Utilização de metodologias XAI para explicar ataques a redes de computadores

Martinho Alfredo Martins Sousa
Mestrado em Segurança Informática
Faculdade de Ciências da Universidade do Porto
2024

Utilização de metodologias XAI para explicar ataques a redes de computadores

Martinho Alfredo Martins Sousa

Dissertação de Projeto realizada(o) no âmbito do Mestrado de Segurança Informática

Faculdade de Ciências da Universidade do Porto

2024

Orientador
Prof. Rita P. Ribeiro, Faculdade de Ciências da Universidade do Porto

Coorientador
Prof. João Gama, Faculdade de Economia da Universidade do Porto

Abstract

Recently, people have become more attentive to cyber security, making the topic more debated. As a result of this popularity, several studies have been carried out to find solutions for consumers. These studies tend to focus on the most diverse topics, specifically computer security, which ranges from the use of previously existing tools with some improvements to them, to the creation of new technologies that satisfy users' needs. A cybersecurity system of great importance in combating cyber attacks is *Intrusion Detection System (IDS)*. One of the reasons why these systems are used is the fact that they have the potential to stop cyber attacks. However, these systems have limitations, one of which is the fact that these methods are based on a black box, which does not allow the logic behind a result to be understood.

These systems are improved with the help of explanation methods, *Explainable Artificial Intelligence (XAI)*, which aim to explain to the user so that they understand what is behind a result.

In this work, we try to demonstrate that these technologies can go further together.

Resumo

Recentemente as pessoas estão mais atentas à sua segurança cibernética o que faz com que o tema seja mais debatido. Como resultado desta popularidade foram realizados vários estudos com o objetivo de encontrar soluções para os consumidores. Estes estudos tendem a focar-se nos mais diversos temas que constituem segurança informática, que vai desde a utilização de ferramentas previamente existentes com recurso a alguns melhoramentos das mesmas, até a criação de novas tecnologias que satisfaçam as necessidades dos utilizadores. Um sistema de cibersegurança que tem uma grande importância no combate a ataques cibernéticos é *Intrusion Detection System (IDS)*. Um dos motivos que leva a utilização destes sistemas é o facto de estes terem o grande potencial para travar ataques cibernéticos. Porém estes sistemas têm desvantagens, em que uma delas é o facto de estes métodos serem baseados em caixa negra, o que não permite que seja percebido a lógica por de trás de um resultado.

Estes sistemas podem ser melhorados com a ajuda de métodos de explicação, *Explainable Artificial Intelligence (XAI)*, que têm como objetivo fornecer uma explicação ao utilizador para que este perceba o que está por de trás de um resultado.

Neste trabalho tenta-se demonstrar que estas tecnologias em conjunto conseguem chegar mais longe.

Agradecimentos

Gostaria de começar por agradecer aos meus orientadores, Prof. Rita P. Ribeiro e Prof. João Gama, por me darem a oportunidade de trabalhar neste tema.

Gostaria também de agradecer à minha família por estar sempre presente quando precisei.

Por fim, gostaria de estender os meus agradecimentos a todos os outros não mencionados que, de uma forma ou de outra, contribuíram para esta incrível jornada.

Conteúdo

Lista de Tabelas

Lista de Figuras

Acrónimos

AI Artificial intelligence

AUC Area Under the ROC Curve

DARPA Defense Advanced Research Projects Agency

DCC Departamento de Ciência de Computadores

DOS Denial of Service

FCUP Faculdade de Ciências da Universidade do Porto

HSTrees Half Space Trees

IDS Intrusion Detection System

IForestASD Isolation Forest Algorithm for Stream Data

IForest Isolation Forest

LIME Local Interpretable Model-Agnostic Explanations

MIT Massachusetts Institute of Technology

MITM Man in the middle attack

ML Machine learning

PySAD Python Streaming Anomaly Detection

RGPD Regulamento Geral sobre a Proteção de Dados

ROC Receiver Operating Characteristic

R2L Root to Local

SHAP SHapley Additive exPlanations

U2R User to Root

XAI Explainable Artificial Intelligence

Capítulo 1

Introdução

1.1 Motivação

Atualmente as pessoas estão mais consciencializadas para os perigos existentes a nível informático, o que faz com estas estejam mais atentas a possíveis sinais de ataque. Porém, esta atualização não fica só pelos utilizadores mas também atinge os sistemas, pois, estes tentam adaptar-se ao desenvolvimento de novas tecnologias, como é o caso dos métodos de deteção de intrusões.

Como resultado da adaptação dos vários sistemas, das novas tecnologias resultam problemas que, numa perícia inicial, não aparentam ser graves, mas que numa segunda perícia conclui-se que é um problema sério. Um problema que advém desta progressão da tecnologia é o uso de métodos de deteção de anomalias que implementam métodos baseados em caixa negra.

Porém, recentemente foram desenvolvidas várias tecnologias que corrigem os vários problemas que são encontrados nos sistemas que se adaptam aos novos tempos, como é o caso dos métodos *XAI*.

1.2 Descrição do problema

A criação de modelos que utilizam metodologias baseadas em caixa negra não são facilmente interpretáveis, mesmo para utilizadores especializados e com conhecimento prévio. Uma questão que pode ser colocada e que tem uma resposta simples é a seguinte: Porque é que é importante haver confiança entre um método de deteção de anomalia e utilizador? A resposta à pergunta anteriormente feita, é facilmente compreendida. É importante haver confiança entre utilizador e método, porque caso não haja segurança o utilizador não irá utilizá-lo, mesmo que este lhe dê grande parte das vezes resultados corretos. Isto acontece porque o utilizador pensa que foi apenas sorte o resultado estar correto.

Recentemente foram desenvolvidas tecnologias com o objetivo de fornecer um nível de interpretabilidade maior, e essas tecnologias designam-se por *Explainable Artificial Intelligence*

(XAI).

1.3 Objetivos

Este estudo tem como objetivo aplicar diferentes métodos de deteção para analisar atividades de rede em fluxos de dados, no âmbito de obter explicações facilmente interpretáveis por humanos com recurso à utilização de métodos *XAI*. Um dos objetivos é saber como é que as explicações são justificadas e qual é a relação entre as justificações das explicações.

Algumas das questões que são auto-impostas são as seguintes: As explicações para o mesmo ataque são iguais independentemente do método utilizado? As explicações são diferentes para ataques diferentes?

1.4 Estrutura do documento

Resumidamente a estrutura da dissertação é a seguinte: no capitulo 2 é realizado um contexto de onde o trabalho se insere, no capitulo 3 é descrita a metodologia utilizada, no capitulo 4 é onde são descritos os testes e resultados obtidos e, por fim, no capitulo 5 em que é feita a conclusão relativa ao trabalho.

Capítulo 2

Trabalho Relacionado

Com o passar do tempo, as pessoas ficam cada vez mais atentas a ataques cibernéticos, o que faz com que o tema seja mais debatido, resultando na realização de vários estudos, aumentando assim a consciencialização e o conhecimento acerca do tema em si. Os temas dos estudos realizados acerca do tema de cibersegurança focam-se nos mais diversos temas que constituem segurança informática, tendo desde a utilização de ferramentas previamente existentes com recurso a alguns melhoramentos das mesmas, até à criação de novas tecnologias que satisfaçam as necessidades dos utilizadores. Algumas destas tecnologias tendem a focar-se em aspetos-chave como o processamento ou armazenamento. Porém, o primeiro aspeto que os académicos e investigadores devem-se focar no tema de cibersegurança é com a construção ou melhoria de sistemas de deteção de intrusão, *IDS*, pois, estes sistemas são e continuarão a ser a linha da frente no combate a ameaças cibernéticas. Um dos motivos que leva à utilização destes sistemas é o facto de estes terem o potencial para travar ataques cibernéticos.

Segundo Mitchell R et al [35], é fundamental distinguir comportamentos de ataque para realizar uma avaliação correta do plano de defesa. Existem dois comportamentos. O comportamento malicioso e o comportamento egoísta. O primeiro comportamento tem como base a desobediência das várias regras, de confidencialidade, integridade, disponibilidade, autenticidade, não repudio e privacidade. O segundo comportamento resume-se numa ideologia de lobo solitário, (não tem um raciocínio em mente de comunidade). Para além disto, o espetro dos sistemas de deteção de intrusão, pode e deve ser dividido em várias categorias, como se pode ver pela Figura 2.1, sendo que estas podem ser descritas da seguinte forma:

- Prevenção de intrusão - Consiste em travar qualquer tentativa de ataque antes de este conseguir entrar no sistema.
- Deteção de intrusão - Consiste em detetar ataques ao qual o sistema está a ser alvo.
- Tolerância de intrusão - Consiste no sistema sobreviver e operar na presença de ataques bem sucedidos ao sistema.

Dito tudo isto, pode-se dizer que o tema deste trabalho encaixa-se nos sistemas de deteção de

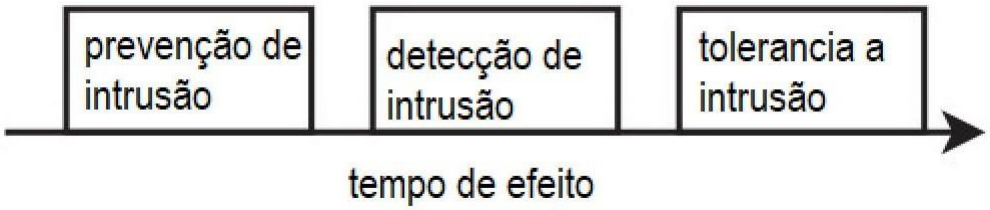


Figura 2.1: Espetro dos sistemas de deteção de intrusão, retirado e adaptado de [35].

intrusões. Um dos aspetos que torna estes sistemas a linha da frente é o facto de estes estarem em constante atualização relativamente ao mundo que nos rodeia. Uma dessas melhorias é o facto de estes sistemas incluírem e utilizarem cada vez mais inteligência artificial, *Artificial intelligence (AI)*.

Como se pode verificar, de acordo com García-Teodoro et al [18], existem várias técnicas de deteção de intrusões, nos modelos de deteção de anomalias, que podem ser separadas em três classes principais que são as seguintes:

- Baseado em estatística - Consiste na criação de perfis da atividade do tráfego / fluxo de dados.
- Baseado em conhecimento - Consiste num sistema que classifica os dados de acordo com um conjunto de regras definidas por um especialista.
- Baseado em *machine learning* - Consiste na utilização de modelos que permite a análise e categorização dos dados.

Associadas a estas técnicas, estão as respetivas vantagens e desvantagens que podem ser vistas nas respetivas Tabelas 2.1 e 2.2.

Técnica	Vantagens
Baseado em estatística	Grande precisão de notificação de ataques.
Baseado em conhecimento	Robusto, flexível e um nível elevado de escalabilidade.
Baseado em *machine learning*	Flexível e adaptável.

Tabela 2.1: Vantagens das técnicas de deteção de intrusão.

Das técnicas acima mencionadas para o tema deste trabalho só tem relevância a última técnica mencionada, sendo que só essa irá ser mencionada mais à frente neste trabalho.

A aplicabilidade destas tecnologias tem vindo a ser exorbitantemente fantasiada no mundo da tecnologia, e com isso tem vindo a ser integrada com a utilização de algoritmos de deteção de anomalias, nas categorias previamente descritas. Mvula et al [38], mostra que existe um interesse no que pode ser alcançado com a conjugação destas tecnologias na area de cibersegurança, em

Técnica	Desvantagens
Baseado em estatística	Suscetível de ser treinado por atacantes. Configuração difícil para parâmetros e métricas.
Baseado em conhecimento	Disponibilidade fraca.
Baseado em *machine learning*	Alto nível de consumo de recursos.

Tabela 2.2: Desvantagens das técnicas de deteção de intrusão.

termos de desempenho das mesmas. Uma das principais preocupações discutidas em [38], é o facto de o conjunto de dados escolhido para treinar o modelo, é considerado como uma parte extraordinariamente crítica no que toca à construção do modelo, pois este vai influenciar a mitigação de ameaças cibernéticas e a segurança do sistema através de vários métodos como reconhecimento de padrões e deteção de ataques em tempo real. Contudo, com a introdução de *machine learning*, *ML*, na área de cibersegurança, sendo esta uma das áreas mais específica da inteligência artificial, é vista como uma alternativa para a realização de deteção de intrusões, ao qual não tem as desvantagens dos métodos convencionais como a perda de precisão ou desempenho. Na maioria dos estudos realizados não é explicado o porquê de um método ter sido selecionado em relação a outro, ou o porquê de terem selecionado um método não supervisionado em relação a um supervisionado.

Contudo, o universo da cibersegurança é um espaço dinâmico, que se encontra em constante alteração, pois para cada defesa que é desenvolvida, os atacantes desenvolvem novas formas de ultrapassar as mesmas, [14]. Tornando desta forma ainda mais importante a escolha do conjunto de dados, apesar das dificuldades que são enfrentadas na seleção do conjunto de dados como se pode ver nos seguintes artigos, [24, 31, 38].

2.1 Anomalias

2.1.1 Definição de anomalia

Antes de se falar sobre deteção de anomalias, é necessário saber o que é uma anomalia, para ser possível compreender o que diferencia entre anomalias e dados normais. O conceito de anomalia pode ser descrito como um ponto que não se enquadra no perfil em relação aos outros pontos, [1, 10].

O processo de deteção de anomalias é muito importante devido a ser possível retirar muita informação das anomalias identificadas. Um exemplo que descreve a importância das anomalias é no caso da área da saúde, por exemplo, um sistema que detete anomalias dos sintomas ou do estado atual de um utente em observação pode indicar se esse mesmo paciente está a responder negativamente ao tratamento ou se houve erro médico [54]. As anomalias também são muito importantes em outras áreas como deteção de fraude, deteção de intrusões, entre outros.

Porém, para as aplicações que utilizam mecanismos de deteção de anomalias, para obterem um melhor resultado relativamente ao processo de deteção de anomalias devem dividir esse processo em deteção de anomalias e explicação de anomalias [43]. Quando é referido que um dos processos deve ser explicação de anomalias, está-se a referir à criação de explicações que forneçam a explicação, por detrás do motivo, que levou àquela anomalia ser classificada como uma anomalia, de forma, que os utilizadores percebam o que são anomalias naquele contexto específico para que seja possível melhorar os processos de deteção. Relativamente ao processo de deteção de anomalia, este vai capturar diferentes tipos de anomalias [36], sendo essas as seguintes:

- Anomalia pontual - Consiste em pontos individuais dos dados que se distinguem relativa- mente ao resto dos pontos.
- Anomalia contextual - Consiste em pontos que são dados como anomalias num contexto específico. Isto pode ser verificado no seguinte exemplo: na realização de transações avultadas num período de tempo curto e numa altura do dia em que é raro haver qualquer transação.
- Anomalia coletiva - Consiste num agrupamento de vários pontos do conjunto de dados serem anómalos aos restantes pontos do conjunto de dados.

2.1.2 Tipos de explicação de anomalia

Segundo a literatura relativa à área das anomalias, existem dois tipos de explicação de anomalias [43], sendo essas: avaliativa e não avaliativa . A explicação avaliativa diz respeito aos critérios que foram utilizados para avaliar as explicações criadas, enquanto a explicação não avaliativa refere-se aos aspetos associados a criação da explicação independentemente de como esta é avaliada.

As explicações não avaliativas têm três tipos: nível de importância da anomalia, interações entre anomalias e os atributos das anomalias. O primeiro tipo de explicação não avaliativa tem como objetivo ordenar por nível as anomalias. Este divide-se em dois tipos de classificação como se pode verificar na Tabela 2.3. Relativamente ao segundo tipo como o nome indica, este tem como objetivo dar a entender como é que as anomalias interagem entre si, um exemplo que pode ajudar a perceber pode ser o seguinte: numa cidade o trânsito no local B pode ter como causa a congestão do tráfego no local A da cidade. O respetivo terceiro tipo tem como objetivo descobrir quais são os atributos que contribuem para a anormalidade dos dados.

Classificação numérica	1º, 2º, 3ª, ...
Classificação categórica	forte, fraco, trivial

Tabela 2.3: Tipos de classificação.

No que concerne ao terceiro tipo descrito acima, este pode ser dividido em duas categorias, sendo essas anomalias individuais e anomalias coletivas. Nas anomalias individuais, a pontuação de cada atributo correspondente aquela anomalia vai ser comparada com um *threshold* em que

o grupo de atributos que for maior que o *threshold* é associado a anormalidade da anomalia. Relativamente às anomalias coletivas, estas funcionam da mesma forma que as anomalias individuais, mas agora num contexto de grupo.

Perante tudo o que foi dito anteriormente, estes métodos de explicação também apresentam algumas desvantagens / desafios na sua explicação, como se pode ver na seguinte Tabela 2.4.

Técnica	Desvantagem
Classificação numérica	Os algoritmos de deteção de anomalias atribuem valores que variam em escala de algoritmo para algoritmo. Incorporação da opinião / experiência do utilizador.
Classificação categórica	Definição dos níveis de importância. Incorporação da opinião / experiência do utilizador. Selecionar métodos eficientes para atribuição da respetiva categoria.
Interações entre anomalias	Existência de disponibilidade temporal para haver uma correlação entre causa - efeito. Existência de uma estrutura de processamento eficiente.
Anomalias individuais	Limitação da procura no subespaço. Criação de descrições facilmente interpretáveis por humanos. Incorporação da opinião / experiência do utilizador.
Anomalias coletivas	Limitação da procura no subespaço. Criação de descrições facilmente interpretáveis por humanos. Incorporação da opinião / experiência do utilizador. Descobrir semelhanças entre as anomalias.

Tabela 2.4: Desafios das explicações de anomalias não avaliativas.

2.1.3 Contextos de implementação de métodos de explicação de anomalias

Como foi descrito na secção 2.1.2 anterior, existem vários tipos de explicação de anomalias que podem ser úteis nos mais diferentes ambientes, pois a explicação de uma anomalia implica a análise das anomalias. Tornando-se desta forma possível aplicar a problemas reais, métodos de deteção de anomalias. Existem vários artigos que descrevem técnicas de explicação genéricas que podem ser implementadas no mundo real [3, 11, 13, 23, 25, 28, 30, 33, 47, 49].

Uma das implementações dos métodos de explicação que pode ser implementada no mundo é no caso de sistemas de deteção de intrusões, *Intrusion Detection System (IDS)*. Segundo Milenkoski et al [34], pode se deferir um sistema de deteção de intrusões como uma ferramenta de segurança que tem como objetivo detetar possíveis ataques ao nível da rede. Quando estamos perante um ambiente na qual é essencial haver segurança têm que ser consideradas certas propriedades como confidencialidade, integridade e disponibilidade, e quando uma destas propriedades é posta em causa implica necessariamente que o sistema está perante um ataque ou uma intrusão. Como

este tipo de comportamento é diferente do que é esperado, pode-se dizer que estamos perante uma anomalia, implicando dessa forma a possibilidade da integração de métodos de explicação de anomalias [10].

Habitualmente o sistema de deteção de intrusão, *IDS*, é definido como um sistema que lida regularmente com grandes quantidades de dados, e que têm como objetivo detetar ataques e responder com uma resposta apropriada e num espaço de tempo propício a ser bem sucedido. Porém, devido ao número possível de ataques que o sistema poderá enfrentar num pequeno espaço de tempo, torna-se aflitivo para os utilizadores no que toca a fornecer respostas convenientes aos problemas. Sendo que desta forma os métodos de explicação encaixam-se perfeitamente neste ambiente, pois, estes conseguem fornecer uma lista dos ataques prioritários com base nas explicações geradas pelo método e assim remover algum do fardo dos utilizadores nestes sistemas.

2.2 Deteção de anomalias

Para além da seleção do conjunto de dados ser muito difícil, o mesmo também pode ser dito da seleção do método de deteção. Um caso que mostra isso mesmo é Garcia et al [17], na qual dá a perceber as dificuldades da seleção do método, em que esta mesma seleção é controlada e dependente do âmbito e características a obter num ambiente com um objetivo pretendido. Por outro lado, uma das limitações associadas à integração e utilização de métodos que envolvam *machine learning* é o volume de dados que estes métodos selecionados necessitam para serem treinados de forma a que consigam ser úteis num dado ambiente. Isto levou com que vários investigadores se focassem em tópicos como *big data* e *cloud computing*, pois estes abriam novas possibilidades na área de cibersegurança, como se pode ver nas seguintes referências [21, 44]. Porém, apesar da capacidade de algumas soluções conseguirem processar grandes volumes de dados, a maioria dos modelos não são capazes de detetar anomalias em contexto de fluxo contínuo de dados. Existem alguns modelos que fornecem opções de processamento de fluxos contínuos de dados, que podem ser vistos nas seguintes referências [19, 40, 52].

Atualmente os trabalhos realizados por académicos e investigadores, tendem a focar-se mais na necessidade de criar soluções sólidas, com o objetivo de complementar a comunidade de cibersegurança no que toca a aplicações de tempo real.

2.2.1 Definição de deteção de anomalias

Na literatura, a definição de deteção de anomalias é uniformemente descrita como a identificação de padrões nos dados que se destingem dos restantes dados [10]. A deteção destes dados é de grande importância porque estes dados podem representar informação crítica e significativa do sistema em questão.

Existem muitas áreas que podem usufruir desta tecnologia para além da área mencionada na secção 2.1.3, e essas áreas vão desde a área da saúde até a área da cibersegurança.

Para haver uma solução do agrado dos utilizadores é necessário haver um cuidado mais profundo, na seleção do método de deteção de anomalia, pois diferentes métodos têm diferentes vantagens e desvantagens dependendo do meio em que vão ser utilizados.

2.2.1.1 Complexidades e desafios de deteção de anomalias

Apesar da definição de deteção de anomalias descrita na secção anterior 2.2.1, ser de fácil compreensão, o mesmo não pode ser dito da implementação dessas técnicas [10]. As dificuldades dessas técnicas são as seguintes:

- Definição de normalidade - Definição de um domínio que contenha todos os pontos normais dentro dos seus limites é extremamente difícil, pois, a distância entre pontos por vezes é muito reduzida fazendo com que certos pontos sejam considerados anomalias ou normais, sem que a sua verdadeira classificação seja essa.
- Dados vulneráveis a fraude - Os modelos de deteção de anomalias podem ser sujeitos a ataques, em que um ator com más intenções treina esse método com o objetivo de os ataques serem classificados como atividade normal. Outro caso que pode ser realizado por um ator maligno é a camuflagem dos ataques para que estes não sejam detetados.
- Implementação de métodos de deteção de anomalias em diferentes ambientes - Dependendo do ambiente e objetivo, o método a selecionar deve ter em consideração vários aspetos. Um caso que pode demonstrar isso é, por exemplo, no caso de um sistema que detete pequenas alterações como anomalias no caso de um medidor de temperatura, e no caso de um sistema que detete pequenas alterações como normais como é o caso do setor financeiro, o mesmo sistema torna-se incompatível.
- Disponibilidade dos dados - A variedade existente de conjuntos de dados para treinar modelos é limitada e alguns destes conjuntos de dados apresentam ruído, o que torna difícil a distinção entre anomalias e pontos normais.

2.2.2 Técnicas de deteção de anomalias

Deteção de anomalias foi um tópico alvo de vários artigos e estudos, nos mais diferentes tipos de técnicas de deteção de anomalias [10, 38, 52], e esta secção tem como objetivo mencionar algumas dessas técnicas, mais especificamente técnicas estatísticas, *nearest neighbor*, *cluster* e isolamento.

2.2.2.1 Baseado em *cluster*

Modelos baseados em *cluster* s são baseados em aproximações entre pontos do conjunto de dados. Os métodos baseados em *cluster* s funcionam através da divisão do conjunto de dados, em conjuntos com diferentes dimensões, em que os conjuntos com menor densidade ou com uma

maior distância em relação a um conjunto de pontos de grandes dimensões são considerados anomalias.

2.2.2.2 Baseado em *nearest neighbor*

Modelos baseados em *nearest neighbor* são baseados em aproximações entre pontos do conjunto de dados, na qual existem duas categorias baseado em distância e baseado em densidade. No caso método baseado em distância, este funciona através do cálculo da distância entre os pontos do conjunto de dados, em que aqueles que obtiverem um valor de distância superior são considerados anomalias. Porém, no caso do método baseado em densidade este funciona através de um raio com um valor previamente definido, em que dependendo do número de pontos normais e anormais presentes dentro desse raio é classificado como anomalia ou normal.

2.2.2.3 Baseado em estatística

Modelos baseados em estatísticas na maior parte dos casos estipulam um modelo ao qual caracterizam os pontos normais do conjunto de dados. Quando são comparados dados com o modelo previamente criado, é atribuído um valor a esses dados, em que um valor probabilístico baixo indica que um dado ponto é classificado como uma anomalia. Estes métodos podem ser divididos em duas subcategorias, sendo essas as seguintes: paramétricos e não paramétricos. Em ambos os casos os métodos aprendem com dados que lhes são fornecidos, a única diferença é que no caso do não paramétrico este aprende com o conjunto de dados atual enquanto o paramétrico tem conhecimento prévio [10].

2.2.2.4 Baseado em isolamento

Modelos baseados em isolamento são baseados em isolamento de pontos anormais do conjunto de dados. O isolamento é possível, pois os pontos anómalos são diferentes dos pontos normais e estão presentes em menor número no conjunto de dados. Alguns dos modelos baseados em isolamento estão presentes nas secções 3.1.1 e 3.1.2.

Dito tudo isto acerca das diferentes técnicas de deteção de anomalias, estas trazem associadas consigo algumas vantagens e desvantagens como podemos ver na Tabela 2.5.

2.3 Tipos de aprendizagem dos métodos de deteção de anomalias

A aprendizagem dos métodos de deteção de anomalias pode ser dividida em duas categorias. A primeira categoria é a metodologia da aprendizagem do algoritmo e a segunda categoria tem a haver com a relação entre a aprendizagem e o conjunto de dados. A primeira categoria pode ser definida em aprendizagem estática ou aprendizagem incremental. Relativamente a

Técnica	Desvantagens	Vantagens
Baseado em estatística	Métodos paramétricos são difíceis de utilizar em contexto de dados contínuos. Métodos não paramétricos só podem ser utilizados em conjuntos de dados com baixo nível de dimensão em ambientes de fluxo de dados contínuos.	Estão adaptados para o contexto de fluxos de dados contínuos.
Baseado em *nearest neighbor*	Custo computacional elevado e baixo nível de desempenho para conjuntos de dados com grandes dimensões em contexto de fluxo de dados.	Métodos baseados em distância estão moldados para anomalias globais, enquanto métodos baseados em densidade estão adaptados para anomalias locais.
Baseado em *cluster*	Não apresenta otimizações para identificação individual de anomalias.	Método moldado para identificação de aglomerados de pontos.
Baseado em isolamento	Desempenho do modelo depende do tamanho da janela escolhida. Difícil adaptação para dados categóricos.	Custo computacional e consumo de memória baixo. Eficiente na deteção de anomalias.

Tabela 2.5: Comparação entre ás varias técnicas de deteção de anomalias.

segunda categoria, esta divide-se em aprendizagem supervisionada, semissupervisionada ou não supervisionada.

2.3.1 Aprendizagem relativa ao conjunto de dados

2.3.1.1 Aprendizagem supervisionada

Os algoritmos que têm uma aprendizagem supervisionada necessitam que no conjunto de dados estejam devidamente identificados os pontos normais e os pontos anormais. Como o processo é feito por um humano, o processo é demoroso e suscetível a erros na realização da identificação dos pontos do conjunto de dados [7, 38, 39]. Este tipo de aprendizagem vai detetar anomalias baseadas nos exemplos que lhe são fornecidos.

2.3.1.2 Aprendizagem semissupervisionada

Os algoritmos que têm uma aprendizagem semissupervisionada treinam num conjunto de dados em que todos esses pontos são normais. Desta forma, qualquer ponto que não seja possível classificar como normal é classificado como anomalia. Este método de aprendizagem é mais comum do que o método supervisionado [7, 38, 39].

2.3.1.3 Aprendizagem não supervisionada

Os algoritmos que têm uma aprendizagem não supervisionada não necessitam que os pontos do conjunto de dados estejam identificados. Devido a este facto estes métodos são extremamente populares na deteção de anomalias [7, 38, 39].

2.3.2 Aprendizagem estática versus aprendizagem incremental

Existem dois tipos de aprendizagem, sendo estes através de métodos estáticos ou métodos incrementais [6, 27, 41]. Cada um destes métodos de aprendizagem apresenta as suas vantagens e desvantagens que são descritas mais abaixo nas respetivas secções. Quando se fala sobre aprendizagem, está-se a referir à utilização de uma tecnologia chamada *machine learning ML*, que segundo Koza et al [26], pode ser definida como uma área de uma tecnologia mais abrangente que neste caso é a inteligência artificial, na qual algoritmos aprendem a partir de dados que lhes são fornecidos. A partir desta aprendizagem, estes algoritmos, são capazes de realizar instruções sem que lhes sejam fornecidos passos específicos a cumprir.

Atualmente já existem tecnologias mais avançadas que conseguem realizar com mais rapidez, eficiência e facilidade do que as tecnologias mais antigas [58].

2.3.2.1 Aprendizagem estática

O método de aprendizagem estático, como o nome indica, baseiam-se na aprendizagem através da utilização de um conjunto de dados estático previamente preparado. A partir deste conjunto de dados um modelo baseado em *machine learning* vai aprender com estes dados para futuramente realizar ações.

Contudo, esta metodologia apresenta vantagens como desvantagens. A vantagem relacionada com os métodos estáticos tem haver com o não exigir capacidades computacionais muito grandes, uma vez que não tem que processar dados em tempo real. Porém, as desvantagens relacionadas com o modelo estático estão relacionadas com a limitação do armazenamento e a flexibilidade de adaptação.

2.3.2.2 Aprendizagem incremental

O método de aprendizagem incremental, ao contrário do método estático, baseia-se na aprendiza- gem em tempo real. Isto quer dizer que a aprendizagem do método ocorre sempre que um novo conjunto de dados é fornecido.

Contudo, este método de aprendizagem apresenta vantagens e desvantagens. As vantagens relacionadas com este método estão relacionadas com a flexibilidade e utilização de memória do método de aprendizagem, pois uma vez que este aprende com a chegada de um novo conjunto de dados não existe a necessidade de haver armazenamento. Porém, a desvantagem relacionada com este método de aprendizagem está relacionada com o custo computacional exigido.

Como se pode ver, em ambos os métodos de aprendizagem existem vantagens e desvantagens, mas, tudo depende do meio em que estes métodos vão ser utilizados. Dito tudo isto, para o objetivo deste trabalho foi escolhido o método de aprendizagem incremental, na qual foram escolhidos dois métodos de deteção sendo esses: Half Space Trees (HSTrees) e Isolation Forest Algorithm for Stream Data (IForestASD) que são respetivamente descritos mais à frente na secção 3.1.

2.4 Interpretabilidade dos modelos

Nos últimos anos tem havido a necessidade de processar uma quantidade muito grande de dados, para o bom funcionamento de serviços e aplicações. Com o objetivo de combater este problema foram desenvolvidos vários sistemas de ajuda à decisão, construídos segundo uma metodologia de caixa negra, baseados em *machine learning*, que dificulta ou impede intencionalmente os utilizadores de perceberem a lógica que este sistema usa para obter os resultados. Esta falta de transparência contribui para a criação de problemas práticos e éticos [20]. Isto cria a seguinte questão: “O que é a interpretabilidade?”. Esta questão pode ser respondida da seguinte forma: interpretabilidade é a facilidade com que uma pessoa consegue perceber uma decisão ou raciocínio. A explicação pode ter um nível de facilidade de interpretação alto ou baixo, e, com isso podemos comparar os vários métodos como bons ou maus, relativamente ao tipo de explicações que estes fornecem [37].

Desta forma, pode-se afirmar que, a importância do nível de interpretação de uma explicação é muito importante, pois este decide se o utilizador pode confiar no modelo ou não. A confiança que o utilizar deposita num modelo não é uma confiança cega, mas sim, justificada porque o modelo tem que justificar o porquê do resultado obtido e como é que este chegou a esta conclusão. Só depois, de estes dois aspetos terem sido respondidos é que o utilizador decide se confia ou não no resultado.

Mais recentemente, com o avanço destas tecnologias, criou-se um problema grave de falta de confiança destes mesmos métodos, pois, apesar destes serem rápidos e precisos, como não existia confiança entre os métodos e utilizadores, estes acabavam por não serem utilizados. Apesar

dos métodos em alguns casos preverem com exatidão o resultado, como não era dada nenhuma explicação da lógica que estava por detrás dessa previsão, os utilizadores podem afirmar que foi um caso de "sorte". Para combater estes defeitos foram desenvolvidos métodos de explicação com o simples objetivo de fornecer explicações, e, com isso, aumentar a confiança que os utilizadores têm para com estes métodos. Existem vários métodos de explicação, os mais conhecidos e utilizados são, o Local Interpretable Model-Agnostic Explanations (LIME) e SHapley Additive exPlanations (SHAP).

Wang et al [55], propõem uma *framework* que deteta intrusões com o auxílio de métodos baseados em *machine learning*, para obtenção de uma maior eficiência e precisão. Contudo, com o aumento da complexidade dos métodos, estes tornam-se difíceis de perceber e entender a lógica por de trás de cada decisão, e com isso em mente são aplicados métodos de explicação, com o propósito de simplificar e explicar o que está por detrás de cada decisão fornecendo desta forma explicações com um nível de interpretabilidade que é facilmente entendido pelos humanos.

2.5 Métodos de explicação

Já foi mencionado na secção 2.4, que o aumento da popularidade e desenvolvimentos na área de inteligência artificial, origina a utilização de métodos baseados em *machine learning* para resolver tarefas cada vez mais complexas. Dito isto, pode-se afirmar que nestes anos recentes tornou-se evidente que é de grande importância haver uma forma de interpretar resultados obtidos a partir de métodos que não são facilmente compreensíveis. Devido a esta necessidade foram criadas soluções ao problema de transparência, que resultou no desenvolvimento de métodos *Explainable Artificial Intelligence (XAI)*.

Estes métodos de *XAI* abrange vários aspetos importantes como: transparência, cumprimento de regulamentos, confiança, ética e justiça [57], que podem ser descritos da seguinte forma:

- Transparência e interpretabilidade - A chave para haver uma transparência em sistemas de *machine learning* é a utilização de *XAI*, em que torna possível haver uma visualização clara e perceber o que está por detrás de uma previsão.
- Cumprimento de regulamentos - Com a utilização de *XAI* é mais fácil cumprir com os regulamentos impostos, como o Regulamento Geral sobre a Proteção de Dados (RGPD) [57].
- Justiça e ética - Com a utilização de *XAI* é assegurada justiça e imparcialidade através da análise de padrões.
- Confiança - Com a utilização de *XAI* é fornecido uma explicação por detrás de cada decisão, assegurando desta forma que o utilizador percebe o porquê de uma previsão.

2.5.1 Categorização dos métodos *XAI*

De acordo com a literatura [5, 9, 57] existem vários tipos de categorização dos métodos de *XAI*. Assim sendo, para haver uma melhor descrição e entendimento irá ser realizada uma descrição através das diferentes categorias.

2.5.1.1 *Intrinsic* ou *Post-Hoc*

Esta categorização distingue entre o uso de limitação da complexidade do modelo ou análise metodológica do modelo após treino, que representa respetivamente *Intrinsic* e *post-Hoc*. Estas categorizações podem ser descritas da seguinte forma:

- *Intrinsic* - O método *Intrinsic* cria a explicação ao mesmo tempo que é criada a previsão.
- *Post-Hoc* - No método *post-Hoc* as explicações são criadas quando o modelo já foi treinado e as previsões já foram feitas. Neste caso um método *XAI* é o LIME, pois é um modelo externo sendo desta forma independente ao modelo relacionado com as previsões.

2.5.1.2 Específico ou agnóstico

A diferença entre estes tipos de classificações é o simples facto de que modelos agnósticos poderem ser utilizados com qualquer tipo de métodos, pelo contrário temos os modelos específicos que como o nome indica estes só podem ser usados por modelos específicos.

2.5.1.3 Explicação local ou global

A diferença entre estas duas categorias é que em modelos de explicação em que as decisões são locais, a explicação tem em conta só os fatores locais relativos à esfera da previsão realizada. Pelo contrário, os modelos de explicação em que as decisões são globais tem em consideração aquela previsão, tendo em conta, o conjunto de dados na sua totalidade. Esta diferença tem um grande impacto nos utilizadores, pois são as explicações baseadas em decisões locais que garantem / fornecem uma maior confiança e transparência aos utilizadores. Pois, as diferenças implicam o seguinte: alguns aspetos que são importantes localmente não são necessariamente importantes para explicações globais, isto implica que explicações locais não implicam explicações globais.

2.5.1.4 Tipo de explicação

A explicação é a parte crucial de qualquer método de explicação. A explicação pode ter várias formas como: gráficos, texto, argumentos e baseado em modelos. Quando é referido que as explicações podem ser argumentos, está-se a referir ao realçar de atributos que levaram àquela

decisão para ajudar a perceber a importância de aquele determinado atributo ou conjunto de atributos. Quando é referido que as explicações podem ser baseados em modelos, está-se a referir à aproximação do modelo de caixa negra através de um modelo mais percetível.

2.5.2 Desafios do *XAI*

Como todas as tecnologias existentes *XAI* não é diferente e também tem as suas dificuldades relacionadas com a segurança, desempenho, aspetos legais e interpretação [57]. Essas dificuldades podem ser descrita da seguinte forma:

- Segurança - Alguns modelos *XAI* têm vulnerabilidades que podem ser exploradas por atacantes.
- Desempenho - O desempenho dos métodos *XAI* pode ser medida de várias formas, mas não existe forma de comparação entre os vários métodos de *XAI*.
- Problemas legais e de privacidade - Estes problemas estão relacionados com a gestão de aspetos como privacidade, qualidade e integridade dos dados e também o acesso aos mesmos. Para combater este problema foram criados vários guiões éticos para serem implementados de forma a fornecer privacidade, qualidade e integridade dos dados e também do acesso aos mesmos.
- Interpretação e precisão - Dependendo do conjunto de dados pode haver complicações com o modelo, como é o caso, quando é usado um conjunto de dados com um grande número de dimensões.

2.5.3 Aplicações de *XAI* em segurança informática

A implementação de *XAI* em segurança informática aumenta a capacidade de combate contra vários tipos de ataques, pois ajuda os utilizadores a perceber os ataques que estão a enfrentar tornando assim, os sistemas mais transparentes. Alguns dos ataques mais conhecidos podem ser vistos na Tabela 2.6.

Malware	Software construido com o intuito de causar danos no computador da vítima.
DOS	É um ataque que consiste na negação do serviço, através do esgotamento dos recursos da máquina alvo.
Spam	É um ataque que consiste no envio de um número elevado de mensagens para um ou mais utilizadores.
Phishing	É um ataque de engenharia social com o objetivo de enganar utilizadores para obter informações confidenciais.

Tabela 2.6: Tipos de ataques mais conhecidos.

Capítulo 3

Metodologia

Nesta secção vão ser descritas as escolhas realizadas para a realização deste trabalho.

3.1 Escolha dos métodos de deteção

Como foi dito anteriormente na secção 2.3.2, para a escolha dos métodos a utilizar têm que ser considerados vários aspetos, sendo alguns desses aspetos os seguintes: ambiente em que esse método vai ser inserido, tipo de aprendizagem, entre outros. Desta forma depois de uma longa ponderação àcerca de que métodos escolher chegou-se a conclusão que os métodos Half Space Trees (HSTrees) e Isolation Forest Algorithm for Stream Data (IForestASD) seriam os mais apropriados. Um dos fatores que influenciou na escolha do método foi o facto destes métodos terem uma aprendizagem não supervisionada, o que permite que estes possam ser utilizados com qualquer tipo de conjunto de dados.

3.1.1 Half Space Trees

Ao contrário dos outros métodos que lidam com conjuntos de dados limitados, existem áreas que necessitam de métodos que detetem anomalias em conjuntos de dados infinitos. Para responder a esta necessidade foram criados vários métodos que lidasse e se adaptem a fluxos de dados que estão em constante mudança como é o caso do método HSTrees [50]. Existem vários problemas associados ao fluxo contínuo de dados, em que alguns desses são os seguintes: o primeiro problema relacionado com o fluxo contínuo de dados, como o nome indica é o facto de o conjunto de dados ser infinito e caso haja a tentativa de armazenar o mesmo irá proporcionar um erro de falta de memória. O segundo problema está relacionado com a percentagem de pontos normais relativamente aos pontos anómalos, e como existe uma grande disparidade, os métodos que necessitem a presença de dados anómalos corretamente identificados estão em desvantagem. O terceiro problema é o facto de o fluxo de dados estar em constante alteração o que implica que o método tem que ser flexível o suficiente para se adaptar ao fluxo.

Em resposta aos problemas mencionados acima foi criado / desenvolvido uma solução que desse resposta aos problemas e necessidades descritas anteriormente. Com esse sentido foi criado o método de deteção de anomalias HSTrees, que contém vários aspetos que o diferenciam dos restantes métodos existentes. Esses aspetos são os seguintes: o primeiro aspeto é o facto de este método necessitar de uma quantidade de memória constante ao longo da sua utilização ao contrário dos seus competidores que tem um consumo de memória variável. O segundo aspeto é o facto de este método não ser influenciado pela discrepância da percentagem entre pontos normais e pontos anómalos. O terceiro aspeto é o facto de o método realizar atualizações em si mesmo com o objetivo de manter o nível de precisão elevado. Para além do que foi dito anteriormente o método HSTrees, pode ser construído com pequenas amostras do conjunto de dados, o que permite ao mesmo ser rápido e flexível no fluxo contínuo de dados. Uma diferença que é facilmente salientada em relação aos outros métodos baseados em árvores é o facto do método HSTrees funcionar com base na árvore criada a partir das dimensões do espaço de dados e não a partir das árvores de treino, o que faz com que as árvores criadas consigam adaptar-se facilmente. Uma consequência positiva que advém desta forma de funcionamento é o facto de garantir um requisito de memória e complexidade de tempo constante.

De uma forma simplificada pode-se dizer que o algoritmo HSTrees é a construção de um conjunto de árvores, em que cada uma dessas consiste num conjunto de nós, que capturam o número de objetos que estão dentro de um subespaço em particular [50]. Em concordância com o número de objetos presentes nos subespaços é criado um perfil utilizado para calcular o nível de anomalia, que em comparação com outros métodos, este torna-se mais rápido e simples de calcular.

Para este mecanismo ser mais flexível, o algoritmo divide o fluxo em janelas de tamanho igual, em que cada janela contém o mesmo número de pontos. O algoritmo funciona com duas janelas em simultâneo. Estas são: a janela de referência e a última janela. Na fase inicial de deteção de anomalias o método aprende na janela de referência com o objetivo de criar um perfil. Com a criação deste perfil, os dados da última janela vão ser avaliados segundo este perfil, em que os subespaços que contenham um número elevado são considerados normais enquanto os subespaços com baixa quantidade são considerados anormais. Quando este processo termina, novos dados são carregados para a última janela e o perfil é guardado na janela de referência, reescrevendo o antigo perfil. Com este método o perfil mais recente é utilizado para avaliar os dados que vão chegando, sendo que este processo repete-se até acabar os dados.

O ponto de vista geral deste método é intuitivo, porém uma definição mais técnica também é necessária para ter um melhor entendimento acerca deste mesmo método.

Dessa forma, a definição deste método pode ser descrita da seguinte forma: O algoritmo HSTrees consiste na criação de uma árvore de profundidade X, em que todas as folhas se encontram à mesma profundidade, X, e que os nós se expandem por meio da seleção de uma dimensão D, que é arbitrariamente escolhida em relação ao nó. De seguida, é dividido a meio a dimensão D, criando desta forma o filho esquerdo e direito do nó original. Esta expansão continua até ser atingida a profundidade X, previamente escolhida.

Cada nó guarda uma série de elementos referentes à dimensão que estão associados, como a quantidade máxima e mínima de objetos, os perfis da janela de referência e da última janela, profundidade do nó atual e os nós filhos do nó atual.

Na Figura 3.1, pode-se visualizar como é que o método HSTrees funciona numa simples janela de dados.

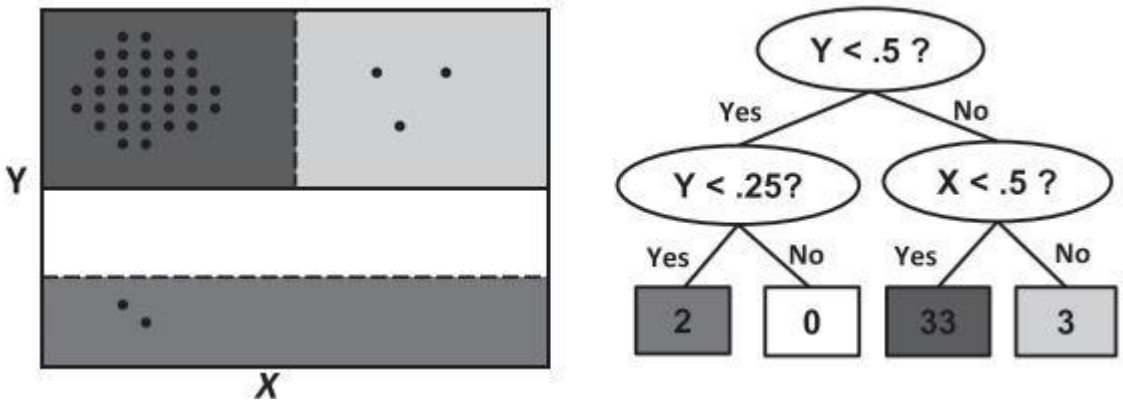


Figura 3.1: Um exemplo de dados (em janela) particionados pelo método HSTrees retirado de [50]

Como se pode ver na Figura 3.1, os eixos X e Y foram divididos múltiplas vezes até ser atingida a profundidade previamente estipulada. Devido a esta divisão foram obtidos os resultados da quantidade máxima e mínima de objetos.

3.1.2 IForestASD - Isolation Forest Algorithm for Stream Data

Ao contrário do método descrito anteriormente, o método IForestASD, tem a sua origem num outro algoritmo, Isolation Forest (IForest), desenvolvido inicialmente para conjuntos de dados estáticos [51]. Como o método IForestASD se trata de uma adaptação do algoritmo original, IForest, esta secção vai descrever o algoritmo original, pois o algoritmo IForestASD apenas adaptou o algoritmo original para aprendizagem incremental através da utilização de janelas.

3.1.2.1 IForest

Toni Liu et al [53], propõem um modelo de deteção de anomalias chamado de IForest, que é diferente dos métodos previamente existentes no seguinte aspeto: enquanto os outros modelos identificam anomalias com base na criação de um perfil, em que se um dado ponto não encaixar nesse perfil é dado como sendo uma anomalia, no caso do IForest, este método isola as anomalias em vez de criar um perfil do que é suposto os pontos normais serem. Este método utiliza técnicas como subsampling, o que faz com que o algoritmo seja rápido e eficiente em modelos de grande

dimensão. A maior parte dos métodos de deteção de anomalias criam um perfil dos pontos normais, e depois comparam os dados com esse perfil para descobrir as anomalias. A utilização desta técnica apresenta algumas desvantagens em relação à técnica utilizada por IForest, que são as seguintes:

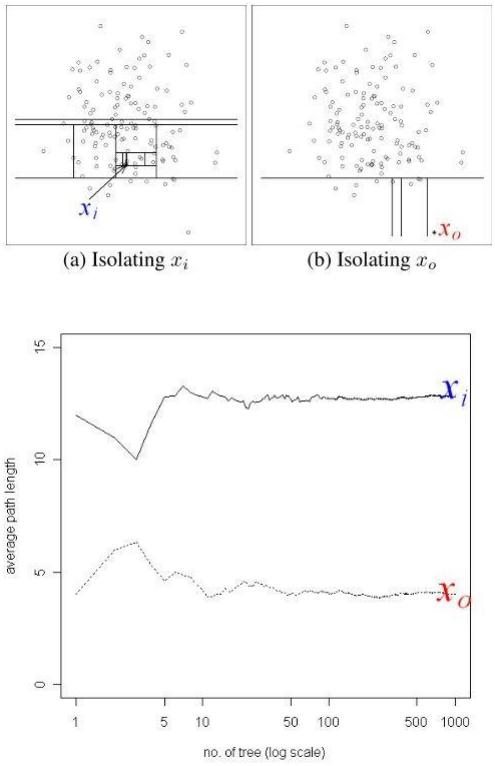
- Ao contrário do que parece, o método não está especificado para encontrar anomalias, mas sim para criar perfis de dados normais, o que pode resultar na fraca qualidade de deteção de anomalias.
- Ao contrário do IForest os outros métodos são computacionalmente caros, o que os restringe no tipo de conjuntos de dados em que podem ou devem ser utilizados.

O algoritmo IForest, é um método estruturado em árvore em que as anomalias estão mais perto da raiz, porque estes são mais fáceis de isolar, enquanto os pontos normais é o contrário. Para além do que foi dito acima, outros aspetos distinguem IForest dos outros métodos que são os seguintes:

- As características de isolamento de pontos permite que só seja necessário criar modelos parciais, pois não é necessário isolar pontos normais, permitindo desta forma também reduzir efeitos de *swamping* e *masking*.
- Não utiliza técnicas de distância e densidade, tendo desta forma um fardo computacional menor.
- Este método não é computacionalmente exigente, sendo desta forma rápido na sua execução.
- IForest tem a capacidade de lidar com conjuntos de dados com grandes dimensões e com um grande número de atributos irrelevantes.

Para alcançar o objetivo de isolar as anomalias, em vez da utilização da técnica utilizada pelos outros métodos, foram aproveitadas duas vantagens que são as seguintes: a primeira vantagem é o facto de o número de anomalias ser em menor número comparativamente aos dados normais, e a segunda vantagem é o facto de os valores dos atributos das anomalias ser diferente relativamente aos dados normais, o que vai fazer com que sejam separados na árvore criada. A árvore está diretamente relacionada com os pontos que são isolados e as anomalias não têm necessidade de ter tantas partições relativamente aos pontos não anómalos. Estas partições são criadas através da seleção aleatória de um atributo e consequentemente a seleção de um valor entre o valor máximo e mínimo desse mesmo atributo. Este processo é recursivo uma vez que as partições podem ser traduzidas numa estrutura em árvore. O número de partições necessárias para isolar um ponto são diretamente proporcionais à profundidade da árvore. Isto pode ser visualizado na seguinte Figura 3.2.

Como se pode ver na Figura 3.2 retirada de [53], podemos verificar que as anomalias são mais vulneráveis a isolamento e em consequência têm uma profundidade reduzida comparativamente



(a) Isolating x_i (b) Isolating x_o

(c) Average path lengths converge

Figura 3.2: Isolamento de anomalias utilizando o método IForest retirado de [53]

aos pontos normais. Nesta Figura, estão enquadradas três Figuras, sendo cada uma delas identificada por (a), (b) e (c) respetivamente. Na Figura 3.2 (a) pode-se ver que o isolamento de um ponto normal necessita de um número de partições superior comparativamente à Figura 3.2 (b) na qual se trata de uma anomalia. Isto também se verifica na Figura 3.2 (c), a qual representa a profundidade média de ambos os pontos retratados nas Figuras 3.2 (a) e (b), em que se chega à conclusão que a árvore está diretamente relacionada com os pontos que são isolados.

Um aspeto que deve ser salientado no algoritmo IForest, é o facto de neste modelo ser preferível amostras de dados pequenas, o que vai contra a metodologia dos métodos até agora desenvolvidos que preferem amostras de dados com grandes dimensões. Isto deve-se ao facto de o algoritmo não ter que isolar todos os pontos normais para construir o modelo. Para além disso, se o conjunto de amostras for de grandes dimensões provoca interferência na habilidade de classificação do modelo. Para evitar este problema o algoritmo utiliza subsampling, que faz com que sejam utilizados apenas pequenas amostras de dados de cada vez.

Contudo, este não é o único problema deste algoritmo, já que este método também é afetado por *swamping* e *masking* [56]. *swamping* consiste na identificação de pontos normais como sendo anomalias, porque a distância entre dois pontos, sendo um deles uma anomalia, é pequena o que torna difícil a sua distinção. Relativamente ao *masking* este consiste na existência de múltiplas anomalias próximas umas das outras o que torna o seu isolamento mais difícil. Estes dois pontos sobrepõem-se no facto de ambos estes problemas terem como raiz do problema o

facto de existirem muitos pontos nas amostras a serem utilizadas. Porém, devido à característica peculiar do IForest, que permite a criação de modelos parciais com a utilização de subsampling, resulta na suavização dos problemas mencionados acima e isto acontece por dois motivos, sendo estes os seguintes:

- Com a utilização de subsampling existe um controlo do tamanho do conjunto de dados.
- Cada árvore criada é especializada nas amostras que podem conter anomalias ou não.

3.2 Escolha dos métodos de explicação

Como foi mencionado na secção 2.5, existem vários tipos de métodos de explicação que dependendo do ambiente que são implementados podem ser mais ou menos úteis. Os métodos de explicação mais conhecidos são o Local Interpretable Model-Agnostic Explanations (LIME) e o SHapley Additive exPlanations (SHAP).

Métodos como o LIME e o SHAP foram criados com um único propósito em vista. Esse é explicar os métodos baseados em caixa negra, pois os utilizadores não percebem como é que o método funciona nem como chegou a um determinado resultado, [4, 29, 45, 55, 57]. Ambos os métodos mencionados anteriormente foram criados com o objetivo de formar a ponte entre os métodos de *machine learning* e os utilizadores. Estes métodos partilham algumas características como o caso de serem ambos agnósticos. Porém, no caso do SHAP existem métodos que são criados a partir deste que são específicos. Outra semelhança que estes dois métodos partilham são as explicações em que no caso do LIME estas são locais e no caso do SHAP as explicações podem ser locais ou globais. Porém, existem algumas diferenças como é o caso do funcionamento dos métodos e o caso da interpretabilidade versus precisão. No caso da interpretabilidade versus precisão o LIME sacrifica um pouco da sua precisão em troca de haver uma maior interpretabilidade. Contudo, no caso do SHAP, este não sacrifica a precisão nem a interpretabilidade. No que toca ao funcionamento dos métodos estes podem ser descritos da seguinte forma:

- LIME - Na seguinte referência [45] esta, descrito este método. Este método pode ser descrito como sendo essencialmente a construção de modelos locais simples, para que seja possível realizar uma substituição entre o modelo complexo do método e o modelo criado à posteriori, com o objetivo de fornecer uma explicação facilmente compreensível. A ideia de funcionamento por de trás do LIME pode ser dividida em duas fases. A primeira fase compreende-se em criar um conjunto de dados consistindo na permutação dos dados originais. A segunda fase consiste em treinar o modelo interpretável a partir do novo conjunto de dados [45].
- SHAP - Na seguinte referência [29] está descrito este método. Este método pode ser descrito como sendo essencialmente uma teoria de jogo em que há a explicação da contribuição de

cada atributo presente. Este conceito é muito usado em jogos, em que a contribuição de cada jogador para um dado objetivo é avaliada, em que conforme maior for a contribuição maior será o ganho desse mesmo jogador. Desta forma pode-se dizer que as explicações são feitas através do cálculo da contribuição de cada atributo [29].

Um outro aspeto que difere entre os dois métodos descritos anteriormente é que na utilização do LIME, para além da visualização gráfica da explicação também são fornecidos alguns dados adicionais como: `right, prediction_local e intercept`. A caracterização destes dados é a seguinte:

- `right` - Este dado indica o nível de previsão àcerca do mesmo.
- `prediction_local` - Este dado indica o valor criado pelo modelo de teste em relação ao número de atributos principais a este dado ponto.
- `intercept` - Este dado é o valor constante fornecido pelo modelo de previsão, naquele vetor de teste.

No que diz respeito à visualização gráfica, as cores laranja e azul correspondem às associações positivas e negativas, respetivamente, em relação ao ponto explicado.

Para a resolução deste trabalho foi decidido utilizar apenas o método LIME.

3.3 Seleção do conjunto de dados

Para a resolução deste trabalho foram selecionados dois conjuntos de dados que são designados por *KDD99* e *ICS-Flow*, que são descritos abaixo.

3.3.1 Conjunto de dados *KDD99*

Para a realização deste trabalho foi escolhido um conjunto de dados muito usado a um nível académico, por ser um conjunto de dados muito completo e que contém um nível de representação do que acontece na vida real, sendo esse conjunto de dados o *KDD Cup 1999* mais conhecido por *KDD99*. Porém, este mesmo conjunto de dados também tem os seus aspetos negativos como a idade e o facto de a diferença entre ligações inofensivas e ligações ofensivas por parte de terceiros não estar devidamente representado neste conjunto de dados. Isto acontece porque existe um número de ataques superior ao número de comunicações inofensivas. Por outro lado, devido a ser um conjunto de dados muito completo, no que diz respeito a ataques, sendo que alguns destes são desproporcionais em número. Este conjunto de dados é utilizado para todo o tipo de investigações relacionadas com os métodos que detetam ataques, *IDS*.

Nome	Tamanho
DARPA 99	10 444 980
KDD99	5 209 460
NSL-KDD	148 517

Tabela 3.1: Tamanho dos conjuntos de dados.

3.3.1.1 História da criação deste conjunto de dados

O conjunto de dados *KDD99* começou com um evento da agência de projetos de pesquisa avançada de defesa mais conhecida como DARPA, com o objetivo de simular um ataque a uma instalação governamental, mais especificamente simular um ataque a uma base da força aérea da América do Norte. Este evento ocorreu em 1998 e foi realizado pelo Massachusetts Institute of Technology (MIT), com o objetivo de avaliar sistemas de deteção de intrusão, *IDS* [42, 48]. Deste evento resultou um conjunto de dados designado por DARPA, que consiste nos dados no seu estado mais natural.

O conjunto de dados *KDD99* é o resultado da transformação do conjunto de dados DARPA ao qual foram retirados os atributos e foi realizado um pré-processamento com o objetivo de tornar este conjunto de dados muito mais fácil de utilizar.

Apesar de o conjunto de dados *KDD99* ser de mais fácil utilização do que o conjunto original, existe um outro conjunto de dados criado a partir do *KDD99* que tem um tamanho reduzido e os pontos duplicados removidos. Este conjunto de dados tem a designação de *NSL-KDD*.

Os respetivos tamanhos dos conjuntos de dados podem ser vistos na Tabela 3.1.

Como foi mencionado anteriormente, este conjunto de dados é desproporcional tendo um número de ataques vastamente superior, comparativamente ao fluxo de dados normais [42, 48], o que contraria o que acontece na vida real, em que uma grande parte do fluxo de dados são inofensivos e uma pequena percentagem representa o fluxo de dados ofensivos. Para além da desproporcionalidade dos fluxos ofensivos e inofensivos, também há uma desproporcionalidade no que toca à variedade de ataques e existem pontos duplicados.

Por outro lado, uma característica que faz com que muitas das utilizações apenas utilizem 10% do conjunto de dados é o facto de este ter grandes dimensões.

Este conjunto de dados apresenta cinco categorias de dados, sendo essas as seguintes:

1. Dados normais.
2. *Denial of Service (DOS)* - Consiste em negação de serviços.
3. *Root to Local (R2L)* - Consiste em acesso não autorizado a privilégios de superusuário local (root).
4. *R2L* - Consiste em acesso não autorizado de uma máquina remota.

back *DOS*
teardrop *DOS*
pod *DOS*
neptune *DOS*
smurf *DOS*
land *DOS*
spy *R2L*
buffer overflow *R2L*
rootKit *R2L*
perl *R2L*
load Module *R2L*
warezmaster *R2L*
warezclient *R2L*
multihop *R2L*
phf *R2L*
imap *R2L*
guess password *R2L*
ftp write *R2L*
satan *Probe*
nmap *Probe*
ipsweep *Probe*
portsweep *Probe*

Tabela 3.2: Tipos de ataques existentes no conjunto de dados *KDD99*.

5. *Probing* - Consiste em vigilância.

Nas categorias de ataques existem no total vinte e dois, (22), tipos de ataques como se pode ver na Tabela 3.2 .

3.3.2 Conjunto de dados *ICS-Flow*

Da mesma forma que foi selecionado o conjunto de dados anteriormente mencionado, este foi escolhido por razões óbvias como o facto de ser um conjunto de dados atualizado, tendo sido criado em 2023, e pelo facto de ser completo e realista para fins de treinamento de mecanismos de *IDS* [15].

Um dos aspetos que levou a criação deste conjunto de dados foi o facto de não haver conjuntos de dados apropriados para a avaliação de algoritmos de mecanismos de *IDS*.

Este conjunto de dados tem 4 (quatro) tipos de ataques presentes, que são os seguintes:

- *Replay Attack*.

- *DOS*.
- *Scan*.
- *Man in the middle attack (MITM)*.

3.3.3 Desafios relacionados com a seleção dos conjuntos de dados

Em todas as tecnologias existem desafios e os conjuntos de dados não estão isentos desses desafios [38]. Os conjuntos de dados enfrentam os seus próprios desafios, que são os seguintes:

- Conjuntos de dados desatualizados, no caso do *KDD99* este já tem mais de 20 anos.
- Número limitado de amostras.
- Distribuição de dados não representativa.
- Grande parte dos conjuntos de dados são privados.
- Falta de métodos de avaliação dos conjuntos de dados.

Para além disso, os conjuntos de dados anteriormente descritos apresentam um número de atributos muito elevado, isso faz com que o processo de treinar os modelos de deteção de anomalias sejam lentos e de certa forma ineficaz. Isto deve-se ao grande número de atributos que os conjuntos de dados têm.

Uma forma de mitigar este problema é através da utilização de mecanismos de seleção de atributos. A seleção de atributos consiste na seleção dos mesmos com base em testes estatísticos [8]. Este processo pode ser definido como um pré-processamento, que é realizado antes do modelo de deteção de anomalias ser treinado, em que são selecionados apenas os atributos mais valorizados de forma a ter um melhor conjunto de dados. Para isso, foi utilizado o algoritmo *SelectKBest*, [8], na qual foram selecionados os 15 atributos com maior valorização.

Nos testes realizados, que estão mais a frente descritos, foram utilizados os conjuntos de dados com e sem a seleção de atributos, com o objetivo de realizar uma comparação entre estes.

Capítulo 4

Experiências e Resultados

Nesta secção são discutidas as experiências que foram realizadas, como é que o modelo foi treinado e o desempenho dos modelos. Isto, para que fosse possível haver respostas a várias perguntas como: As explicações são iguais para o mesmo ataque? Para diferentes ataques existem diferentes explicações? Existem semelhanças entre as explicações do mesmo método de deteção de anomalias? Entre outros pontos que são importantes para a realização deste trabalho.

Este capitulo esta dividido em três partes. A primeira parte trata-se da avaliação do desempenho dos métodos escolhidos. A segunda parte trata-se dos testes realizados no conjunto de dados *KDD99*, enquanto a terceira parte diz respeito aos testes realizados ao conjunto de dados *ICS-Flow*. Relativamente aos testes realizados nas secções 4.2.1, 4.2.2, 4.2.3, 4.3.1, 4.3.2 e 4.3.3 foram utilizados os conjuntos de dados dos respetivos sites oficiais, na ordem original, na qual só foi utilizado o segundo exemplo de cada ataque para a explicação. Porém, nas secções 4.2.4 e 4.3.4 foram utilizados todos os ataques presentes nos respetivos conjunto de dados.

Para a realização deste trabalho foi utilizado a *framework* Python Streaming Anomaly Detection (PySAD), na qual foram utilizados os métodos anteriormente descritos, na qual ambos utilizam janelas de tamanho 100. Para além disso, no método HSTrees também foi especificado o número de atributos máximo e mínimo, sendo esses respetivamente 15 e 20. Para os restantes atributos foram utilizados os valores padrão.

4.1 Avaliação de desempenho

Para avaliar o desempenho foram recolhidas as seguintes métricas: tempo de treino de cada método utilizado e *Area Under the ROC Curve (AUC)*. Para explicar o que é *AUC* tem que se explicar o que é *Receiver Operating Characteristic (ROC)* [16, 22, 32].

ROC é um gráfico de duas dimensões que ilustra o desempenho do modelo. O gráfico *ROC* é construido com a taxa de verdadeiros positivos no eixo Y e a taxa de falsos positivos no eixo X, como se pode ver na Figura 4.1. A curva *ROC* pode ser descrita como um limiar entre os

verdadeiros positivos e os falsos positivos do método em todos os pontos limiares de classificação. Isto pode ser visualizado na seguinte Figura 4.1. Como a comparação das curvas *ROC* seria um processo muito dispendioso a nível de recursos, foi desenvolvido uma forma de comparação mais eficiente designada por *AUC*. *AUC*, não é nada mais, do que, a área de baixo da curva *ROC*.

Os valores variam entre 0, (zero), e 1, (um), em que, quanto maior for o resultado, melhor é o modelo. Dito tudo isto, pode-se comparar os valores obtidos na Tabela 4.3 em que se chega à conclusão que o método IForestASD é melhor comparativamente ao método HSTrees.

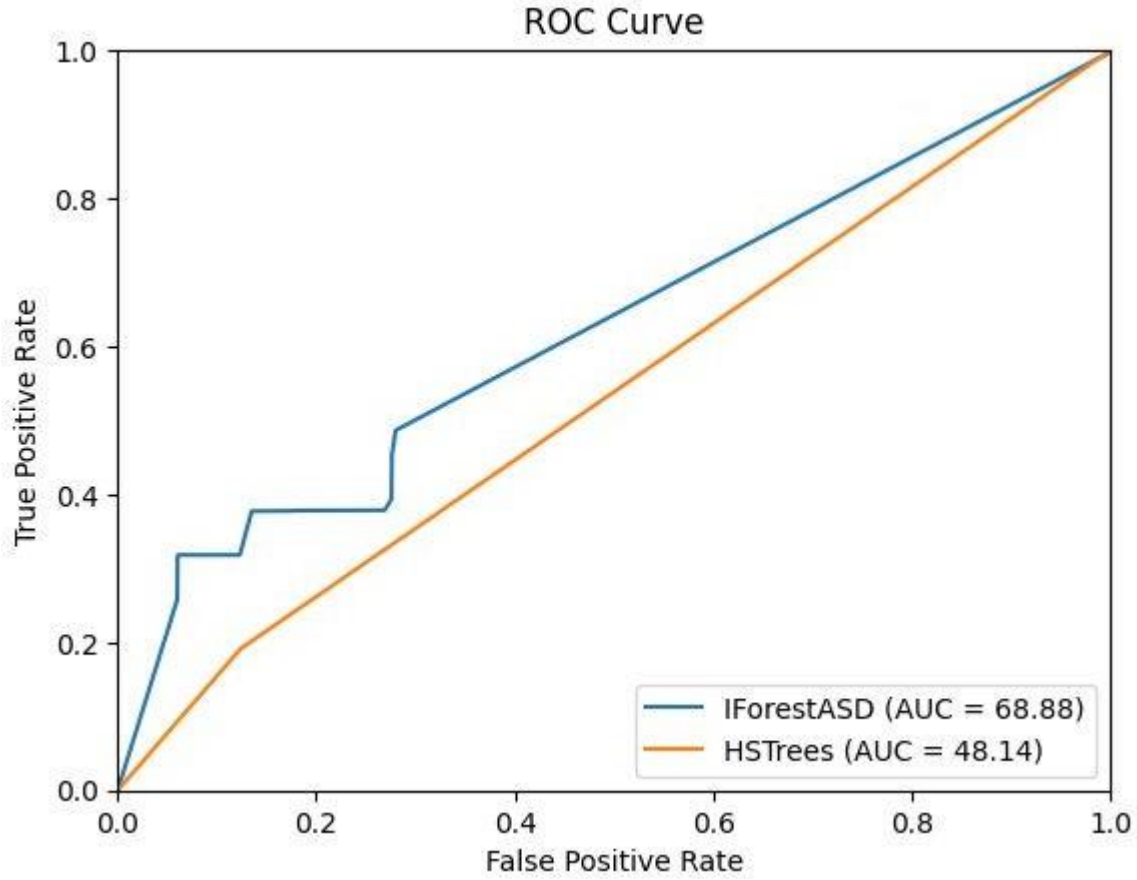


Figura 4.1: Curva *ROC*

Relativamente ao tempo de treino dos métodos, pode-se dizer que neste caso o contrário se verifica comparativamente ao que foi dito anteriormente. Neste caso, o método HSTrees demonstrou resultados vastamente superiores comparativamente ao método IForestASD, como se pode ver nas Tabelas 4.1 e 4.2. Porém, esta métrica de comparação é irrelevante, uma vez, que o tempo de treino dos métodos está associada ao equipamento utilizado, por isso, não é dada grande importância a este método de comparação ao longo deste trabalho.

Modelo	10% *KDD99*	100% *KDD99*	100% *KDD99* (utilização de seleção de atributos)
HSTrees	0h:4min:4sec	4h:32min:10sec	5h:39min:17sec
IForestASD	0h:2min:53sec	8h:23min	5h:37min:20sec

Tabela 4.1: Tempo de treino dos respetivos métodos no conjunto de dados *KDD99*.

Modelo	*ICS-Flow*	*ICS-Flow* (utilização de seleção de atributos)
HSTrees	0h:1min:31sec	0h:1min:26sec
IForestASD	0h:2min:18sec	0h:2min:19sec

Tabela 4.2: Tempo de treino dos respetivos métodos no conjunto de dados *ICS-Flow*.

Modelo	*AUC* %
HSTrees	48
IForestASD	68.88

Tabela 4.3: Métrica *AUC* dos respetivos métodos.

4.2 Resultados acerca da explicação local relativamente ao con- junto de dados *KDD99*

Os resultados àcerca da explicação dos métodos anteriormente descritos, são debatidos nesta secção. Para além disso, também são examinadas as diferenças entre as explicações provenientes dos vários métodos de deteção que neste caso são o IForestASD e HSTrees. A organização desta secção vai ser a seguinte: primeiro vai ser comparada a explicação dos modelos em dois pontos distintos, em que numa primeira parte, os modelos são treinados só com 10% do total do conjunto de dados, na qual são descritas as diferenças e semelhanças em ambas as explicações, de ambos os pontos. Numa segunda fase, o processo descrito anteriormente é repetido, mas, para a totalidade do conjunto de dados, em que de seguida são comparados os resultados obtidos com os dados anteriormente obtidos. Na terceira fase, vai ser comparada a explicação de dois tipos de ataques, com os modelos treinados com a totalidade do conjunto de dados. Numa quarta fase vai ser feita a comparação dos restantes ataques.

Para a realização destas comparações foram selecionados dois pontos do conjunto de dados que são dados como anomalias, devido ao valor de ambos os pontos se aproximarem do valor máximo possível, do que, caracteriza uma anomalia no respetivo método. As Figuras abaixo são o resultado da explicação de cada um dos métodos relativamente a cada um dos pontos, sendo que as Figuras 4.2 e 4.3 e as Figuras 4.4 e 4.5 são as respetivas explicações para os respetivos pontos.

Relativamente à comparação do primeiro ponto pode-se observar na Figura 4.2 que os quatros principais atributos com mais importância para categorização do ponto como anomalia são *duration, is_host_login, num_outbound_cmds e root_shell.* Porém, na Figura 4.3 os quatro

principais atributos que contribuem para a categorização de anomalia são *num_files_creations, duration, num_acess_files e is_host_login.*

Como se pode ver nas respetivas Figuras 4.2 e 4.3 existem atributos coincidentes que são os seguintes: *duration, num_acess_files, wrong_fragment, is_host_login, num_outbound_cmds e num_files_creations.* Da mesma forma que acontece para os atributos que contribuem para uma categorização de anomalia, o inverso também se verifica, em que para ambas as explicações o mesmo atributo é igual. Porém, não se encontra com a mesma categorização, sendo neste caso os atributos *num_acess_files e num_files_creations.*

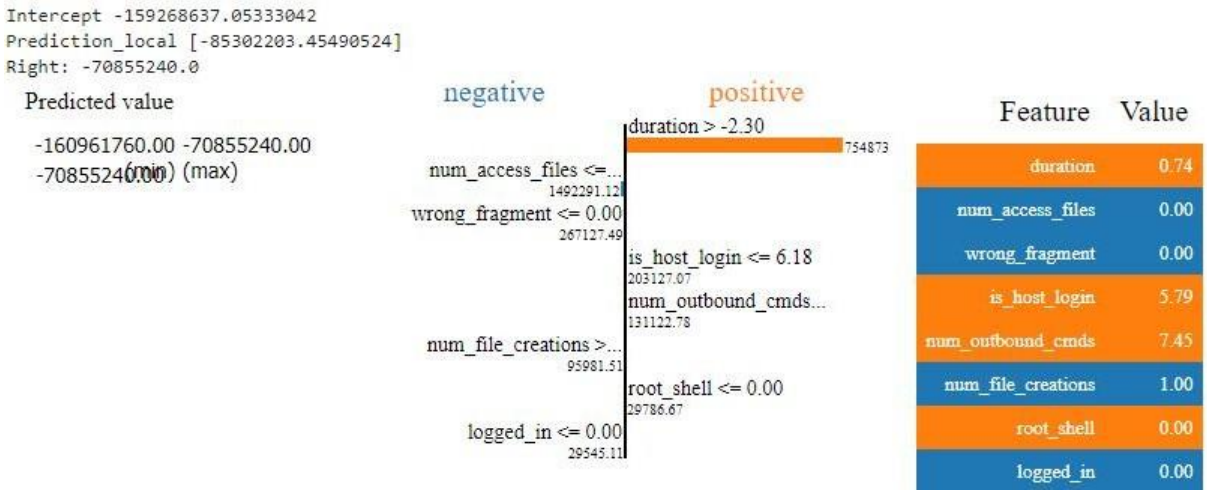


Figura 4.2: Explicação local do algoritmo LIME do primeiro ponto relativo ao método HSTrees (10%)

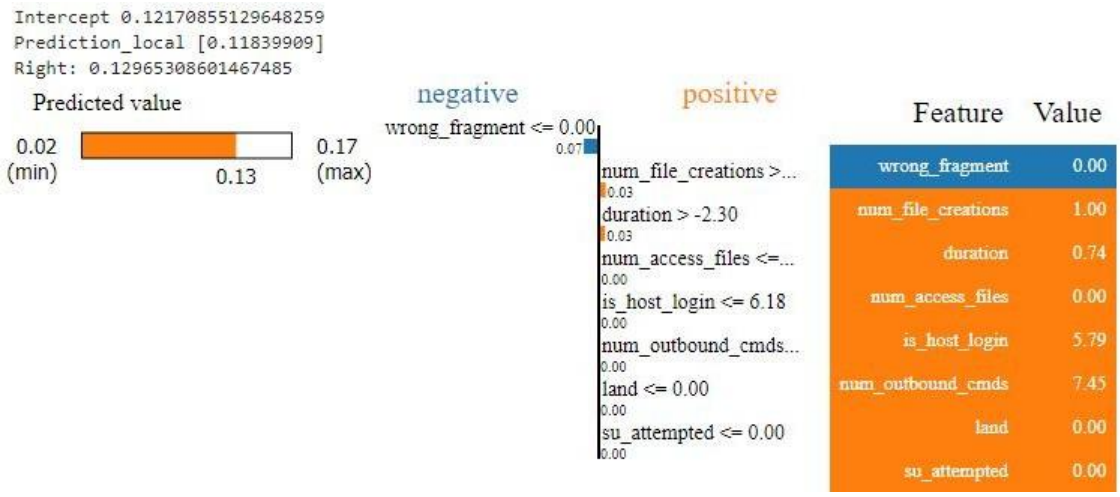


Figura 4.3: Explicação local do algoritmo LIME do primeiro ponto relativo ao método IForestASD (10%)

Relativamente à comparação do segundo ponto pode-se ver na Figura 4.4 que os quatros prin- cipais atributos com mais importância para categorização do ponto como anomalia são *duration, num_file_creations, num_outbound_cmds e is_host_login.* Porém, na Figura 4.5 os quatro

principais atributos que contribuem para a categorização de anomalia são *num_files_creations, duration, is_host_login e dst_bytes.*

Como se pode ver nas respetivas Figuras 4.4 e 4.5 existem atributos coincidentes que são os seguintes: *duration, num_file_creations, num_outbound_cmds, is_host_login, wrong_fragment e su_attempted.* Da mesma forma que acontece para os atributos que contribuem para uma categorização de anomalia, o inverso também se verifica, em que para ambas as explicações o mesmo atributo é igual, porém, não se encontra com a mesma caracterização, sendo neste caso o atributo *su_attempted.*

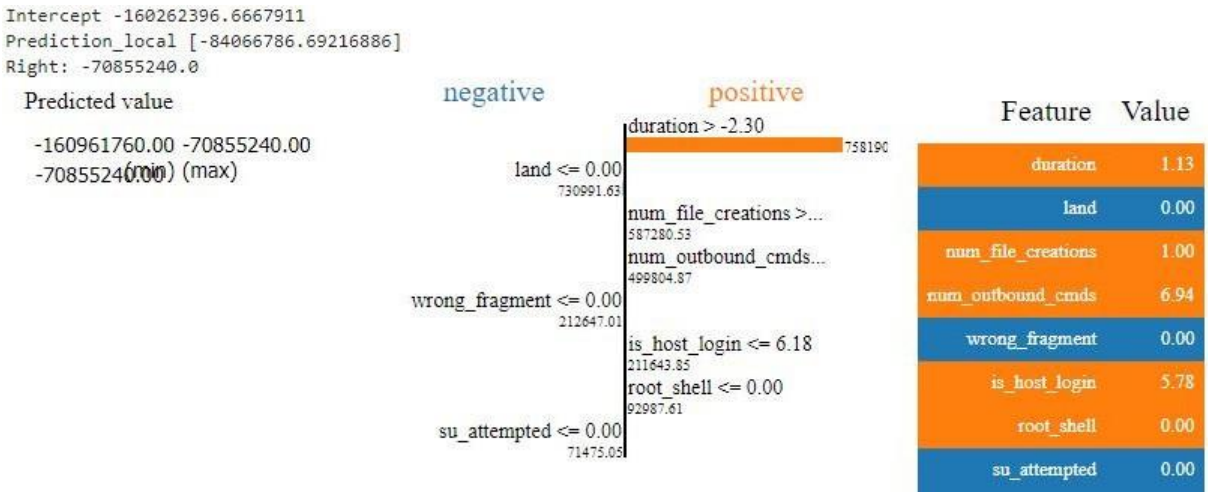


Figura 4.4: Explicação local do algoritmo LIME do segundo ponto relativo ao método HSTrees (10%)

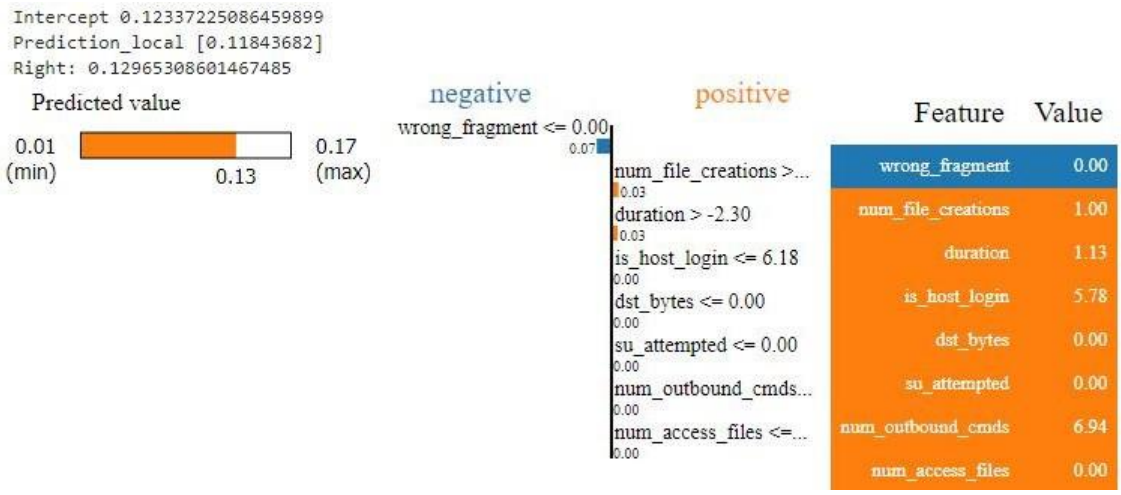


Figura 4.5: Explicação local do algoritmo LIME do segundo ponto relativo ao método IForestASD (10%)

Em relação à comparação anteriormente realizada, essa mesma comparação vai ser feita novamente com os mesmos pontos, mas, com uma única diferença que é o tamanho do conjunto de dados em que os modelos são treinados. A partir desta comparação para a frente todas as

comparações e descrições vão ser realizadas com os modelos treinados na totalidade do tamanho do conjunto de dados.

Relativamente à comparação do primeiro ponto pode-se ver na Figura 4.6 que os quatros principais atributos com mais importância para categorização do ponto como anomalia são: *dst_bytes, num_shells, num_acess_files e wrong_fragment*, enquanto que na Figura 4.7 pode-se ver que os principais atributos são: *srv_diff_host_rate, logged_in e dst_bytes.*

Como se pode ver nas respetivas Figuras 4.6 e 4.7 existe um atributo coincidente e que apresenta a mesma classificação em ambas as explicações, sendo esse atributo o seguinte: *dst_bytes.*

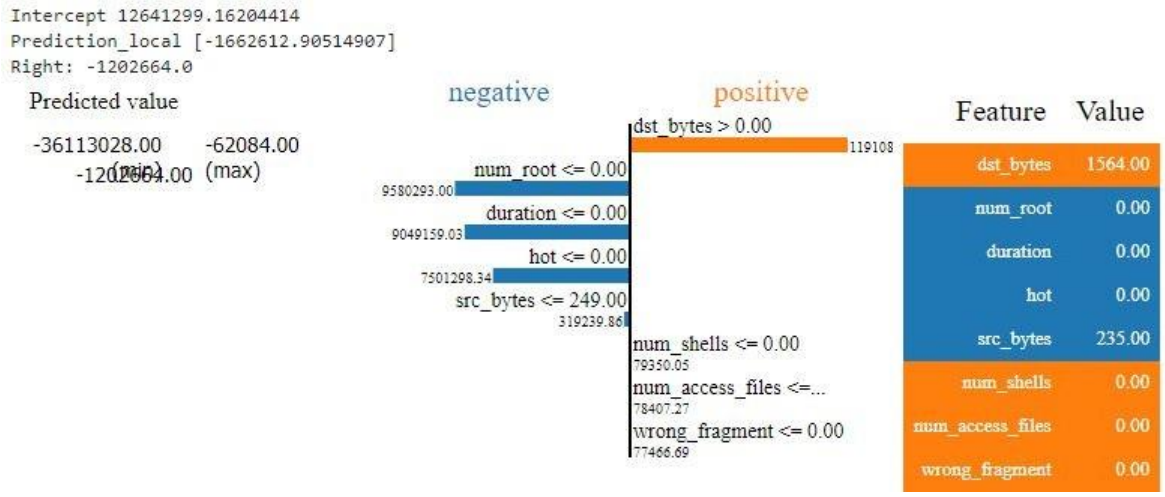


Figura 4.6: Explicação local do algoritmo LIME do primeiro ponto relativo ao método HSTrees (100%)

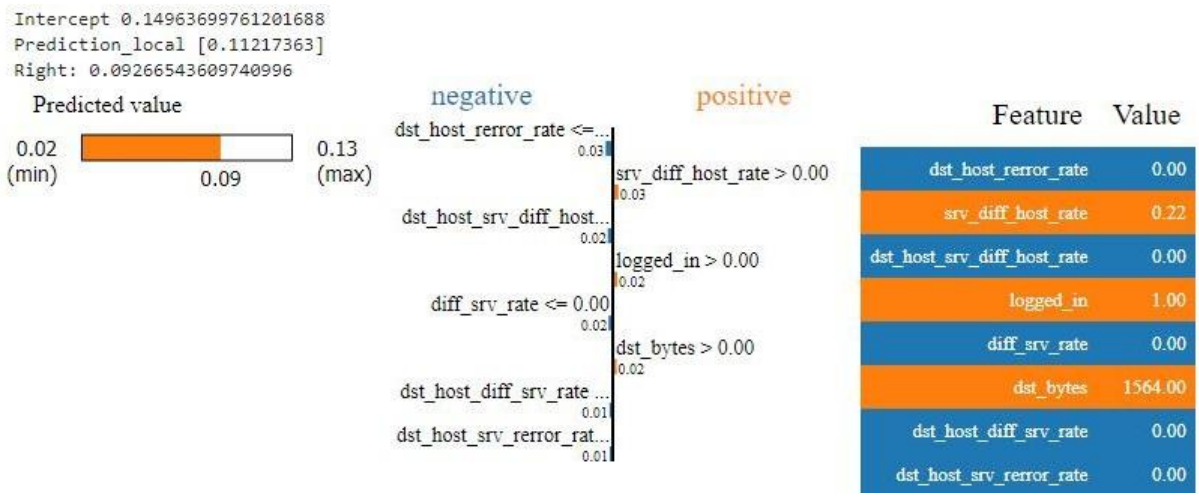


Figura 4.7: Explicação local do algoritmo LIME do primeiro ponto relativo ao método IForestASD (100%)

Relativamente à comparação do segundo ponto pode-se ver na Figura 4.8 que os principais atri- butos com mais importância para categorização do ponto como anomalia são *num_file_creation*

e src_bytes, enquanto que na Figura 4.9 pode-se ver que o principal atributo é *diff_srv_rate.*

Como se pode ver nas respetivas Figuras 4.8 e 4.9 existem atributos coincidentes e que apresenta a mesma classificação em ambas as explicações, sendo esses atributos os seguintes: *duration e num_root.*

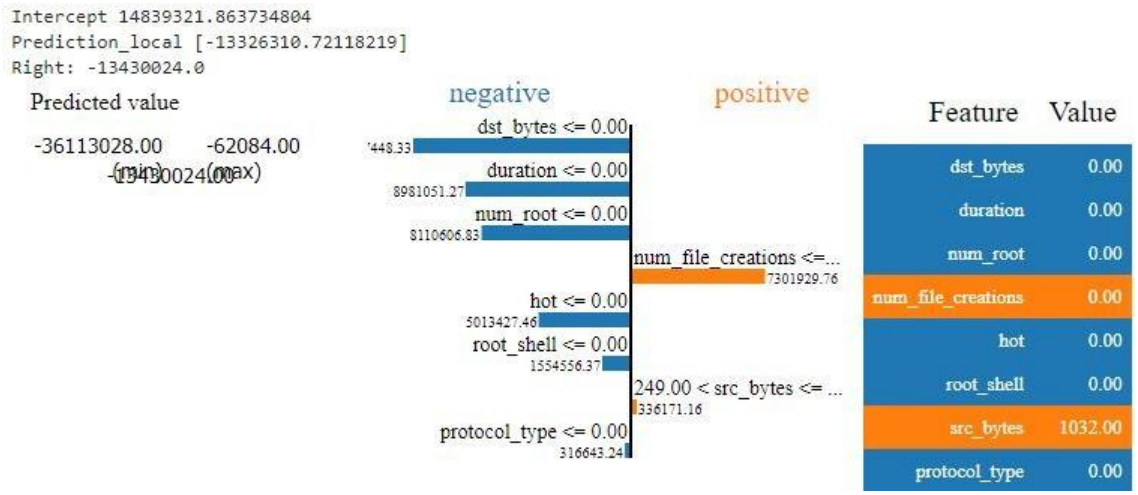


Figura 4.8: Explicação local do algoritmo LIME do segundo ponto relativo ao método HSTrees (100%)

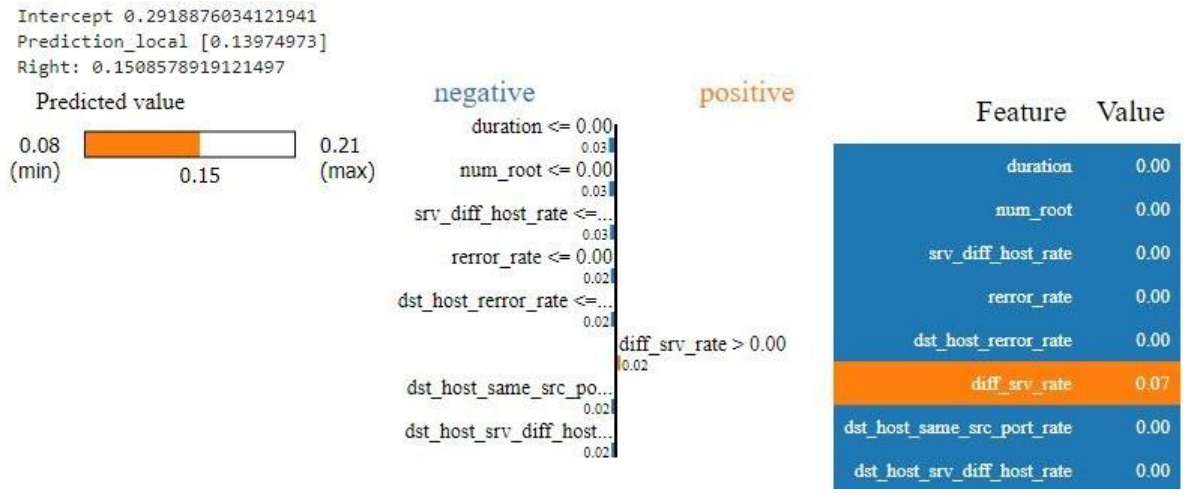


Figura 4.9: Explicação local do algoritmo LIME do segundo ponto relativo ao método IForestASD (100%)

A partir das comparações realizadas pode-se chegar às seguintes conclusões: A explicação proveniente do modelo LIME, não é consistente no que diz respeito à comparação entre o modelo HSTrees treinado com 10% e o modelo que é treinado com 100%, pois os atributos que são dados como favoráveis em ambos os pontos não coincidem com a exceção do atributo *duration.* No caso do modelo IForestASD as explicações também não são consistentes, como acontece no caso do modelo HSTrees, na comparação dos modelos treinados com 10% e 100% respetivamente. Porém, as explicações relativas aos modelos treinados com a mesma percentagem, apresentam

uma consistência elevada relativa à utilização dos mesmos atributos para realizar as explicações, aos respetivos métodos.

4.2.1 Comparação da explicação entre dois ataques diferentes

Nesta secção vai ser realizada a comparação entre dois ataques distintos, sendo esses os seguintes: *buffer overflow* e *neptune*. A comparação vai se centrar no que as explicações têm de comum entre os diferentes modelos e quais são as diferenças entre as explicações do mesmo modelo, com o sentido de perceber quais são os atributos que são mais importantes para a categorização de um tipo de ataque em relação a outro tipo de ataque.

Comparativamente entre os modelos IForestASD e HSTrees, no ataque *neptune* existem dois atributos que são comuns em ambas as explicações e que apresentam o mesmo sentido de classificação, como se pode ver nas Figuras 4.10 e 4.11, sendo esses atributos os seguintes *duratio e num_root*.

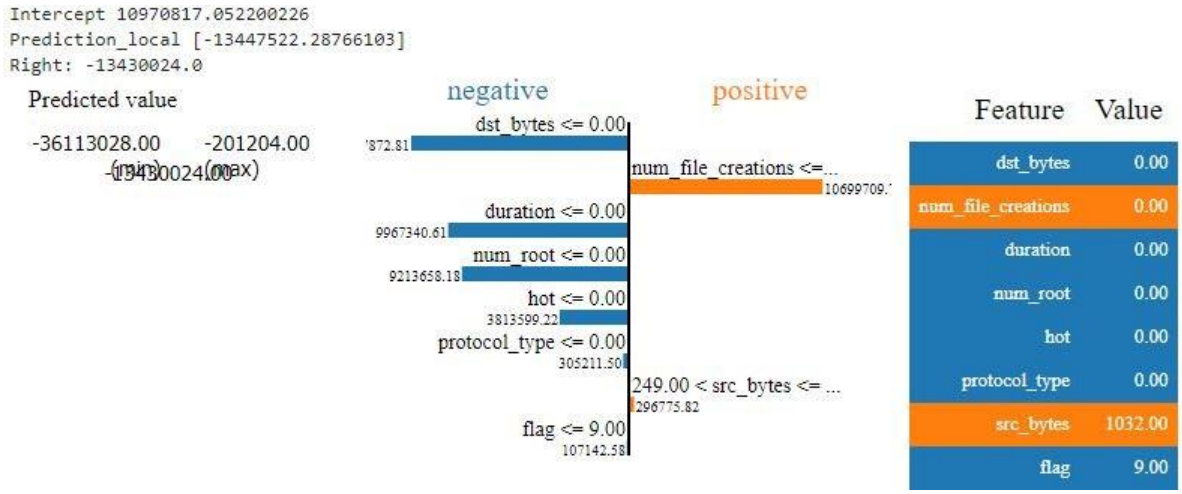


Figura 4.10: Explicação local do algoritmo LIME relativo ao método HSTrees do ataque *neptune*

Comparativamente entre os modelos IForestASD e HSTrees, no ataque *buffer overflow* só existe um atributo que é comum em ambas as explicações e que apresenta o mesmo sentido de classificação, como se pode ver nas Figuras 4.12 e 4.13, sendo esse atributo o atributo *duratio*.

No que toca à comparação das explicações dos diferentes ataques, do modelo HSTrees pode- se concluir que ambas as explicações têm pontos em comum como *duration, num_root , hot, src_bytes e num_file_creations*, que apresentam contribuições no mesmo sentido. Porém, existem mais atributos, coincidentes, que estão presentes em ambas as explicações que são *dst_bytes e flag*, que apresentam contribuições em sentidos opostos nas respetivas explicações.

No que toca à comparação das explicações dos diferentes ataques, do modelo IForestASD estão presentes atributos que apresentam o mesmo sentido de classificação em ambas as explicações, sendo esses atributos os seguintes: *duration, rerror_rate e dst_host_srv_diff_host_rate*. Os restantes atributos não coincidem.

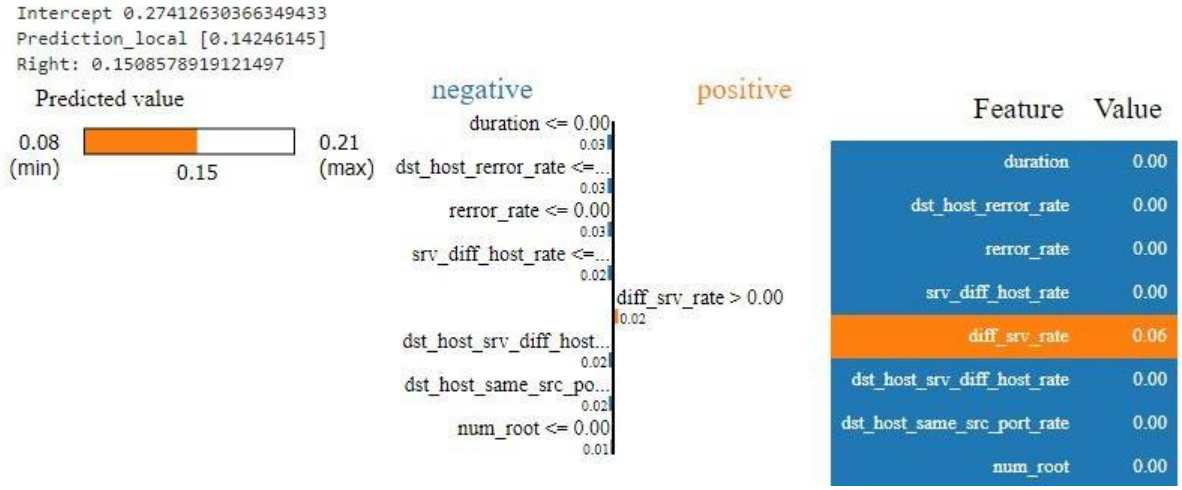


Figura 4.11: Explicação local do algoritmo LIME relativo ao método IForestASD do ataque *neptune*

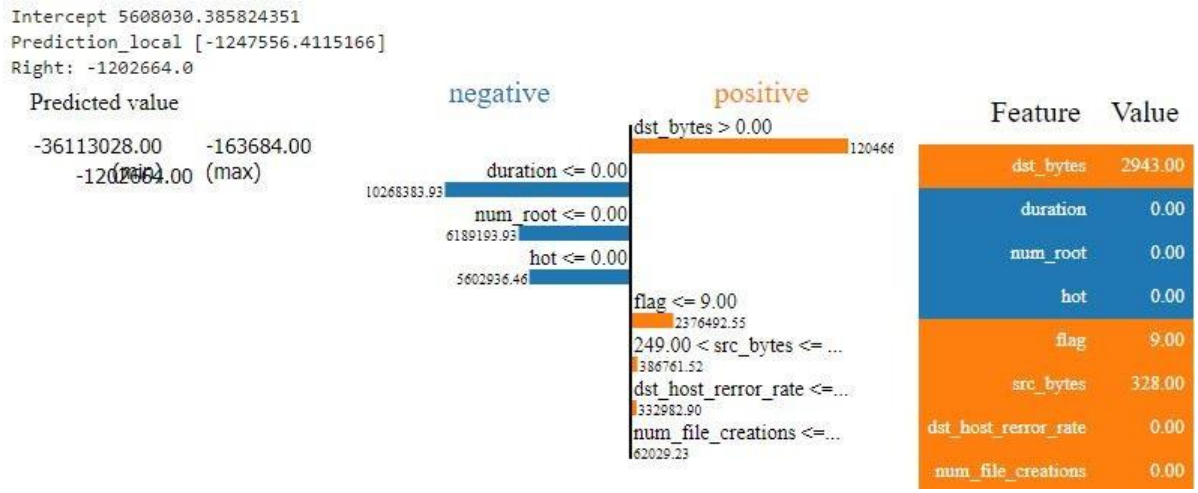


Figura 4.12: Explicação local do algoritmo LIME relativo ao método HSTrees do ataque *buffer overflow*

No entanto, no que diz respeito a comparação das explicações de ambos os métodos de deteção de anomalias só é encontrado um atributo em comum em todas as explicações que é *duration*, em que este se apresenta com o mesmo sentido de classificação em todas as explicações. Dito isto, pode-se afirmar que o atributo *duration*, é de grande importância na explicação das anomalias.

4.2.2 Comparação das explicações dos restantes ataques

Nesta secção vai ser realizada uma comparação entre as explicações fornecidas dos restantes ataques existentes. Para haver uma melhor visualidade das Tabelas 4.4 e 4.5, foi criada a Tabela A.1 com os atributos relevantes para a explicação dos ataques, em que têm o respetivo atributo e um nome atribuído ao atributo. O objetivo desta substituição é permitir que as dimensões das

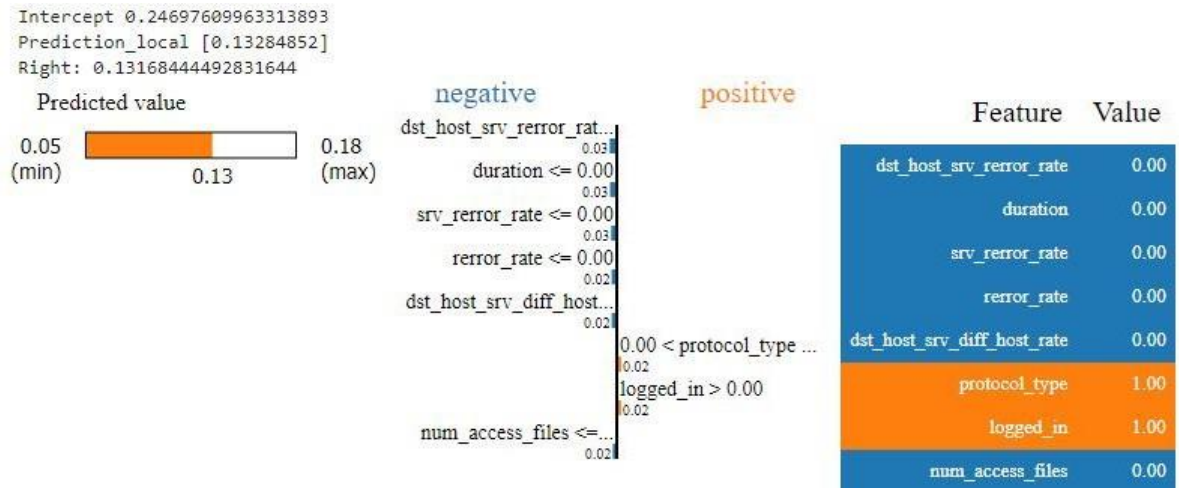


Figura 4.13: Explicação local do algoritmo LIME relativo ao método IForestASD do ataque *buffer overflow*

tabelas sejam reduzidas, permitindo desta forma uma visualização de qualidade dessas mesmas tabelas.

Como se pode ver na Tabela 4.4, os atributos utilizados para descrever os ataques estão mais distribuídos em comparação à Tabela 4.5, em que os atributos utilizados nesta, são mais coincidentes. Uma forma simples de comprovar esta simples comparação é através do número total de atributos referidos nas explicações, que são 24 e 11 para as Tabelas 4.4 e 4.5 respetivamente. Porém, comparativamente entre os modelos IForestASD e HSTrees existem atributos coincidentes, como se pode ver nas Tabelas 4.4 e 4.5, que são os seguintes: *dst_bytes, duration, srv_rerror_rate e num_failed_logins* , designados por A1, A2, A13 e A23 referentes a Tabela A.1.

Relativamente à Tabela 4.4, os atributos mais coincidentes nas explicações dos ataques do método HSTrees são: *dst_bytes, duration, num_root, hot e src_bytes*, designados por A1, A2, A3, A4 e A6 , referentes à Tabela A.1. Por outro lado, pode-se ver que na Tabela 4.5, os atributos menos utilizados nas explicações dos ataques do método IForestASD são: *num_failed_logins, diff_srv_rate e land* , designados por A23, A30 e A31, referentes à Tabela A.1.

4.2.3 Comparação das explicações dos ataques com recurso ao uso de seleção de atributos

Nesta secção vai ser realizada uma comparação entre as explicações, ao qual houve uma seleção prévia dos atributos antes do treinamento dos modelos de deteção de anomalias. Para além disso, também pode ser feita uma comparação com os resultados anteriormente obtidos na secção 4.2.2.

Como se pode ver nas Tabelas 4.6 e 4.7, estas apresentam um número de atributos coincidentes, em maior número comparativamente com os resultados obtidos nos testes anteriormente realizados. Em relação aos atributos que não coincidem nas explicações, esses atributos, são os atributos designados por A1, A4, A6, A10 e A15 referentes a Tabela A.1, ao qual os quatro primeiros

HSTrees ataque	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24
land (DOS)	+	-	-	-	-	+	-	+																
smurf (DOS)	-	-	-	-		+			+	-	+													
pod (DOS)	-	-	-	-								+	+	-	-									
teardrop (DOS)	-	-	-	-		-									+	+	+							
back (DOS)	+	-	-	+								+		+	-			+						
load module (U2R)	+	+	-	-								+			-				+	+				
perl (U2R)	+	+	-	-		-		+										-			+			
rootKit (U2R)	+	+	-	-		-		+										-	+					
spy (U2R)	+	+	-	-		-			-			+									+			
ftp write (R2L)	+	+	-	-		-										-					+	-		
guess password (R2L)	+	-	-	+		-											+						-	+
imap (R2L)	+	-	-	-		+							+							-	+			
phf (R2L)	-	-	-	-		-		+		-							+							
multihop (R2L)	+	+	-	+		-					-								+	-				
warezclient (R2L)	-	-	-	-		+						+	-						+					
warezmaster (R2L)	+	-	-	-			+	-				-								+				
satan (Probe)	-	-	-	-		-		+		-								+						
nmap (Probe)	-	-	-	-		-			+	-											+			
ipsweep (Probe)	-	-	-	-						-		-							+			+		
portsweep (Probe)	+	-	-	-		-							+					+		+				

Tabela 4.4: Explicações dos ataques relativos ao método HSTrees. Utilizando o *KDD99* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

IForestASD ataque	A1	A2	A13	A23	A25	A26	A27	A28	A29	A30	A31
back (DOS)	+	-	+		+	-	-	+	-		
teardrop (DOS)		-	-		-	-	+	-	-	-	
pod (DOS)	-	-	-		-	-	+	-	-		
smurf (DOS)	-	-	-		-	-	+	-	-		
land (DOS)		-	-		+	+	-	-	+		-
spy (U2R)	+	+	-		-	-	+	-	-		
rootKit (U2R)	+	+	-		-	-	+	-	-		
perl (U2R)		+	-		-	+	-	+	-	-	
load Module (U2R)		-	-		+	-	+	-	-	-	
warezmaster (R2L)	+	+	-		-	-	-	-	-		
warezclient (R2L)	-	-	-		-	-	-	+	-		
multihop (R2L)	+	+	-		-	-	+	+	-		
phf (R2L)	-	-	-		-	-	+	-	+		
imap (R2L)	+	-	-		-	-	+	+	-		
guess password (R2L)		-	+	+	-	-	-	-	+		
ftp write (R2L)	+	+	-		-	-	+	+	-		
satan (Probe)		+	-		-	-	+	-	-	+	
nmap (Probe)	-	-	-		-	-	+	-	-		
ipsweep (Probe)	-	-	-		+	+	-	-	-		
portsweep (Probe)	-	+	+		-	-	+	-	-		

Tabela 4.5: Explicações dos ataques relativos ao método IForestASD. Utilizando o *KDD99* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

dizem respeito às explicações da Tabela 4.6 e os restantes atributos dizem respeito às explicações da Tabela 4.7.

Comparativamente entre os resultados obtidos na secção 4.2.2 e os resultados obtidos nesta secção, existem alguns apontamentos que devem ser realizados. O primeiro apontamento é relativo ao número de atributos utilizados nas explicações relativas ao método HSTrees, ao qual apresentam uma diferença significativa de 24 para 15 atributos utilizados nas Tabelas 4.4 e 4.6 respetivamente. Relativamente às explicações do método IForestASD o número de atributos utilizados é constante. O segundo apontamento é o mais importante, pelo facto dos atributos utilizados nas explicações não serem constantes. Comparativamente entre as explicações que utilizam seleção de atributos e as explicações que não utilizam seleção de atributos, os únicos atributos que são constantes entre as Tabelas 4.4, 4.5, 4.6 e 4.7, são os atributos designados por A2 e A23 referentes à Tabela A.1.

Desta forma, chega-se a conclusão que a utilização de seleção de atributos faz com que os métodos forneçam explicações com graus de semelhança muito superiores no que toca à utilização

dos mesmos atributos para a explicação. Isto pode-se comprovar na visualização das Tabelas 4.6 e 4.7, comparativamente à visualização das Tabelas 4.4 e 4.5. Porém, as explicações são diferentes para o mesmo ataque ou para diferentes ataques, tanto na comparação de ataques referentes ao mesmo método de deteção de anomalia ou na comparação das explicações relativas a diferentes métodos de deteção de anomalias.

4.2.4 Comparação da utilização de atributos nas explicações

Nesta secção vai se discutir as diferenças e semelhanças da utilização de atributos nas explicações, relativamente à totalidade do conjunto de dados *KDD99* original.

Como se pode ver pelas Figuras 4.14 e 4.15, existem atributos que são coincidentes, mas, grande parte destes não são coincidentes.

Na comparação das Figuras 4.16, 4.17, 4.18 e 4.19, pode-se verificar que na comparação da distribuição dos atributos com classificação negativa e positiva nas explicações de ambos os métodos utilizados, é diferente. Porém, na comparação dos atributos utilizados nas explicações de cada método, pode-se afirmar com recurso às Figuras previamente mencionadas, que alguns dos atributos apresentam uma utilização semelhante como se pode ver na comparação das Figuras 4.16 e 4.18, e entre as Figuras 4.17 e 4.19.

Para haver uma melhor comparação entre as explicações de cada tipo de ataque, é utilizado *silhouette* e *kMeans*. A análise de *silhouette* [46], é utilizada para avaliar a qualidade de um agrupamento *clustering* ao medir quão bem cada ponto se agrupa com outros pontos no mesmo *cluster* em comparação com outros *clusters*, ajudando a determinar o número ideal de *clusters*. No caso do método *kMeans* [2, 12], este algoritmo de *clustering* tem como objetivo particionar um conjunto de dados em k *clusters*, minimizando a variância dentro de cada *cluster* e identificando os centroides que representam o centro de cada *cluster*. Estes dois métodos em conjunto, têm como objetivo dividir o conjunto de dados em *clusters* de acordo com a semelhança dos dados, permitindo desta forma haver uma comparação mais realista.

A combinação da análise de *silhouette* e do algoritmo *kMeans* é uma abordagem eficaz para descobrir os centroides dos *clusters* e avaliar a qualidade do agrupamento num conjunto de dados. Com a utilização destas ferramentas obteve-se as seguintes Figuras: 4.20,4.21, 4.22,4.23, 4.24, 4.25, 4.26,4.27, 4.28,4.29, 4.30 e 4.31.

Outros dados que podem ser obtidos são a contribuição relativa dos atributos relativos para descrever cada *cluster*. Com esta informação é possível obter vários tipos de informação, como:

- Identificação dos atributos mais importantes;
- Caracterização dos *clusters* - É possível caracterizar e descrever os *clusters* de forma detalhada, como acontece em casos em que os *clusters* têm altos valores em determinadas variáveis e baixos em outras;

- Diferenciação entre *clusters*;
- Informações para ações e decisões - Com base nos atributos mais utilizados, é possível executar decisões informadas.

Desta forma conclui-se que também é importante haver uma descrição dos atributos utilizados nos *clusters*, para haver uma melhor visualização, foi criada a Figura A.1 com os atributos relevantes para a explicação dos ataques, e com isso obteve-se as seguintes Figuras: 4.32, 4.33, 4.34, 4.35, 4.36 e 4.37.

Nas figuras anteriormente mencionadas, algumas apresentam semelhanças no que diz respeito ao numero de *clusters*, porem as contribuições dos atributos nas explicações são diferentes para métodos diferentes, HSTrees e IForestASD, e também são diferentes relativamente a ataques diferentes.

Com base nas Figuras anteriormente mencionadas, é possível mostrar que para os diferentes métodos, HSTrees e IForestASD, os centroides e *clusters* são diferentes, na distribuição e utilização de atributos. Para além disso, nas Figuras utilizadas para mostrar a distribuição dos atributos com classificação negativa e positiva das explicações, mostra-se a diferença entre os métodos. Com estes dados chega-se a conclusão que dependendo do método utilizado as explicações variam consideravelmente entre métodos diferentes, independentemente de os tipos de ataques serem iguais.

4.3 Resultados acerca da explicação local relativamente ao con- junto de dados *ICS-Flow*

Nesta secção vai ser realizada a comparação entre explicações com recurso ao uso de seleção de atributos e explicações sem utilização de seleção de atributos como aconteceu anteriormente nas secções 4.2.1, 4.2.2, 4.2.3 e 4.2.4.

4.3.1 Comparação da explicação entre dois ataques diferentes

Nesta secção vai ser realizada a comparação entre dois ataques distintos, sendo esses os seguintes: *DOS* e *IP scan*. A comparação vai se centrar no que as explicações têm de comum entre os diferentes modelos e quais são as diferenças entre as explicações do mesmo modelo, com o sentido de perceber quais são os atributos que são mais importantes para a categorização entre dois tipos de ataques diferentes.

Comparativamente entre os modelos IForestASD e HSTrees, no ataque *DOS* existem dois atributos que são comuns em ambas as explicações, mas, um destes atributos não apresenta o mesmo sentido de classificação, como se pode ver nas Figuras 4.38 e 4.39, sendo esses atributos os seguintes: *sAddress e rAddress*.

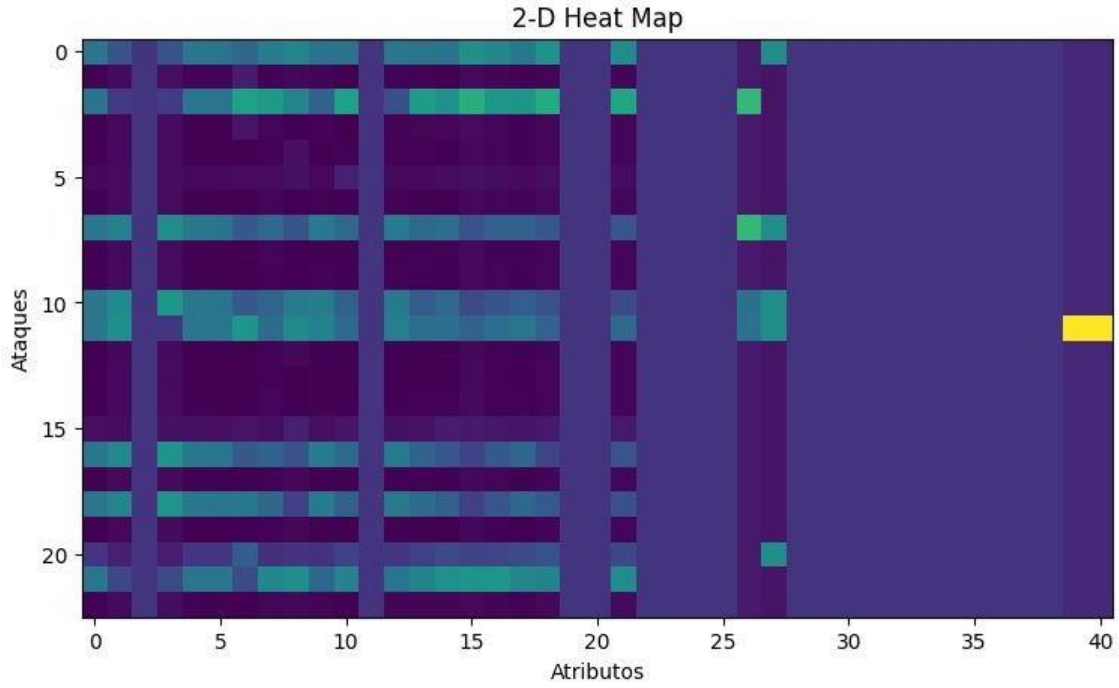


Figura 4.14: Mapa de calor do método HSTrees, relativo à utilização dos atributos, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor as cores mais vivas, (cores claras), significam que aquele atributo é muito utilizado enquanto as cores mais frias, (cores escuras), é o inverso. Os atributos que não são utilizados apresentam a mesma cor

Comparativamente entre os modelos IForestASD e HSTrees, no ataque *IP scan* não existem atributos em comum nas explicações, como se pode ver nas Figuras 4.40 e 4.41.

No que toca à comparação das explicações dos diferentes ataques, do modelo HSTrees pode-se concluir que ambas as explicações têm pontos em comum como *rAddress, sPackets, rPackets, rFinRate e sAddress*, que apresentam contribuições no mesmo sentido. Porém, existe mais um atributo, coincidente, que está presente em ambas as explicações que é o *sRstRate*, que apresenta contribuições no sentido oposto nas respetivas explicações.

No que toca à comparação das explicações dos diferentes ataques, do modelo IForestASD só existe um atributo em comum e que apresenta o mesmo sentido de classificação em ambas as explicações, sendo esse atributo o seguinte: *protocol*. Os restantes atributos não coincidem.

No entanto, no que diz respeito à comparação das explicações de ambos os métodos de deteção de anomalias não é encontrado um atributo que seja comum em todas as explicações.

4.3.2 Comparação das explicações dos restantes ataques

Para haver uma melhor visualidade das Tabelas 4.8, 4.9, 4.10 e 4.11, foi criada a Tabela B.1 com os atributos relevantes para a explicação dos ataques, em que têm o respetivo atributo e

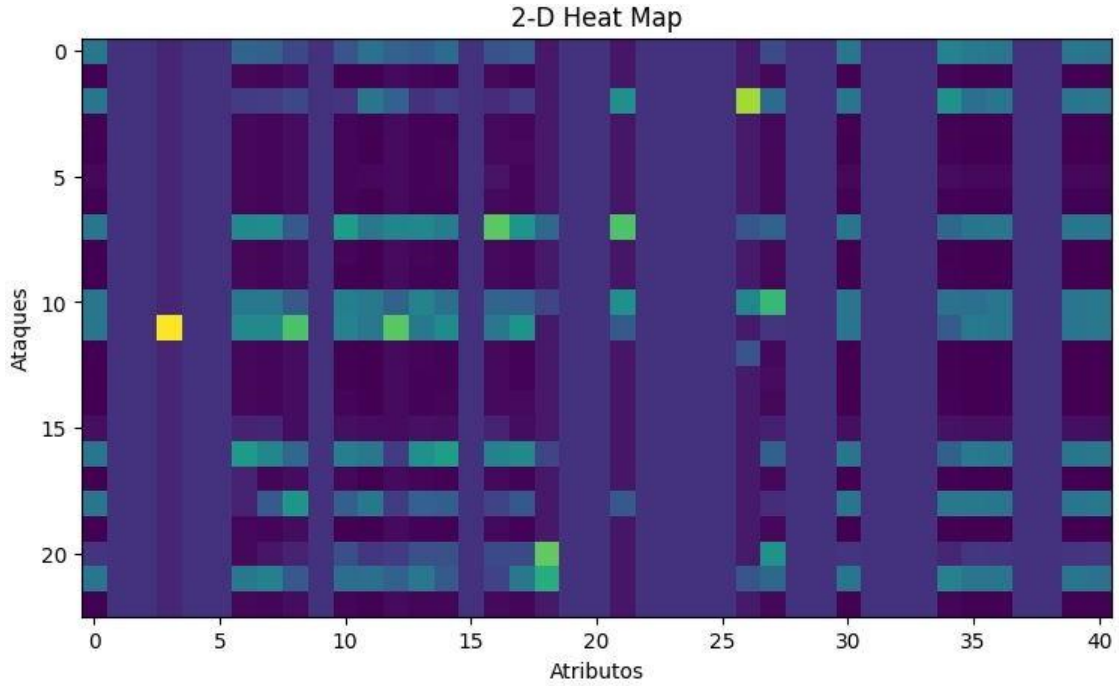


Figura 4.15: Mapa de calor do método IForestASD, relativo à utilização dos atributos, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor as cores mais vivas, (cores claras), significam que aquele atributo é muito utilizado enquanto as cores mais frias, (cores escuras), é o inverso. Os atributos que não são utilizados apresentam a mesma cor

um nome atribuído ao atributo. O objetivo desta substituição é permitir que as dimensões das Tabelas sejam reduzidas, permitindo desta forma uma visualização de qualidade dessas mesmas Tabelas.

Como se pode ver na Tabela 4.9, os atributos utilizados para descrever os ataques estão mais distribuídos em comparação à Tabela 4.8, em que os atributos utilizados nesta, são mais coincidentes. Comparativamente entre as Tabelas 4.8 e 4.9 o número de atributos é semelhante, 14 e 15 respetivamente, porém, apenas os atributos definidos por B1, B7 e B12 referentes à Tabela B.1, são coincidentes.

Relativamente à Tabela 4.8, os atributos mais coincidentes nas explicações dos ataques do método HSTrees são os atributos definidos por B1, B2, B3, B4 e B5 referentes à Tabela B.1. Por outro lado, pode-se ver que na Tabela 4.9, os atributos menos utilizados nas explicações dos ataques do método IForestASD são os atributos definidos por B20, B21, B22, B23, B24, B25 e B26 referentes à Tabela B.1.

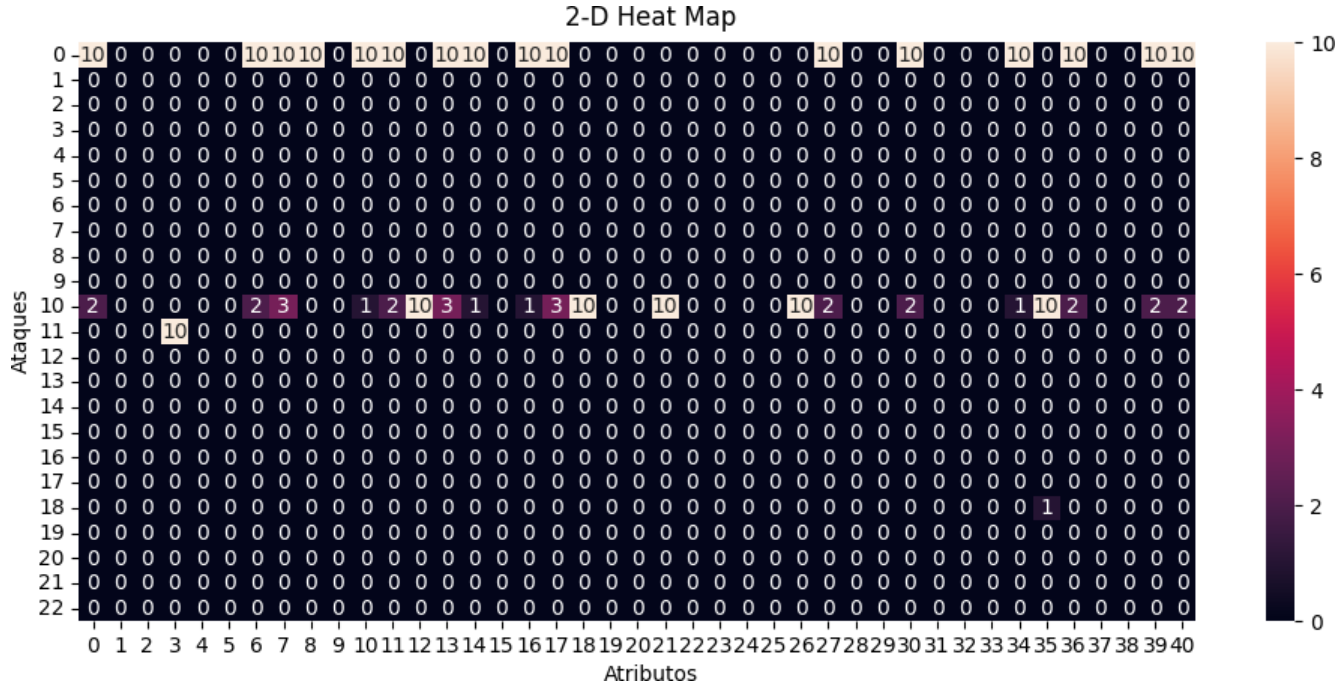


Figura 4.16: Mapa de calor do método IForestASD, relativo à utilização dos atributos com classificação negativa, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

4.3.3 Comparação das explicações dos ataques com recurso ao uso de seleção de atributos

Nesta secção vai ser realizada uma comparação entre as explicações, ao qual houve uma seleção prévia dos atributos antes do treinamento dos modelos de deteção de anomalias. Para além disso, também pode ser feita uma comparação com os resultados anteriormente obtidos na secção 4.3.2.

Comparativamente entre as Tabelas 4.10 e 4.11 os atributos utilizados para as explicações coincidem perfeitamente em ambos os casos. Porém, as explicações fornecidas são diferentes para os mesmos tipos de ataques.

Comparativamente entre os resultados obtidos na secção 4.3.2 e os resultados obtidos nesta secção, existem alguns apontamentos que devem ser realizados. O primeiro apontamento é relativo ao número de atributos utilizados nas explicações relativas aos métodos HSTrees e IForestASD, ao qual apresentam uma diferença significativa de 14 e 15 respetivamente, para 11 atributos utilizados nas Tabelas 4.4 e 4.6. O segundo apontamento é o mais importante, pelo facto dos atributos utilizados nas explicações não serem constantes. Comparativamente entre as explicações que utilizam seleção de atributos e as explicações que não utilizam seleção de atributos, pode-se dizer que das Tabelas 4.8, 4.9, 4.10 e 4.11 apenas 3, (três), atributos designados por B1, B7 e B12 referentes a Tabela B.1, são coincidentes.

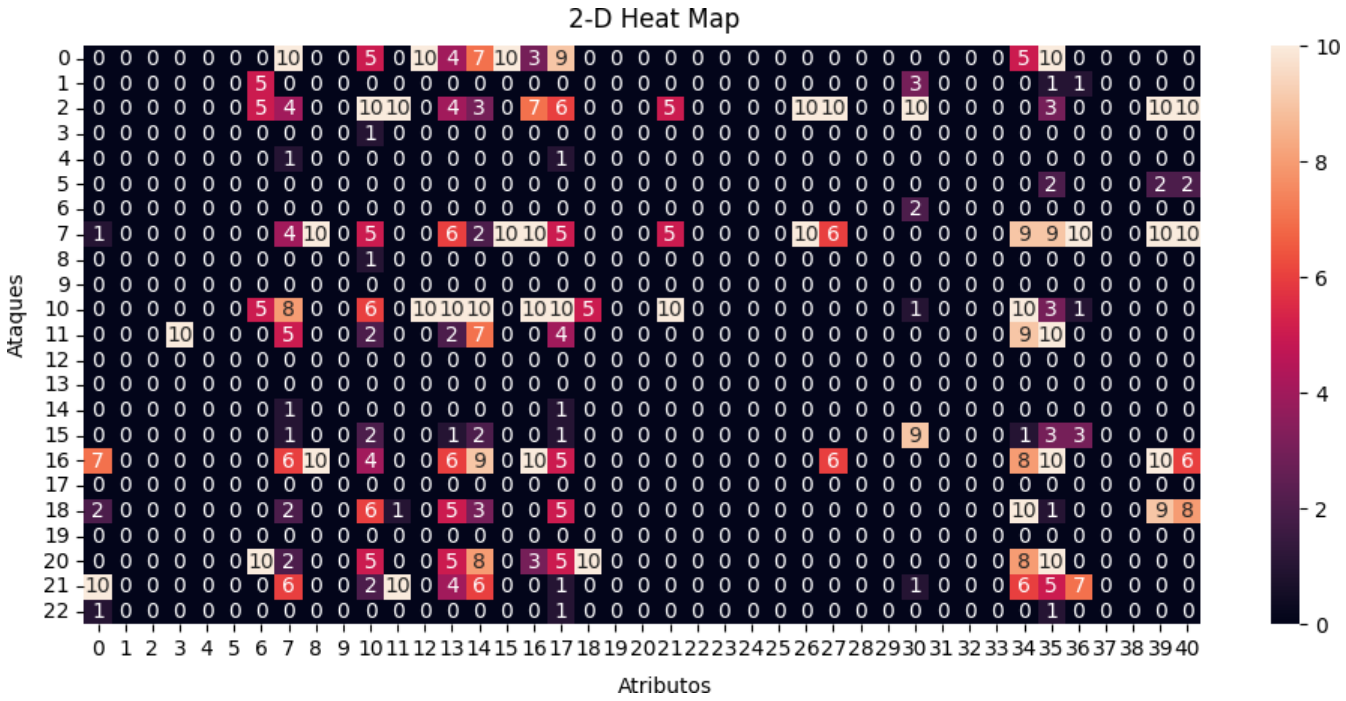


Figura 4.17: Mapa de calor do método IForestASD, relativo à utilização dos atributos com classificação positiva, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

Desta forma, é possível chegar à conclusão que a utilização de seleção de atributos faz com que os métodos forneçam explicações com graus de semelhança muito superiores no que toca a utilização dos mesmos atributos para a explicação. Isto, pode-se comprovar na visualização das Tabelas 4.10 e 4.11, comparativamente à visualização das Tabelas 4.8 e 4.9. Porém, as explicações são diferentes para o mesmo ataque ou para diferentes ataques, tanto na comparação de ataques referentes ao mesmo método de deteção de anomalia ou na comparação das explicações relativas a diferentes métodos de deteção de anomalias.

4.3.4 Comparação da utilização de atributos nas explicações

Nesta secção vai se discutir as diferenças e semelhanças da utilização de atributos nas explicações, relativamente a totalidade do conjunto de dados *ICS-Flow* original.

Como se pode ver pelas Figuras 4.42 e 4.43, existem atributos que são coincidentes mas, grande parte destes não são coincidentes.

Na comparação das Figuras 4.44, 4.45, 4.46 e 4.47, pode-se verificar que na comparação da distribuição dos atributos com classificação negativa e positiva nas explicações de ambos os métodos utilizados é diferente. Porém, na comparação dos atributos utilizados nas explicações de

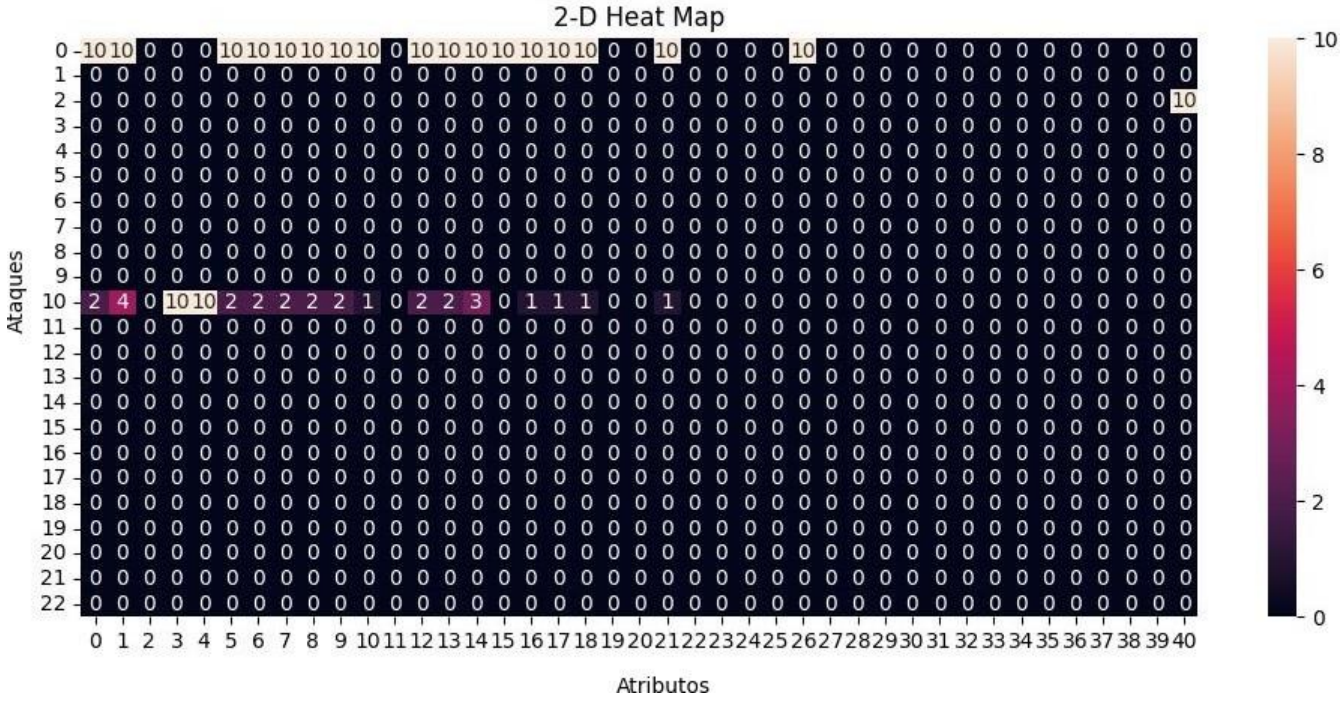


Figura 4.18: Mapa de calor do método HSTrees, relativo à utilização dos atributos com classificação negativa, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

cada método pode-se afirmar com recurso às Figuras previamente mencionadas, que os atributos utilizados apresentam uma distribuição semelhante como se pode ver na comparação das Figuras 4.44 e 4.46, e entre as Figuras 4.45 e 4.47.

Como foi referido anteriormente os testes realizados nesta secção são os mesmos realizados na secção 4.2.4, mas para o conjunto de dados *ICS-Flow*. Com a realização desses testes foram obtidas as seguintes Figuras: 4.48, 4.49,4.50,4.51, 4.52, 4.53,4.54,4.55.

Relativamente a descrição dos atributos utilizados nos *clusters*, para haver uma melhor visualização, foi criada a Figura B.1 com os atributos relevantes para a explicação dos ataques, e com isso obteve-se as seguintes Figuras: 4.56, 4.57, 4.58 e 4.59.

Nas figuras anteriormente mencionadas, elas apresentam uma semelhança que é o numero de *clusters*, porem as contribuições dos atributos nas explicações são diferentes para métodos diferentes, HSTrees e IForestASD, e também são diferentes relativamente a ataques diferentes.

Com base nas Figuras anteriormente mencionadas, é possível mostrar que para os diferentes métodos, HSTrees e IForestASD, os centroides e *clusters* são diferentes, na distribuição e utilização de atributos. Para além disso, nas Figuras utilizadas para mostrar a distribuição dos atributos com classificação negativa e positiva das explicações, mostra-se a diferença entre os métodos.

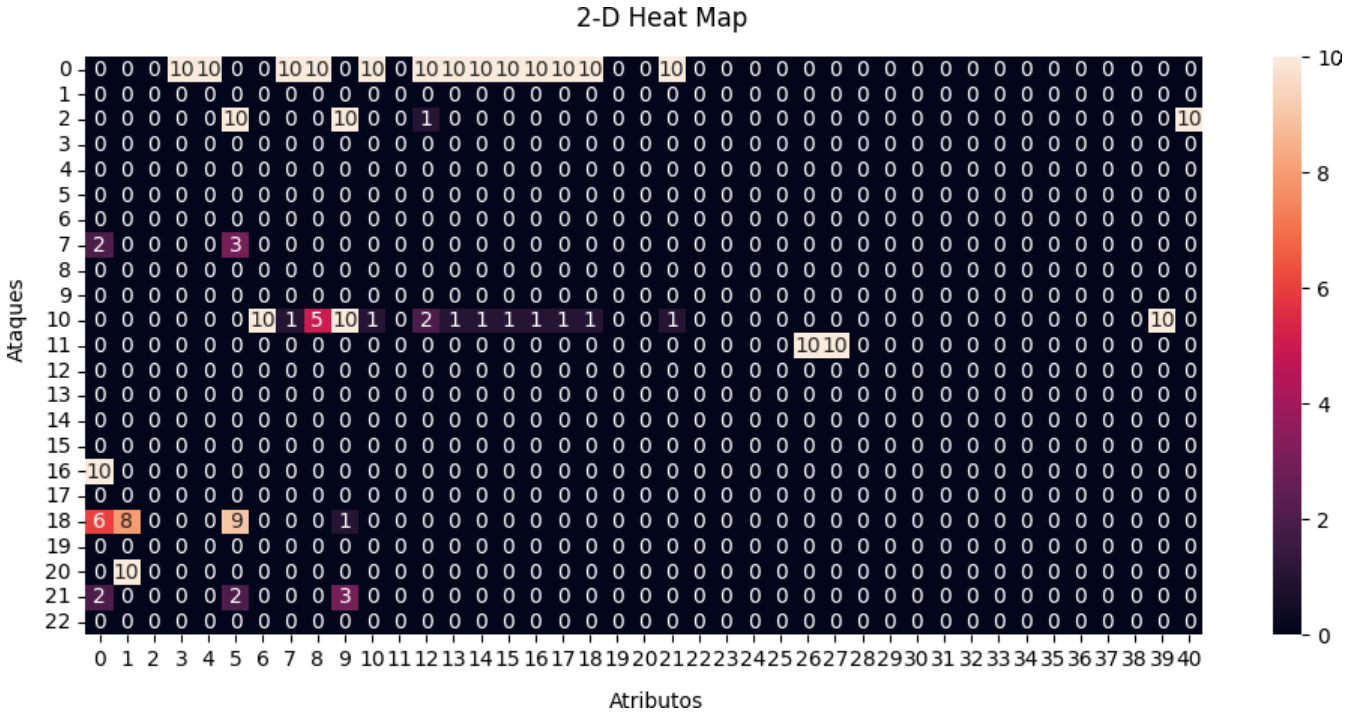


Figura 4.19: Mapa de calor do método HSTrees, relativo à utilização dos atributos com classificação positiva, na explicação dos ataques envolventes na totalidade do *KDD99* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

Com estes dados chega-se a conclusão que dependendo do método utilizado as explicações variam consideravelmente entre métodos diferentes, independentemente de os tipos de ataques serem iguais.

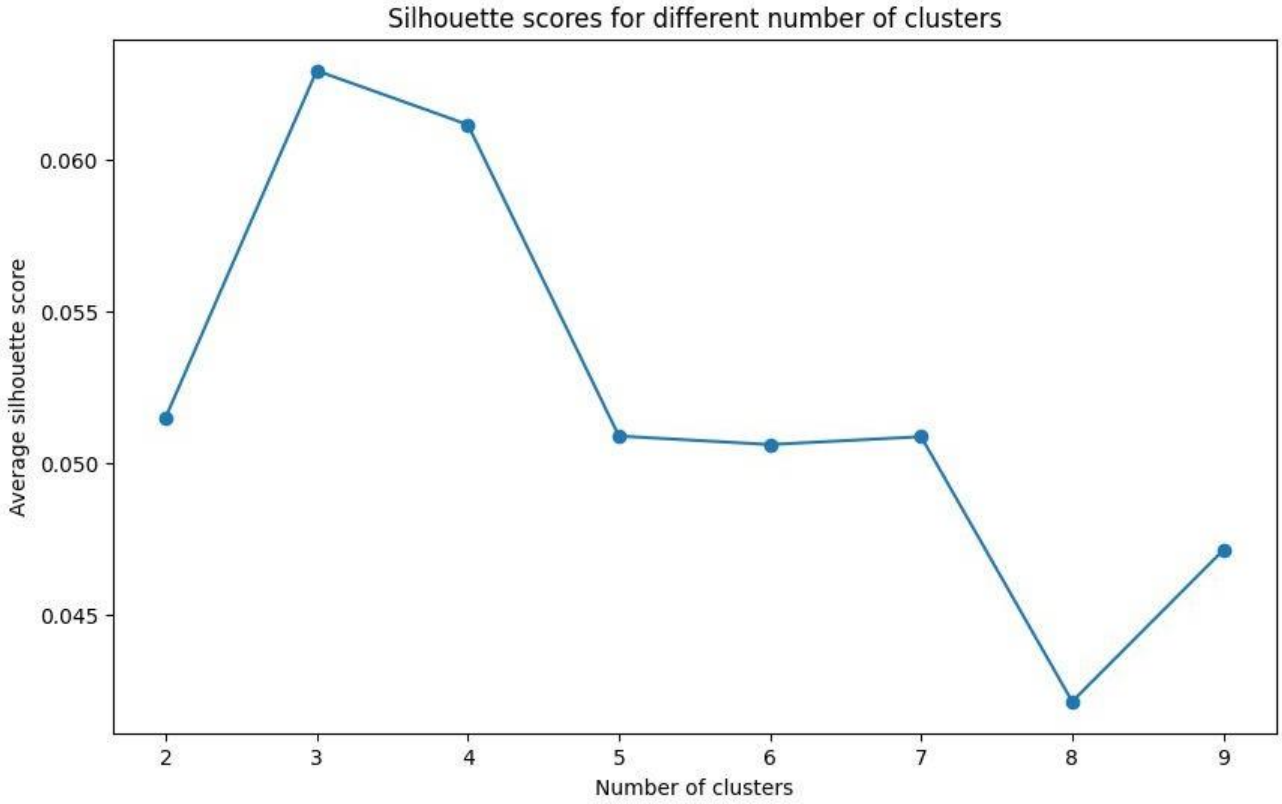


Figura 4.20: Silhouette do tipo de ataque *Back* do método HSTrees. Teste relativo à utilização do conjunto de dados *KDD99* original.

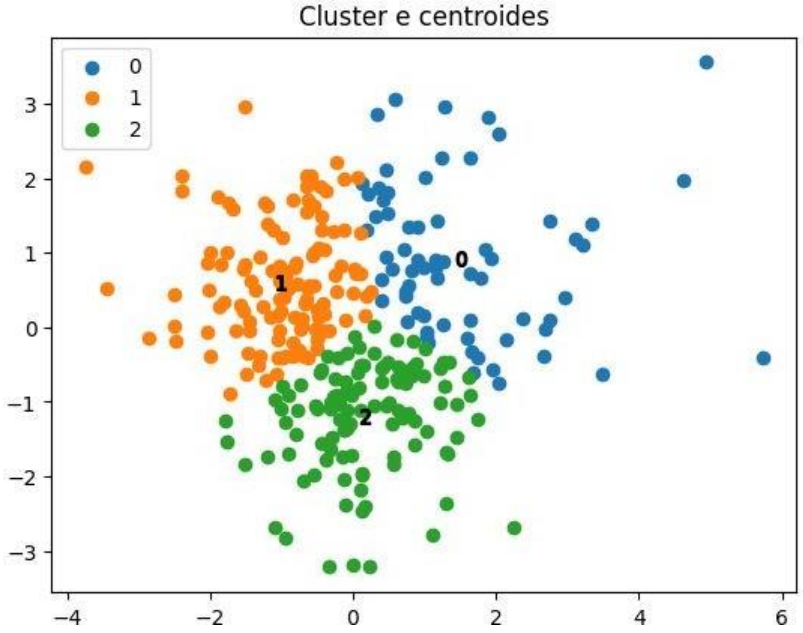


Figura 4.21: *Cluster* e centroides do tipo de ataque *Back* do método HSTrees. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

HSTrees ataque	A1	A2	A4	A6	A8	A10	A17	A18	A22	A23	A24	A28	A31	A32	A33
land (DOS)	-	-	-	-	+	-					+				-
neptune (DOS)	-	+	-	+				-			+			-	-
smurf (DOS)	+	+	-	+				-		-				-	+
pod (DOS)	-		-	-				+	+			+		-	+
teardrop (DOS)	-	+	-	-	+						+			+	+
back (DOS)	-	+	-	-	+					+				-	-
bufferOverflow (U2R)	-	-	-	-	+					-				-	-
load module (U2R)	-	-	+	-			+			+				+	-
perl (U2R)	-	+	-	-	+				-		+				-
rootKit (U2R)	-	-	-	-						+		+		+	+
spy (U2R)	-	-	+	-				+	+					+	+
ftp write (R2L)	-	+	-	-			+	+				-			-
guess pasword (R2L)	-		-	-	+			+		+	+				-
imap (R2L)	-	+	+	-			+				+	+			-
phf (R2L)	-	+	-	-	+		+			-					+
multihop (R2L)	-	+	+	-				-		+	+				-
warezclient (R2L)	-	-	+	-				+	+					-	-
warezmaster (R2L)	-	+	-	-			+				-			+	+
satan (Probe)	-	+	-	-					+			+		+	+
nmap (Probe)	-	+	-	-					+			-	+		-
ipsweep (Probe)	-	+	-	-				+					-	+	-
portsweep (Probe)	-	+	-	-								+	+	+	+

Tabela 4.6: Explicações dos ataques relativos ao método HSTrees com recurso a seleção de atributos. Utilizando o *KDD99* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

IForestASD ataque	A2	A8	A15	A17	A18	A22	A23	A24	A28	A31	A32
land	+	-	-		+	+		+	+		-
neptune (DOS)	+	-			+	+		+	+	+	+
smurf (DOS)	-	-			-	+		-	+	-	+
pod (DOS)	-	-		+	-	+		-	+	-	
teardrop (DOS)	+	-			-	+		-	+	-	+
back (DOS)	+	+			-	+	+	-	+	-	
load module (U2R)	+	+			-	+	+	-	+	-	
perl (U2R)	+	+			-	+	+	-	+	-	
rootKit (U2R)	+	+		+	-	+		-	-	-	
spy (U2R)	+	+		+	+	+		-	-	-	
bufferOverflow (U2R)	+	+			-	+	+	-	+	-	
ftp write (R2L)	-	-		+	-	+		-	+	-	
guess pasword (R2L)	+	-			+	+	+	+	+	-	
imap (R2L)	+	+	+		-	+		-	-		-
phf (R2L)	+	+		+	-	+		+	-	-	
multihop (R2L)	+	+		+	-	+		-	-	-	
warezclient (R2L)	+	+		+	-	+		-	+	-	
warezmaster (R2L)	+	-			-	+	+	-	+	-	
satan (Probe)	+	-			-	+		-	-	-	-
nmap (Probe)	-	-			-	+		-	+	-	+
ipsweep (Probe)	-	-			-	+		-	+	-	-
portsweep (Probe)	+	+	+		-	+		+	+	-	

Tabela 4.7: Explicações dos ataques relativos ao método IForestASD com recurso a seleção de atributos. Utilizando o *KDD99* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

HST	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14
IP scan	-	+	-	-	+	-	+	+						
MITM	-	+		-	-	+			-	-	+			
Port scan	+	+	+	+	-							-	-	+
Replay	+	-	+	+	+	+				-			+	
DDOS	-	+	-	-	-				+				+	

Tabela 4.8: Explicações dos ataques do conjunto de dados ICSFlow relativos ao método HSTrees. Utilizando o *ICS-Flow* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

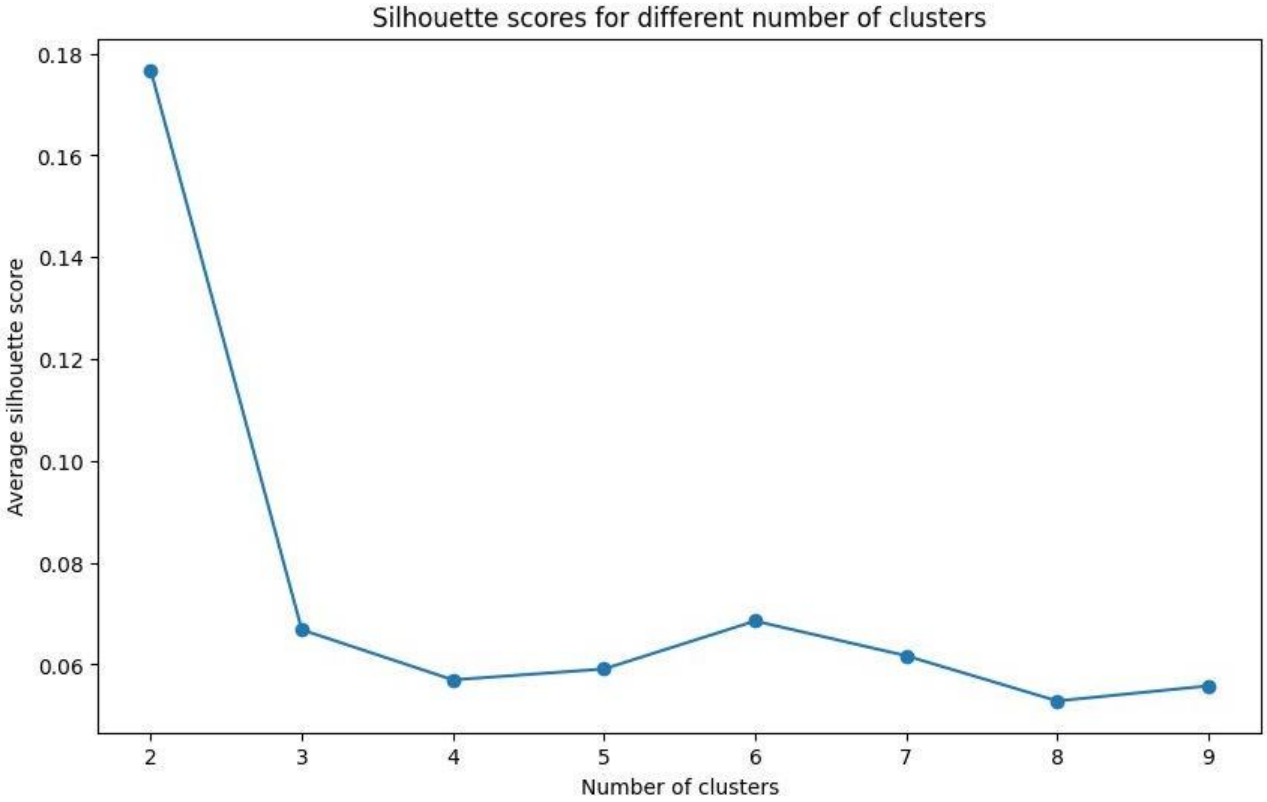


Figura 4.22: Silhouette do tipo de ataque *Smurf* do método HSTrees. Teste relativo à utilização do conjunto de dados *KDD99* original.

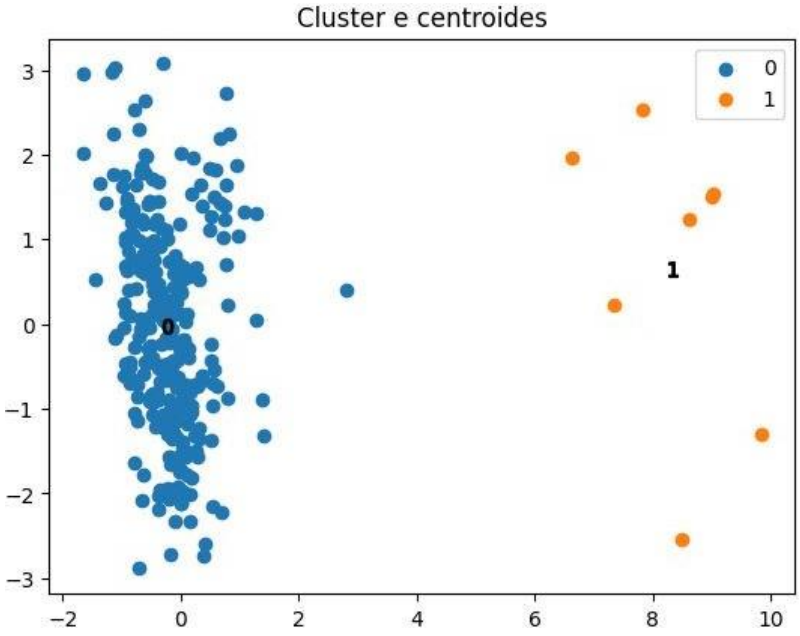


Figura 4.23: *Cluster* e centroides do tipo de ataque *Smurf* do método HSTrees. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

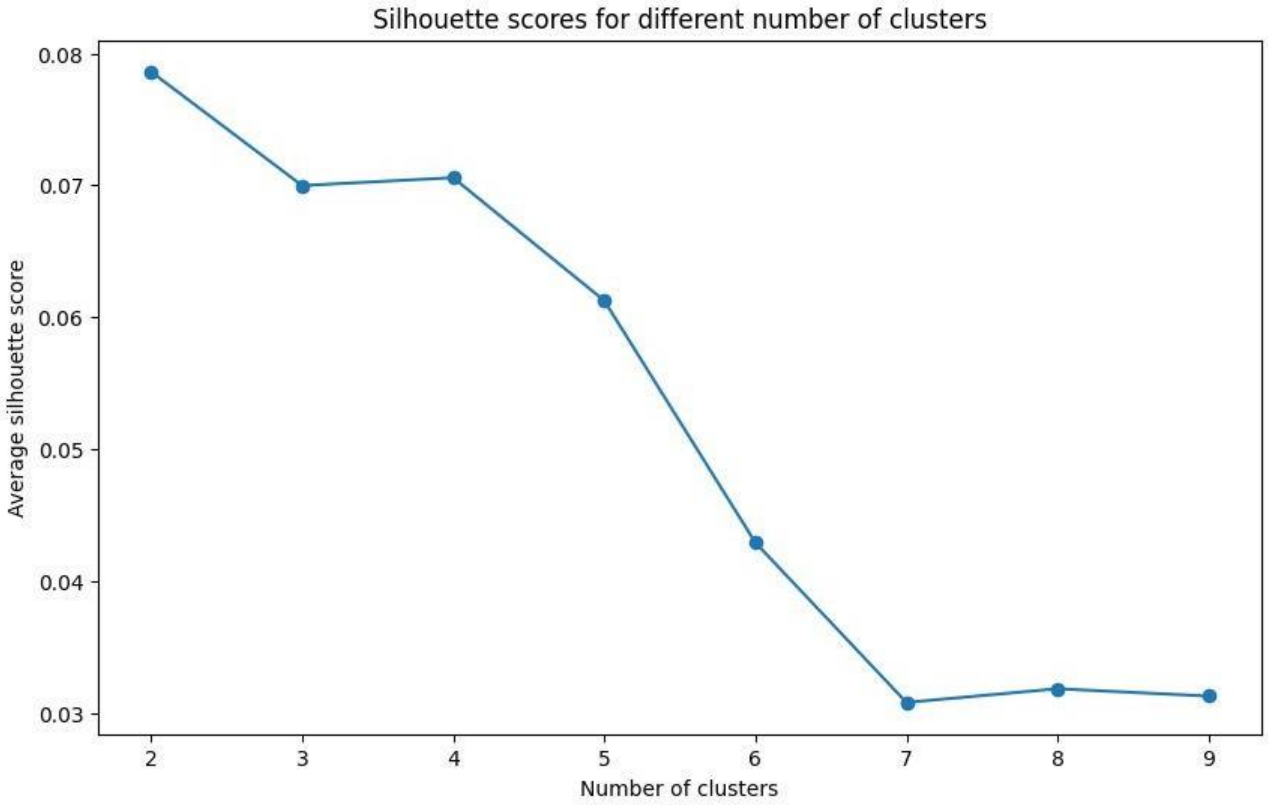


Figura 4.24: Silhouette do tipo de ataque *WarezClient* do método HSTrees. Teste relativo à utilização do conjunto de dados *KDD99* original.

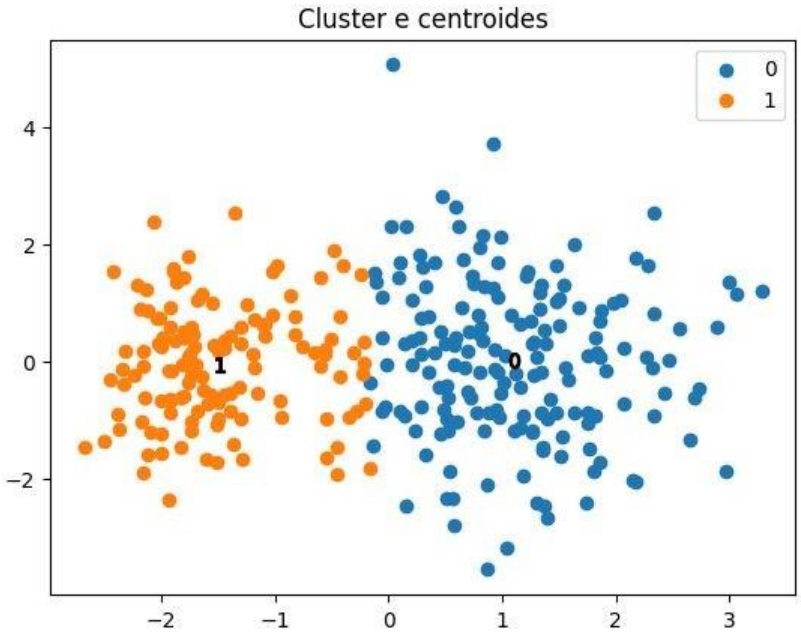


Figura 4.25: *Cluster* e centroides do tipo de ataque *WarezClient* do método HSTrees. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

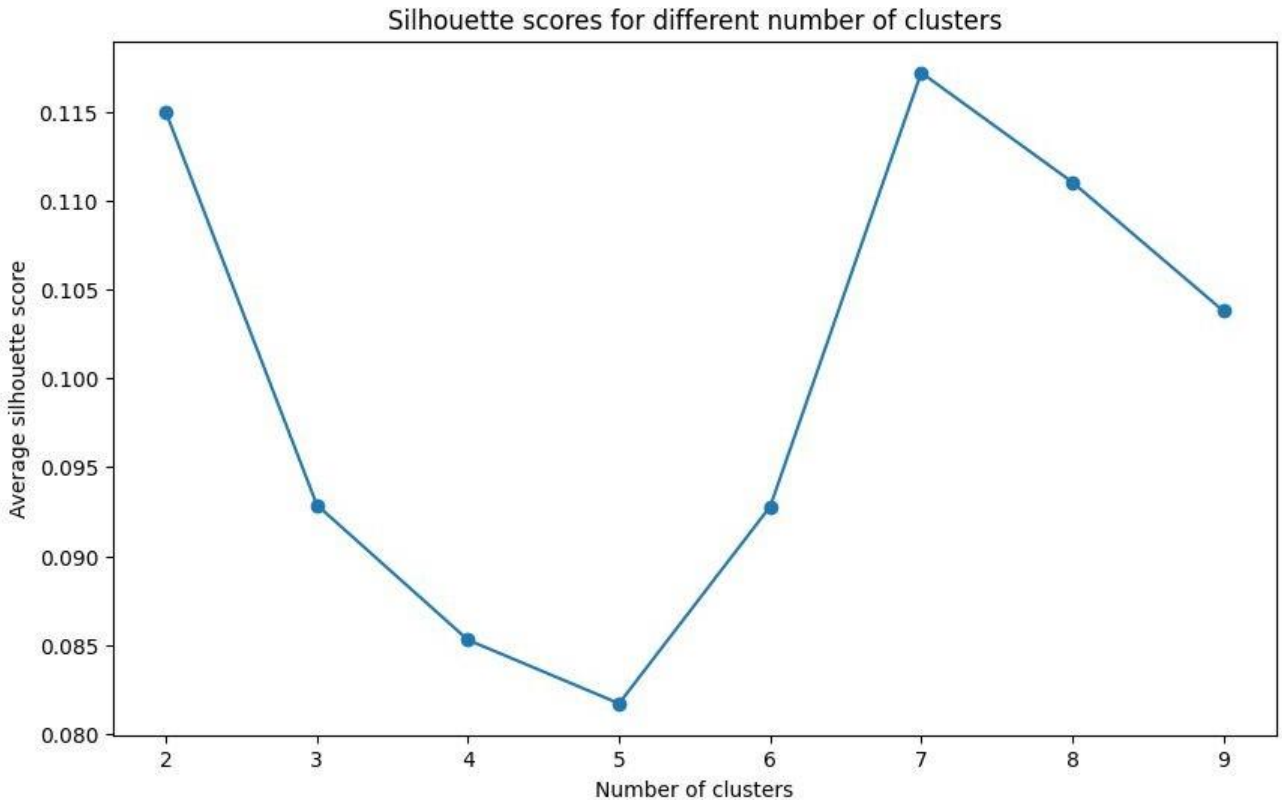


Figura 4.26: Silhouette do tipo de ataque *Back* do método IForestASD. Teste relativo à utilização do conjunto de dados *KDD99* original.

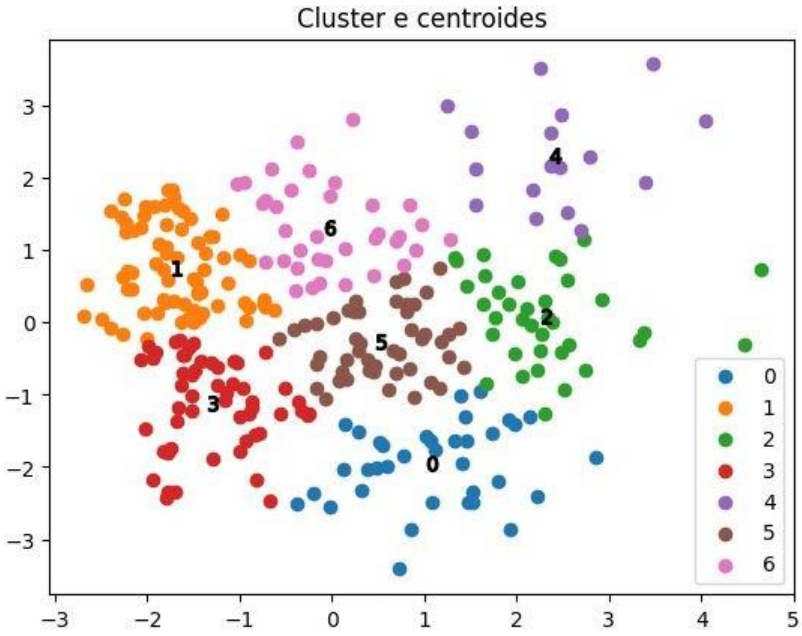


Figura 4.27: *Cluster* e centroides do tipo de ataque *Back* do método IForestASD. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

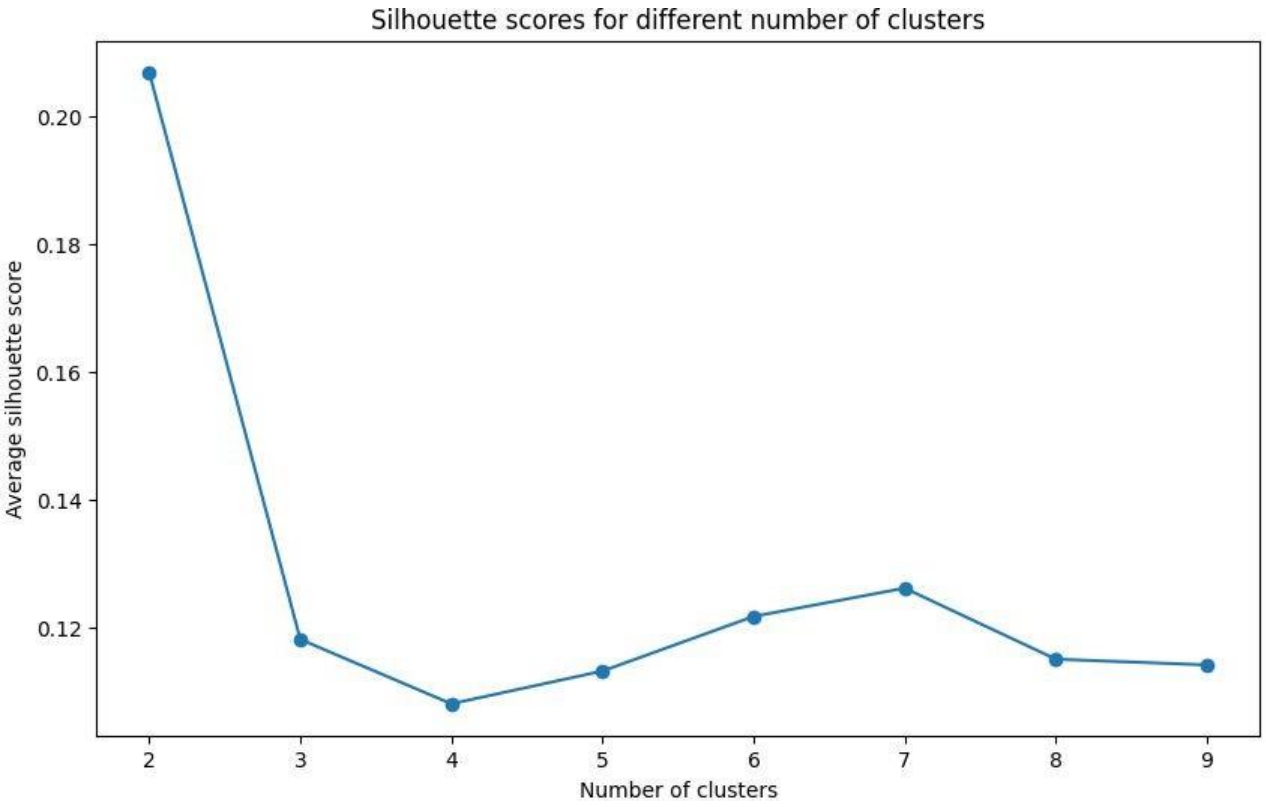


Figura 4.28: Silhouette do tipo de ataque *Smurf* do método IForestASD. Teste relativo à utilização do conjunto de dados *KDD99* original.

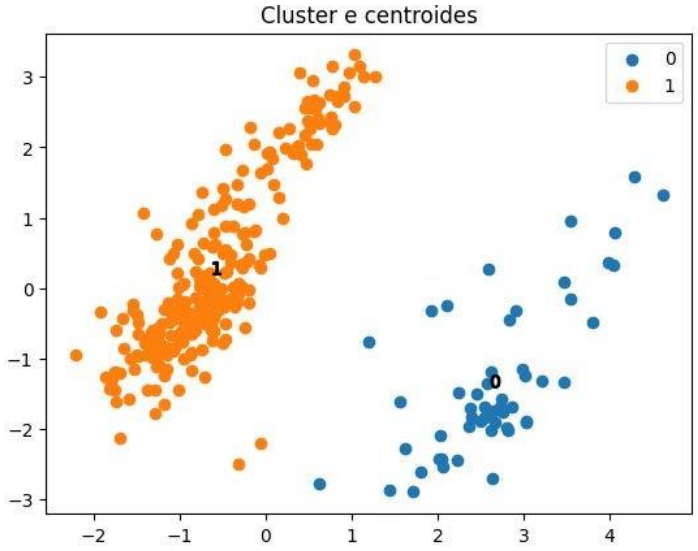


Figura 4.29: *Cluster* e centroides do tipo de ataque *Smurf* do método IForestASD. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

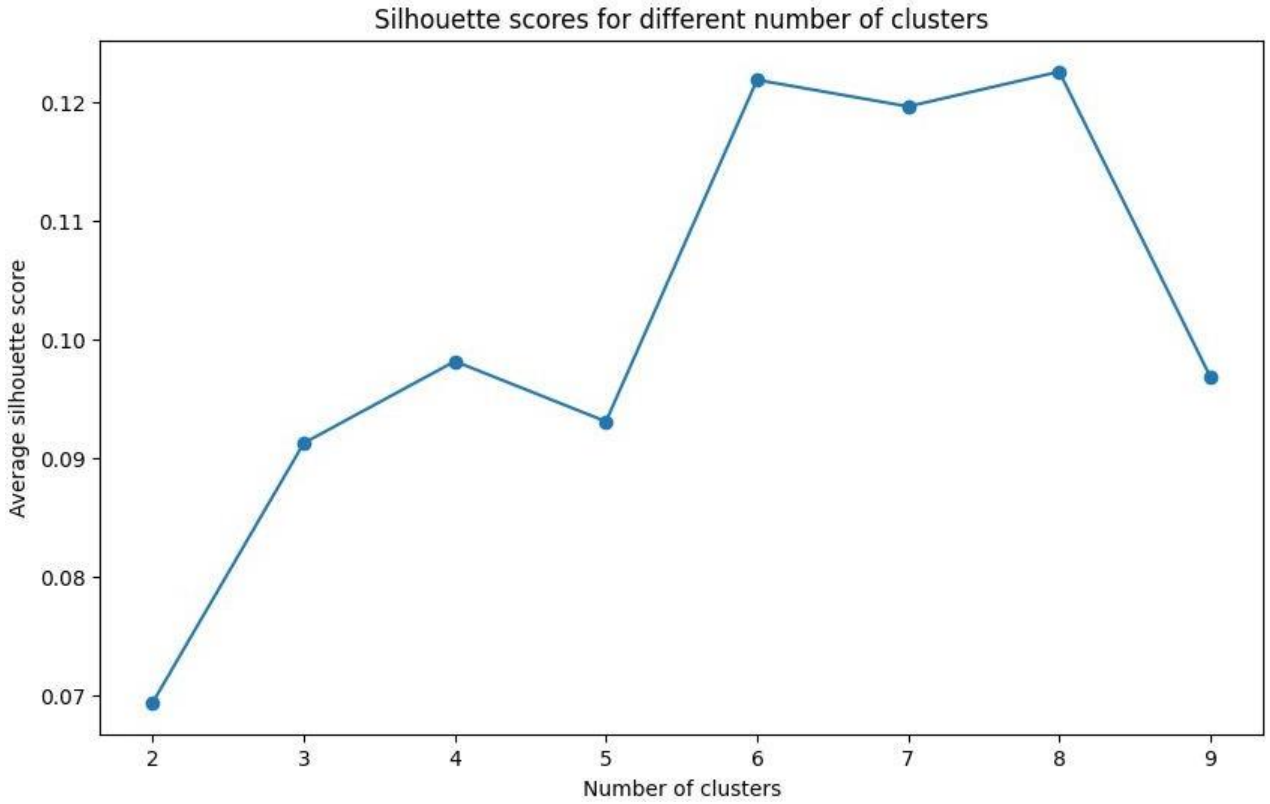


Figura 4.30: Silhouette do tipo de ataque *WarezClient* do método IForestASD. Teste relativo à utilização do conjunto de dados *KDD99* original.

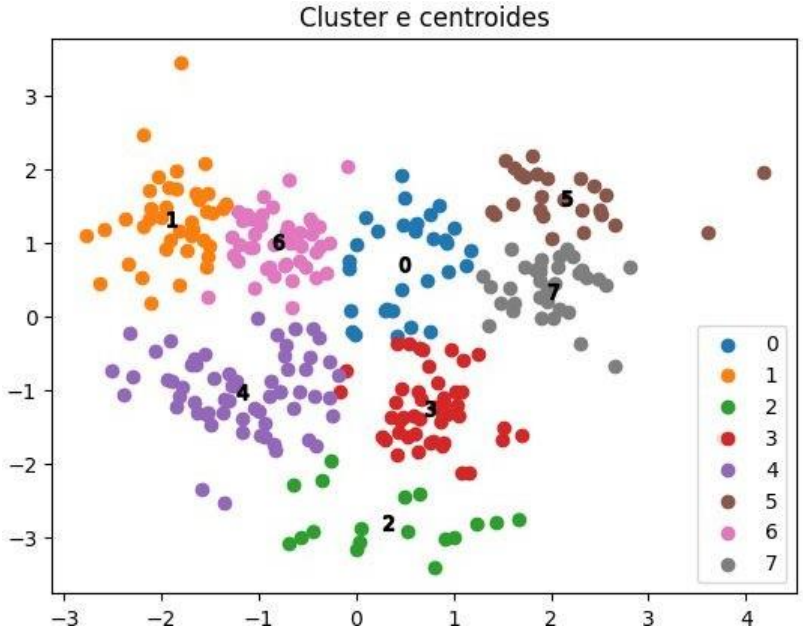


Figura 4.31: *Cluster* e centroides do tipo de ataque *WarezClient* do método IForestASD. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *KDD99* original.

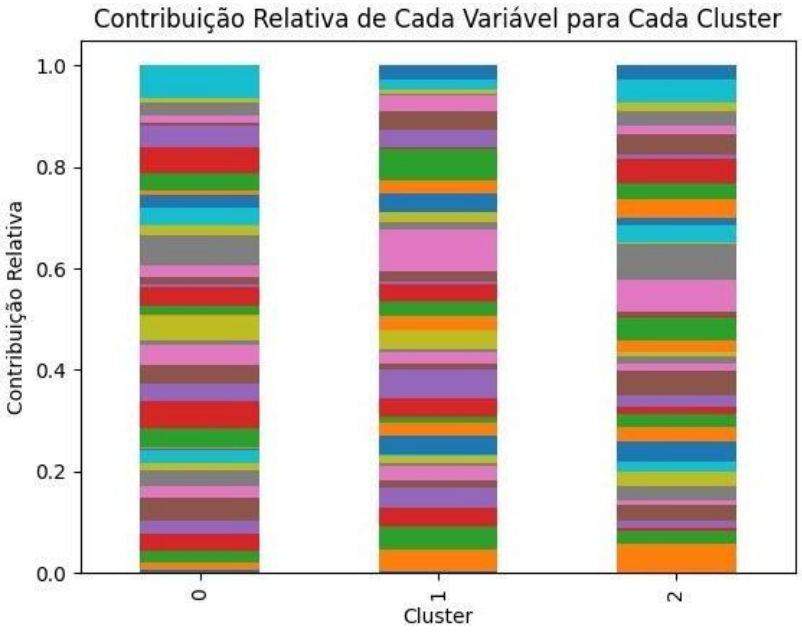


Figura 4.32: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método HSTrees.

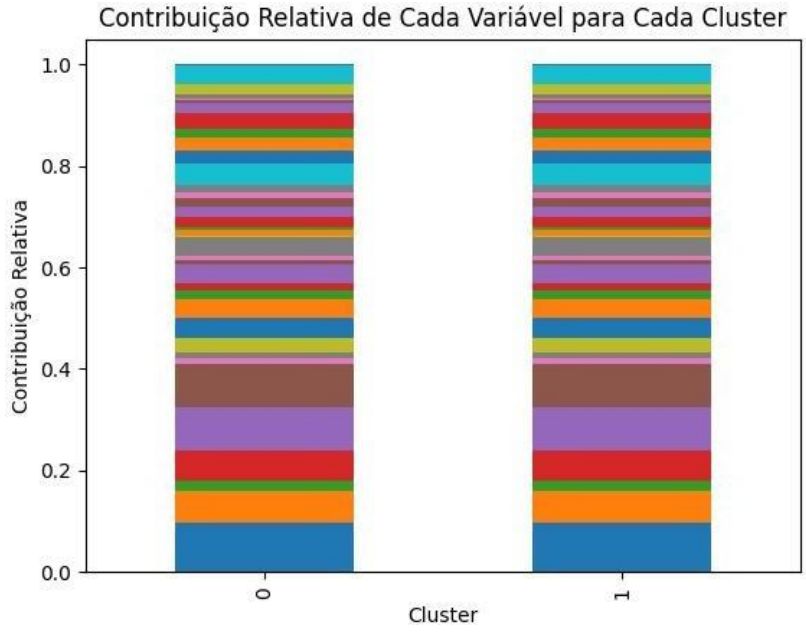


Figura 4.33: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método HSTrees.

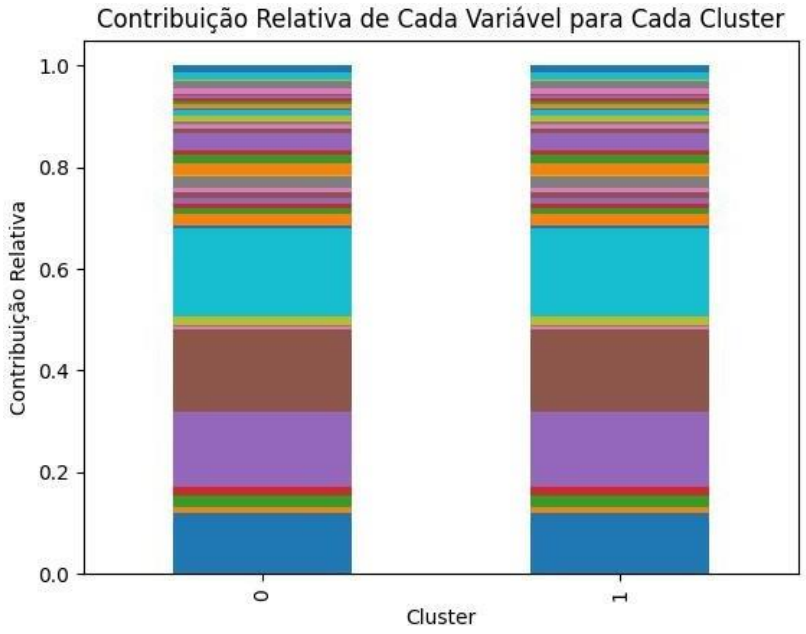


Figura 4.34: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método HSTrees.

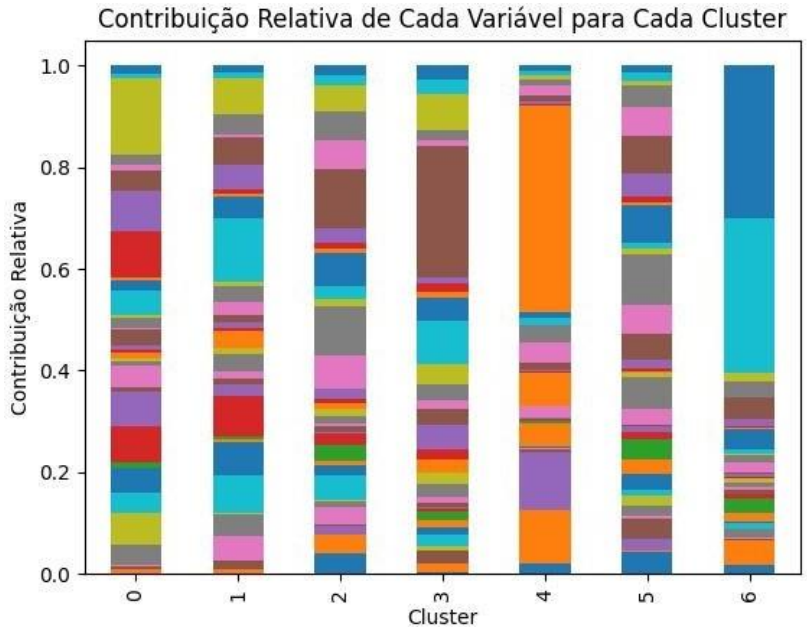


Figura 4.35: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método IForestASD.

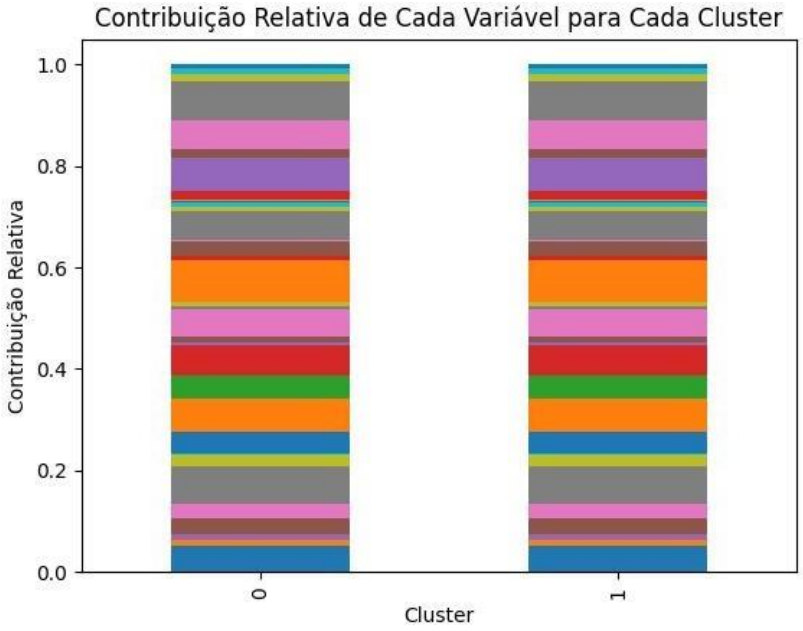


Figura 4.36: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método IForestASD.

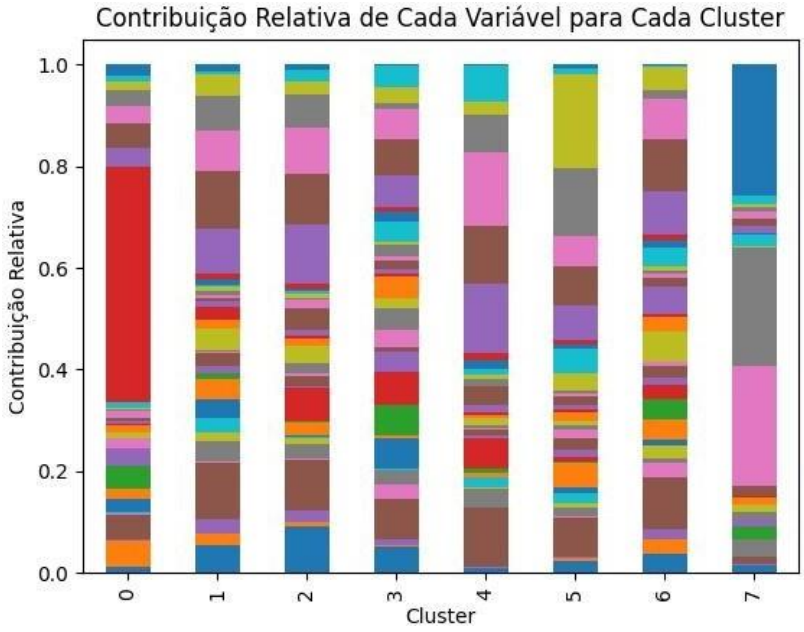


Figura 4.37: Explicação dos *Clusters* do tipo de ataque *WarezClient* do método IForestASD.

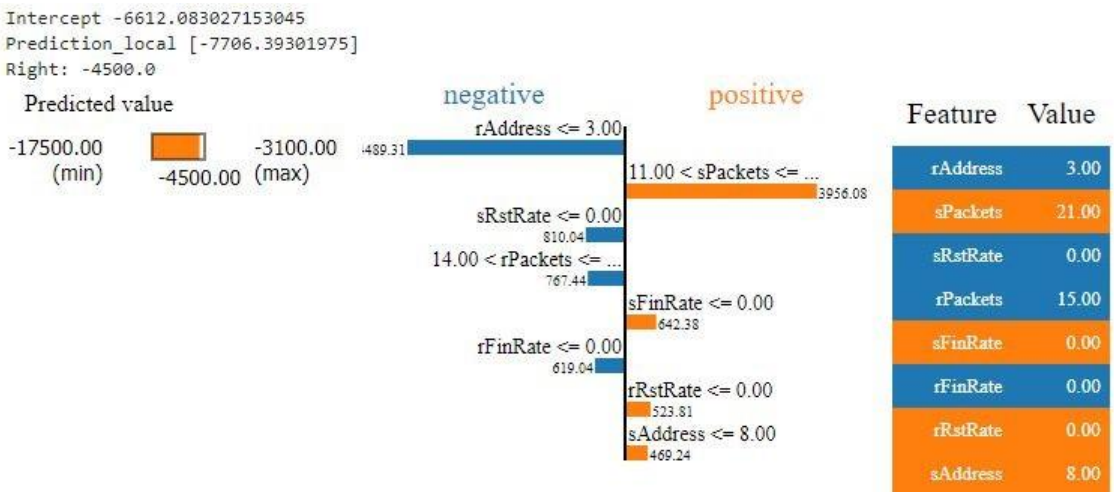


Figura 4.38: Explicação local do algoritmo LIME relativo ao método HSTrees do ataque *DOS*

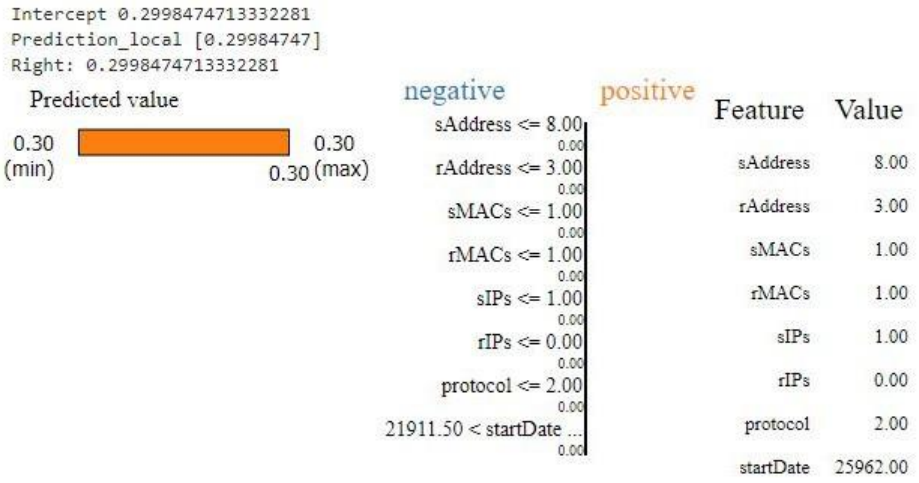


Figura 4.39: Explicação local do algoritmo LIME relativo ao método IForestASD do ataque *DOS*

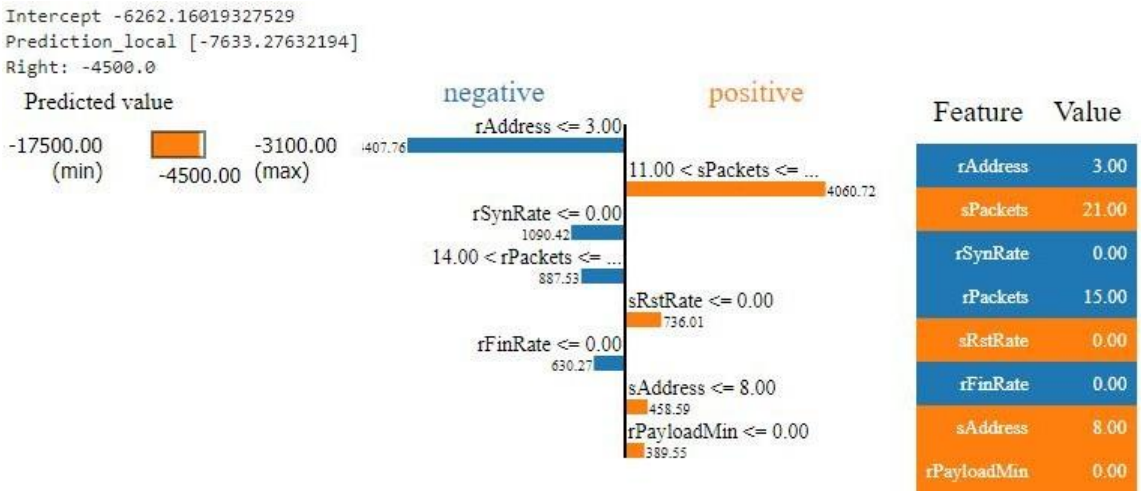


Figura 4.40: Explicação local do algoritmo LIME relativo ao método HSTrees do ataque *IP scan*

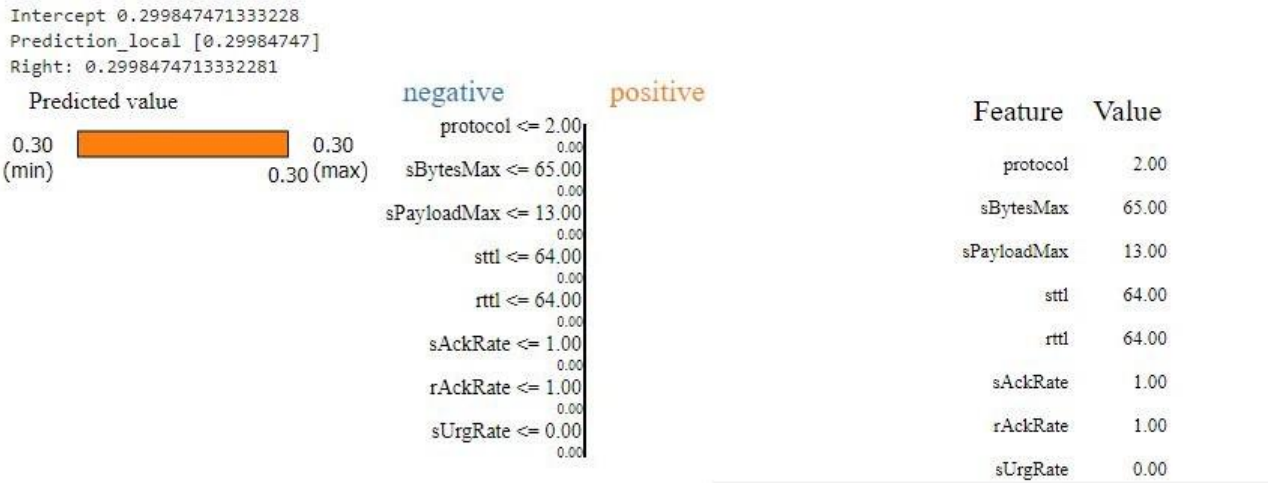


Figura 4.41: Explicação local do algoritmo LIME relativo ao método IForestASD do ataque *IP scan*

IForesASD	B1	B7	B12	B15	B16	B17	B18	B19	B20	B21	B22	B23	B24	B25	B26
IP scan							-		-	-	-	-	-	-	-
MITM	-	-	-	-	-	-	-	-							
Port scan							-		-	-	-	-	-	-	-
Replay	-	-	-	-	-	-	-	-							
DDOS	-	-	-	-	-	-	-	-							

Tabela 4.9: Explicações dos ataques do conjunto de dados *ICS-Flow* relativos ao método IForestASD. Utilizando o *ICS-Flow* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

HST	B1	B2	B7	B12	B15	B16	B18	B19	B27	B28	B29
IP scan		-	+		-	+	+		-	-	-
MITM	+	-	+		-	-		-	-	-	
Port scan	+	+	-			-	-	-	+		-
Replay	-		-	+	-			+	+	-	+
DDOS		+	+	+		-	+		-	-	-

Tabela 4.10: Explicações dos ataques do conjunto de dados *ICS-Flow* relativos ao método HSTrees com recurso a seleção de atributos. Utilizando o *ICS-Flow* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

IForestASD	B1	B2	B7	B12	B15	B16	B18	B19	B27	B28	B29
IP scan		-	-	+	+	+			-	-	-
MITM		-	-		+		+	+	-	-	-
Port scan		-	+			+	-	+	-	-	-
Replay	-	-	+	+				+	-	-	-
DDOS		-	-			+	-	+	-	-	-

Tabela 4.11: Explicações dos ataques do conjunto de dados *ICS-Flow* relativos ao método IForestASD com recurso a seleção de atributos. Utilizando o *ICS-Flow* do site oficial, na ordem original, foi utilizado o segundo exemplo do ataque para a explicação. A cor laranja diz respeito a contribuição positiva do atributo para que o ponto seja classificado como anomalia, enquanto a cor azul é o inverso.

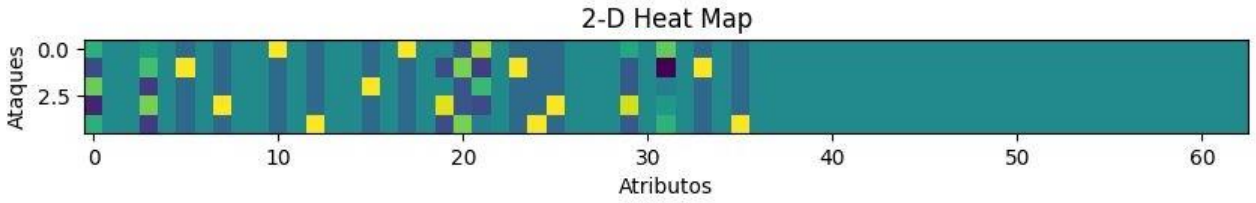


Figura 4.42: Mapa de calor do método HSTrees, relativo à utilização dos atributos, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor as cores mais vivas, (cores claras), significam que aquele atributo é muito utilizado enquanto as cores mais frias, (cores escuras), é o inverso. Os atributos que não são utilizados apresentam a mesma cor.

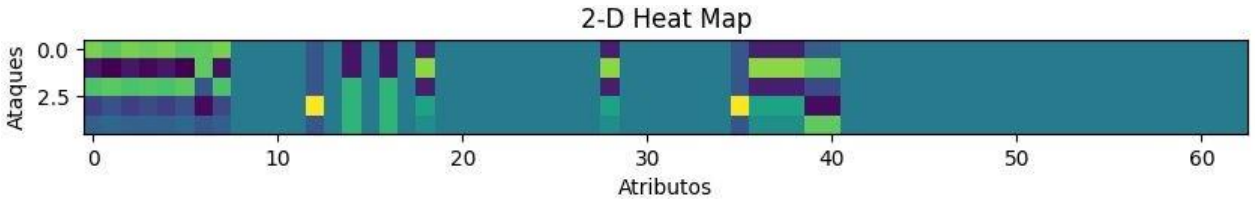


Figura 4.43: Mapa de calor do método IForestASD, relativo à utilização dos atributos, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor as cores mais vivas, (cores claras), significam que aquele atributo é muito utilizado enquanto as cores mais frias, (cores escuras), é o inverso. Os atributos que não são utilizados apresentam a mesma cor

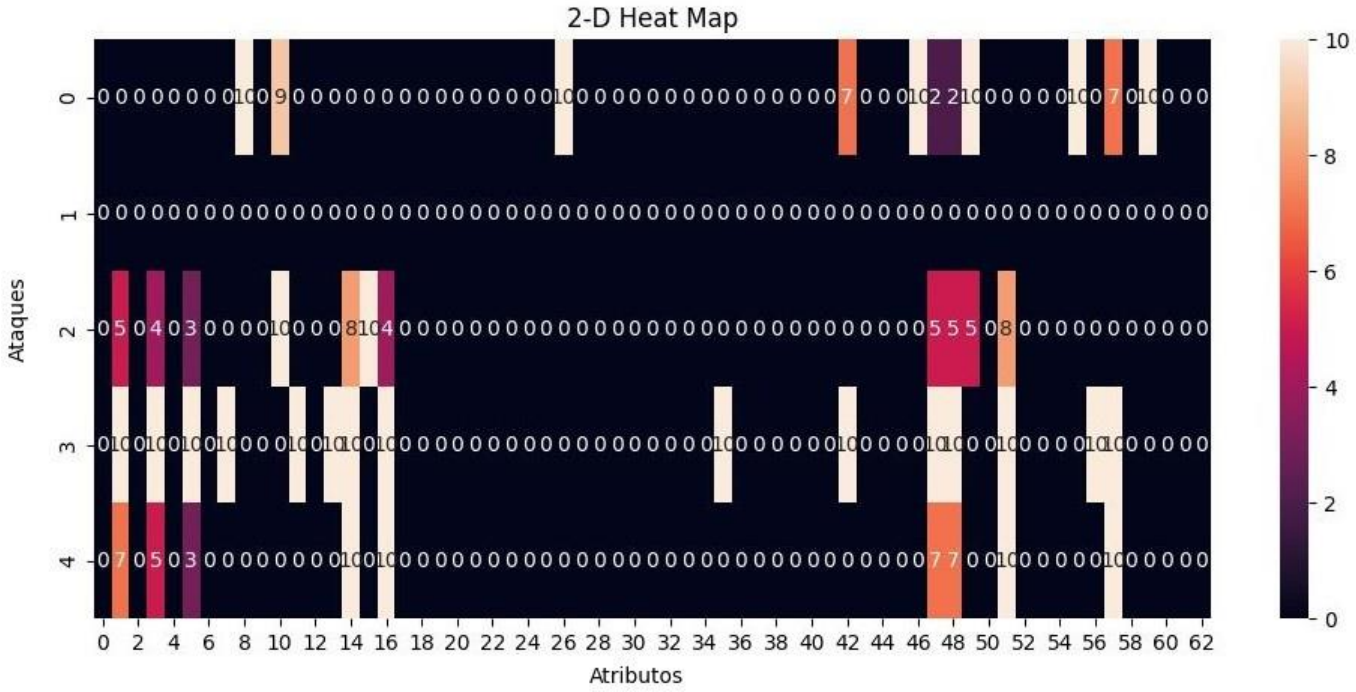


Figura 4.44: Mapa de calor do método IForestASD, relativo à utilização dos atributos com classificação negativa, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

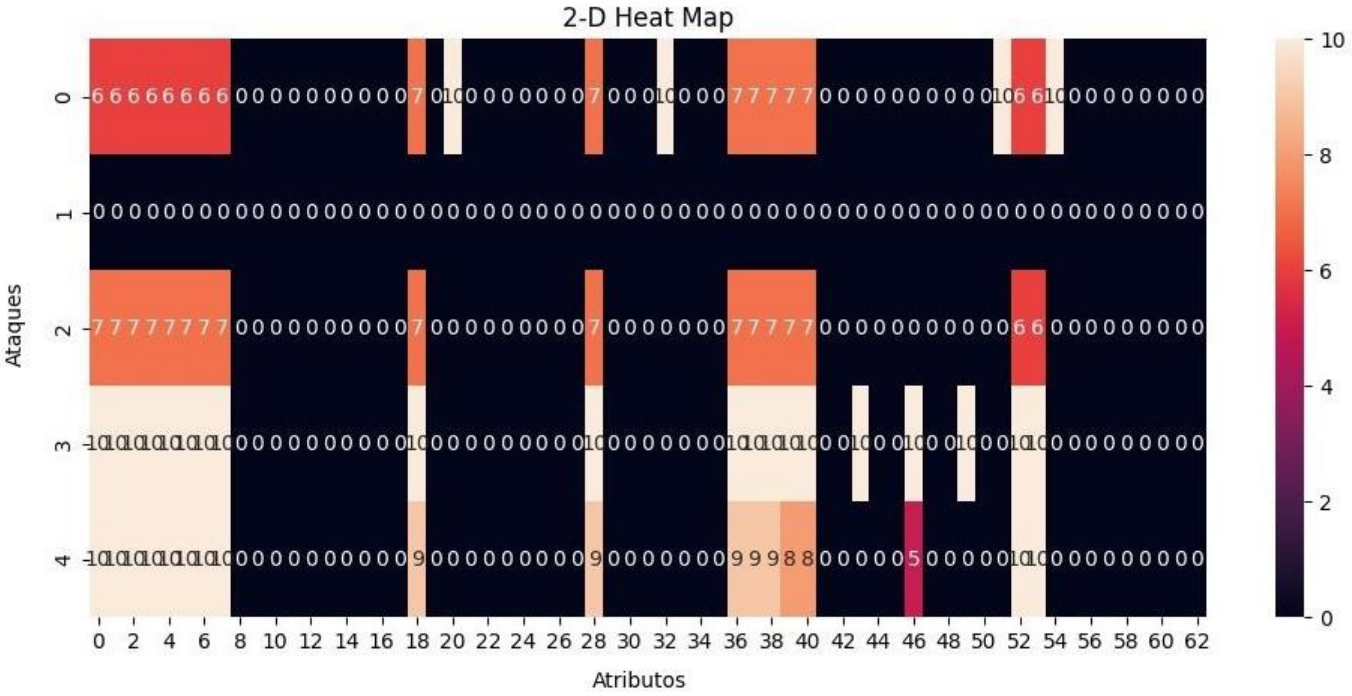


Figura 4.45: Mapa de calor do método IForestASD, relativo à utilização dos atributos com classificação positiva, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

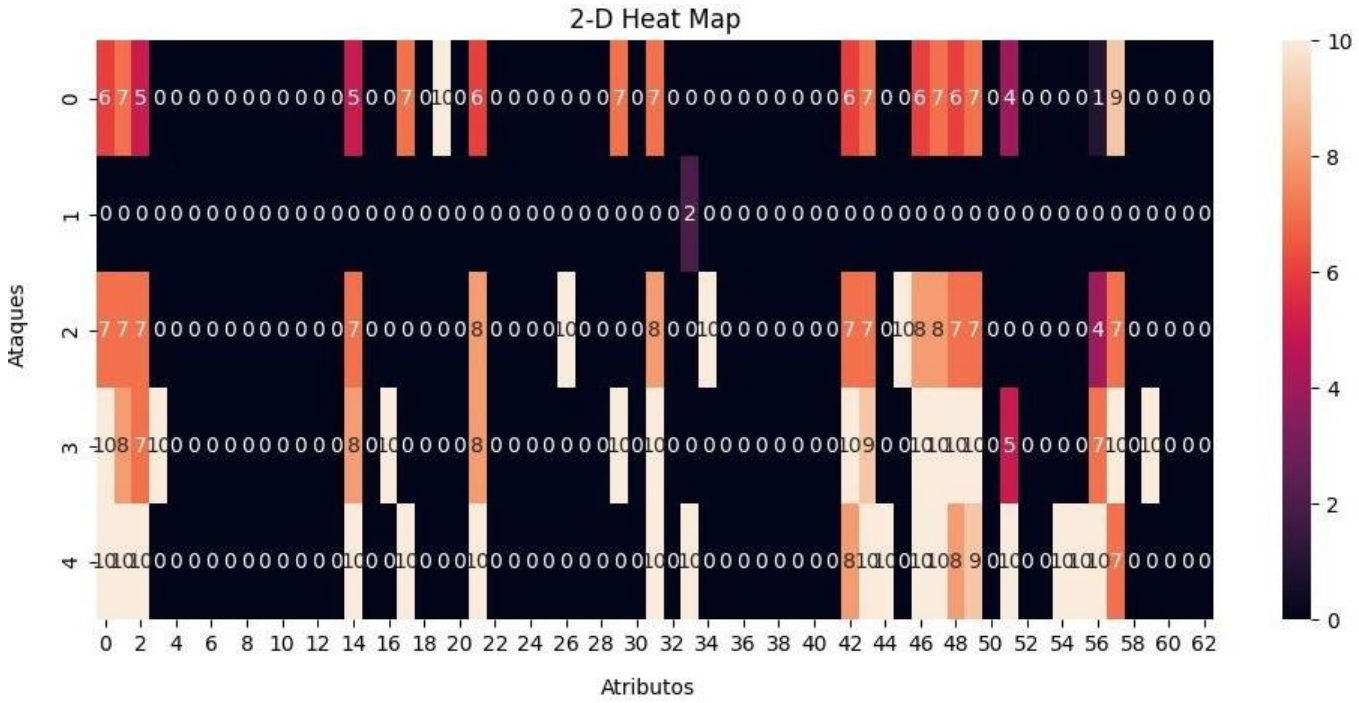


Figura 4.46: Mapa de calor do método HSTrees, relativo à utilização dos atributos com classificação negativa, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

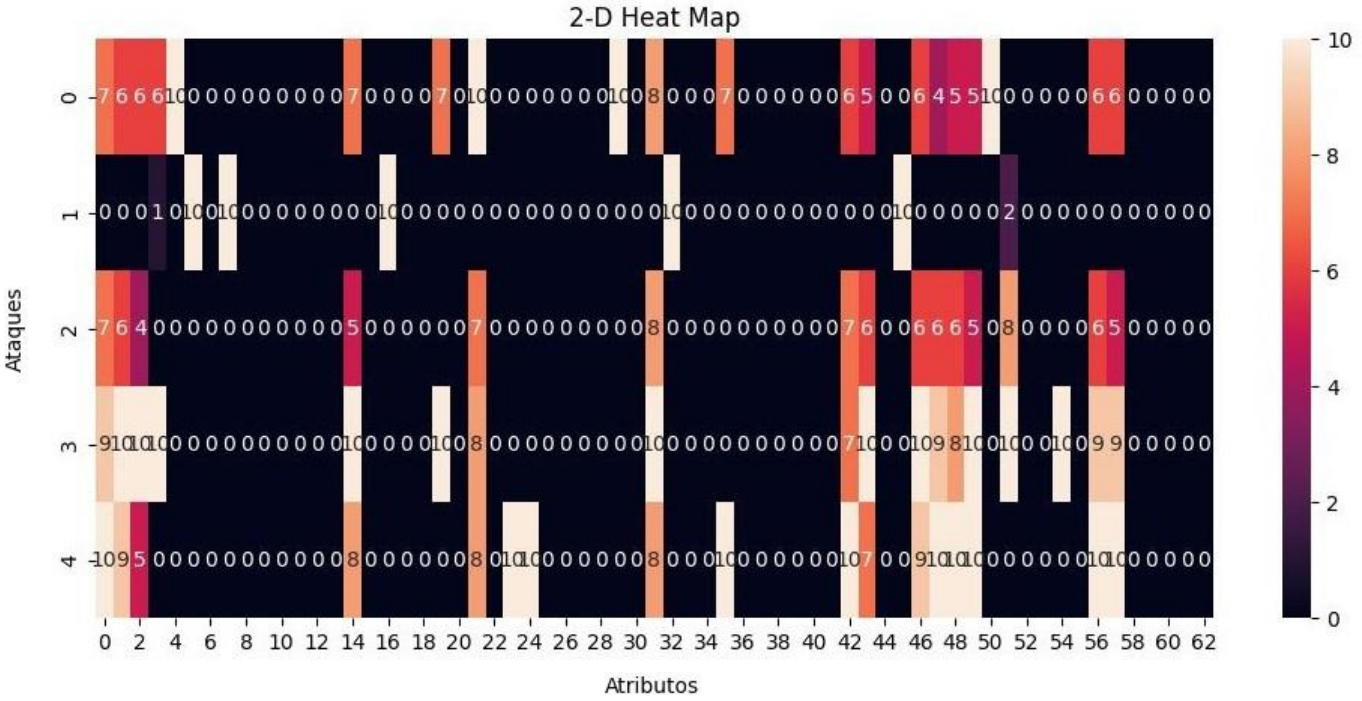


Figura 4.47: Mapa de calor do método HSTrees, relativo à utilização dos atributos com classificação positiva, na explicação dos ataques envolventes na totalidade do *ICS-Flow* original. No mapa de calor está presente uma escala de zero a dez, na qual zero significa pouco ou sem utilização e dez significa muita utilização.

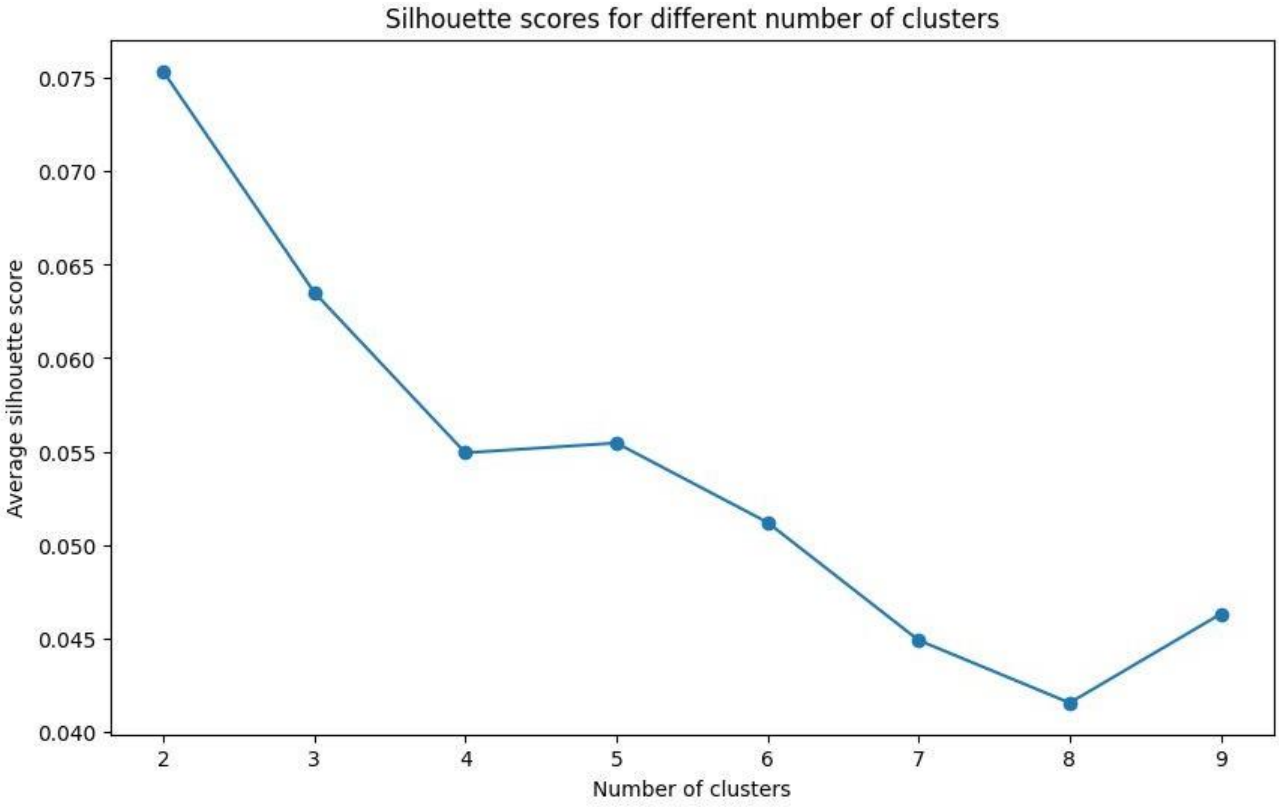


Figura 4.48: Silhouette do tipo de ataque *MITM* do método HSTrees. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

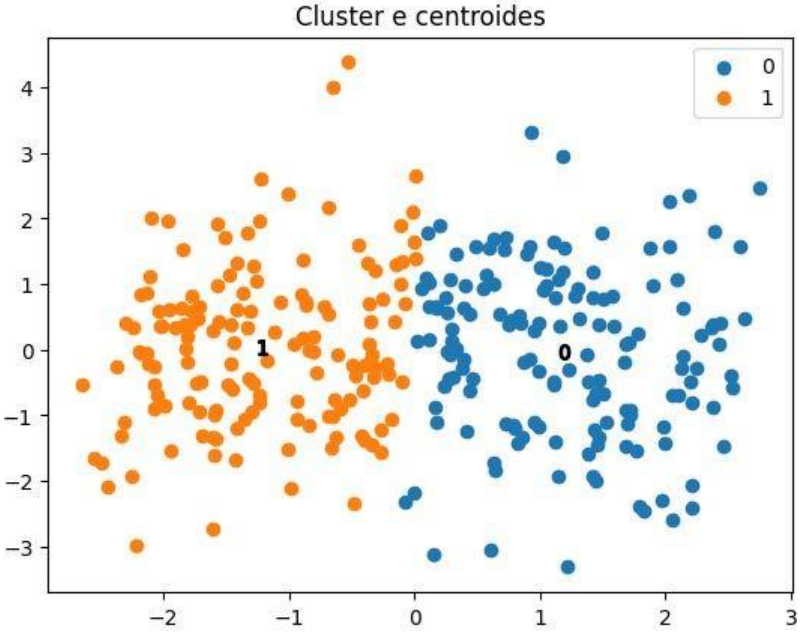


Figura 4.49: *Cluster* e centroides do tipo de ataque *MITM* do método HSTrees. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

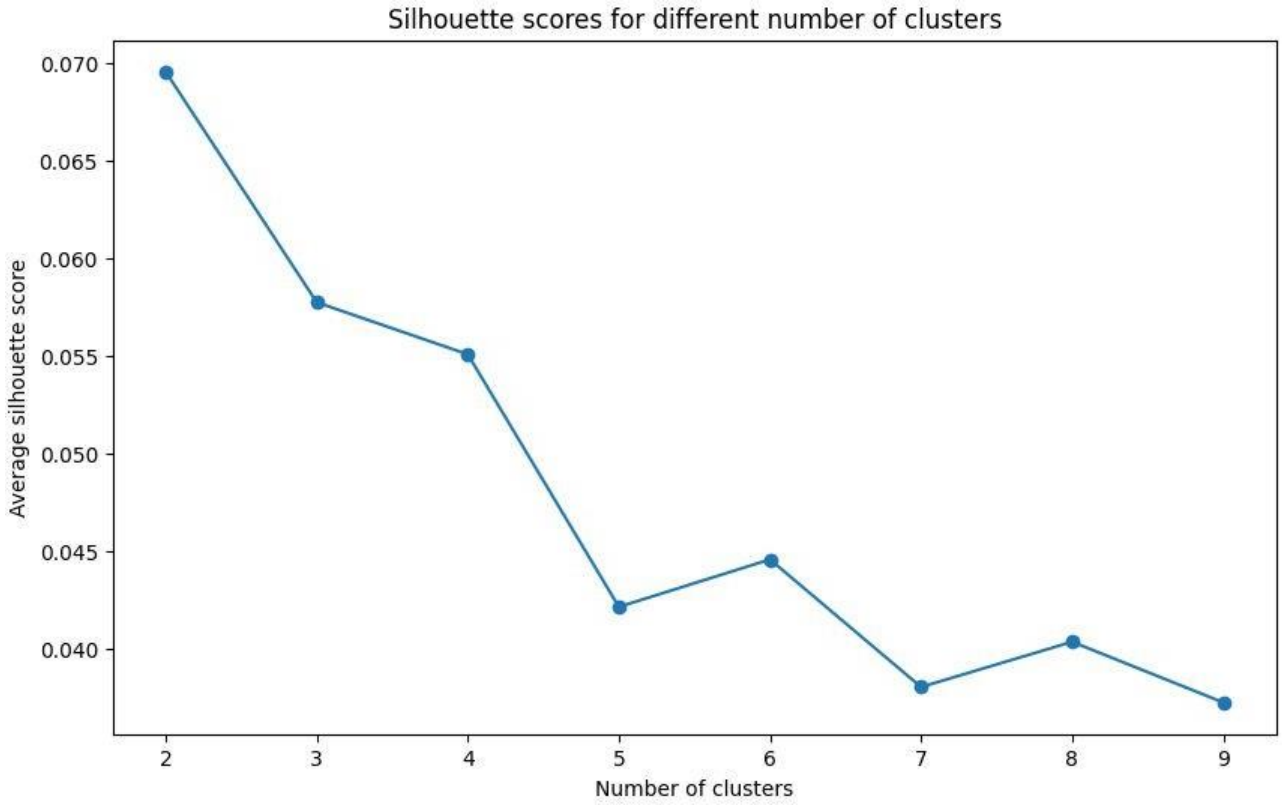


Figura 4.50: Silhouette do tipo de ataque *IPScan* do método HSTrees. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

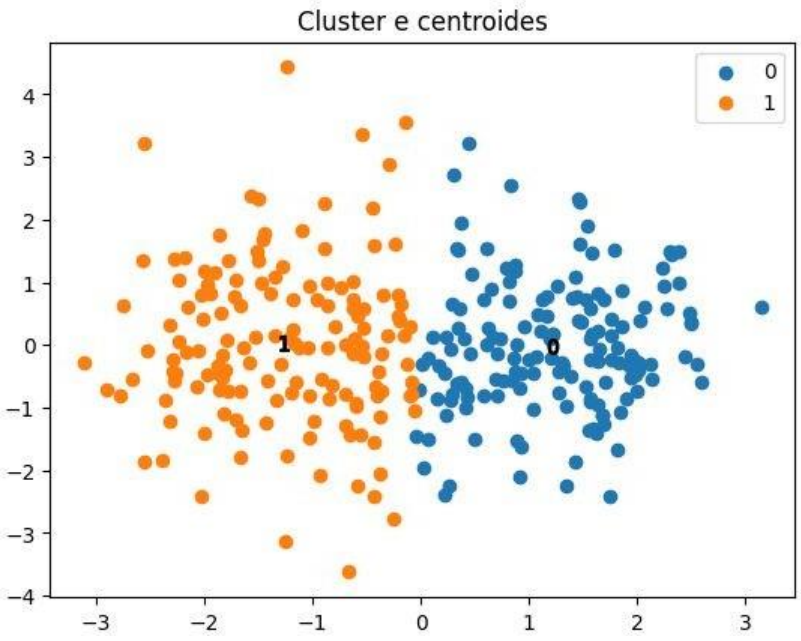


Figura 4.51: *Cluster* e centroides do tipo de ataque *IPScan* do método HSTrees. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

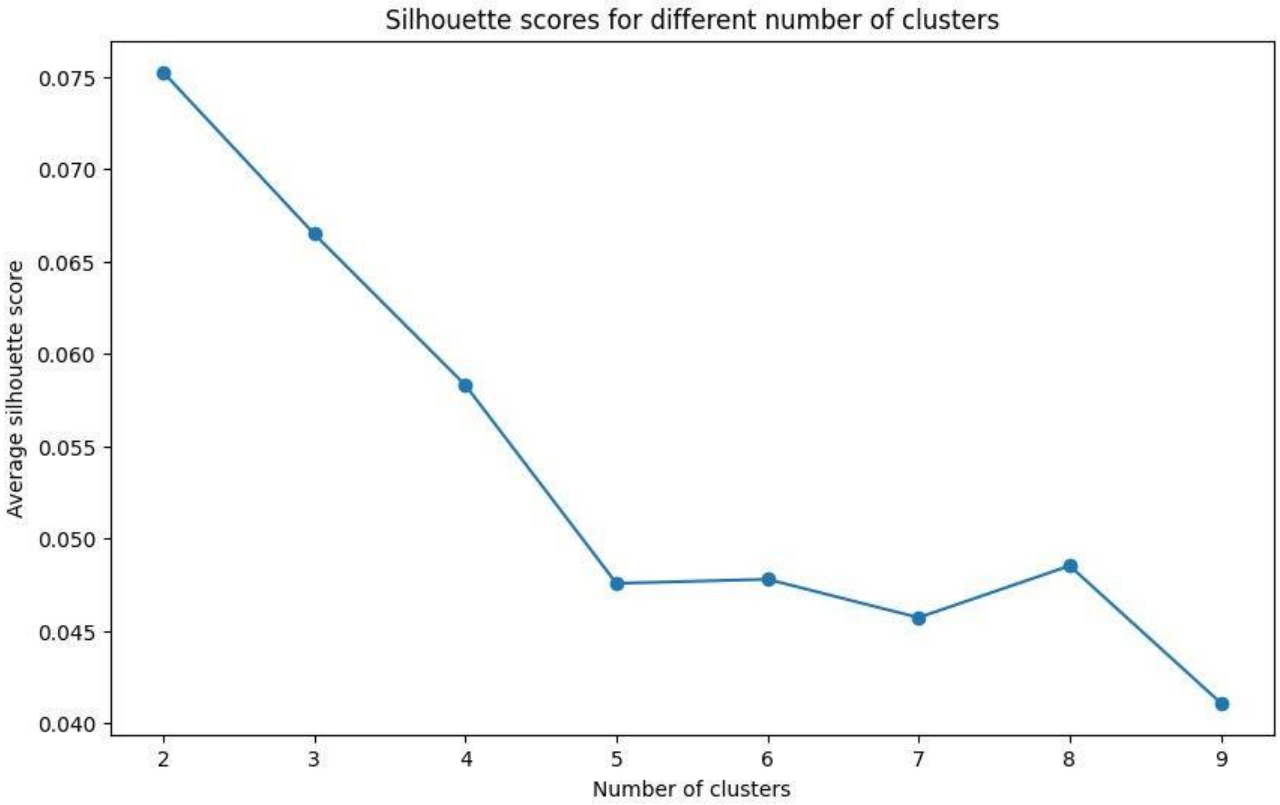


Figura 4.52: Silhouette do tipo de ataque *MITM* do método IForestASD. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

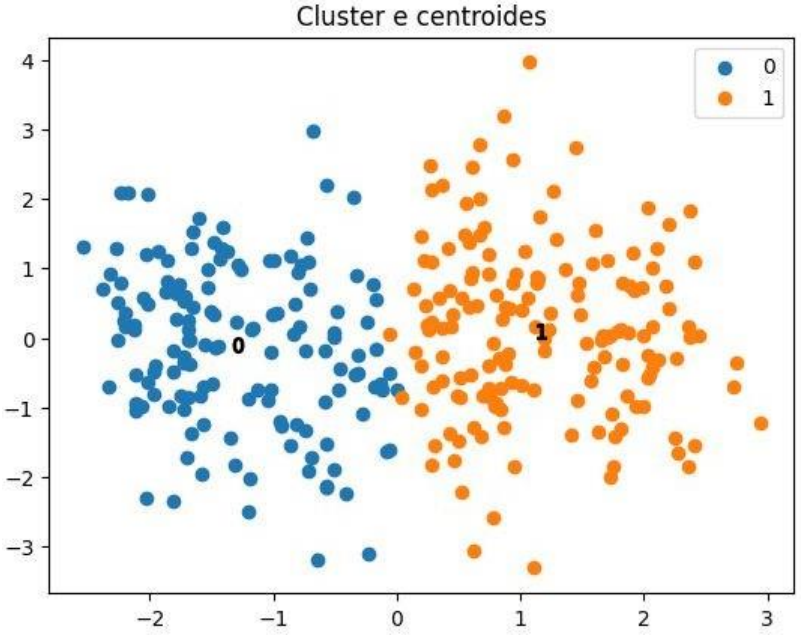


Figura 4.53: *Cluster* e centroides do tipo de ataque *MITM* do método IForestASD. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

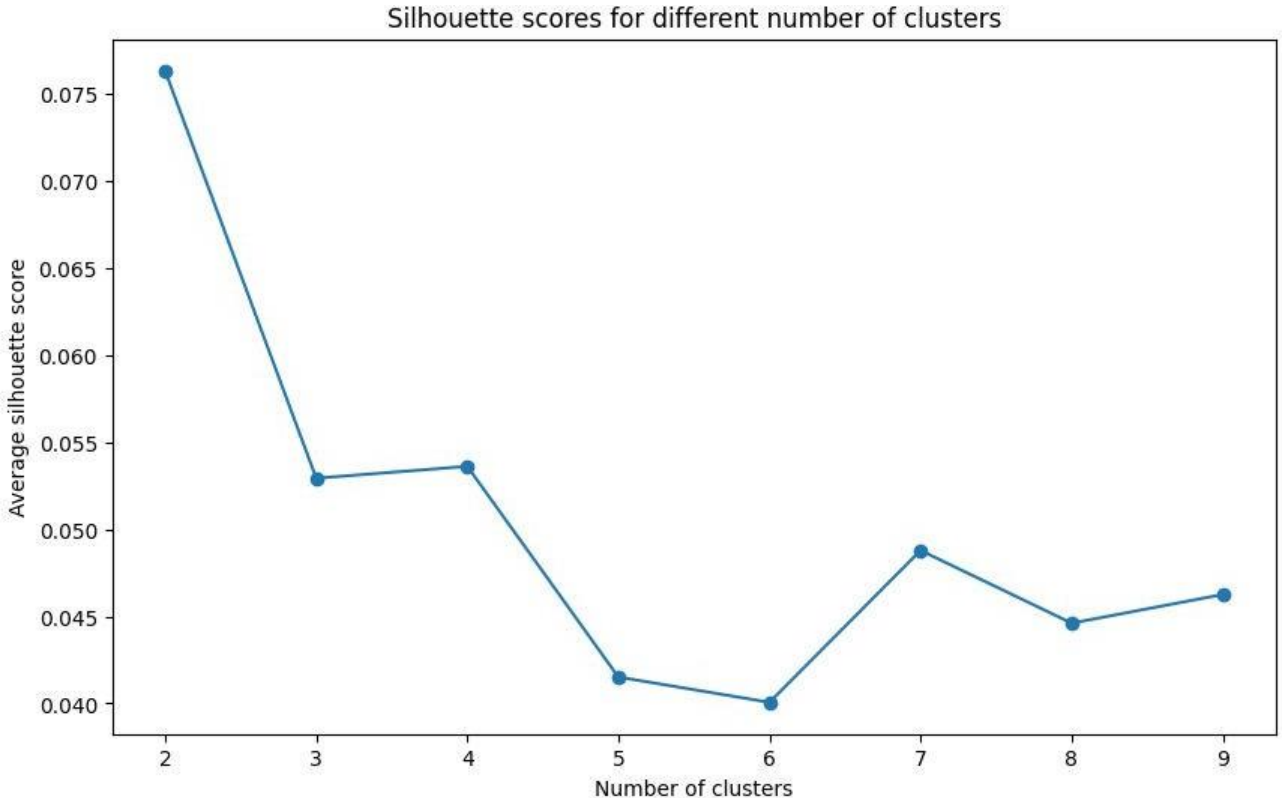


Figura 4.54: Silhouette do tipo de ataque *IPScan* do método IForestASD. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

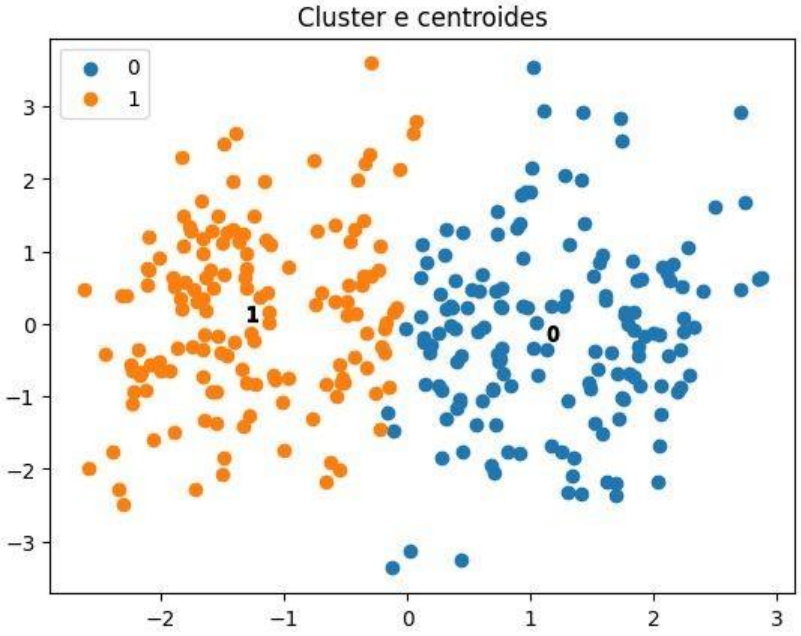


Figura 4.55: *Cluster* e centroides do tipo de ataque *IPScan* do método IForestASD. Os centroides estão marcados numericamente nas figuras. Teste relativo à utilização do conjunto de dados *ICS-Flow* original.

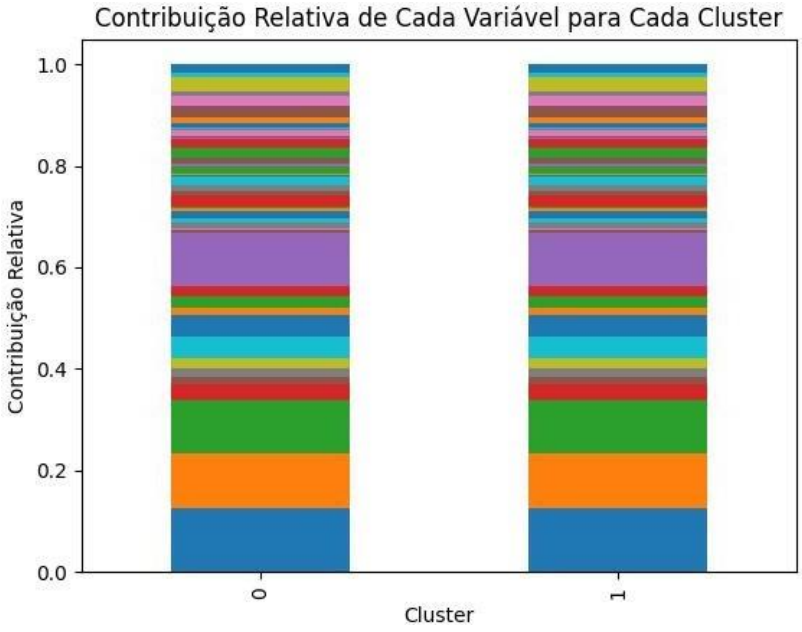


Figura 4.56: Explicação dos *Clusters* do tipo de ataque *MITM* do método HSTrees.

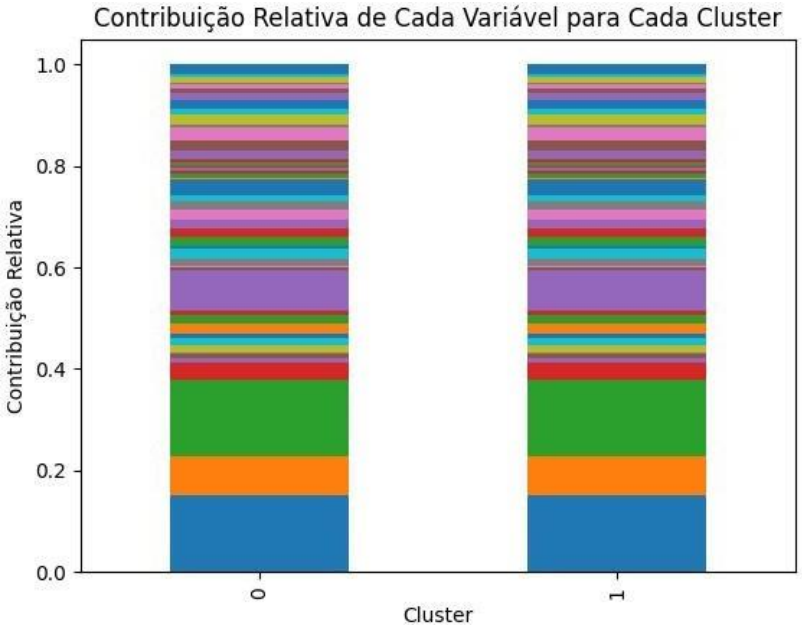


Figura 4.57: Explicação dos *Clusters* do tipo de ataque *IPScan* do método HSTrees.

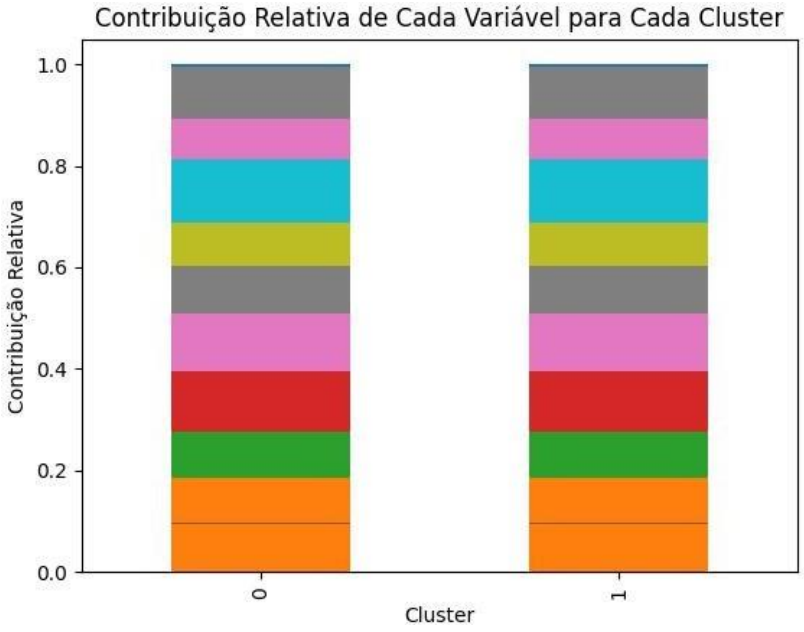


Figura 4.58: Explicação dos *Clusters* do tipo de ataque *MITM* do método IForestASD.

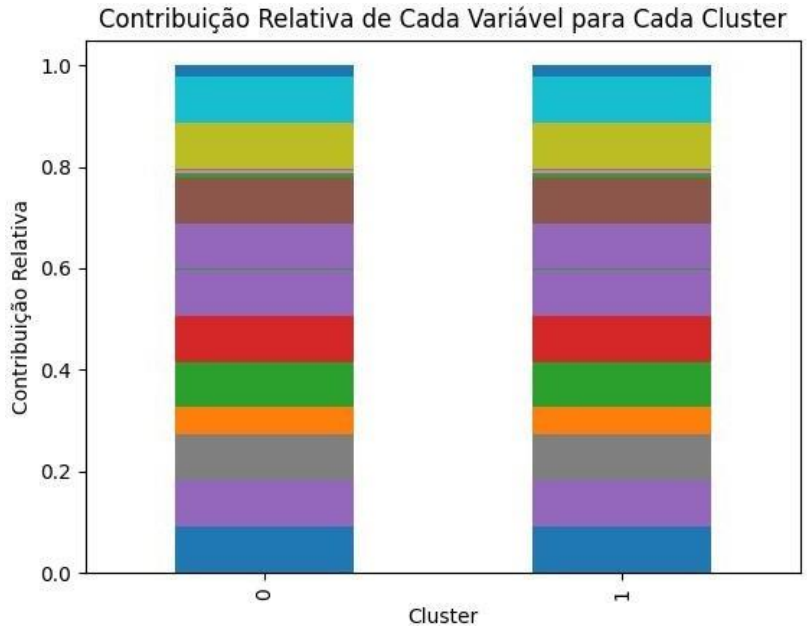


Figura 4.59: Explicação dos *Clusters* do tipo de ataque *IPScan* do método IForestASD.

Capítulo 5

Conclusão

Atualmente as pessoas estão mais consciencializadas acerca das ameaças que enfrentam de forma constante no espaço digital. Como resultado desta compreensão tem havido um aumento de estudos relacionados com a segurança informática. Para atingir este objetivo existem vários caminhos a seguir que podem ser a modernização de ferramentas já existentes ou criação de novas ferramentas.

O foco desta dissertação é a explicação de anomalias de rede, ao qual foram respondidas várias questões importantes.

Como foi dito anteriormente, os conjuntos de dados escolhidos são de extrema relevância no tema, e isso, é demonstrado pela utilização do mesmo na área de investigação. Com estes conjuntos de dados foi possível realizar várias experiências fundamentais, com objetivo de dar resposta às questões anteriormente lançadas. Para além, dos conjuntos de dados, foram utilizados dois métodos não supervisionados para a realização das experiências, sendo esses métodos os seguintes: HSTrees e IForestASD. Com base nas medidas de desempenho recolhidas pode-se chegar à conclusão que o método IForestASD é mais fidedigno em relação ao HSTrees, devido aos resultados obtidos pela medida *AUC*. Apesar do que foi dito anteriormente, as explicações obtidas de ambos os métodos fornecem de forma consistente pontos comuns entre as várias explicações.

Ao longo das experiências realizadas, foi determinado que a utilização destas tecnologias em conjunto constitui uma solução para o problema deteção de intrusões, pois estas tecnologias em conjunto conseguem fornecer resultados muito mais satisfatórios. Por último pode-se dizer que a utilização da ferramenta LIME permitiu que fosse possível haver uma perceção sobre a relação dos atributos e o processo de deteção de anomalias. Sendo assim, foi possível chegar à conclusão, relativamente aos testes realizados no capitulo 4, que os mesmos ataques podem ter explicações diferentes, porém, ataques diferentes podem ter explicações semelhantes.

Em conclusão, as descobertas desta dissertação contribuem no âmbito da integração de ferramentas de explicação, com o objetivo de ser possível ter resultados com um nível de interpretabilidade superiores em áreas que até agora tinham um nível de interpretabilidade baixo.

Para além disto, os resultados mostram a importância da implementação desta tecnologia nos sistemas atualmente utilizados.

5.1 Trabalho Futuro

Este trabalho lançou algumas luzes sobre as dificuldades dos utilizadores em relação a métodos baseados em caixa negra. No entanto, existe a possibilidade de obter melhores resultados. Um dos pontos que deve ser explorado é a comparação das explicações obtidas de vários métodos de explicação diferentes, com o sentido de saber se existem pontos que coincidem ou não.

No futuro, uma das soluções possíveis pode ser a possibilidade de conceder várias explicações da mesma anomalia com a utilização de vários métodos de explicação, na qual, o utilizador pode comparar e decidir com base na comparação destas.

Apêndice A

Atributos do Conjunto de dados KDD99

Tabela A.1: Lista de atributos relevantes na explicação dos ataques do KDD.

Nome do atributo	Nome atribuído ao atributo
dst_bytes	A1
duration	A2
num_root	A3
hot	A4
num_file_creations	A5
src_bytes	A6
dst_host_error_rate	A7
flag	A8
rerror_rate	A9
protocol_type	A10
is_guest_login	A11
dst_host_srv_rerror_rate	A12
srv_rerror_rate	A13
dst_host_rerror_rate	A14
protocol_type	A15
is_guest_login	A16

Nome do atributo	Nome atribuído ao atributo
wrong_fragment	A17
num_compromised	A18
num_shells	A19
num_file_creations	A20
num_access_files	A21
urgent	A22
num_failed_logins	A23
root_shell	A24
srv_dif_host_rate	A25
dst_host_srv_diff_host_rate	A26
dst_host_diff_srv_rate	A27
logged_in	A28
srv_serror_rate	A29
diff_srv_rate	A30
land	A31
service	A32
su_attempted	A33

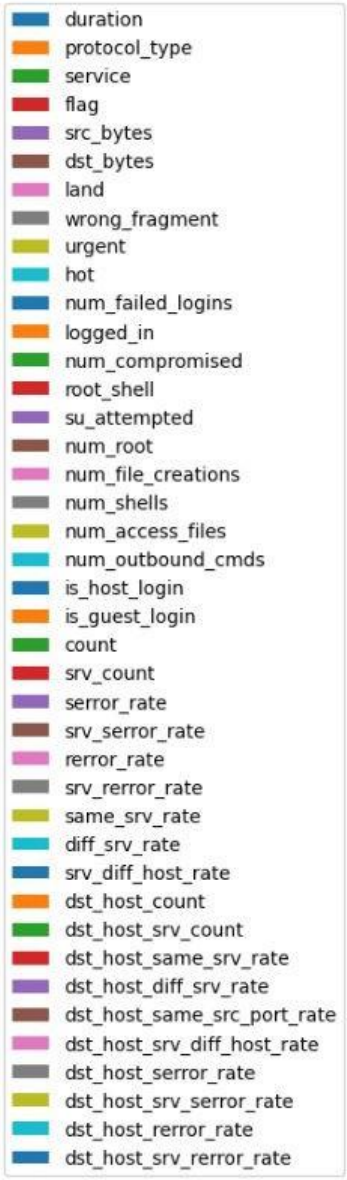


Figura A.1: Lista de atributos utilizados nas explicações dos *clusters* relacionados com o conjunto de dados *KDD99*.

Apêndice B

Atributos do Conjunto de dados ICS- Flow

Tabela B.1: Lista de atributos relevantes na explicação dos ataques do ICS-Flow.

Nome do atributo	Nome atribuído ao atributo
rAddress	B1
sPackets	B2
rSynRate	B3
rPackets	B4
sRstRate	B5
rFinRate	B6
sAddress	B7
rPayloadMin	B8
rRstRate	B9
sSynRate	B10
rBytesMin	B11
rMACs	B12
sFinRate	B13
sAckDelayMin	B14
sMACs	B15
sIPs	B16
rIPs	B17
protocol	B18

Nome do atributo	Nome atribuído ao atributo
startDate	B19
sBytesMax	B20
sPayloadMax	B21
sttl	B22
rttl	B23
sAckRate	B24
rAckRate	B25
sUrgRate	B26
startOffset	B27
duration	B28
endOffset	B29

Figura B.1: Lista de atributos utilizados nas explicações dos *clusters* relacionados com o conjunto de dados *ICS-Flow*.

Bibliografia

[1] C.C Aggarwal. An introduction to outlier analysis. pages 1–34, 2017. doi:https://doi.org/10.1007/978-3-319-47578-3_1.

[2] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

[3] Kasun Amarasinghe, Kevin Kenney, and Milos Manic. Toward explainable deep neural network based anomaly detection. pages 311–317, 2018. doi:10.1109/HSI.2018.8430788.

[4] Liat Antwarg, RM Miller, B Shapira, and L Rokach. Explaining anomalies detected by autoencoders using shap. arxiv 2019. *arXiv preprint arXiv:1903.02407*.

[5] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[6] Kushilevitz Eyal Mansour Yishay Ben-David, Shai. Online learning versus offline learning. pages 1–19, 1997.

[7] Dhruba Kumar Bhattacharyya and Jugal Kumar Kalita. *Network anomaly detection: A machine learning perspective*. Crc Press, 2013.

[8] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[9] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[10] Kumar Vipin Chandola Varun, Banerjee Arindam. Anomaly detection : A survey. pages 1–74, 2009.

[11] Zhixiang Chen, Jian Tang, and Ada Wai-Chee Fu. Modeling and efficient mining of intentional knowledge of outliers. pages 44–53, 2003.

[12] Bao Chong et al. K-means clustering algorithm: a brief review. *vol*, 4:37–40, 2021.

[13] Xuan Hong Dang, Ira Assent, Raymond T. Ng, Arthur Zimek, and Erich Schubert. Discriminative features for identifying and interpreting outliers. pages 88–99, 2014. doi:10.1109/ICDE.2014.6816642.

[14] Collins Joseph B Dasgupta, Prithviraj. A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks. 1:1–13, 2019. ISSN: 0738-4602.

[15] Alireza Dehlaghi-Ghadim, Mahshid Helali Moghadam, Ali Balador, and Hans Hansson. Anomaly detection dataset for industrial control systems, 2023.

[16] Google for Developers. Classificação: curva roc e auc, 2022-09-27 UTC.

[17] Garrigues Carles Rifà-Pous Helena Garcia-Font, Victor. Difficulties and challenges of anomaly detection in smart cities: A laboratory analysis. 1:1–21, 2018. ISSN: 14248220. doi:10.3390/s18103198.

[18] Díaz-Verdejo J.-Maciá-Fernández G. Vázquez E. García-Teodoro, P. Anomaly-based network intrusion detection: Techniques, systems and challenges. pages 1–11, 2009. ISSN: 01674048. doi:10.1016/j.cose.2008.08.003.

[19] Lobato Pastana-Andreoni Lopez-Martin-Sanz Igor Jochem Cardenas Alvaro A Carlos Otto- Duarte M B Pujolle Guy Gonzalez, Antonio. An adaptive real-time architecture for zero-day threat detection. pages 1–6, 2018.

[20] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *CoRR*, abs/1802.01933, 2018.

[21] Mansour Sheikhan Hamid Bostani. Hybrid of anomaly-based and specification-based ids for internet of things using unsupervised opf based on mapreduce approach. 98:52–71, 2017. ISSN: 0140-3664.

[22] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45:171–186, 2001.

[23] Erich Schubert Hans-Peter Kriegel, Peer Kr¨oger and Arthur Zimek. Interpreting and unifying outlier scores. pages 1–12, 2011.

[24] John M Hugh. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. 3:262–294, 2000.

[25] Edwin M. Knorr and Raymond T. Ng. Finding intensional knowledge of distance-based outliers. pages 1–12, 1999.

[26] Forrest H.; Andre-David; Keane Martin A. Koza, John R.; Bennett. Artificial intelligence in design '96. pages 151–170, 1996. ISSN: 978-94-010-6610-5. doi:10.1007/978-94-009-0279-4$_9$.

[27] Rikard Laxhammar and Göran Falkman. Online learning and sequential anomaly detection in trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1158–1173, 2013.

[28] Ninghao Liu, Donghwa Shin, and Xia Hu. Contextual outlier interpretation. 2018.

[29] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[30] Meghanath Macha and Leman Akoglu. Explaining anomalies in groups with characterizing subspace rules. 2018.

[31] Chan Philip K Mahoney Matthew V. An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. 1:1–21, 2003.

[32] Donna Katzman McClish. Analyzing a portion of the roc curve. *Medical decision making*, 9(3): 190–195, 1989.

[33] Barbora Micenková, Raymond T. Ng, Xuan-Hong Dang, and Ira Assent. Explaining outliers by subspace separability. pages 518–527, 2013. doi:10.1109/ICDM.2013.132.

[34] Vieira Marco Kounev Samuel Avritzer-Alberto Payne Bryan D. Milenkoski, Aleksandar. Evaluating computer intrusion detection systems: A survey of common practices. pages 1– 41, 2015. ISSN: 15577341. doi:10.1145/2808691.

[35] Ing-Ray Mitchell Robert, Chen. A survey of intrusion detection in wireless network applications. 42:1–23, 2014. ISSN: 01403664. doi:10.1016/j.comcom.2014.01.012.

[36] Jiankun Hu Mohiuddin Ahmed, Abdun Naser Mahmood. A survey of network anomaly detection techniques. 60:19–31, 2016,. ISSN: 1084-8045.

[37] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[38] Jourdan Guy-Vincent Viktor Herna L Mvula Paul K, Branco Paula. A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning. 1:1–33, 2023. doi:10.1007/s44248-023-00003-x.

[39] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. Machine learning for anomaly detection: A systematic review. *Ieee Access*, 9:78658–78700, 2021.

[40] Adams Niall M Noble, Jordan. Real-time dynamic network anomaly detection. 33:1–7, 2018.

[41] Murugaraj Odiathevar, Winston KG Seah, and Marcus Frean. A hybrid online offline system for network anomaly detection. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2019.

[42] Atilla Özgür and Hamit Erdem. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. 2016.

[43] Gruenwald Le Leal Eleazar Nguyen-Christopher Silvia Shejuti Panjei, Egawati. A survey on outlier explanations. pages 1–32, 2022. ISSN: 0949877X. doi:10.1007/s00778-021-00721-1.

[44] M. Mazhar Rathore, Anand Paul, Awais Ahmad, Seungmin Rho, Muhammad Imran, and Mohsen Guizani. Hadoop based realtime intrusion detection for high-speed networks. pages 1–6, 2016. doi:10.1109/GLOCOM.2016.7841864.

[45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?"explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[46] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[47] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, and Weng-Keen Wong. Sequential feature explanations for anomaly detection. 2015.

[48] J Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K Chan. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection. pages 1–15, 2000.

[49] Naoya Takeishi. Shapley values of reconstruction errors of pca for explaining anomaly detection. pages 1–6, 2020.

[50] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *Twenty-second international joint conference on artificial intelligence*. Citeseer, 2011.

[51] Bagheri Zadeh Pooneh Thornton, Greg. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. pages 1–6, 2013. ISSN: 26662817. doi:10.1016/j.fsidi.2022.301379.

[52] Barry Mariam Boly Aliou-Chabchoub Yousra Chiky Raja Montiel Jacob Tran Vinh-Thuy Togbe, Maurras Ulbricht. Anomaly detection for data streams based on isolation forest using scikit- multiflow. pages 1–17, 2020.

[53] Ming Ting Kai Zhou-Zhi-Hua Tony Liu, Fei. Isolation forest. pages 413–422, 2008.

[54] Seybert A Visweswaran S-Saul M Hauskrecht M. Valko M, Cooper G. Conditional anomaly detection methods for patient-management alert systems. pages 1–11, 2008.

[55] Yang yanqin Wang Xiujuan Wang Maonan, Zheng Kanfeng. An explainable machine learning framework for intrusion detection systems. 8:73127–73141, 2020. ISSN: 21693536. doi:10.1109/ACCESS.2020.2988359.

[56] Thomas W. Woolley. An investigation of the effect of the swamping phenomenon on several block procedures for multiple outliers in univariate samples. pages 1–6, 2013. ISSN: 2161-718X. doi:10.4236/ojs.2013.35035.

[57] Zhibo Zhang, Hussam Al Hamadi, Ernesto Damiani, Chan Yeob Yeun, and Fatma Taher. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access*, 2022.

[58] Victor Zhou. Machine learning for beginners: An introduction to neural networks. *Towards Data Science*, 12, 2019.