equação, com isso tem-se:

$$Ganho\ de\ informação(Sexo) = 1 - \left(\left(\frac{1}{2}\right) * 0{,}92 + \left(\frac{1}{2}\right) * 0{,}92\right) = 0{,}08$$

O próximo passo é calcular o ganho de informação para o atributo “Altura”. Por sua vez este atributo é composto por duas instâncias sendo elas: “Alto(a)” e “Baixo(a)”. Com isso calcula-se as entropias dessas instâncias, resultando em:

$$Entropia\ (Altura\ baixo(a)) = -\left(\left(\frac{2}{3}\right) * log_2\left(\frac{2}{3}\right) + \left(\frac{1}{3}\right) * log_2\left(\frac{1}{3}\right)\right) = 0{,}92$$

$$Entropia\ (Altura\ alta(a)) = -\left(\left(\frac{1}{3}\right) * log_2\left(\frac{1}{3}\right) + \left(\frac{2}{3}\right) * log_2\left(\frac{2}{3}\right)\right) = 0{,}92$$

Aplica-se então a fórmula geral do ganho de informação para esse atributo, resultando em:

$$Ganho\ de\ informação(Altura) = 1 - \left(\left(\frac{3}{6}\right) * 0{,}92 + \left(\frac{3}{6}\right) * 0{,}92\right) = 0{,}08$$

Agora o próximo passo é calcular o ganho de informação para o atributo “Cabelo” e como este atributo também possui 2 instâncias: “Claro” e “Escuro” deve-se ser calculado suas respectivas entropias, resultando em:

$$Entropia\ (Cabelo\ claro) = -\left(\left(\frac{1}{2}\right) * log_2\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) * log_2\left(\frac{1}{2}\right)\right) = 1$$

$$Entropia\ (Cabelo\ escuro) = -\left(\left(\frac{2}{4}\right) * log_2\left(\frac{2}{4}\right) + \left(\frac{2}{4}\right) * log_2\left(\frac{2}{4}\right)\right) = 1$$

Aplica-se a fórmula geral do ganho de informação para o atributo “Cabelo”, resultando em:

$$Ganho\ de\ informação(Cabelo) = 1 - \left(\left(\frac{2}{6}\right) * 1 + \left(\frac{4}{6}\right) * 1\right) = 0$$

Com isso já pode definir o primeiro nó da árvore como neste exemplo tanto o atributo “Sexo” como “Altura” obtiveram o maior ganho de informação um destes poderá ser o primeiro nó desta árvore e fará a construção da árvore recalculando novamente um novo nó de onde parou para ser considerado o primeiro com maior ganho a partir do ponto em que parou.

3 DESENVOLVIMENTO

Neste Capítulo será apresentado o ambiente WEKA utilizado para fazer a simulação, o dataset escolhido para análise, as simulações dos algoritmos e os resultados e comparativos de ambos.

3.1 Ambiente de simulação WEKA

WEKA é um software de aprendizado de máquinas utilizado para mineração de dados. Ele é de código aberto emitido sob a GNU General Public License, este software possui ferramentas para realizar a preparação, classificação, regressão, agrupamento, mineração de regras de associação e visualização de dados. Começou seu desenvolvimento em 1993, utilizando a linguagem java, na Universidade de Waikato, Nova Zelândia (WEKA, 1999).

Existem 2 principais tipos de arquivos que são possíveis abrir no WEKA. O CSV que são arquivos de texto em que os campos de dados são delimitados por vírgula e o ARFF *(Attribute-Relation File Format)* que aceita 2 tipos de dados que são as strings que serão convertidas em nominais e os dados numéricos para com isso construir o cabeçalho do arquivo de acordo com o que a ferramenta necessitar. Importante ressaltar que no formato ARFF tem-se os dados padrão desse arquivo que são a relação, os atributos e os dados.

Na própria ferramenta já é disponibilizada alguns algoritmos como: NaiveBayes, ZeroR, J48, Floresta e Árvores aleatórias, K-means, FarthestFirst, LVQ, Apriori, dentre outros que são possíveis baixar o pacote.

3.2 Dataset

O dataset CICIDS2017 utilizado foi baixado na página da UNB (Universidade de New Brunswick) no Canadá, esta universidade possui o CIC *(Canadian Institute for Cybersecurity)* onde reúne pesquisadores de diversas áreas, como Ciências sociais, negócios, ciência da computação, engenharia, direito e ciência (SHARAFALDIN et al. 2018). O arquivo utilizado do dataset é composto por registros de pacotes de diversos serviços como porta de destino, fluxo, tamanho de pacotes,

tempo entre pacotes, flags e os ataques benignos e os que se caracterizam como DDoS. Essa base de dados é composta por um total de 79 atributos e 225.745 instâncias, o tamanho total do arquivo é de 75.537 KB, este arquivo encontra-se originalmente no formato em .csv. No Quadro 2 é exemplificado alguns dos atributos do data set CICID2017 com suas respectivas descrições, todos os atributos que constam no dataset podem ser vistos no Apêndice A.

Quadro 2 - Alguns atributos do CICID2017

Nome dos atributos	Descrição
Destination Port	Porta de destino
Flow Duration	Duração do fluxo
Total Fwd Packets	Total de pacotes enviados
Total Backward Packets	Total de pacotes recebidos
Total Length of Fwd Packets	Tamanho total de pacotes enviados
Total Lenght of Bwd Packets	Tamanho total de pacotes recebidos
Fwd Packet Length Max	Tamanho máximo dos pacotes enviados
Fwd Packet Length Min	Tamanho mínimo dos pacotes enviados
Fwd Packet Length Mean	Tamanho médio dos pacotes enviados
Fwd Packet Length Std	Tamanho do desvio padrão dos pacotes enviados
Flow Bytes/s	Fluxo de bytes por segundo
Flow Packets/s	Fluxo de pacotes por segundo
Fwd PSH Flags	Flag’s enviadas dos envios imediato de dados
Bwd PSH Flags	Flag’s recebidas dos envios imediato de dados
Label	Referente ao tipo de ataque se é Benigno ou DDOS.

Fonte: Autoria própria (2022)

Após a seleção do arquivo deve-se realizar o tratamento dos dados, realizando a exclusão de todas as colunas em que todos os respectivos elementos de sua célula sejam vazios ou iguais a 0. O arquivo é composto por um total de 79 atributos, desses atributos foram excluídos 12 em que os dados eram iguais a 0, restando com isso 67 atributos.

3.3 Simulação Random Tree

Após selecionar o algoritmo nas opções de teste define-se a porcentagem Split. Este tipo de teste permite que seja definido o tamanho que será utilizado para treino e o tamanho para testar a rede, neste caso define-se a porcentagem 70%, isso indicará ao algoritmo que seja utilizado 70% para treinar e 30% para testar. Com isso o algoritmo gera as regras de decisão para identificar quando os dados são benignos ou quando são caracterizados como DDoS. Como Mostrado na Figura

6 uma parte das regras de decisão gerada, as regras inteiras podem ser observadas no Apêndice B.

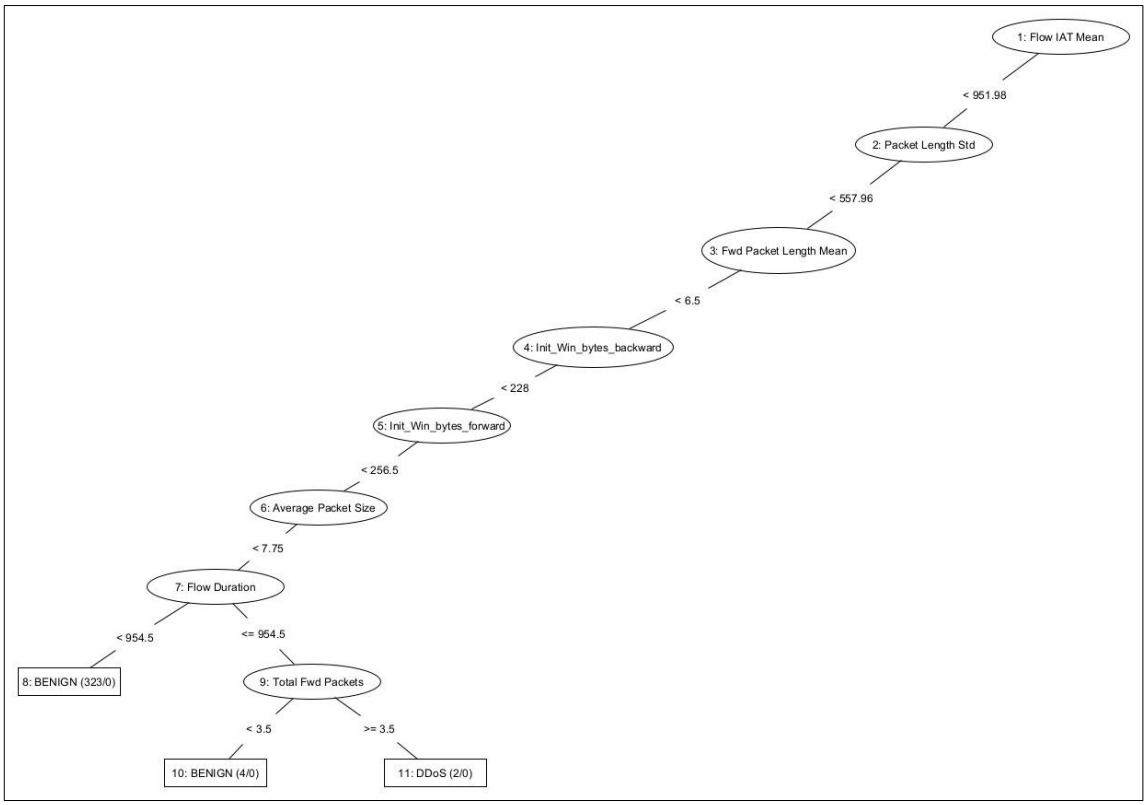
Figura 6 - Regras de Decisão Random Tree

```
RandomTree
==========

 Flow IAT Mean < 951.98
|    Packet Length Std < 557.96
|   |    Fwd Packet Length Mean < 6.5
|   |   |    Init_Win_bytes_backward < 228
|   |   |   |   Init_Win_bytes_forward < 256.5
|   |   |   |   |    Average Packet Size < 7.75
|   |   |   |   |   |    Flow Duration < 954.5 : BENIGN (323/0)
|   |   |   |   |   |    Flow Duration >= 954.5
|   |   |   |   |   |   |    Total Fwd Packets < 3.5 : BENIGN (4/0)
|   |   |   |   |   |   |    Total Fwd Packets >= 3.5 : DDoS (2/0)
```

Fonte: Autoria própria (2022)

Após a execução do algoritmo é possível visualizar a árvore de decisão gerada para o dataset analisado como mostra na Figura 7 uma parte da árvore gerada, a árvore de decisão completa pode-se observar no Apêndice C.

Figura 7 - Árvore de Decisão



Fonte: Autoria própria (2022)

Com a árvore gerada é possível identificar por meio das condições geradas quando o dado será ou não propício a ser considerado como benigno ou como ataque DDoS. Este trecho de árvore gerada analisando o último nó aponta que foram identificados 2 ataques caracterizados como DDoS e 4 benignos.

3.3.1 Resultados com diferentes porcentagens de tamanho de treino e teste

Após a execução da base de dados CICID2017 com tamanhos de teste e treino distintos obteve-se o seguinte resultado como mostrado na Tabela 1.

Tabela 1 - Resultados do algoritmo Random Tree

Tamanho da base de treino e teste (%)	Tempo de execução para treinar e testar (s)	Instâncias classificadas corretamente (%)	Instâncias classificadas incorretamente (%)
60 - 40	4,38 – 0,14	90280 (99.9801)	18 (0.0199)
65 - 35	4,1 – 0,33	78995 (99.9797)	16 (0.0203)
70 - 30	4,34 – 0,19	67698 (99.9631)	25 (0.0369)
75 - 25	4,21 – 0,06	56426 (99.9823)	10 (0.0177)
80 - 20	4,26 – 0,05	45140 (99.9801)	9 (0.0199)

Fonte: Autoria própria (2022)

Com isso o melhor tamanho de treino e teste para essa base de dados é 75% para treinar e 25% para testar, pois foi a que obteve o melhor resultado de instâncias classificadas corretamente.

Na Tabela 2 pode-se observar os falsos positivos e falsos negativos através da matriz de confusão que foram geradas em cada modelo da base de treino e teste citados na Tabela 1.

Tabela 2 - Matrizes de confusão

	Benigno	DDoS
60 – 40	...	...
Benigno	39100	12
DDoS	6	51180
65 – 35	...	...
Benigno	34210	6
DDoS	10	44785
70 – 30	...	...
Benigno	29291	12
DDoS	13	38407
75 – 25	...	...
Benigno	24464	6
DDoS	4	31962
80 – 20	...	...
Benigno	19504	6
DDoS	3	25636

Fonte: Autoria própria (2022)

Para a base de dados testada em 40% obteve-se um total de 18 instâncias classificadas incorretamente sendo elas 12 falsos positivos e 6 falsos negativos,

para a base de dados testada em 35% teve-se 6 instâncias falsos positivos e 10 falsos negativos, para a base de dados testada em 30% teve-se 12 instâncias falsos positivos e 13 falsos negativos, para a base testada em 25% 6 instâncias foram classificadas como falsos positivos e 4 falsos negativos e por fim a base de dados testada em apenas 20% obteve 6 instâncias classificadas como falsos positivos e 3 falsos negativos.

3.4 Simulação Random Forest

Selecionado o algoritmo na opção de teste deve-se definir a porcentagem Split para a simulação, essa opção permite ao usuário definir a porcentagem dos dados que serão utilizados como treino e consequentemente o restante como teste.

3.4.1 Resultados com diferentes porcentagens de tamanho de treino e teste

Utilizando o mesmo dataset da simulação Random Tree obteve-se o seguinte resultado como mostrado na Tabela 3.

Tabela 3 - Resultados do algoritmo Random Forest

Tamanho da base de treino e teste (%)	Tempo de execução para treinar e testar (s)	Instâncias classificadas corretamente (%)	Instâncias classificadas incorretamente (%)
60 – 40	184,09 – 1	90294 (99.9956)	4 (0.0044)
65 – 35	184,53 – 0,86	79005 (99.9924)	6 (0.0076)
70 – 30	187,77 – 0,97	67718 (99.9926)	5 (0.0074)
75 - 25	185,76 – 0,66	56433 (99.9947)	3 (0.0053)
80 – 20	186,22 – 0,49	45147 (99.9956)	2 (0.0044)

Fonte: Autoria própria (2022)

Analisando os resultados observa-se que o melhor tamanho de teste e treino para a base de dados simulada neste algoritmo serão 60% para treinar e 40% para testar e também 80% para treinar e 20% para testar, pois em ambos os testes tanto a porcentagem de instâncias classificadas corretamente como as classificadas incorretamente obteve-se exatamente a mesma porcentagem.

Na Tabela 4 pode-se observar os falsos positivos e falsos negativos por meio da matriz de confusão que foram geradas em cada modelo da base de treino e teste citados na Tabela 3.

Tabela 4 - Matrizes de confusão Random Forest

(continua)

	Benigno	DDoS
60 – 40	...	...
Benigno	39111	1
DDoS	3	51183
65 – 35	...	...

Tabela 4 – Matrizes de confusão Random Forest

(conclusão)

	Benigno	**DDoS**
Benigno	34215	1
DDoS	5	44790
70 – 30	...	...
Benigno	29302	1
DDoS	4	38416
75 – 25	...	...
Benigno	24469	1
DDoS	2	31964
80 – 20	...	...
Benigno	19510	0
DDoS	2	25637

Fonte: Autoria própria (2022)

Por fim tem-se que para a base de dados testada em 40% obteve-se um total de 4 instâncias classificadas incorretamente sendo elas 1 falso positivo e 3 falsos negativos, para a base testada em 35% dos dados teve-se 6 instâncias classificadas incorretamente sendo 1 falso positivo e 5 falsos negativos, para a base testada em 30% obteve-se 5 instâncias classificadas incorretamente sendo elas 1 falso positivo e 4 falsos negativos, para a base de dados testada em 25% obteve-se 3 instâncias classificadas incorretamente sendo 1 falso positivo e 2 falsos negativos, por fim a base testada em 20% foram encontradas 2 instâncias classificadas incorretamente sendo elas 2 falsos negativos.

3.5 Simulação J48

Após selecionado o algoritmo na opção teste, define-se a porcentagem Split para iniciar a simulação, esta porcentagem como vistos nas simulações anteriores define para o algoritmo qual será a quantidade de dados para treino e a quantidade para teste. Com isso é gerado as regras de decisão que se comparadas as regras do algoritmo random tree disponível no Apêndice B são bem menores, isto se dá pelo fato de que ao fazer a simulação utilizando o algoritmo j48 ele irá escolher as instâncias que mais se adequem a subdivisão dos dados, tornando com isso o conjunto de regras bem menores em relação ao algoritmo random tree. Na Figura 8 é possível observar as regras de decisão do j48.

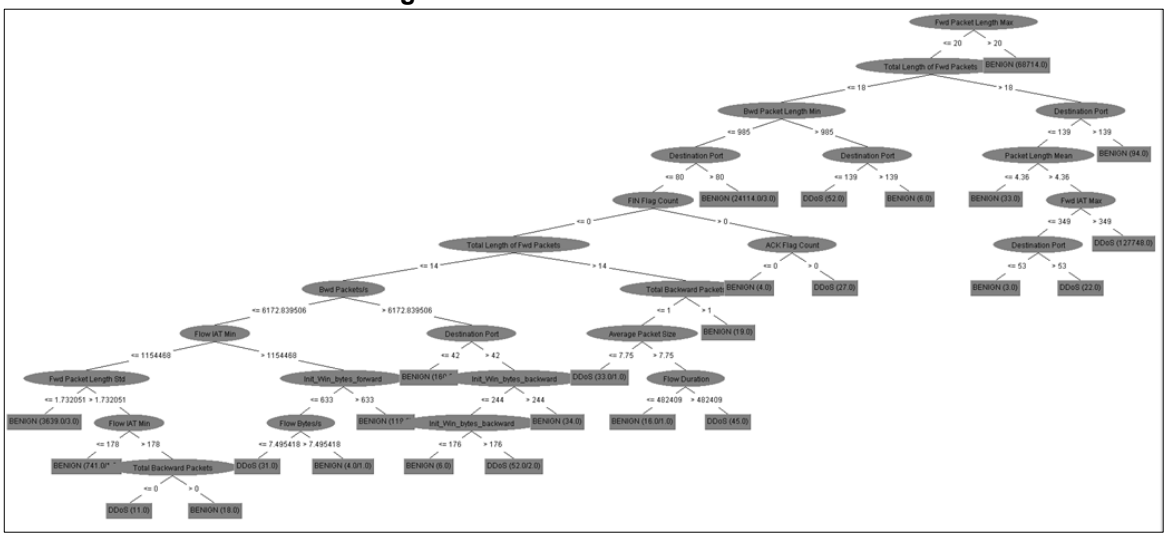
Figura 8 - Regras de Decisão J48

```
J48 pruned tree
------------------

 Fwd Packet Length Max <= 20
|   Total Length of Fwd Packets <= 18
|   |   Bwd Packet Length Min <= 985
|   |   |   Destination Port <= 80
|   |   |   |   FIN Flag Count <= 0
|   |   |   |   |   Total Length of Fwd Packets <= 14
|   |   |   |   |   |   Bwd Packets/s <= 6172.839506
|   |   |   |   |   |   |   Flow IAT Min <= 1154468
|   |   |   |   |   |   |   |   Fwd Packet Length Std <= 1.732051: BENIGN (3639.0/3.0)
|   |   |   |   |   |   |   |   Fwd Packet Length Std > 1.732051
|   |   |   |   |   |   |   |   |   Flow IAT Min <= 178: BENIGN (741.0/1.0)
|   |   |   |   |   |   |   |   |   Flow IAT Min > 178
|   |   |   |   |   |   |   |   |   |   Total Backward Packets <= 0: DDoS (11.0)
|   |   |   |   |   |   |   |   |   |   Total Backward Packets > 0: BENIGN (18.0)
|   |   |   |   |   |   |   |   Flow IAT Min > 1154468
|   |   |   |   |   |   |   |   |   Init_Win_bytes_forward <= 633
|   |   |   |   |   |   |   |   |   |   Flow Bytes/s <= 7.495418: DDoS (31.0)
|   |   |   |   |   |   |   |   |   |   Flow Bytes/s > 7.495418: BENIGN (4.0/1.0)
|   |   |   |   |   |   |   |   |   Init_Win_bytes_forward > 633: BENIGN (119.0)
|   |   |   |   |   |   |   Bwd Packets/s > 6172.839506
|   |   |   |   |   |   |   |   Destination Port <= 42: BENIGN (160.0)
|   |   |   |   |   |   |   |   Destination Port > 42
|   |   |   |   |   |   |   |   |   Init_Win_bytes_backward <= 244
|   |   |   |   |   |   |   |   |   |   Init_Win_bytes_backward <= 176: BENIGN (6.0)
|   |   |   |   |   |   |   |   |   |   Init_Win_bytes_backward > 176: DDoS (52.0/2.0)
|   |   |   |   |   |   |   |   |   Init_Win_bytes_backward > 244: BENIGN (34.0)
|   |   |   |   |   |   Total Length of Fwd Packets > 14
|   |   |   |   |   |   |   Total Backward Packets <= 1
|   |   |   |   |   |   |   |   Average Packet Size <= 7.75: DDoS (33.0/1.0)
|   |   |   |   |   |   |   |   Average Packet Size > 7.75
|   |   |   |   |   |   |   |   |   Flow Duration <= 482409: BENIGN (16.0/1.0)
|   |   |   |   |   |   |   |   |   Flow Duration > 482409: DDoS (45.0)
|   |   |   |   |   |   |   Total Backward Packets > 1: BENIGN (19.0)
|   |   |   |   FIN Flag Count > 0
|   |   |   |   |   ACK Flag Count <= 0: BENIGN (4.0)
|   |   |   |   |   ACK Flag Count > 0: DDoS (27.0)
|   |   |   Destination Port > 80: BENIGN (24114.0/3.0)
|   |   Bwd Packet Length Min > 985
|   |   |   Destination Port <= 139: DDoS (52.0)
|   |   |   Destination Port > 139: BENIGN (6.0)
|   Total Length of Fwd Packets > 18
|   |   Destination Port <= 139
|   |   |   Packet Length Mean <= 4.36: BENIGN (33.0)
|   |   |   Packet Length Mean > 4.36
|   |   |   |   Fwd IAT Max <= 349
|   |   |   |   |   Destination Port <= 53: BENIGN (3.0)
|   |   |   |   |   Destination Port > 53: DDoS (22.0)
|   |   |   |   Fwd IAT Max > 349: DDoS (127748.0)
|   |   Destination Port > 139: BENIGN (94.0)
 Fwd Packet Length Max > 20: BENIGN (68714.0)

Number of Leaves  :     26
```

Fonte: Autoria própria (2022)

Por conta do algoritmo J48 pegar as instâncias que mais se ajustem as suas respectivas subdivisões a sua árvore gerada será bem menor em relação a árvore gerada no algoritmo Random Tree, pois ao realizar a simulação é utilizado o conceito de poda de árvore, aonde vai cortando tudo aquilo que julgar não ser trivial, com isso é possível observar na Figura 9 que a árvore gerada diminuiu consideravelmente, cerca de 65% em relação a Random Tree.

Figura 9 - Árvore de Decisão J48



Fonte: Autoria própria (2022)

3.5.1 Resultados com diferentes porcentagens de tamanho de treino e teste

Para esta simulação pode-se observer os resultados obtidos como mostrados na Tabela 5, tendo como referência o mesmo dataset utilizado nas simulações dos algoritmos anteriores.

Tabela 5 - Resultados do algoritmo J48

Tamanho da base de treino e teste (%)	Tempo de execução para treinar e testar (s)	Instâncias classificadas corretamente (%)	Instâncias classificadas incorretamente (%)
60 – 40	27,35 – 0,11	90288 (99.9889)	10 (0.0111)
65 – 35	27,69 – 0,08	78998 (99.9835)	13 (0.0165)
70 – 30	27,48 – 0,27	67711 (99.9823)	12 (0.0177)
75 - 25	26,96 – 0,08	56423 (99.977)	13 (0.023)
80 – 20	27,06 – 0,05	45144 (99.9889)	5 (0.0111)

Fonte: Autoria própria (2022)

Com base nos resultados obtidos, a simulação com o melhor resultado de treino e teste para esta simulação foram a base de dados treinada em 60% e testada em 40% e também a base treinada em 80% e testada em 20%, pois ambas obtiveram a mesma porcentagem de instâncias classificadas corretamente.

Na Tabela 6 observa-se os falsos positivos e negativos através da matriz de confusão gerada no modelo de treino e teste citados na Tabela 5 do algoritmo J48.

Tabela 6 - Matrizes de confusão J48

	Benigno	DDoS
60 – 40	...	...
Benigno	39106	6
DDoS	4	51182
65 – 35	...	...
Benigno	34210	6
DDoS	7	44788
70 – 30	...	...
Benigno	29298	5
DDoS	7	38413
75 – 25	...	...
Benigno	24465	5
DDoS	8	31958
80 – 20	...	...
Benigno	19506	4
DDoS	1	25638

Fonte: Autoria própria (2022)

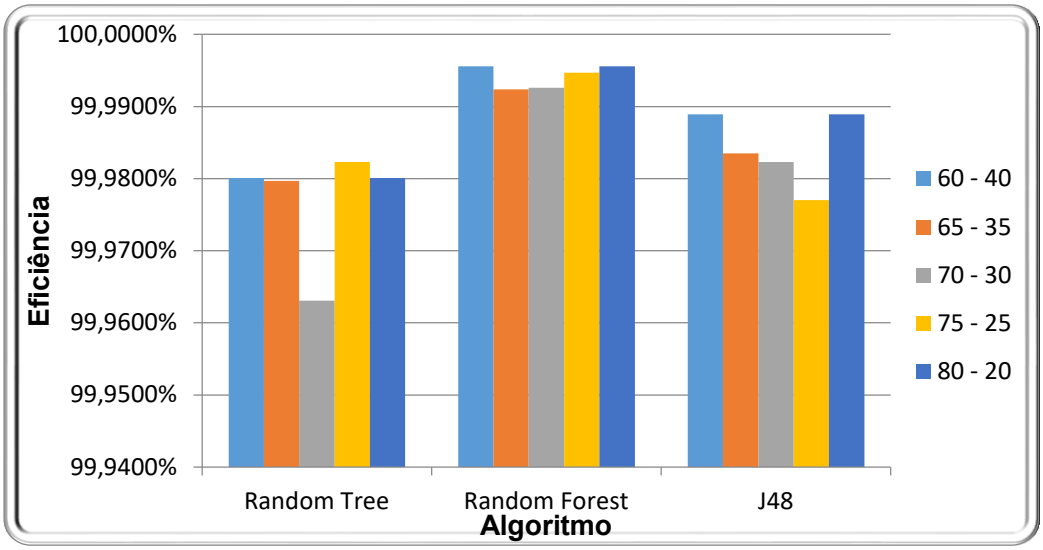
Com base nos dados testados em 40% obteve-se um total de 10 instâncias classificadas incorretamente sendo elas 6 falsos positivos e 4 falsos negativos, para a base testada em 35% foram identificadas 13 instâncias incorretas sendo elas 6 falsos positivos e 7 falsos negativos, para base testada em 30% teve-se 12 instâncias incorretas sendo elas 5 falsos positivos e 7 falsos negativos, para base

testada em 25% foram encontradas 13 instâncias incorretas sendo elas 5 falsos positivos e 8 falsos negativos, por fim a base testada em 20% foram identificadas 5 instâncias incorretas sendo elas 4 falsos positivos e 1 falso negativo.

3.6 Comparação do Random Tree, Random Forest e J48

Para comparação dos testes realizados é possível observar no Gráfico 1, dentre os 3 algoritmos qual foi o que obteve um melhor desempenho em relação aos demais.

Gráfico 1 - Comparação dos algoritmos



Fonte: Autoria própria (2022)

Com base no Gráfico 1, o algoritmo com melhor desempenho foi o Random Forest, pois em todas as simulações realizadas obteve-se uma porcentagem de acertos superiores, e o algoritmo com o pior desempenho foi o Random Tree.

É possível observar na Tabela 7 o tempo de execução de cada algoritmo.

Tabela 7 - Tempo de Execução dos Algoritmos

(continua)

Algoritmo	Tamanho da Base (%) (Treino e Teste)	Tempo de Execução (s)
Random Tree	60 - 40	4.52
	65 - 35	4.43
	70 - 30	4.53
	75 - 25	4.27
	80 - 20	4.31
Random Forest	60 - 40	185.09
	65 - 35	185.39
	70 - 30	188.74
	75 - 25	186.42
	80 - 20	186.71
	60 - 40	27.46

Tabela – 7 Tempo de Execução dos Algoritmos

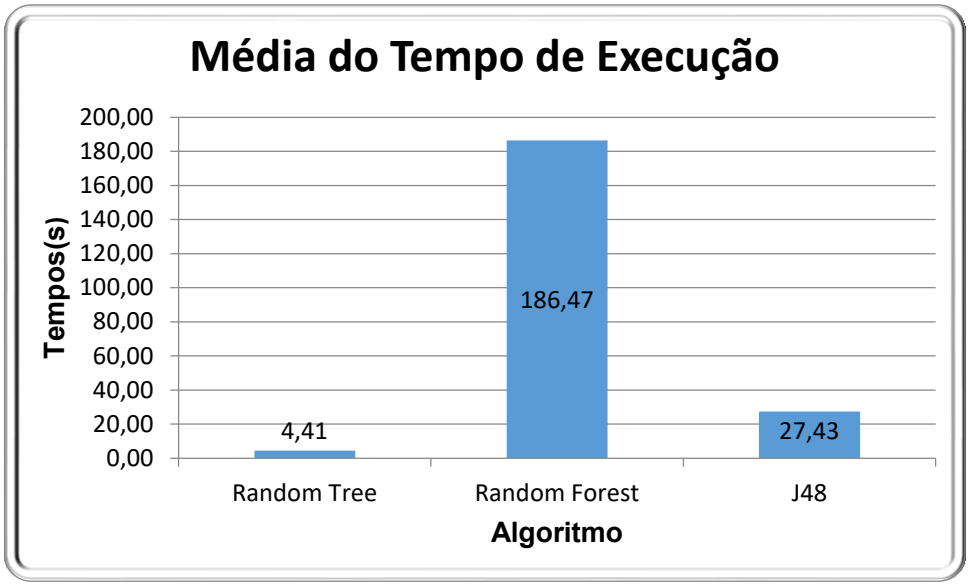
(conclusão)

Algoritmo	Tamanho da Base (%) (Treino e Teste)	Tempo de Execução (s)
J48	65 - 35	27.77
	70 - 30	27.75
	75 - 25	27.04
	80 - 20	27.11

Fonte: Autoria própria (2022)

Com base na Tabela 7 o algoritmo Random Tree foi executado em menos tempo em relação aos outros porém, foi o menos eficiente, já o J48 foi executado com um tempo maior em relação ao Random Tree e seus resultados foram melhores entretanto, o algoritmo que obteve o maior tempo de execução foi o Random Forest e seus resultados foram os melhores. Com base nestes dados foi calculado o tempo médio de execução de cada algoritmo como mostrado no Gráfico 2.

Gráfico 2 - Tempo médio de execução



Fonte: Autoria própria (2022)

É observável que quanto maior o tempo do algoritmo executado melhor será seu resultado e quanto menor o tempo de execução pior será o resultado.

4 CONCLUSÃO

Este trabalho mostra como identificar ataques DDoS utilizando três algoritmos de classificação de dados por meio de árvores de decisão: Random Tree, Random Forest e J48. Através deles foram feitas simulações com diferentes porcentagens de treino e testes para cada simulação realizada.

Através dos testes pode-se observar qual algoritmo obteve melhor tempo e melhor resultado para as simulações. Com isso os resultados revelaram que o algoritmo Random Tree levou o menor tempo para criação e execução do modelo em média (4,41 segundos), porém em contrapartida obteve o menor desempenho em relação aos demais (99,97%) de eficiência. O J48 se comparado ao algoritmo Random Tree foi mais eficiente, pois obteve um desempenho de (99,98%), mas seu tempo médio de execução foi maior (27,43 segundos). O Random Forest foi o algoritmo que teve o melhor desempenho, pois sua eficiência foi (99,99%), já seu tempo médio de execução foi bem maior se comparado aos algoritmos Random Tree e J48 (186,47 segundos).

Por fim a partir dos resultados, pode-se concluir quando necessário um algoritmo que execute em menor tempo deve-se optar pelo algoritmo Random Tree, se for necessário um bom desempenho de sua eficiência o algoritmo ideal é o Random Forest.

Como trabalhos futuros é sugerido realizar os testes utilizando validações cruzadas e comparar os resultados obtidos com os resultados deste trabalho, assim como também realizar testes com outros algoritmos de classificação. As regras de decisão geradas a partir das árvores de decisão deste trabalho também podem ser utilizadas para criações de novas regras e implementa-las na ferramenta Snort. É esperado observar quando os resultados de identificação do IDS melhoram ou não utilizando a simulação seja num ambiente de rede real ou virtual.

REFERÊNCIAS

ANALYTICS VIDHYA. **Understanding Random Forest**. Disponível em: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. Acesso em: 29 set. 2022.

BULUSU, N.; *et al*. Scalable coordination for wireless sensor networks: self-configuring localization systems. In: INTERNATIONAL SYMPOSIUM ON COMMUNICATION THEORY AND APPLICATIONS, 2009, Ambleside Reino Unido. **Anais…** Ambleside (Reino Unido): ISCTA, 2001.

CANALTECH. **Ataques DDoS geram preocupação no setor financeiro**. Disponível em: https://canaltech.com.br/seguranca/ataques-ddos-geram-preocupacao-no-setor-financeiro-156861/. Acesso em: 14 set. 2022.

CHEN, Z.; QIAN, P. Application of PSO-RBF neural network in network intrusion detection. In: THIRD INTERNATIONAL SYMPOSIUM ON INTELLIGENT INFORMATION TECHNOLOGY APPLICATION, 2009, Shangai China. **Anais…** Shangai (China): IITA, 2009.

CISCO. **O que é segurança de rede**. Disponível em: https://www.cisco.com/c/pt_br/products/security/what-is-network-security.html. Acesso em: 29 ago. 2022.

COPPIN, B. **Inteligência artificial**. 1. ed. Rio de Janeiro: Gen, 2010

GANAME, A. K.; *et al*. A global security architecture for intrusion detection on computer networks. **Computers & Security**, v. 27, n. 1-2, p. 30-47, mar. 2008. A2

JIE-HAO, C.; et al. DDoS defense system with test and neural network. In: IEEE INTERNATIONAL CONFERENCE ON GRANULAR COMPUTING, 2012, Hangzhou China. Proceedings… Hangzhou (China): GrC, 2012.

MACEDO, R. *et al*. **Redes de computadores**. 1. ed. Santa Maria: UFSM, NTE, 2018. *E-book* (196 p.) ISBN 978-85-8341-225-0. Disponível em: https://www.ufsm.br/app/uploads/sites/358/2019/08/MD_RedesdeComputadores.pdf. Acesso em: 25 set. 2020.

MEDINA, R. D. **ASTERIX - Aprendizagem significativa e tecnologias aplicadas no ensino de redes de computadores: integrando e explorando possibilidades.** 174 f. 2004. Tese (Doutorado) – Programa de Pós-Graduação em Tecnologia, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2004.

MARIN, M. C.; *et al.* Caracterização e classificação do tráfego da darknet com modelos baseados em árvores de decisão. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 2021, Uberlândia Minas Gerais. **Anais…** Uberlândia (Minas Gerais): SBRC, 2021.

MENDES, D. R. **Redes de computadores**: teoria e prática. 1. ed. São Paulo: Novatec, 2007

NORVIG, P.; RUSSELL, S. **Inteligência artificial**. 3. ed. Rio de Janeiro: Gen, 2021

OO, T. T.; PHYU, T. A Statistical Approach to Classify and Identify DDoS Attacks using UCLA Dataset. **International Journal of Advanced Research in Computer Engineering & Technology**, Índia, v. 2, n. 20, p. 1766-1700, mai. 2018.

PRIBERAM INFORMÁTICA. **Segurança**. Disponível em: https://dicionario.priberam.org/seguran%C3%A7a. Acesso em: 29 ago. 2022.

RIGHI, M. A.; NUNES, R. C. Detecção de ddos através da análise da recorrência baseada na extração de características dinâmicas. In: Simpósio Brasileiro em Segurança da Informação e Sistemas Computacionais, 2015, Florianópolis. Santa Catarina.

SARAVANAN, N.; GAYATHRI, V. Performance and classification evaluation of j48 algorithm and kendall's based j48 algorithm (knj48). **International Journal of Computer Trends and Technology**, n.2, v. 59. 2018. p. 73-80.

SERPRO. Snort: ferramenta livre garante segurança na Rede Serpro. **Governo federal.** 10 set. 2008. Disponível em: <http://intra.serpro.gov.br/noticias/snort-ferramenta-livre-garante-seguranca-na-rede-serpro>. Acesso em: 11 ago. 2020.

SHARAFALDIN, I.; *et al*. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS SECURITY AND PRIVACY, 2018, Funchal Portugal. **Proceedings…** Funchal (Portugal): SciTePress, 2018.

TANENBAUM, A. S. **Redes de computadores**. 4. ed. São Paulo: Pearson, 2011

TAROUCO, L. M. R. **Inteligência Artificial Aplicada a Rede de Computadores.** 203 f. 1990. Tese (Doutorado) – Programa de Pós-Graduação em Tecnologia, Universidade Federal do Rio Grande do Sul. São Paulo, 1990.

VALORINVESTE. **Ciberataques a dados bancários cresceram 141% no Brasil em 2021**. Disponível em: https://valorinveste.globo.com/mercados/renda-variavel/empresas/noticia/2022/03/24/ciberataques-a-dados-bancarios-cresceram-141percent-no-brasil-em-2021.ghtml. Acesso em: 14 set. 2022.

VERISIGN**, Como ocorre um ataque ddos**. Disponível em: https://www.verisign.com/pt_br/security-services/ddos-protection/how-does-a-ddos-attack-work/index.xhtml. Acesso em 13 set 2022.

WEKA, Machine Learning Software in Java. Disponível em: https://www.cs.waikato.ac.nz/ml/weka/. Acesso em 20 jun 2022.

WITTEN, I.; FRANK, E. **Data mining: practical machine learning tools and techiniques**. 2. ed. San Francisco: Mk, 2006

APÊNDICE A - Atributos do CICID2017

Atributos do CICID2017

Nome dos atributos	Descrição
1. Destination Port	Porta de destino
2. Flow Duration	Duração do fluxo
3. Total Fwd Packets	Total de pacotes enviados
4. Total Backward Packets	Total de pacotes recebidos
5. Total Length of Fwd Packets	Tamanho total de pacotes enviados
6.Total Lenght of Bwd Packets	Tamanho total de pacotes recebidos
7.Fwd Packet Length Max	Tamanho máximo dos pacotes enviados
8. Fwd Packet Length Min	Tamanho mínimo dos pacotes enviados
9. Fwd Packet Length Mean	Tamanho médio dos pacotes enviados
10. Fwd Packet Length Std	Tamanho do desvio padrão dos pacotes enviados
11. Bwd Packet Length Max	Tamanho máximo dos pacotes recebidos
12. Bwd Packet Length Min	Tamanho mínimo dos pacotes recebidos
13. Bwd Packet Length Mean	Tamanho médio dos pacotes recebidos
14. Bwd Packet Length Std	Tamanho do desvio padrão dos pacotes recebidos
15. Flow Bytes/s	Fluxo de bytes por segundo
16. Flow Packets/s	Fluxo de pacotes por segundo
17. Flow IAT Mean	
18.Flow IAT Std	
19.Flow IAT Max	
20.Flow IAT Min	
21. Fwd IAT Total	
22. Fwd IAT Mean	
23. Fwd IAT Std	
24. Fwd IAT Max	
25. Fwd IAT Min	
26. Bwd IAT Total	
27. Bwd IAT Mean	
28. Bwd IAT Std	
29. Bwd IAT Max	
30. Bwd IAT Min	
31. Fwd PSH Flags	Flag’s enviadas dos envios imediato de dados
32. Bwd PSH Flags	Flag’s recebidas dos envios imediato de dados
33. Fwd URG Flags	Urgências enviadas das flags
34. Bwd URG Flags	Urgências recebidas das flags
35. Fwd Header Length	Tamanho do cabeçalho enviado
36. Bwd Header Length	Tamanho do cabeçalho recebido
37. Fwd Packets/s	Pacotes enviados por segundo
38. Bwd Packets/s	Pacotes recebidos por segundo
39. Min Packet Length	Tamanho mínimo dos pacotes
40. Max Packet Length	Tamanho máximo dos pacotes
41. Packet Length Mean	Tamanho médio dos pacotes

42. Packet Length Std	Tamanho do desvio padrão dos pacotes
43. Packet Length Variance	Tamanho da variância dos pacotes
44. FIN Flag Count	Flag’s das quantidades de conexões finalizadas
45. SYN Flag Count	Flag’s das quantidades de conexões estabelecidas
46. RST Flag Count	Flag’s das quantidades de conexões reiniciadas
47. PSH Flag Count	Flag’s das quantidades de envio imediato de dados
48. ACK Flag Count	Flag’s das quantidades de sincronização dos números de sequência válidos
49. URG Flag Count	Flag’s das quantidades de urgências
50. CWE Flag Count	
51. ECE Flag Count	
52. Down/UP Ratio	Relação de bytes enviados e recebidos
53. Average Packet Size	Tamanho médio de pacotes em cada fila
54. Avg Fwd Segment Size	Tamanho médio de pacotes enviados em cada fila
55. Avg Bwd Segment Size	Tamanho médio de pacote recebidos em cada fila
56. Fwd Header Length1	Tamanho do cabeçalho enviado
57. Fwd Avg Bytes/Bulk	Tamanho médio do volume de bytes enviados.
58. Fwd Avg Packets/Bulk	Tamanho médio do volume de pacotes recebidos
59. Fwd Avg Bulk Rate	Tamanho médio da taxa de volume enviado
60. Bwd Avg Bytes/Bulk	Tamanho médio do volume de bytes recebidos
61. Bwd Avg Packets/Bulk	Tamanho médio do volume de pacotes recebidos
62. Bwd Avg Bulk Rate	Tamanho médio da taxa de volume recebido
63. Subflow Fwd Packets	Subfluxo de pacotes enviados
64. Subflow Fwd Bytes	Subfluxo de bytes enviados
65. Subflow Bwd Packets	Subfluxo de pacotes recebidos
66 Subflow Bwd Bytes	Subfluxo de bytes recebidos
67. Init_Win_bytes_forward	
68. Init_Win_bytes_backward	
69. act_data_pkt_fwd	
70. min_seg_size_forward	
71. Active Mean	Ativo médio
72. Active Std	Ativo de desvio padrão
73. Active Max	Ativo máximo
74. Active Min	Ativo minímo
75. Idle Mean	Inatividade média
76. Idle Std	Inatividade de desvio padrão
77. Idle Max	Inatividade máxima
78. . Idle Min	Inatividade mínima
79. Label	Referente ao tipo de ataque se é Benigno ou DDOS.

APÊNDICE B - Regras de Decisão RandomTree Completa

Regras de Decisão RandomTree

Regras de decisão RandomTree
==========

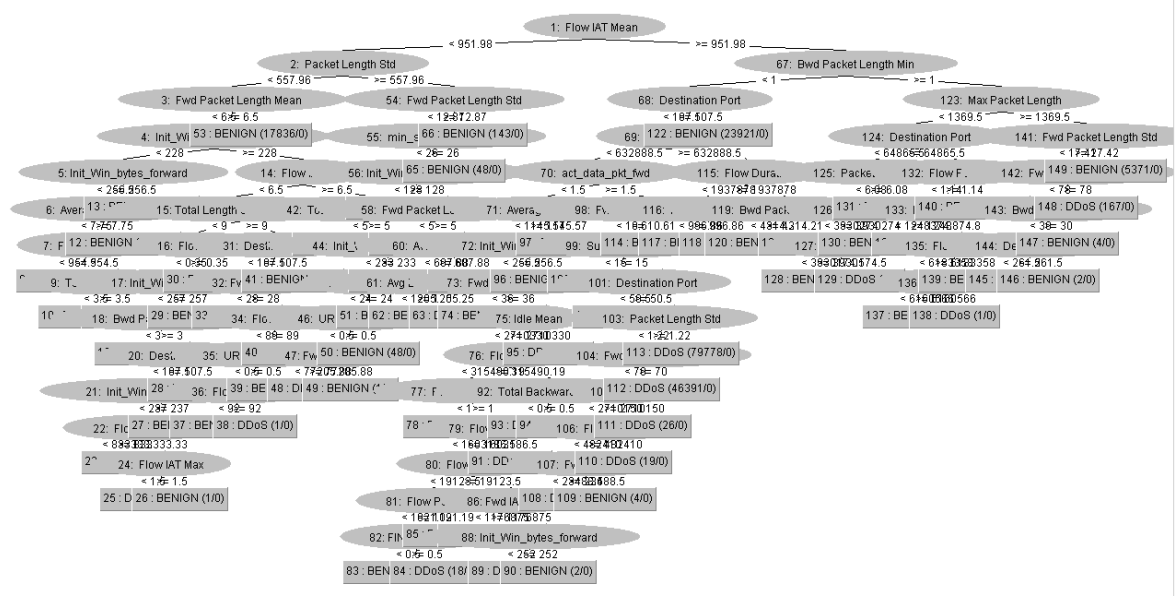
```
 Flow IAT Mean < 951.98
|   Packet Length Std < 557.96
|   |   Fwd Packet Length Mean < 6.5
|   |   |   Init_Win_bytes_backward < 228
|   |   |   |   Init_Win_bytes_forward < 256.5
|   |   |   |   |   Average Packet Size < 7.75
|   |   |   |   |   |   Flow Duration < 954.5 : BENIGN (323/0)
|   |   |   |   |   |   Flow Duration >= 954.5
|   |   |   |   |   |   |   Total Fwd Packets < 3.5 : BENIGN (4/0)
|   |   |   |   |   |   |   Total Fwd Packets >= 3.5 : DDoS (2/0)
|   |   |   |   |   Average Packet Size >= 7.75 : BENIGN (1989/0)
|   |   |   |   Init_Win_bytes_forward >= 256.5 : BENIGN (2722/0)
|   |   |   Init_Win_bytes_backward >= 228
|   |   |   |   Flow IAT Min < 6.5
|   |   |   |   |   Total Length of Fwd Packets < 9
|   |   |   |   |   |   Flow IAT Std < 0.35
|   |   |   |   |   |   |   Init_Win_bytes_forward < 257
|   |   |   |   |   |   |   |   Bwd Packet Length Min < 3 : BENIGN (6/0)
|   |   |   |   |   |   |   |   Bwd Packet Length Min >= 3
|   |   |   |   |   |   |   |   |   Destination Port < 107.5
|   |   |   |   |   |   |   |   |   |   Init_Win_bytes_backward < 237
|   |   |   |   |   |   |   |   |   |   |   Flow Packets/s < 833333.33 : DDoS (6/0)
|   |   |   |   |   |   |   |   |   |   |   Flow Packets/s >= 833333.33
|   |   |   |   |   |   |   |   |   |   |   |   Flow IAT Max < 1.5 : DDoS (2/0)
|   |   |   |   |   |   |   |   |   |   |   |   Flow IAT Max >= 1.5 : BENIGN (1/0)
|   |   |   |   |   |   |   |   |   |   Init_Win_bytes_backward >= 237 : BENIGN (11/0)
|   |   |   |   |   |   |   |   |   Destination Port >= 107.5 : BENIGN (37/0)
|   |   |   |   |   |   |   Init_Win_bytes_forward >= 257 : BENIGN (81/0)
|   |   |   |   |   |   Flow IAT Std >= 0.35 : BENIGN (251/0)
|   |   |   |   |   Total Length of Fwd Packets >= 9
|   |   |   |   |   |   Destination Port < 107.5
|   |   |   |   |   |   |   Fwd IAT Total < 28 : DDoS (51/0)
|   |   |   |   |   |   |   Fwd IAT Total >= 28
|   |   |   |   |   |   |   |   Flow IAT Max < 89
|   |   |   |   |   |   |   |   |   URG Flag Count < 0.5
|   |   |   |   |   |   |   |   |   |   Flow Duration < 92 : BENIGN (1/0)
|   |   |   |   |   |   |   |   |   |   Flow Duration >= 92 : DDoS (1/0)
|   |   |   |   |   |   |   |   |   URG Flag Count >= 0.5 : BENIGN (1/0)
|   |   |   |   |   |   |   |   Flow IAT Max >= 89 : BENIGN (18/0)
|   |   |   |   |   |   Destination Port >= 107.5 : BENIGN (31/0)
|   |   |   |   Flow IAT Min >= 6.5
|   |   |   |   |   Total Length of Bwd Packets < 5 : BENIGN (1408/0)
|   |   |   |   |   Total Length of Bwd Packets >= 5
|   |   |   |   |   |   Init_Win_bytes_backward < 233
```

```
|   |   |   |   |   |   |   Bwd IAT Max < 24
|   |   |   |   |   |   |   |   URG Flag Count < 0.5
|   |   |   |   |   |   |   |   |   Fwd Packets/s < 77205.88 : DDoS (15/0)
|   |   |   |   |   |   |   |   |   Fwd Packets/s >= 77205.88 : BENIGN (1/0)
|   |   |   |   |   |   |   |   URG Flag Count >= 0.5 : BENIGN (48/0)
|   |   |   |   |   |   |   Bwd IAT Max >= 24 : BENIGN (27/0)
|   |   |   |   |   |   Init_Win_bytes_backward >= 233 : BENIGN (2515/0)
|   |   Fwd Packet Length Mean >= 6.5 : BENIGN (17836/0)
|    Packet Length Std >= 557.96
|   |   Fwd Packet Length Std < 12.87
|   |   |   min_seg_size_forward < 26
|   |   |   |   Init_Win_bytes_forward < 128 : BENIGN (31/0)
|   |   |   |   Init_Win_bytes_forward >= 128
|   |   |   |   |   Fwd Packet Length Mean < 5 : BENIGN (4/0)
|   |   |   |   |   Fwd Packet Length Mean >= 5
|   |   |   |   |   |   Average Packet Size < 607.88
|   |   |   |   |   |   |   Avg Bwd Segment Size < 1205.25 : BENIGN (2/0)
|   |   |   |   |   |   |   Avg Bwd Segment Size >= 1205.25 : DDoS (25/0)
|   |   |   |   |   |   Average Packet Size >= 607.88 : DDoS (65/0)
|   |   |   min_seg_size_forward >= 26 : BENIGN (48/0)
|   |   Fwd Packet Length Std >= 12.87 : BENIGN (143/0)
 Flow IAT Mean >= 951.98
|    Bwd Packet Length Min < 1
|   |   Destination Port < 107.5
|   |   |   Bwd IAT Mean < 632888.5
|   |   |   |   act_data_pkt_fwd < 1.5
|   |   |   |   |   Average Packet Size < 1145.57
|   |   |   |   |   |   Init_Win_bytes_forward < 256.5
|   |   |   |   |   |   |   Fwd Header Length1 < 36 : BENIGN (271/0)
|   |   |   |   |   |   |   Fwd Header Length1 >= 36
|   |   |   |   |   |   |   |   Idle Mean < 2710330
|   |   |   |   |   |   |   |   |   Flow IAT Std < 315490.19
|   |   |   |   |   |   |   |   |   |   Packet Length Mean < 1 : BENIGN (84/0)
|   |   |   |   |   |   |   |   |   |   Packet Length Mean >= 1
|   |   |   |   |   |   |   |   |   |   |   Flow IAT Max < 1603186.5
|   |   |   |   |   |   |   |   |   |   |   |   Flow IAT Mean < 19123.5
|   |   |   |   |   |   |   |   |   |   |   |   |   Flow Packets/s < 1021.19
|   |   |   |   |   |   |   |   |   |   |   |   |   |   FIN Flag Count < 0.5 : BENIGN (105/0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   FIN Flag Count >= 0.5 : DDoS (18/0)
|   |   |   |   |   |   |   |   |   |   |   |   |   Flow Packets/s >= 1021.19 : BENIGN (27/0)
|   |   |   |   |   |   |   |   |   |   |   |   Flow IAT Mean >= 19123.5
|   |   |   |   |   |   |   |   |   |   |   |   |   Fwd IAT Total < 1176875 : BENIGN (220/0)
|   |   |   |   |   |   |   |   |   |   |   |   |   Fwd IAT Total >= 1176875
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Init_Win_bytes_forward < 252 : DDoS (1/0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Init_Win_bytes_forward >= 252 : BENIGN (2/0)
|   |   |   |   |   |   |   |   |   |   |   Flow IAT Max >= 1603186.5 : DDoS (8/0)
|   |   |   |   |   |   |   |   |   Flow IAT Std >= 315490.19
|   |   |   |   |   |   |   |   |   |   Total Backward Packets < 0.5 : DDoS (7/0)
|   |   |   |   |   |   |   |   |   |   Total Backward Packets >= 0.5 : BENIGN (28/0)
|   |   |   |   |   |   |   |   Idle Mean >= 2710330 : DDoS (28/0)
```

```
|   |   |   |   |   |   Init_Win_bytes_forward >= 256.5 : BENIGN (2124/0)
|   |   |   |   |   Average Packet Size >= 1145.57 : DDoS (1360/0)
|   |   |   |   act_data_pkt_fwd >= 1.5
|   |   |   |   |   Fwd Packet Length Std < 10.61
|   |   |   |   |   |   Subflow Fwd Bytes < 15 : BENIGN (620/0)
|   |   |   |   |   |   Subflow Fwd Bytes >= 15
|   |   |   |   |   |   |   Destination Port < 50.5 : BENIGN (1/0)
|   |   |   |   |   |   |   Destination Port >= 50.5
|   |   |   |   |   |   |   |   Packet Length Std < 1.22
|   |   |   |   |   |   |   |   |   Fwd Header Length1 < 70
|   |   |   |   |   |   |   |   |   |   Idle Min < 2710150
|   |   |   |   |   |   |   |   |   |   |   Flow Duration < 482410
|   |   |   |   |   |   |   |   |   |   |   |   Fwd IAT Mean < 23488.5 : DDoS (10/0)
|   |   |   |   |   |   |   |   |   |   |   |   Fwd IAT Mean >= 23488.5 : BENIGN (4/0)
|   |   |   |   |   |   |   |   |   |   |   Flow Duration >= 482410 : DDoS (19/0)
|   |   |   |   |   |   |   |   |   |   Idle Min >= 2710150 : DDoS (26/0)
|   |   |   |   |   |   |   |   |   Fwd Header Length1 >= 70 : DDoS (46391/0)
|   |   |   |   |   |   |   |   Packet Length Std >= 1.22 : DDoS (79778/0)
|   |   |   |   |   Fwd Packet Length Std >= 10.61 : BENIGN (1107/0)
|   |   |   Bwd IAT Mean >= 632888.5
|   |   |   |   Flow Duration < 1937878
|   |   |   |   |   Packet Length Mean < 996.86 : BENIGN (5/0)
|   |   |   |   |   Packet Length Mean >= 996.86 : DDoS (28/0)
|   |   |   |   Flow Duration >= 1937878
|   |   |   |   |   Bwd Packet Length Std < 4314.21 : BENIGN (2457/0)
|   |   |   |   |   Bwd Packet Length Std >= 4314.21 : DDoS (8/0)
|   |   Destination Port >= 107.5 : BENIGN (23921/0)
|   Bwd Packet Length Min >= 1
|   |   Max Packet Length < 1369.5
|   |   |   Destination Port < 64865.5
|   |   |   |   Packet Length Mean < 6.08
|   |   |   |   |   Flow IAT Max < 3930274
|   |   |   |   |   |   Flow IAT Max < 3930174.5 : BENIGN (4935/0)
|   |   |   |   |   |   Flow IAT Max >= 3930174.5 : DDoS (1/0)
|   |   |   |   |   Flow IAT Max >= 3930274 : BENIGN (9649/0)
|   |   |   |   Packet Length Mean >= 6.08 : BENIGN (18999/0)
|   |   |   Destination Port >= 64865.5
|   |   |   |   Flow Packets/s < 1.14
|   |   |   |   |   Bwd IAT Mean < 1248374.8 : DDoS (1/0)
|   |   |   |   |   Bwd IAT Mean >= 1248374.8
|   |   |   |   |   |   Flow Duration < 6183358
|   |   |   |   |   |   |   Idle Mean < 6160566 : BENIGN (14/0)
|   |   |   |   |   |   |   Idle Mean >= 6160566 : DDoS (1/0)
|   |   |   |   |   |   Flow Duration >= 6183358 : BENIGN (97/0)
|   |   |   |   Flow Packets/s >= 1.14 : BENIGN (132/0)
|   |   Max Packet Length >= 1369.5
|   |   |   Fwd Packet Length Std < 17.42
|   |   |   |   Fwd Header Length1 < 78
|   |   |   |   |   Bwd Header Length < 30
|   |   |   |   |   |   Destination Port < 261.5 : DDoS (8/0)
```

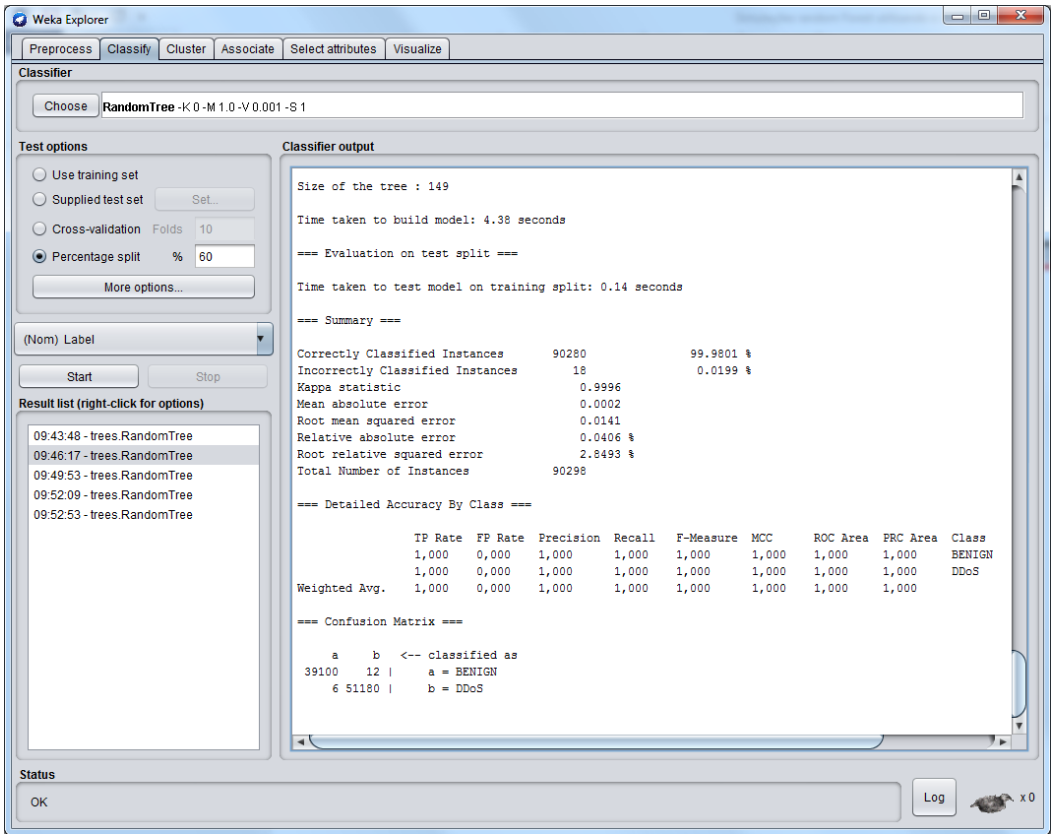
```
|   |   |   |   |   |    Destination Port >= 261.5 : BENIGN (2/0)
|   |   |   |   |    Bwd Header Length >= 30 : BENIGN (4/0)
|   |   |   |    Fwd Header Length1 >= 78 : DDoS (167/0)
|   |   |    Fwd Packet Length Std >= 17.42 : BENIGN (5371/0)
```

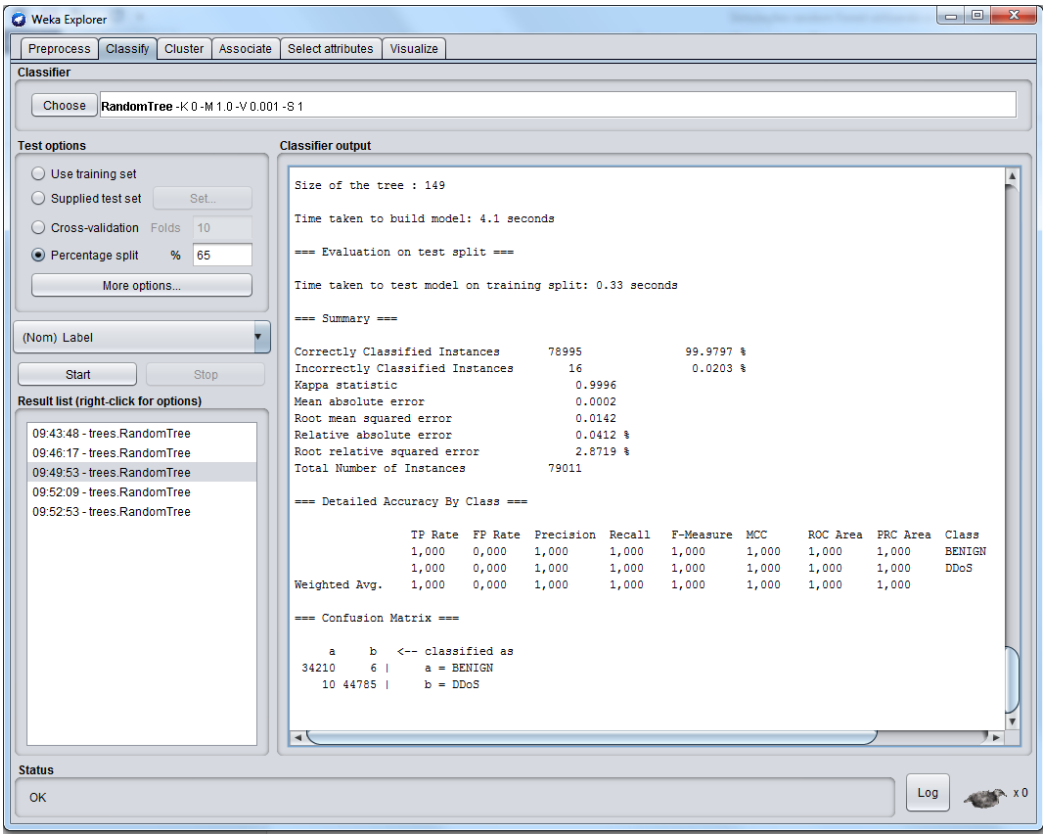
APÊNDICE C - Árvore de Decisão RandomTree Completa

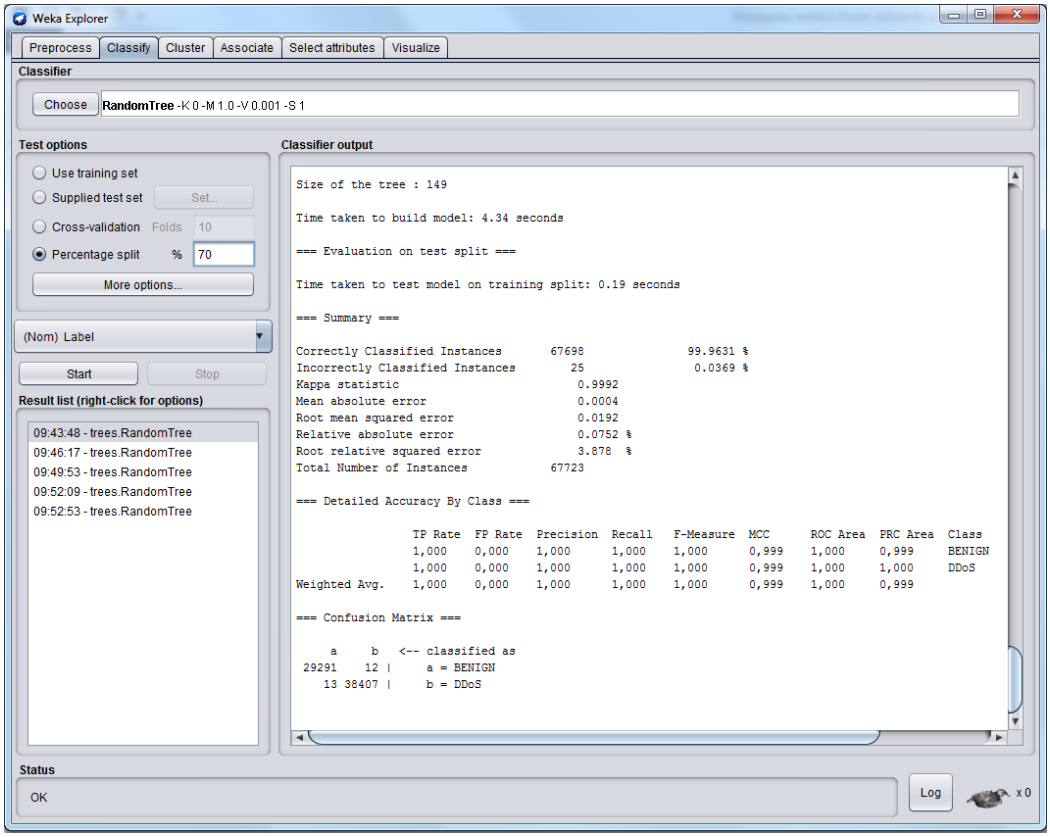

1: Flow IAT Mean
< 951.98
>= 951.98
2: Packet Length Std
67: Bwd Packet Length Min
< 557.96
>= 557.96
< 1
>= 1
3: Fwd Packet Length Mean
54: Fwd Packet Length Std
68: Destination Port
123: Max Packet Length
< 6.5
< 1272.87
< 1369.5
>= 1369.5
4: Init_Wi
53 : BENIGN (17836/0)
55: min_s
66 : BENIGN (143/0)
69:
122 : BENIGN (23921/0)
124: Destination Port
141: Fwd Packet Length Std
< 228
>= 228
< 26
< 632888.5
>= 632888.5
< 17.42
5: Init_Win_bytes_forward
14: Flow
56: Init_Wi
65 : BENIGN (48/0)
70: act_data_pkt_fwd
115: Flow Dura
125: Packe
132: Flow F
142: Fw
149 : BENIGN (5371/0)
< 6.5
>= 6.5
< 128
< 1.5
>= 1.5
< 1.14
< 78
>= 78
6: Aver
15: Total Length
42: To
58: Fwd Packet L
71: Averag
98: Fv
116:
119: Bwd Pac
126
131
133:
140 : DD
143: Bwd
148 : DDoS (167/0)
< 9
>= 9
< 5
>= 5
< 30
>= 30
7: F
12 : BENIGN
16: Flo
31: Dest
44: Init_
60: A
72: Init_Wi
97
99: Su
114 : B
117 : BI
118
120 : BEN
127:
130 : BEN
135: Fl
144: De
147 : BENIGN (4/0)
< 15
>= 15
9: T
17: Init_Wi
30 :
32: Fv
41 : BENIGN
61: Avg
73: Fwd
96 : BENIC
101: Destination Port
128 : BEN
129 : DDoS
136
139 : BE
145 :
146 : BENIGN (2/0)
< 3.5
>= 3.5
< 257
>= 257
< 28
>= 28
< 24
>= 24
< 36
>= 36
< 58
>= 58
18: Bwd P
29 : BEN
34: Flo
46: UR
51 : B
62 : BE
63 :
74 : BE
75: Idle Mean
103: Packet Length Std
137 : BE
138 : DDoS (1/0)
< 3
>= 3
< 89
>= 89
< 0.5
>= 0.5
< 1.21
>= 1.21
20: Dest
35: UR
40
47: Fw
50 : BENIGN (48/0)
76: Flo
95 : DD
104: Fwd
113 : DDoS (79778/0)
< 78
>= 78
21: Init_Win
28
36: Flo
39 : BE
48 : DI
49 : BENIGN (
77: F
92: Total Backwar
10
112 : DDoS (46391/0)
< 237
>= 237
< 92
>= 92
< 1
>= 1
22: Flo
27 : BEI
37 : BEI
38 : DDoS (1/0)
78
79: Flo
93 : D
106: Fl
111 : DDoS (26/0)
24: Flow IAT Max
80: Flow
91 : DD
107: Fv
110 : DDoS (19/0)
< 1.5
>= 1.5
25 : D
26 : BENIGN (1/0)
81: Flow P
86: Fwd IA
108 :
109 : BENIGN (4/0)
82: Fl
85
88: Init_Win_bytes_forward
< 0.5
>= 0.5
< 252
>= 252
83 : BEN
84 : DDoS (18/
89 : D
90 : BENIGN (2/0)

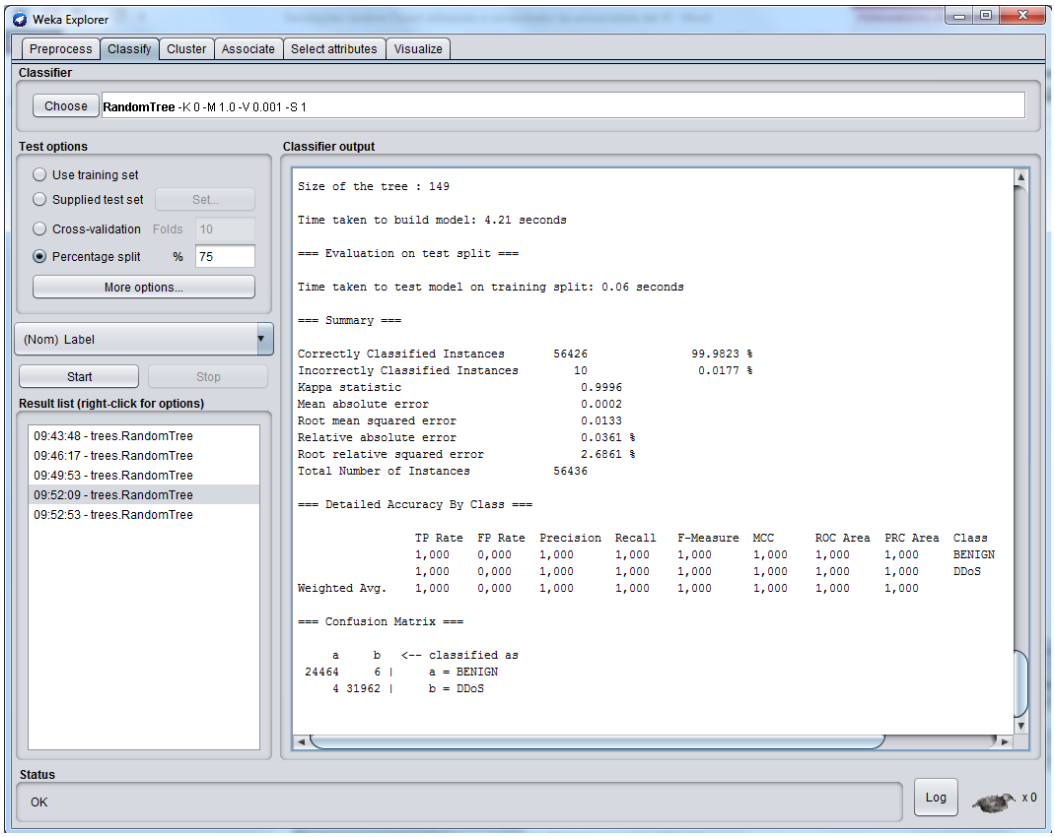
APÊNDICE D - Testes utilizando RandomTree

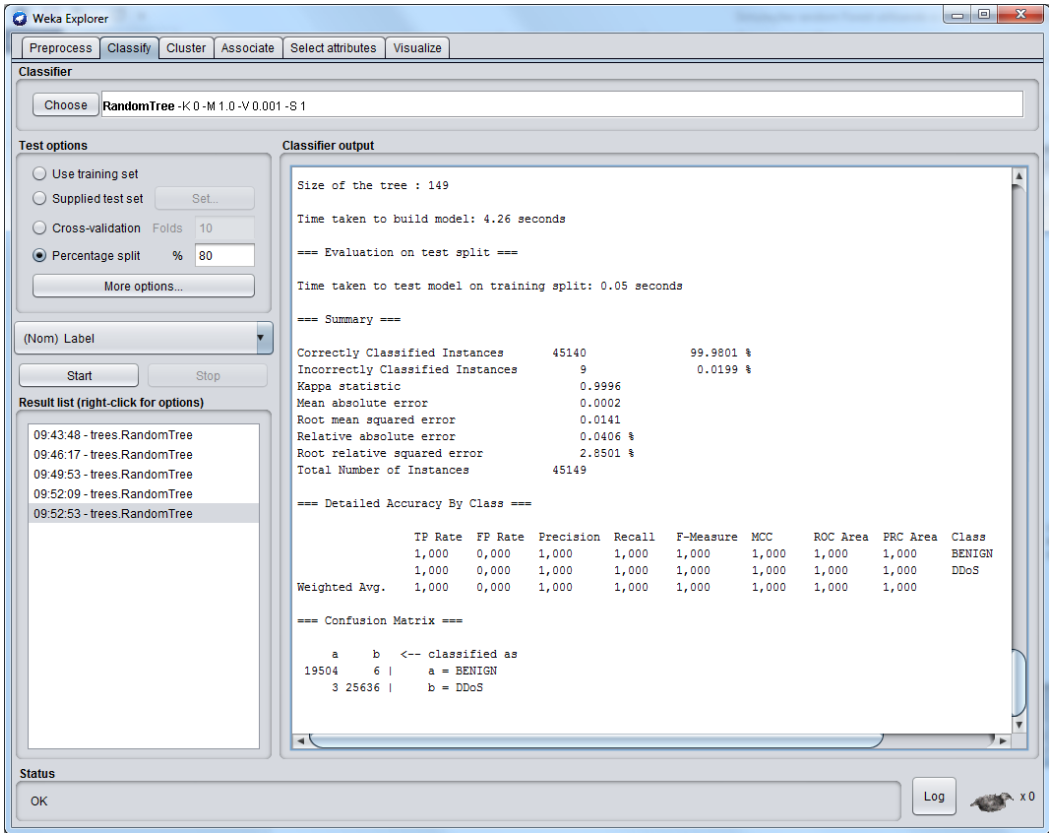
Testes do dataset CICIDs2017 utilizando Random Tree






Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomTree -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
70
More options...
(Nom) Label
Start
Stop
Result list (right-click for options)
09:43:48 - trees.RandomTree
09:46:17 - trees.RandomTree
09:49:53 - trees.RandomTree
09:52:09 - trees.RandomTree
09:52:53 - trees.RandomTree
Classifier output
Size of the tree : 149
Time taken to build model: 4.34 seconds
=== Evaluation on test split ===
Time taken to test model on training split: 0.19 seconds
=== Summary ===
Correctly Classified Instances 67698 99.9631 %
Incorrectly Classified Instances 25 0.0369 %
Kappa statistic 0.9992
Mean absolute error 0.0004
Root mean squared error 0.0192
Relative absolute error 0.0752 %
Root relative squared error 3.878 %
Total Number of Instances 67723
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1,000 0,000 1,000 1,000 1,000 0,999 1,000 0,999 BENIGN
1,000 0,000 1,000 1,000 1,000 0,999 1,000 1,000 DDoS
Weighted Avg. 1,000 0,000 1,000 1,000 1,000 0,999 1,000 0,999
=== Confusion Matrix ===
a b <-- classified as
29291 12 | a = BENIGN
13 38407 | b = DDoS
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomTree -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
75
More options...
(Nom) Label
Start
Stop
Result list (right-click for options)
09:43:48 - trees.RandomTree
09:46:17 - trees.RandomTree
09:49:53 - trees.RandomTree
09:52:09 - trees.RandomTree
09:52:53 - trees.RandomTree
Classifier output
Size of the tree : 149
Time taken to build model: 4.21 seconds
=== Evaluation on test split ===
Time taken to test model on training split: 0.06 seconds
=== Summary ===
Correctly Classified Instances 56426 99.9823 %
Incorrectly Classified Instances 10 0.0177 %
Kappa statistic 0.9996
Mean absolute error 0.0002
Root mean squared error 0.0133
Relative absolute error 0.0361 %
Root relative squared error 2.6861 %
Total Number of Instances 56436
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 BENIGN
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 DDoS
Weighted Avg. 1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000
=== Confusion Matrix ===
a b <-- classified as
24464 6 | a = BENIGN
4 31962 | b = DDoS
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
RandomTree -K 0 -M 1.0 -V 0.001 -S 1
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
80
More options...
(Nom) Label
Start
Stop
Result list (right-click for options)
09:43:48 - trees.RandomTree
09:46:17 - trees.RandomTree
09:49:53 - trees.RandomTree
09:52:09 - trees.RandomTree
09:52:53 - trees.RandomTree
Classifier output
Size of the tree : 149
Time taken to build model: 4.26 seconds
=== Evaluation on test split ===
Time taken to test model on training split: 0.05 seconds
=== Summary ===
Correctly Classified Instances 45140 99.9801 %
Incorrectly Classified Instances 9 0.0199 %
Kappa statistic 0.9996
Mean absolute error 0.0002
Root mean squared error 0.0141
Relative absolute error 0.0406 %
Root relative squared error 2.8501 %
Total Number of Instances 45149
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 BENIGN
1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000 DDoS
Weighted Avg. 1,000 0,000 1,000 1,000 1,000 1,000 1,000 1,000
=== Confusion Matrix ===
a b <-- classified as
19504 6 | a = BENIGN
3 25636 | b = DDoS
Status
OK
Log
x 0

APÊNDICE E - Testes utilizando Random Forest

Testes do dataset CICIDs2017 utilizando Random Forest

```
Test mode:    split 60.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 184.09 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 1 seconds

=== Summary ===

Correctly Classified Instances       90294               99.9956 %
Incorrectly Classified Instances         4                0.0044 %
Kappa statistic                          0.9999
Mean absolute error                      0.0004
Root mean squared error                  0.0085
Relative absolute error                  0.0744 %
Root relative squared error              1.7207 %
Total Number of Instances            90298

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     BENIGN
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     DDoS
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

     a     b   <-- classified as
 39111     1 |     a = BENIGN
     3 51183 |     b = DDoS
```

```
Test mode:    split 65.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 184.53 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 0.86 seconds

=== Summary ===

Correctly Classified Instances       79005               99.9924 %
Incorrectly Classified Instances         6                0.0076 %
Kappa statistic                          0.9998
Mean absolute error                      0.0004
Root mean squared error                  0.0092
Relative absolute error                  0.0766 %
Root relative squared error              1.8467 %
Total Number of Instances            79011

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     BENIGN
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     DDoS
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

     a     b   <-- classified as
 34215     1 |     a = BENIGN
     5 44790 |     b = DDoS
```

```
Test mode:    split 70.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 187.77 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 0.97 seconds

=== Summary ===

Correctly Classified Instances       67718               99.9926 %
Incorrectly Classified Instances         5                0.0074 %
Kappa statistic                          0.9998
Mean absolute error                      0.0003
Root mean squared error                  0.0079
Relative absolute error                  0.0605 %
Root relative squared error              1.5955 %
Total Number of Instances            67723

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     BENIGN
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     DDoS
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

     a     b   <-- classified as
 29302     1 |     a = BENIGN
     4 38416 |     b = DDoS
```

```
Test mode:    split 75.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 185.76 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 0.66 seconds

=== Summary ===

Correctly Classified Instances       56433               99.9947 %
Incorrectly Classified Instances         3                0.0053 %
Kappa statistic                          0.9999
Mean absolute error                      0.0003
Root mean squared error                  0.0074
Relative absolute error                  0.0563 %
Root relative squared error              1.4924 %
Total Number of Instances            56436

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     BENIGN
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     DDoS
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

     a     b   <-- classified as
 24469     1 |     a = BENIGN
     2 31964 |     b = DDoS
```

```
Test mode:    split 80.0% train, remainder test

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 186.22 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 0.49 seconds

=== Summary ===

Correctly Classified Instances       45147               99.9956 %
Incorrectly Classified Instances         2                0.0044 %
Kappa statistic                          0.9999
Mean absolute error                      0.0003
Root mean squared error                  0.0071
Relative absolute error                  0.0534 %
Root relative squared error              1.4358 %
Total Number of Instances            45149

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     BENIGN
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     DDoS
Weighted Avg.    1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000

=== Confusion Matrix ===

     a     b   <-- classified as
 19510     0 |     a = BENIGN
     2 25637 |     b = DDoS
```