**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO VITOR ANDREOSSI

# APLICAÇÃO DAS TÉCNICAS EXPLAINABLE ARTIFICIAL INTELLIGENCE E ENSEMBLE LEARNING PARA DETECÇÃO DE ATAQUES DDOS

BAURU
Fevereiro/2022

JOÃO VITOR ANDREOSSI

# APLICAÇÃO DAS TÉCNICAS EXPLAINABLE ARTIFICIAL INTELLIGENCE E ENSEMBLE LEARNING PARA DETECÇÃO DE ATAQUES DDOS

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

BAURU
Fevereiro/2022

A559a

Andreossi, João Vitor

Aplicação das técnicas Explainable Artificial Intelligence e Ensemble Learning para detecção de ataques DDoS / João Vitor Andreossi. -- Bauru, 2022

33 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru

Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

1. Sistemas de segurança. 2. Aprendizado do computador. 3. Engenharia de software. 4. Teoria da informação. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

João Vitor Andreossi

# Aplicação das técnicas Explainable Artificial Intelligence e Ensemble Learning para detecção de ataques DDoS

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

---

**Prof. Dr. Kelton Augusto Pontara da Costa**
Orientador
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

---

**Profa. Dra. Simone das Graças Domingues Prado**
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

---

**Profa. Dra. Márcia A. Zanoli Meira e Silva**
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Bauru, _____ de _____ de _____.

*Dedico este trabalho a todos que defendem a ciência diante do obscurantismo.*

# Agradecimentos

Agradeço meus pais por todo o amor, suporte, compreensão e confiança, sem vocês nada disso seria possível.

Agradeço a minha namorada pelo companheirismo, paciência, apoio e, principalmente, por acreditar em mim.

Agradeço aos professores e servidores da UNESP Bauru por trabalharem incansavelmente para expor jovens mentes às belezas da ciência.

E por fim, agradeço a todas as pessoas que cruzaram meu caminho de alguma forma durante todos esses (muitos) anos de universidade.

*Nada é permanente, exceto a mudança.*
Heráclito de Éfeso

# Resumo

Com o aumento exponencial de serviços ofertados digitalmente nas últimas décadas, muitas pessoas passaram a depender de serviços armazenados em servidores web, de maneira que a queda desses serviços, mesmo que por alguns minutos, pode afetar toda uma cadeia produtiva e gerar enormes prejuízos. Isso torna ataques do tipo DDoS especialmente perigosos, pois são difíceis de serem detectados e não necessitam de muitos recursos para serem executados. Esse trabalho propõe um sistema de detecção de ataques DDoS baseado na detecção de anomalias através do conceito de entropia da informação. As conexões do *dataset* CIDDS-001 foram separadas em janelas de tempo de dois minutos e categorizadas com o auxílio de modelos de classificação do tipo *ensemble*, caso a entropia da janela exceda o intervalo estabelecido. Por fim, foi realizada uma análise baseada em conceitos de Explainable Artificial Intelligence para entender melhor o funcionamento interno do modelo.

**Palavras-chave:** Sistemas de segurança. Aprendizado de computador. Engenharia de software. Teoria da informação.

# Abstract

With the exponential increase in digital services being offered in the last decades, a lot of people began to depend on services hosted on web servers, and this means that when the servers become unavailable, even for a few minutes, a whole supply chain can be affected and incur major economic losses. This makes DDoS attacks specially dangerous, as they are hard to detect and don't need many resources to carry out. This project proposes a DDoS detection system based on information theoretic entropy anomaly detection. The connection data from dataset CIDDS-001 was divided in two minutes time windows and categorized via ensemble classifiers, if the time window entropy exceeds a predetermined interval. Finally, Explainable Artificial Intelligence techniques are applied as a way to better understand the model inner workings.

**Keywords:** Security systems. Machine learning. Software engineering. Information theory.

# Lista de figuras

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| CIDDS | Coburg Intrusion Detection Data Sets |
| DDoS | Distributed Denial-of-Service |
| DoS | Denial-of-Service |
| ICMP | Internet Control Message Protocol |
| IP | Internet Protocol |
| RAM | Random Access Memory |
| SHAP | Shapley Additive Explanations |
| SSH | Secure Shell |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| XAI | Explainable Artificial Intelligence |

# Sumário

# 1 Introdução

Ataques do tipo DoS (negação de serviço, em inglês) tem como objetivo impedir a utilização de algum serviço ou funcionalidade específica da vítima. Sua variação, DDoS (negação de serviço distribuída, em inglês), tenta alcançar esse objetivo se utilizando de uma grande quantidade de computadores que executam vários ataques DoS simultaneamente (MIRKOVIC; REIHER, 2004).

A ideia do ataque é abrir um grande número de conexões com a máquina alvo, de forma a ocupar todos os seus recursos e impedir que o sistema possa atender a requisições de usuários legítimos. Como o ataque não viola diretamente a segurança do sistema, apenas ocupa todos os recursos disponíveis, métodos tradicionais de segurança da informação são ineficazes em prevenir esses tipos de ataques (DOULIGERIS; MITROKOTSA, 2004).

Em muitos casos é difícil diferenciar um ataque DDoS de um aumento de tráfego natural, mas existem vários métodos na literatura que tentam mitigar esse problema, como expostos no trabalho de Jaafar et al. (2019), onde foram analisadas várias técnicas de detecção de ataques DDoS. Muitas dessas técnicas detectam variações nos padrões das conexões, por exemplo, fazendo uso de análises estatísticas (H.ABORUJILAH; MUSA, 2017) ou classificação por algoritmos de aprendizado de máquina (SINGH; SINGH; KUMAR, 2018).

Este trabalho baseia-se na técnica proposta por Idhammad et al. (2018), que consiste em detectar anomalias no tráfego da rede dentro de uma janela de tempo pré-determinada utilizando o conceito de entropia da informação (SHANNON, 1948). Os dados de conexão são então classificados por modelos de aprendizado de máquina do tipo *ensemble* (OPITZ; MACLIN, 1999), com o intuito de confirmar se o sistema está de fato sob ataque.

O capítulo 2 é apresentada toda a fundamentação teórica em que o trabalho foi baseado. No capítulo 3 é descrita a metodologia e ferramentas utilizada nos experimentos. O capítulo 4 apresenta os resultados obtidos a partir dos experimentos e, finalmente, o capítulo 5 descreve as conclusões obtidas a partir da análise dos resultados.

## 1.1 Objetivos

A seguir, estão listados os objetivos gerais e específicos deste trabalho.

### 1.1.1 Objetivo Geral

Desenvolver um sistema de detecção de ataques DDoS utilizando conceitos de deteção de anomalias através de entropia e empregando modelos ensemble para classificação da base de

dados, se baseando no trabalho de Idhammad et al. (2018), além de analisar o funcionamento interno dos modelos com o auxílio de técnicas de *Explainable Artificial Intelligence*.

### 1.1.2 Objetivos Específicos

- Implementar e aplicar o conceito de entropia da informação para detectar anomalias no *dataset* escolhido;
- Processar e classificar o *dataset* utilizando modelos de *ensemble learning*;
- Avaliar o desempenho dos modelos através de matrizes de confusão e valores de acurácia;
- Analisar o processo de classificação com conceitos de *Explainable Artificial Intelligence*.

# 2 Fundamentação Teórica

## 2.1 Distributed Denial-of-Service

Um ataque DDoS é caracterizado pelo uso de muitos computadores para lançar um ataque coordenado a algum alvo específico, de maneira a consumir todos os recursos do servidor e impedir o acesso dos usuários legítimos de um serviço. A primeira etapa de um ataque é a fase de recrutamento, onde o perpetrador, abusa de falhas de segurança para infectar máquinas vulneráveis com o código de ataque. Esse processo costuma ser automatizado, e os computadores infectados podem ser usados para infectar outros computadores, e assim por diante (MIRKOVIC; REIHER, 2004).

De forma coordenada, todos esses computadores passam a emitir pacotes com o intuito de inundar o servidor de requisições, ocupando todos os recursos disponíveis, o que se reflete nos pedidos de conexão de outros usuários, que não conseguem mais serem atendidos. Os ataques DDoS não possuem nenhuma característica específica que os diferenciam de um pico de acessos, por isso a detecção não é simples. Além da similaridade com conexões normais, o tamanho dos pacotes também podem ser modificados, o que pode dificultar a classificação (DOULIGERIS; MITROKOTSA, 2004).

## 2.2 Detecção de anomalias

Detecção de anomalias se refere ao processo de encontrar padrões em um conjunto de dados que não condizem com o comportamento esperado. Esse padrões incomuns são chamados de anomalias. Essas anomalias podem ser pistas que revelam um problema maior, como por exemplo, uma anomalia no consumo de um cartão de crédito pode indicar fraude, ou uma anomalia nos dados de leitura de um sensor pode indicar um componente com defeito (CHANDOLA; BANERJEE; KUMAR, 2009).

O campo de deteção de anomalias é muito importante para a segurança da informação, e por isso, foi alvo de diversos estudos nas últimas décadas, levando ao desenvolvimento de vários métodos de detecção, incluindo os baseados no conceito de teoria de informação, que é o caso da detecção por entropia (AHMED; MAHMOOD; HU, 2016).

### 2.2.1 Entropia da Informação

O conceito de entropia da informação foi estabelecido por Shannon (1948) em seu famoso estudo *A Mathematical Theory of Communication*. A ideia é que, a partir de um

conjunto de eventos possíveis com probabilidades $p_i$, para que exista uma medida H que representa a incerteza do resultado, é necessário satisfazer três condições:

- H precisa ser contínua em $p_i$.
- Se todos os $p_i$ são igualmente prováveis, então $p_i = \frac{1}{n}$. Portanto H é uma função monótona crescente de n.
- H deve ser a soma ponderada dos valores anteriores.

A fórmula que satisfaz todas as condições anteriores é conhecida como Entropia de Shannon (SHANNON, 1948):

$$H = -\sum_{i=1}^{n} p_i log p_i \tag{2.1}$$

Lakhina, Crovella e Diot (2005) utilizaram a Entropia de Shannon para detectar anomalias utilizando a distribuição das características em uma amostra de tráfego de rede (IP de origem, IP de destino, portas, etc.), provando ser um método eficaz para encontrar anomalias em um conjunto de dados de conexão.

## 2.3 Modelos Ensemble

Segundo Thomas G. Dietterich, algoritmos de aprendizado de máquina tradicionais funcionam procurando todo o espaço de funções possíveis, chamadas hipóteses, buscando por uma função *h* que melhor aproxima uma função desconhecida *f*. Para determinar qual *h* é a melhor, o algoritmo testa a função com os dados de treinamento e compara quão próximo os valores são, e também quão consistente *h* é com funções selecionadas anteriormente (DIETTERICH, 2002).

Já modelos *ensemble*, ao invés de procurar apenas uma hipótese que melhor representa os dados, eles estabelecem um conjunto de hipóteses $h_1$, ... , $h_K$ e um peso para cada uma delas $w_1$, ... , $w_K$, e o próximo classificador é dado pela fórmula:

$$H(\mathbf{x}) = w_1 h_1(\mathbf{x}) + \cdots + w_K h_K(\mathbf{x}) \tag{2.2}$$

Sendo que o valor do classificador agregado H é $+1$ se H(x) $= 0$, ou -1 caso contrário. Por exemplo, em uma árvore de decisão, -1 indicaria a folha da esquerda e 1 a folha da direita (DIETTERICH, 2002).

Dietterich ainda diz que modelos *ensemble* resolvem três problemas importantes de modelos tradicionais:

- Quando o algoritmo precisa decidir entre várias hipóteses similares que resultam na mesma precisão para os dados de teste, não há garantia de que a hipótese funcione bem para valores futuros. Um voto entre todas as hipóteses possíveis pode ajudar a mitigar o problema.

- Quando o algoritmo não consegue encontrar a hipótese ótima dentro do espaço das hipóteses possíveis. Algumas heurísticas podem ficar presas no mínimo local e nunca encontrar uma solução ótima. Esse problema é minimizado com um *ensemble* que seleciona entre vários mínimos locais.

- Quando o espaço das hipóteses não contém nenhuma hipótese que é uma boa representação da função real. Nesses casos, a soma ponderada das hipóteses pode estender o espaço das hipóteses que podem ser representadas. Naturalmente, um voto ponderado das hipóteses pode ajudar o algoritmo a escolher uma aproximação mais precisa.

Assim, modelos *ensemble* podem reduzir problemas comuns de modelos de classificação mais tradicionais, gerando um resultado mais preciso (DIETTERICH, 2002).

## 2.4 Explainable Artificial Intelligence

Os recentes avanços no campo de aprendizado de máquina trouxeram um grande aumento no interesse em aplicações que implementam algoritmos de aprendizado. Mas esses modelos tendem a agir como uma caixa preta em que não se sabe ao certo quais passos foram tomados até um resultado ser alcançado. Em muitas situações não existe a necessidade de toda essa transparência, mas em áreas sensíveis como defesa nacional, saúde pública ou economia, é imprescindível que os usuários possam confiar no resultado apresentado (GUNNING et al., 2019).

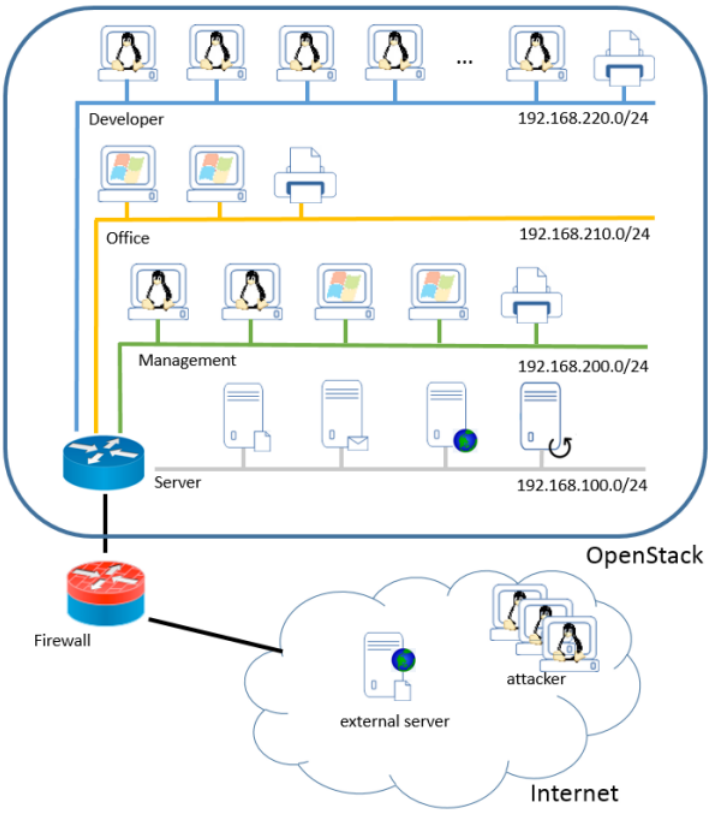
É possível dividir o conceito de interpretabilidade de um modelo em dois tipos: *ante-hoc* e *post-hoc*. Interpretabilidade *ante-hoc* ("antes disso", em latim) busca estabelecer um modelo do tipo "caixa de vidro", onde o modelo é naturalmente transparente e todas as informações são claras e apresentadas de uma maneira em que um humano pode facilmente interagir e entender o algoritmo (HOLZINGER et al., 2017).

*Post-hoc* ("depois disso", em latim) significa explicar o resultado de um modelo após sua execução, sem necessariamente elucidar seu funcionamento interno, apenas utilizando as informações disponíveis para chegar em alguma conclusão. Algumas técnicas que empregam interpretabilidade *post-hoc* são visualizações por mapas de saliência em redes neurais, ou "explicação por exemplo", onde resultados anteriores são utilizados como evidência do porque o modelo chegou em tal resultado (LIPTON, 2018).

## 2.5 Conjunto de dados CIDDS-001

O conjunto de dados CIDDS-001 é um conjunto de dados de teste criado para sistemas de detecção de intrusão baseados em anomalias. Ele foi desenvolvido em um ambiente virtual, com o objetivo de ser um *dataset* baseado em fluxos, atualizado e catalogado. Os dados foram criados simulando um pequeno ambiente corporativo, onde existem vários clientes e servidores, como de *e-mail* e *web*. Um servidor externo com IP público também foi posto *online*, assim foi possível capturar tipos de ataques reais e recentes (RING et al., 2017).

Figura 1 – Arquitetura do ambiente simulado do *dataset* CIDDS-001.




Fonte: Ring et al. (2017)

Como é possível ver na Figura 1, a arquitetura da rede simulada contém um roteador que conecta quatro sub redes entre elas e também à internet, contendo também quatro servidores internos. Existe também um servidor externo que oferece serviços via internet (RING et al., 2017).

Os atributos de número 1 a 10 na Tabela 1 são atributos padrões da conexão, enquanto os atributos 11 a 14 foram adicionados pelos pesquisadores como forma de categorização do *dataset*. O atributo *class* determina a qual categoria cada conexão pertence, para fins de classificação, e apenas conexões do tipo *attacker* utilizam os outros três atributos, que trazem mais informações sobre a natureza do ataque (RING et al., 2017).

Tabela 1 – Atributos do *dataset* CIDDS-001

| **Nr.** | **Nome** | **Descrição** |
|---|---|---|
| *1* | Src IP | Endereço IP de origem |
| *2* | Src Port | Porta de origem |
| *3* | Dest IP | Endereço IP de destino |
| *4* | Dest Port | Porta de destino |
| *5* | Proto | Protocolo de transporte (ex: ICMP, TCP ou UDP) |
| *6* | Date first seen | Horário em que fluxo foi avistado |
| *7* | Duration | Duração do fluxo |
| *8* | Bytes | Número de bytes transmitidos |
| *9* | Packets | Número de pacotes transmitidos |
| *10* | Flags | Concatenação de todas as flags TCP |
| *11* | Class | *Label* de classe (normal, attacker, victim, suspicious or unknown) |
| *12* | AttackType | Tipo do ataque (portScan, dos, bruteForce, ...) |
| *13* | AttackID | Idêntificador único de ataque. |
| *14* | AttackDescription | Informações adicionais sobre os parâmetros de ataque. |

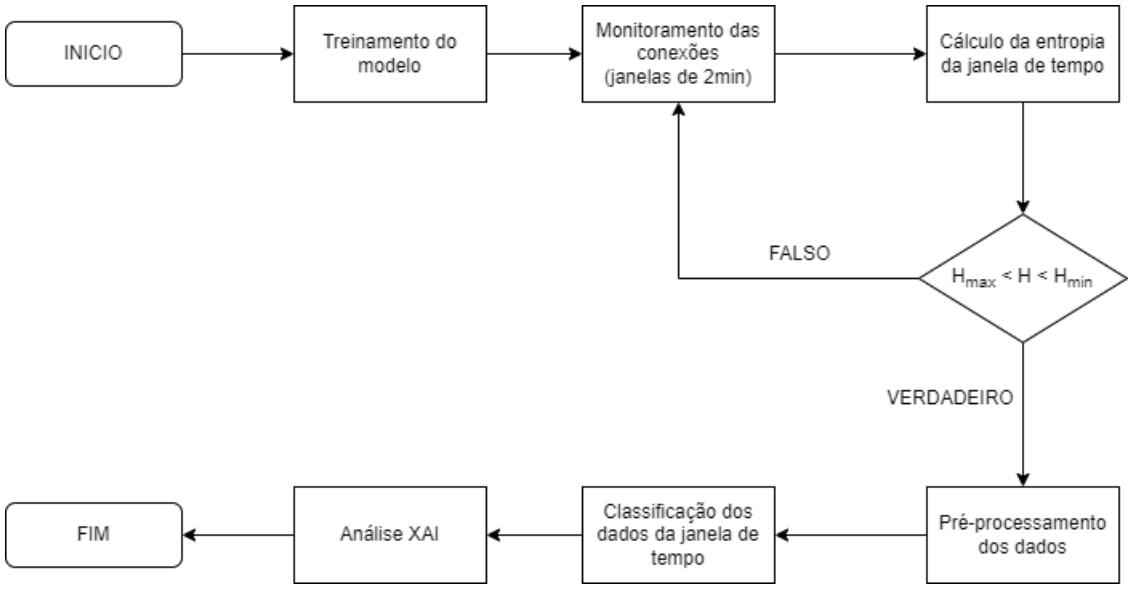
Fonte: Elaborado pelo autor

# 3 Metodologia

O trabalho segue o modelo proposto por Idhammad et al. (2018), que estabelece um sistema de detecção de anomalias por entropia da informação e classificação dos dados selecionados por um modelo de aprendizado de máquina, com o objetivo de determinar se as conexões analisadas podem indicar um ataque DDoS.

Como é possível ver na Figura 2, o sistema é dividido em várias etapas. Primeiramente, os dados da amostra são pré-processados e utilizados no treinamento do modelo de classificação. Com o modelo treinado, é iniciado o monitoramento das conexões, onde é calculada a entropia das conexões dentro de uma janela de tempo de dois minutos e seus valores são comparados a um intervalo de valores pré-estabelecidos, e, caso o limite seja excedido, acontece uma classificação desses dados através do modelo treinado na primeira etapa. Durante todas as etapas, o modelo é análisado com conceitos de *Explainable Artificial Intelligence*, e os resultados são apresentados com gráficos e informações adicionais.

Figura 2 – Diagrama do sistema de detecção proposto.




Fonte: Elaborado pelo autor

## 3.1 Ferramentas

A seguir estão listadas todas as ferramentas utilizadas no desenvolvimento do trabalho e alguns dos motivos para que fossem escolhidas.

### 3.1.1 Python

*Python* é uma linguagem de programação interpretada e orientada a objetos, que possui uma sintaxe simples e extensa seleção de bibliotecas. Dentre os muitos motivos dessa ter sido a linguagem de programação escolhida para o desenvolvimento do trabalho, um dos principais foi permitir acesso às bibliotecas *scikit-learn* e *Matplotlib*, que foram imprescindíveis durante todo o projeto.

### 3.1.2 scikit-learn

*Scikit-learn* é uma biblioteca de aprendizado de máquina para *Python*, que apresenta implementações de vários algoritmos populares, e tem o intuito de permitir que aprendizado de máquina seja acessível a todos (PEDREGOSA et al., 2011). A biblioteca também conta com o módulo *metrics* (medição, em inglês), que contém várias funções auxiliares para cálculos de erro, precisão, etc.

### 3.1.3 Matplotlib

*Matplotlib* é uma biblioteca gráfica para *Python*, com suporte a maioria dos tipos de gráficos 2D. É simples de utilizar e permite exportar gráficos em alta qualidade (HUNTER, 2007). Todas as visualizações deste projeto foram geradas com o *Matplotlib*.

### 3.1.4 SHAP

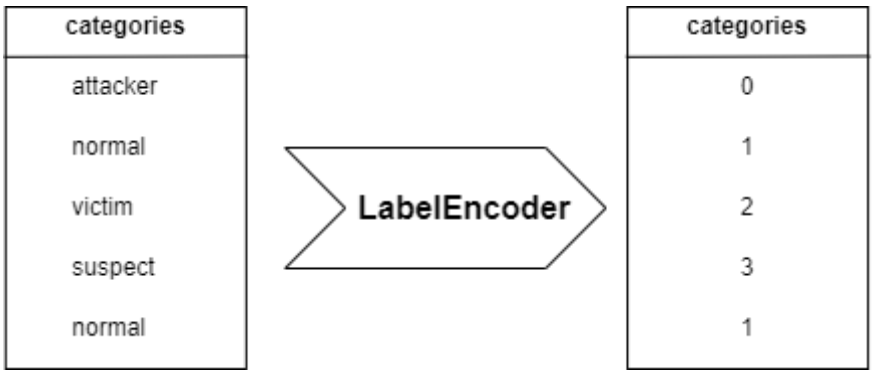
SHAP é um *framework* unificado de interpretação de modelos de *machine learning*, onde a cada atributo é dado um valor de importância que representa sua influência na classificação, aumentando a interpretabilidade do modelo (LUNDBERG; LEE, 2017).

## 3.2 Pré-processamento dos dados

Antes de utilizar o *dataset* escolhido, é preciso realizar um pré-processamento dos dados para que eles possam ser interpretados corretamente pelos modelos utilizados durante os experimentos.

O *dataset* CIDDS-001 contém uma mistura de dados numéricos e categóricos, mas são necessários apenas valores numéricos. As categorias *Proto*, *Src IP Addr*, *Src Pt*, *Dst IP Addr* e *Dst Pt* são codificadas para variáveis numéricas com a ajuda de um método da biblioteca *scikit-learn* chamado *LabelEncoder*, que mapeia toda variável categórica para um valor numérico inteiro único que a represente. A Figura 3 descreve o funcionamento desse método.

Figura 3 – Exemplo de funcionamento de um Label Encoder




Fonte: Elaborado pelo autor

As colunas *Date first seen*, *Flags*, *Tos*, *attackType*, *attackID* e *attackDescription* são descartadas nesta etapa. A coluna *class* é removida da amostra e utilizada como parâmetro *label* durante o treinamento e a classificação.

Os dados são processados uma última vez pelo método *MinMaxScaler* da biblioteca *scikit-learn*, que normaliza a amostra dentro de um intervalo de valores determinados, que nesse caso é o intervalo [0, 1]. Isso significa que o valor máximo de um atributo é 1 e o valor mínimo é 0, eliminando o problema de proporcionalidade da amostra, visto que é necessário, por exemplo, comparar números de pacotes que podem ter valor 1 ou 2, com números de bytes que podem chegar a mais de 6 dígitos.

## 3.3 Cálculo da entropia

A entropia é um valor numérico que representa a incerteza do resultado de um evento, ou no contexto deste trabalho, os atributos da conexão da qual se deseja descobrir o valor de entropia.

É possível analisar o comportamento da entropia em um conjunto de dados e estabelecer quais são os períodos de "normalidade". Isso permite definir um intervalo de entropia que representa uma conexão com características dentro das esperadas.

Seguindo o que foi proposto por Idhammad et al. (2018), as conexões foram monitoradas e separadas em janelas de tempo de dois minutos, agrupando todas que possuírem o *timestamp* que as coloquem dentro deste intervalo de tempo. Ao fim desses dois minutos, a entropia da janela de tempo é calculada utilizando a distribuição dos valores do atributo escolhido com o algoritmo da entropia de Shannon, como é possível ver no Algoritmo 1.

Se os valores excederem o intervalo pré-determinado, a janela de tempo é marcada como suspeita e seus dados são encaminhados para a etapa de classificação, caso contrário, é marcada como normal e nada precisa ser feito.

**Algoritmo 1** – *Entropia Média*

Entrada: distA = Distribuição do atributo A na janela de tempo

1. ***Para cada*** $x$ ***em*** $distA$, ***faça***
2. $\quad p(x) \leftarrow quantidade(x)/tamanho(distA)$
3. $\quad h(x) \leftarrow p(x)log(p(x))$
4. $\quad H(x) \leftarrow H(x) + h(x)$
5. $\quad$ ***Fim***
6. ***Retorne*** $-H(x)$

## 3.4 Classificação

Quando uma janela de tempo é marcada como suspeita, seus dados passam por um processo de classificação pelo modelo treinado na primeira etapa. Em termos práticos, isso quer dizer que as conexões serão classificadas com as mesmas categorias do dataset CIDDS-001: *normal*, *attacker* e *victim*.

Para os testes, foram utilizados os modelos *Random Forest*, *AdaBoost* e *Bagging*, e, para cada um deles, é comparada sua precisão média de classificação para todas as janelas de tempo em relação aos outros, de forma a medir sua performance.

## 3.5 Utilizando SHAP para análise de modelos

A biblioteca SHAP permite fazer uma análise do impacto de cada atributo no resultado final do *ensemble*, sendo possível visualizar essa informação através de gráficos gerados pela própria biblioteca. Com isso, obtêm-se maiores informações sobre o funcionamento do modelo, aumentando muito a transparência da classificação.

# 4 Resultados

Essa seção apresenta os resultados obtidos através dos experimentos realizados ao longo da criação do sistema de detecção. Os experimentos foram realizados em um sistema com processador *Intel i9 10850k* e *16GB* de memória RAM, executando em um ambiente *Windows 10*.

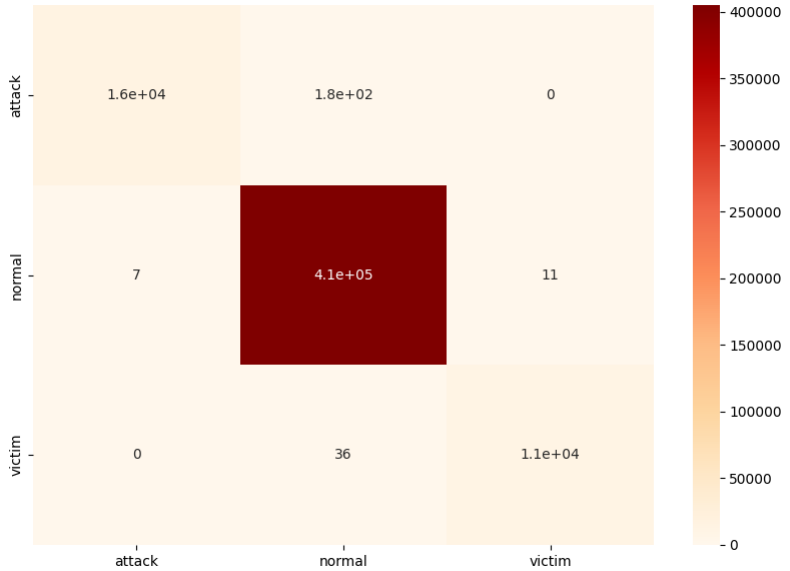
## 4.1 Treinamento

O modelo foi inicialmente treinado utilizando como amostra todos os dados referentes ao segundo dia da primeira semana do período monitorado pelo *dataset* CIDDS-001, enquanto a amostra de treino seriam os dados referentes ao primeiro dia da primeira semana.

Os dados do período analisado, em sua totalidade, são bastante desbalanceados, com 93.8% de conexões categorizadas como *normal*, 3.7% *attacker* e 2.5% *victim*. Essa desproporcionalidade pode se traduzir em um enviesamento do modelo, onde o resultado da classificação é sempre a classe mais predominante (HUANG et al., 2004).

É possível observar melhor essa discrepância através da matriz de confusão do treinamento de um modelo *Random Forest* na Figura 4, onde é claro o excesso de representação da classe *normal* se sobrepor as outras duas categorias.

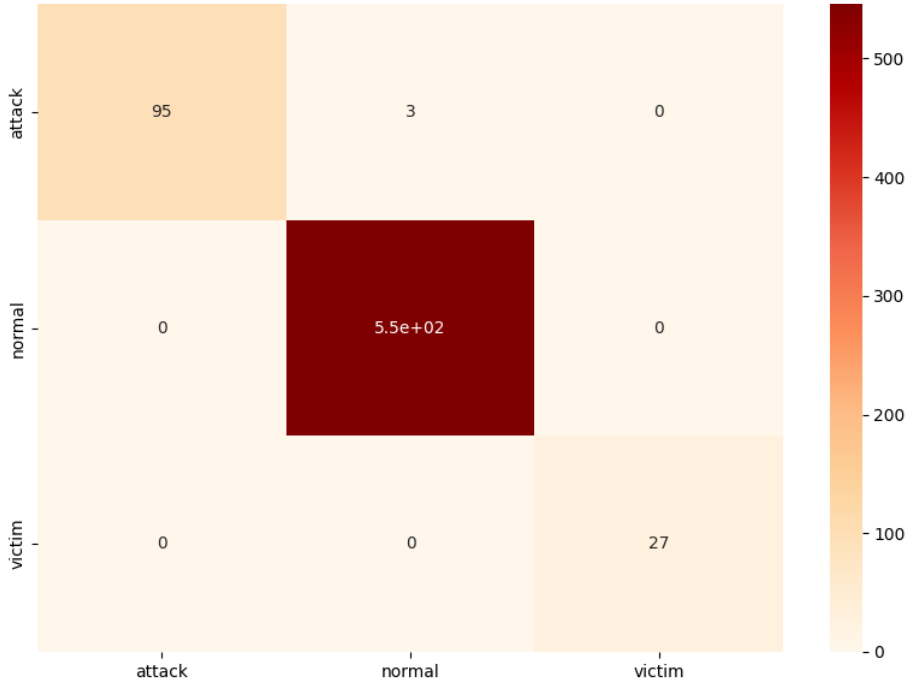
Figura 4 – Matriz de confusão do treinamento de um modelo *Random Forest* com *dataset* desbalanceado.




Fonte: Elaborado pelo autor

Portanto, foi criada uma amostra do *dataset* original com uma maior proporção das outras categorias, passando a ser 82.6% categorias *normal*, 13.8% *attacker* e 3.6% *victim*. Na matriz de confusão da Figura 5, é possível observar as outras categorias com maior destaque e também uma maior acurácia no treino, chegando a uma acurácia de 99%.

Figura 5 – Matriz de confusão do treinamento de um modelo Random Forest com dataset balanceado.

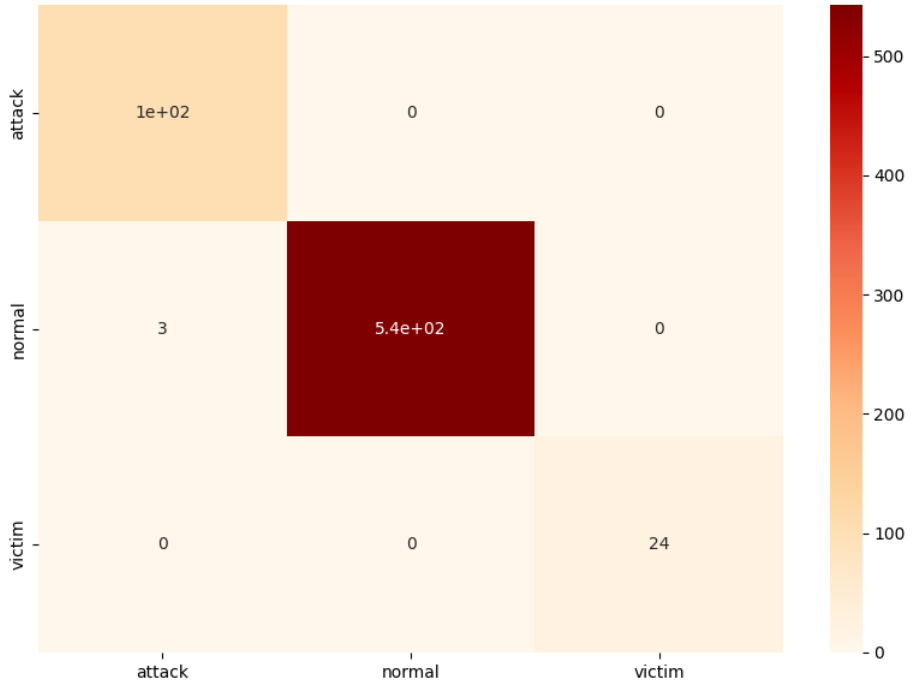



Fonte: Elaborado pelo autor

Analisando o comportamento dos outros algoritmos de classificação, algumas diferenças de comportamento durante o treinamento dos modelos foram observadas. Essas diferenças estão expostas pelas matrizes de confusão das Figuras 6 e 7. Os três algoritmos escolhidos tiveram uma boa precisão durante o treinamento, com *AdaBoost* demonstrando um número ligeiramente maior de erros de treinamento.

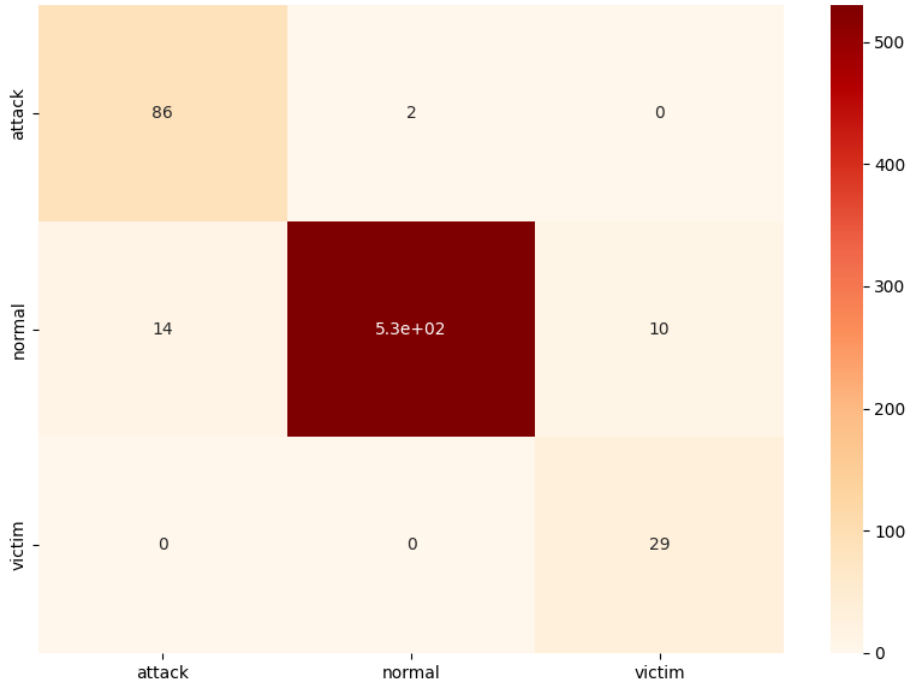
Em questão de performance, como o *dataset* balanceado é um *subset* do original e tem apenas uma fração do número de linhas, o tempo de treino de todos os modelos foi curto, geralmente levando menos de um segundo por conta do número reduzido de dados do *dataset* balanceado.

Figura 6 – Matriz de confusão do treinamento de um modelo *Bagging* com *dataset* balanceado.




Fonte: Elaborado pelo autor

Figura 7 – Matriz de confusão do treinamento de um modelo *AdaBoost* com *dataset* balanceado.

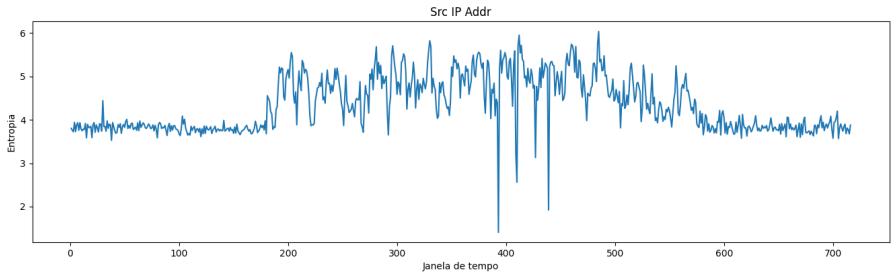



Fonte: Elaborado pelo autor

## 4.2 Entropia

A entropia foi calculada para cada janela de tempo de acordo com o algoritmo apresentado na Seção 3.3. Os resultados estão expostos na Figura 8 e foram bem consistentes com os apresentados por Idhammad et al. (2018).

Figura 8 – Gráfico da entropia pela janela de tempo para o atributo *Src IP Addr*




Fonte: Elaborado pelo autor

É possível observar que a entropia se manteve estável desde o começo até aos arredores da janela 190, e os dados confirmam que nas primeiras horas do dia não houveram muitos ataques. Portanto, toda janela de tempo que destoar muito dos valores dessas primeiras horas de monitoramento pode ser considerada suspeita. Assim, é possível estabelecer um intervalo de entropia em que as características das conexões estão dentro do esperado. Descartando os valores mais extremos do período, o intervalo aceitável da entropia é definido como sendo [3.5, 4.2].

Com o intervalo estabelecido, todas as conexões contidas no primeiro dia do *dataset* são analisadas e agrupadas em janelas de tempo de dois minutos, calculando a entropia de um atributo específico das conexões desse intervalo, como descrito na Seção 3.3.

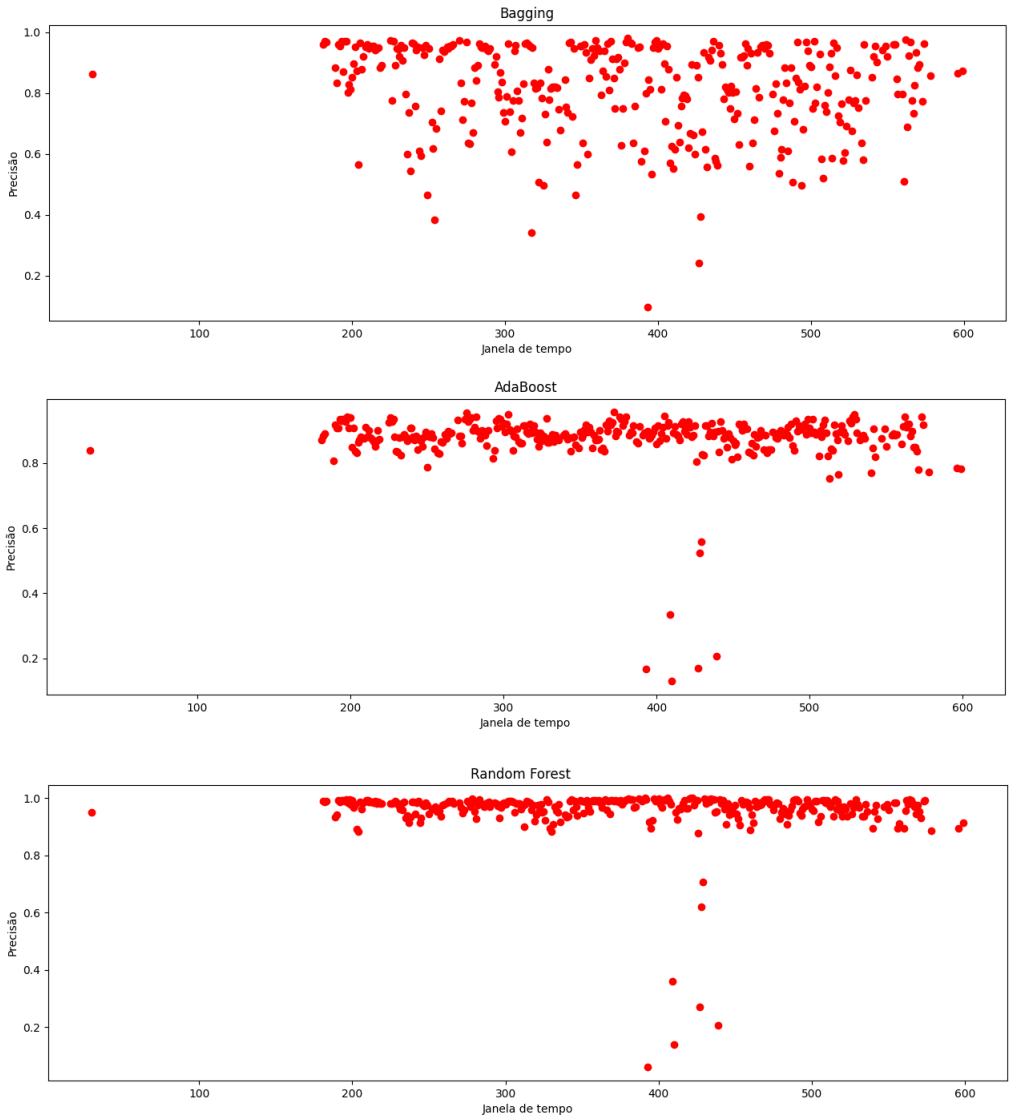
## 4.3 Classificação

Quando uma janela de tempo é marcada como suspeita, seus dados passam por um processo de classificação pelo mesmo modelo treinado na Seção 4.1. A Figura 9 mostra os gráficos da precisão da classificação pela janela de tempo para todos os modelos usados no experimento.

Nos gráficos, é possível ver a precisão de cada janela de tempo que passou pelo processo de classificação, representadas pelas bolas vermelhas. O algoritmo *Bagging*, em específico, apresenta um gráfico de precisão bem disperso quando comparado com os outros, o que resulta em uma precisão média menor que os outros que apresentam uma maior concentração de janelas entre 0.8 e 1.0, com *Random Forest* mostrando um desempenho consideravelmente melhor que *AdaBoost*, sendo 8.3% mais preciso. Os valores exatos de precisão estão expostos

na Tabela 2. Os resultados foram consistentes com os obtidos por Idhammad et al. (2018), que obtiveram uma precisão de 97% com o modelo *Random Forest*.

Figura 9 – Gráfico da precisão pela janela de tempo dos modelos.




Fonte: Elaborado pelo autor

Tabela 2 – Precisão média da classificação das janelas de tempo pelos modelos.

| **Modelo** | **Precisão média (%)** |
|---|---|
| *Bagging* | 81.9 |
| *AdaBoost* | 87.4 |
| *Random Forest* | 95.7 |

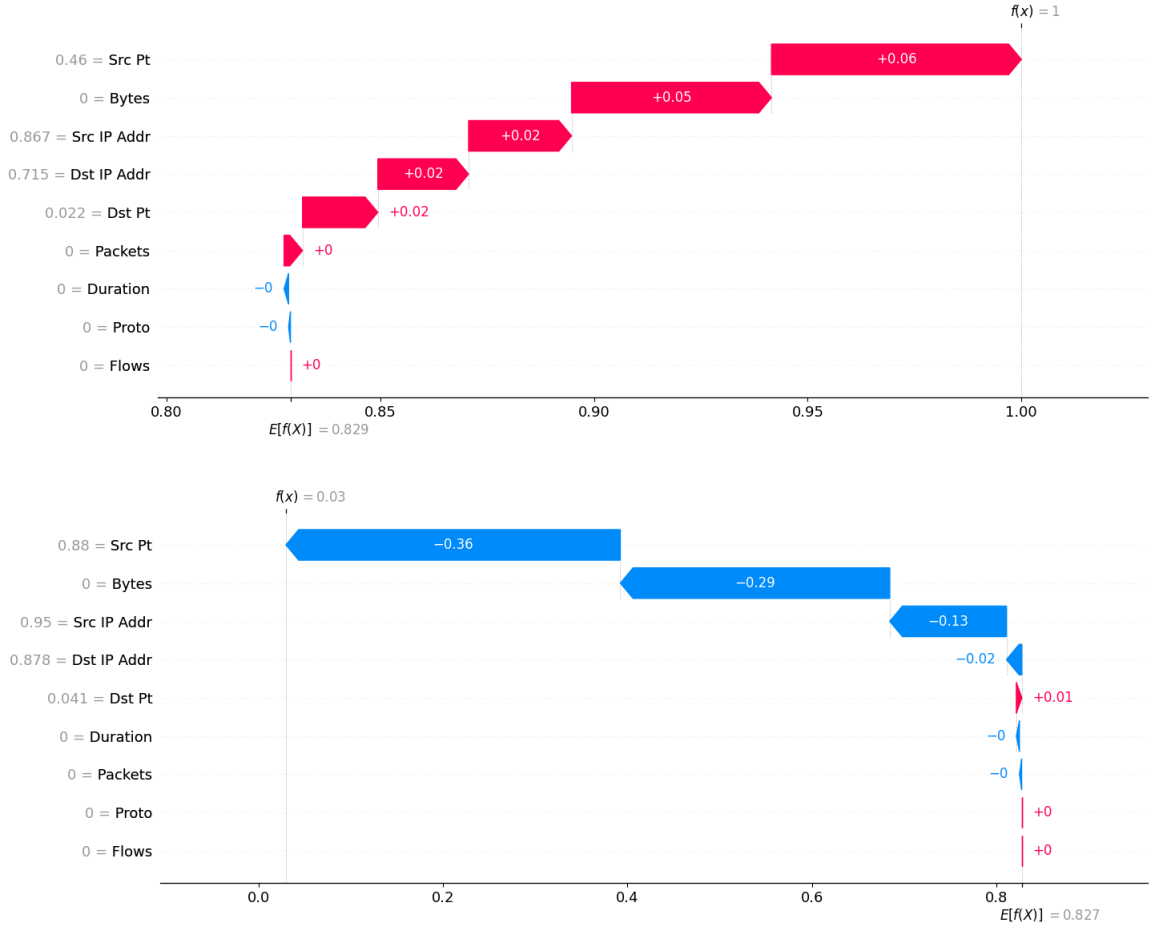
Fonte: Elaborado pelo autor

## 4.4 Análise por *Explainable Artificial Intelligence*

Com o auxílio da biblioteca SHAP, o modelo *Random Forest* foi analisado para entender quais atributos mais influenciam na hora de realizar a classificação. A Figura 10 apresenta dois gráficos em cascata da classificação de duas conexões diferentes, o gráfico na cor vermelha representa uma conexão sendo categorizada como categoria *normal*, representado pelo número 1, e a outra, em azul, é categorizada como *attacker*, representado pelo número 0.

Cada seta indica o quanto o atributo em questão influenciou o resultado final, por exemplo, é possível observar que o atributo *Src Pt* foi o que mais contribuiu para a classificação tanto para *normal* quanto para *attacker*, enquanto atributos como *Proto* e *Flows* praticamente não tiveram nenhum impacto na classificação.

Figura 10 – Gráfico da classificação de duas janelas de tempo pelo modelo *Random Forest*

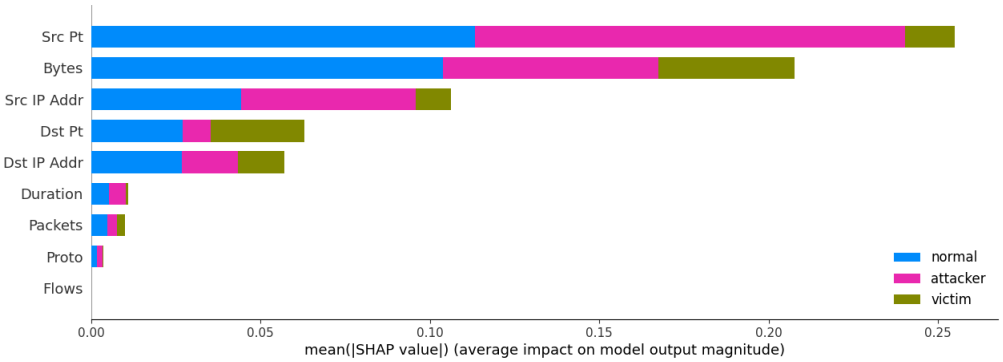



Fonte: Elaborado pelo autor

A Figura 11 apresenta uma visão geral do processo de classificação, considerando todas as conexões analisadas. O atributo *Src Pt* é o principal determinante das categorias *normal* e *attacker*, seguido de *Bytes* e *Src IP Addr*. *Bytes* e *Dst Pt* tem maior influência em determinar a categoria *victim*, enquanto *Flows* não exerce nenhuma influência na classificação, indicando

que esse atributo pode ser removido durante o pré-processamento.

Figura 11 – Gráfico do impacto médio de cada atributo na classificação do modelo *Random Forest*




Fonte: Elaborado pelo autor

# 5 Conclusão

Neste trabalho, foi desenvolvido um sistema de detecção de ataques DDoS baseado no trabalho de Idhammad et al. (2018), que fez a combinação de detecção de anomalia através do cálculo da entropia de Shannon de um atributo em um conjunto de dados, com a categorização dos dados com modelos *ensemble*.

Os resultados obtidos foram bastante promissores, com alguns modelos apresentando altas taxas de precisão, além de rápido treinamento e classificação. Também foi possível observar que o balanceamento do *dataset* foi imprescindível para alcançar um bom resultado e evitar enviesamento do modelo, o que pode significar que é possível alcançar resultados melhores com um conjunto de dados proporcional e de alta qualidade.

Como todos os cálculos de entropia e teste da amostra são feitos em pouquíssimo tempo, é possível utilizar o sistema para detectar os primeiros sinais de um ataque e estabelecer medidas de controle como *throttling* e *rate limiting*.

A análise por *Explainable Artificial Intelligence* com o auxílio da biblioteca SHAP, revelou informações interessantes a respeito da importância dos atributos do *dataset*, indicando quais são os mais influentes e os que podem ser removidos por não contribuírem com as previsões, o que pode salvar recursos de memória e acelerar o processo de treino e classificação.

Trabalhos futuros podem incluir a integração e monitoramento do sistema em um servidor real, de forma a medir sua eficiência em um ambiente de produção. A análise por XAI também pode ser extendida para outros tipos de modelos e *datasets*, podendo ser utilizada como parâmetro para otimizações da base de dados.

# Referências


AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, Elsevier, v. 60, p. 19–31, 2016.

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

DIETTERICH, T. G. Ensemble learning. In: ARBIB, M. (Ed.). *The Handbook of Brain Theory and Neural Networks*. [S.l.]: MIT Press, 2002. p. 405–408.

DOULIGERIS, C.; MITROKOTSA, A. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, v. 44, n. 5, p. 643–666, 2004. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128603004250>. Acesso em: 04 Mar. 2022.

GUNNING, D.; STEFIK, M.; CHOI, J.; MILLER, T.; STUMPF, S.; YANG, G.-Z. Xai-explainable artificial intelligence. *Science Robotics*, v. 4, n. 37, 2019. Disponível em: <https://www.science.org/doi/abs/10.1126/scirobotics.aay7120>. Acesso em: 04 Mar. 2022.

H.ABORUJILAH, A.; MUSA, S. Cloud-based ddos http attack detection using covariance matrix approach. *Journal of Computer Networks and Communications*, v. 2017, p. 1–8, 01 2017.

HOLZINGER, A.; BIEMANN, C.; PATTICHIS, C. S.; KELL, D. B. What do we need to build explainable ai systems for the medical domain? *arXiv preprint:1712.09923*, 2017.

HUANG, K.; YANG, H.; KING, I.; LYU, M. R. Learning classifiers from imbalanced data based on biased minimax probability machine. In: IEEE. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* [S.l.], 2004. v. 2, p. II–II.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science and Engg.*, IEEE Educational Activities Department, USA, v. 9, n. 3, p. 90–95, may 2007. ISSN 1521-9615. Disponível em: <https://doi.org/10.1109/MCSE.2007.55>. Acesso em: 04 Mar. 2022.

IDHAMMAD, M.; AFDEL, K.; BELOUCH, M.; LI, H. Detection system of http ddos attacks in a cloud environment based on information theoretic entropy and random forest. *Sec. and Commun. Netw.*, John Wiley and Sons, Inc., USA, v. 2018, jan 2018. ISSN 1939-0114. Disponível em: <https://doi.org/10.1155/2018/1263123>. Acesso em: 04 Mar. 2022.

JAAFAR, G. A.; ABDULLAH, S. M.; ISMAIL, S.; FERRARI, G. Review of recent detection methods for http ddos attack. *J. Comput. Netw. Commun.*, Hindawi Limited, London, GBR, v. 2019, jan 2019. ISSN 2090-7141. Disponível em: <https://doi.org/10.1155/2019/1283472>. Acesso em: 04 Mar. 2022.

LAKHINA, A.; CROVELLA, M.; DIOT, C. Mining anomalies using traffic feature distributions. In: *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, NY, USA: Association for Computing Machinery, 2005. (SIGCOMM '05), p. 217–228. ISBN 1595930094. Disponível em: <https://doi.org/10.1145/1080091.1080118>. Acesso em: 04 Mar. 2022.

LIPTON, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, Association for Computing Machinery, New York, NY, USA, v. 16, n. 3, p. 31–57, jun 2018. ISSN 1542-7730. Disponível em: <https://doi.org/10.1145/3236386.3241340>. Acesso em: 04 Mar. 2022.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 4768–4777. ISBN 9781510860964.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 34, n. 2, p. 39–53, apr 2004. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/997150.997156>. Acesso em: 04 Mar. 2022.

OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. *J. Artif. Int. Res.*, AI Access Foundation, El Segundo, CA, USA, v. 11, n. 1, p. 169–198, jul 1999. ISSN 1076-9757.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, JMLR.org, v. 12, p. 2825–2830, 2011.

RING, M.; WUNDERLICH, S.; GRUDL, D.; LANDES, D.; HOTHO, A. Flow-based benchmark data sets for intrusion detection. In: *Proceedings of the 16th European Conference on Cyber Warfare and Security. ACPI*. [S.l.: s.n.], 2017. p. 361–369.

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948.

SINGH, K.; SINGH, P.; KUMAR, K. User behavior analytics-based classification of application layer http-get flood attacks. *J. Netw. Comput. Appl.*, Academic Press Ltd., GBR, v. 112, n. C, p. 97–114, jun 2018. ISSN 1084-8045. Disponível em: <https://doi.org/10.1016/j.jnca.2018.03.030>. Acesso em: 04 Mar. 2022.