Revista brasileira de iniciação científica
ISSN: 2359-232X

Um estudo de caso da inteligência artificial explicável aplicada na detecção de intrusão em redes IoT

A case study on explainable artificial intelligence applied to intrusion detection in IoT networks

Un estudio de caso de la inteligencia artificial explicable aplicada a la detección de intrusiones en IoT

Matheus Buschermoehle[1]
Fernando Santos[2]

Resumo: O crescimento de dispositivos de Internet das Coisas (IoT) desafia a cibersegurança e demanda mecanismos eficazes para detecção de intrusão baseados em modelos de aprendizado de máquina. Diante da importância de explicar decisões automatizadas, surge a pergunta de pesquisa: como a inteligência artificial explicável (XAI) explica intrusões detectadas com aprendizado de máquina? Este trabalho investiga o uso de XAI com *Shapley Additive Explanations* (SHAP) em três modelos para detecção de intrusão em redes IoT: Random Forest, LightGBM e *Multilayer Perceptron* (MLP). Utilizando o *dataset* NSL-KDD, foram geradas explicações globais para identificar os atributos relevantes para as predições. O SHAP permitiu identificar atributos relevantes para as predições, evidenciando maior similaridade entre os modelos Random Forest e LightGBM, enquanto o MLP apresenta menor correlação e se apoia em atributos distintos. A pesquisa reforça a importância da interpretabilidade na detecção de intrusão em redes IoT automatizada com modelos de aprendizado de máquina.
Palavras-chave: Inteligência Artificial Explicável. Shapley Additive Explanations. Internet das Coisas. Random Forest. LightGBM. Multilayer Perceptron.

Abstract: The growth of Internet of Things (IoT) devices challenges cybersecurity and demands effective intrusion-detection mechanisms based on machine learning models. Given the importance of explaining automated decisions, the following research question arises: how does explainable artificial intelligence (XAI) interpret intrusions detected by machine learning? This study investigates the use of XAI with Shapley Additive Explanations (SHAP) in three models for intrusion detection in IoT networks: Random Forest, LightGBM, and Multilayer Perceptron (MLP). Using the NSL-KDD dataset, global explanations were generated to identify the features most relevant to predictions. SHAP revealed key features driving model decisions, showing greater similarity between Random Forest and LightGBM, while MLP presents lower correlation and relies on distinct features. The research reinforces the importance of interpretability in automated intrusion detection in IoT networks using machine learning.
Keywords: Explainable Artificial Intelligence. Shapley Additive Explanations. Internet of Things. Random Forest. LightGBM. Multilayer Perceptron.

[1] Bacharel em Engenharia de Software. Universidade do Estado de Santa Catarina (UDESC). ORCID: https://orcid.org/0009-0001-9875-0947. E-mail: matheusbuschermoehle@gmail.com
[2] Doutor em Ciência da Computação. Universidade do Estado de Santa Catarina (UDESC). ORCID: https://orcid.org/0000-0002-8182-2246. E-mail: fernando.santos@udesc.br.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Resumen: El crecimiento de los dispositivos de Internet de las Cosas (IoT) representa un desafío para la ciberseguridad y exige mecanismos eficaces de detección de intrusiones basados en modelos de aprendizaje automático. Dada la importancia de explicar las decisiones automatizadas, surge la pregunta de investigación: ¿cómo explica la inteligencia artificial explicable (XAI) las intrusiones detectadas mediante aprendizaje automático? Este estudio investiga el uso de XAI con Shapley Additive Explanations (SHAP) en tres modelos de detección de intrusiones en redes IoT: Random Forest, LightGBM y Multilayer Perceptron (MLP). Utilizando el *dataset* NSL-KDD, se generaron explicaciones globales para identificar los atributos más relevantes en las predicciones. SHAP permitió identificar las características clave, mostrando mayor similitud entre Random Forest y LightGBM, mientras que MLP presenta menor correlación y se basa en atributos distintos. La investigación refuerza la importancia de la interpretabilidad en la detección automatizada de intrusiones en redes IoT mediante aprendizaje automático.

Palabras-clave: Inteligencia Artificial Explicable. Shapley Additive Explanations. Internet de las Cosas. Random Forest. LightGBM. Multilayer Perceptron.

Submetido 21/01/2026 **Aceito 29/07/2026** **Publicado 07/08/2026**

Revista brasileira de iniciação científica
ISSN: 2359-232X

Considerações Iniciais

Nas últimas décadas, a Internet das Coisas (IoT) tornou-se parte integral em infraestruturas tecnológicas modernas, viabilizando a interconexão digital entre objetos. Estima-se que em 2025 o número de dispositivos IoT em uso ultrapasse os 30 bilhões globalmente (Statista, 2024).

Um desafio que surge com o crescimento e adoção cada vez mais ampla deste modelo de conexão é a segurança cibernética dos dispositivos conectados. Sistemas de Detecção de Intrusão (IDS, do inglês *intrusion detection systems*) são mecanismos frequentemente adotados em aplicações IoT para identificar e mitigar ameaças à segurança da rede.

Técnicas de aprendizado de máquina têm sido aplicadas para automatizar a detecção de atividades atípicas no tráfego da rede. Entretanto, um desafio do uso de aprendizado de máquina é a "caixa-preta" associada a muitos algoritmos, que oferecem pouca capacidade de interpretação. Isto representa um desafio para analistas de segurança responsáveis por examinar os eventos registrados pelos IDS e tomar decisões operacionais. A falta de transparência compromete a capacidade dos analistas de validar, justificar e responder rapidamente às ameaças detectadas, o que é crucial para ambientes de segurança cibernética nos quais erros podem ter consequências graves. Este desafio pode comprometer a compreensão e a confiabilidade do aprendizado de máquina, prejudicando as decisões e colocando em risco eventuais auditorias. Isto evidencia a necessidade de soluções que promovam compreensão e confiabilidade do aprendizado de máquina em IDS.

A Inteligência Artificial Explicável (XAI, do inglês e*xplainable AI*) surge como uma abordagem promissora. O objetivo da XAI é proporcionar transparência e interpretabilidade aos modelos de aprendizado de máquina, facilitando a compreensão de suas predições (Ali et al., 2023).

Adotar XAI, em IDS para redes IoT, pode oferecer a analistas de segurança melhor compreensão acerca de como o sistema identifica possíveis ameaças, aumentando a confiabilidade e favorecendo decisões mais assertivas. Essa transparência e interpretabilidade em modelos preditivos, aplicados em IDS, não apenas facilitaria a validação e auditoria das decisões automatizadas, como também aceleraria a identificação de ameaças emergentes bem como reduziria os riscos de falsas classificações. Em ambientes corporativos, onde a segurança

da informação é uma prioridade estratégica, a capacidade de justificar o funcionamento interno de modelos preditivos é um fator-chave para sua aceitação e adoção (Gunning et al, 2017; Doshi-Velez e Kim, 2017).

Os estudos existentes sobre segurança em redes IoT através de IDS frequentemente concentram-se apenas no uso de aprendizado de máquina para identificar padrões de tráfego que caracterizam ameaças, sem preocupações com a transparência e com a interpretabilidade dos modelos preditivos utilizados. O uso de XAI tem sido observado em outras áreas. Por exemplo, Band et al. (2023) apresentam uma visão abrangente das aplicações de XAI na área de saúde. Černevičienė e Kabašinskas (2024) exploram o setor financeiro, enquanto Mankodiya et al. (2022) discutem o uso de XAI no setor automotivo.

Diante deste contexto, o presente estudo busca responder à seguinte questão de pesquisa: como a XAI explica a identificação do tráfego anômalo (intrusões) em redes IoT? O objetivo deste artigo é, portanto, investigar o uso de XAI em modelos de aprendizado de máquina treinados para identificar intrusões em rede IoT.

Três modelos de aprendizado de máquina cuja aplicação em IDS é reportada na literatura foram selecionados: *Random Forest* (RF), *LightGBM*, e *Multilayer Perceptron* (MLP). RF e LGBM foram utilizados por Arreche, Guntur e Abdallah (2024), enquanto MLP por Souza (2023). Estes três modelos foram treinados no conjunto de dados NSL-KDD (Unb, 2009), amplamente utilizado na literatura em problemas relacionados à segurança em redes IoT. O método Shapley *Additive Explanations* (SHAP) foi aplicado nestes modelos para explicar suas decisões através da identificação dos atributos relevantes às predições. A similaridade entre as explicações produzidas pelo SHAP nestes modelos foi verificada a partir da correlação de Spearman, sendo este um diferencial em relação aos trabalhos existentes que utilizam o mesmo conjunto de dados.

Os resultados obtidos evidenciam similaridade nas explicações dos modelos RF e LightGBM (Spearman 0,91), cujas predições são impactadas por atributos relacionados ao volume de tráfego. Já as predições do MLP são impactadas por atributos relacionados aos serviços da rede, afetando a similaridade de suas explicações em relação aos modelos RF (Spearman 0,60) e LightGBM (Spearman 0,67). Estes resultados evidenciam que, embora estes

Revista brasileira de iniciação científica
ISSN: 2359-232X

modelos preditivos sejam eficazes na detecção de intrusões, a natureza das suas explicações é variada.

Metodologia

O presente trabalho se caracteriza como uma pesquisa descritiva, que busca investigar o uso do método *Shapley Additive Explanations* (SHAP) para explicar as predições dos modelos preditivos considerados. De acordo com Wazlawick (2020), a pesquisa descritiva busca obter dados consistentes sobre determinada realidade, visando descrever os fatos como eles são e sendo caracterizada pelo levantamento de dados. O procedimento técnico adotado é um estudo de caso, considerando exclusivamente o conjunto de dados NSL-KDD e os modelos preditivos *Random Forest*, LightGBM, e *Multilayer Perceptron* (Wazlawick, 2020).

O fluxo geral de desenvolvimento do trabalho seguiu as etapas descritas na Figura 01. Este fluxo foi replicado para cada um dos modelos de aprendizado de máquina utilizados no presente estudo para investigar a explicabilidade dos modelos. A explicabilidade de cada modelo é investigada através da aplicação de uma camada de XAI que utiliza o método SHAP.

Figura 01: Fluxo de desenvolvimento do trabalho

Fonte: Adaptado de Arreche, Guntur e Abdallah (2024)

O conjunto de dados NSL-KDD (Unb, 2009) foi utilizado neste estudo. Este conjunto é amplamente reconhecido na literatura para utilização em problemas relacionados à segurança e redes IoT. O conjunto contém atributos que descrevem conexões de rede, além da classe, que indica se uma conexão é normal ou corresponde a algum tipo de ataque. Este conjunto de dados

Revista brasileira de iniciação científica
ISSN: 2359-232X

foi escolhido devido à sua popularidade. A relação dos atributos deste conjunto de dados é apresentada no Anexo I.

Embora existam trabalhos que utilizam o método SHAP em modelos de detecção de intrusão com o conjunto de dados NSL-KDD, a literatura carece de estudos que comparem sistematicamente a consistência das explicações SHAP entre modelos baseados em árvores e redes neurais. Este trabalho contribui nesse sentido, ao analisar quantitativamente a similaridade das explicações por meio da correlação de Spearman. Apesar de ser um conjunto de dados clássico, o NSL-KDD permanece relevante como *benchmark* acadêmico, permitindo a comparação controlada de técnicas de explicabilidade em um cenário amplamente documentado na literatura.

No total, o conjunto NSL-KDD contém 125.973 registros, dos quais 67.343 correspondem a tráfego de rede legítimo (normal) e 58.630 são classificados como algum tipo de ataque. Os registros de ataque são rotulados em categorias distintas, representando diferentes vetores de ameaça comumente observados em redes computacionais. A Tabela 1 apresenta a distribuição geral dos tipos de tráfego, estando segmentado nas categorias de ataque descritas a seguir.

Tabela 1 - Distribuição por tipo de tráfego no conjunto de dados NSL-KDD

TIPO DE TRÁFEGO	REGISTROS	PORCENTAGEM (%)
Normal	67.343	53,45%
DoS	45.927	36,46%
Probe	11.656	9,25%
R2L	995	0,79%
U2R	52	0,04%

Fonte: Elaborado pelos autores (2025)

- *Denial of Service (DoS).* Ataques de negação de serviço têm como objetivo tornar um recurso de rede indisponível para seus usuários legítimos, geralmente por meio do envio massivo de requisições. Exemplos clássicos dessa categoria no conjunto de dados incluem *smurf*, *neptune* e *back*. Estes ataques comprometem a disponibilidade, uma das principais dimensões da segurança da informação (Cloudflare, 2024).

- *Probe*. Ataques dessa categoria envolvem tentativas de coleta de informações sobre a rede, como varreduras de portas ou como identificação de dispositivos ativos. São considerados uma forma de reconhecimento e frequentemente utilizados como etapa preliminar a ataques mais invasivos. Exemplos incluem *satan, portsweep, nmap e ipsweep* (Khamphakdee, Benjamas e Saiyod, 2014).
- *Remote to Local (R2L).* Ataques R2L ocorrem quando um invasor externo tenta obter acesso a um sistema local, explorando vulnerabilidades em serviços de rede, geralmente por meio do envio de pacotes maliciosos. Esses ataques são mais difíceis de detectar, pois envolvem tráfego aparentemente legítimo. Exemplos incluem *guess_passwd, ftp_write, imap e warezclient* (Tavallaee et al., 2009).
- *User to Root (U2R).* Esse tipo de ataque envolve um usuário comum que já possui algum acesso ao sistema e tenta escalar privilégios para obter acesso como administrador. São ataques altamente críticos e de baixa frequência, geralmente realizados por meio de exploração de vulnerabilidades em processos do sistema. Exemplos incluem *buffer_overflow, perl, loadmodule e rootkit* (Cynet, 2025).

O pré-processamento dos dados foi realizado para viabilizar o treinamento dos modelos de aprendizado de máquina. No conjunto de dados, três colunas (p*rotocol_type, service* e *flag*) possuem valores categóricos. Essas variáveis foram codificadas numericamente por meio de um *LabelEncoder*, que converteu cada categoria em um valor inteiro. As variáveis numéricas foram padronizadas com um *StandardScaler*, que ajusta os dados para que tenham média igual a 0 e desvio padrão igual a 1. Estes mecanismos estão disponíveis na biblioteca Scikit-learn (2025).

Uma abordagem de classificação binária foi adotada para o treinamento dos modelos de aprendizado de máquina. Dessa forma, os tipos de tráfego DoS, Probe, R2L e U2R foram rotulados como *ataque*. O rótulo *normal* foi mantido para os demais registros. A biblioteca Pandas foi utilizada para realizar essa manipulação nos dados (Pandas, 2025).

A técnica de validação cruzada *hold-out* foi utilizada para mitigar eventual superadaptação. O conjunto de dados foi dividido em dois subconjuntos com a proporção de 80% para treinamento e 20% para teste. Esta operação foi realizada utilizando a função *train_test_split* da biblioteca Scikit-learn, que garantiu uma divisão estratificada dos registros.

Revista brasileira de iniciação científica
ISSN: 2359-232X

O conjunto de treinamento foi utilizado para o ajuste dos parâmetros internos dos modelos de aprendizado durante o treinamento. Além disso, no processo de treinamento ainda foram utilizados internamente uma parcela dos dados para validação, servindo para a escolha dos melhores hiperparâmetros. Já o conjunto de teste foi reservado exclusivamente para a avaliação final do desempenho dos modelos, assegurando que os resultados reflitam a capacidade de generalização para dados não vistos.

Os seguintes modelos de aprendizado de máquina foram adotados neste trabalho:

- *Random Forest* (RF). Modelo baseado em múltiplas árvores de decisão, reconhecido por sua robustez contra superadaptação e por lidar eficientemente com dados de alta dimensionalidade (Breiman, 2001). A implementação deste modelo foi inspirada no estudo de Arreche, Guntur e Abdallah (2024), que demonstrou sua eficácia em sistemas de detecção de intrusão com interpretabilidade das decisões.
- *LightGBM*. Modelo de gradient boosting otimizado para velocidade e desempenho em grandes volumes de dados tabulares (Ke et al., 2017). Este modelo foi adotado também com base no estudo de Arreche, Guntur e Abdallah (2024), reforçando seu potencial para a detecção eficiente de anomalias em redes com baixo custo computacional e alta precisão.
- *Multi-Layer Perceptron* (MLP). Rede neural com múltiplas camadas ocultas, capaz de modelar padrões não lineares complexos (Goodfellow, Bengio e Courville, 2016). A escolha do MLP foi fundamentada no trabalho relacionado de Souza (2023), que destaca a capacidade de aprendizado do MLP em contextos de segurança cibernética, embora seja sensível à escolha de hiperparâmetros e exigente em pré-processamento.

O desenvolvimento destes modelos foi realizado com as seguintes ferramentas: Scikit-learn (2025); Keras (2025); TensorFlow (2025); e LightGBM (2025). Além disso, as bibliotecas Pandas (2025) e NumPy (2025) foram utilizadas para manipulação de dados e a Matplotlib (Hunter, 2007) e Seaborn (Waskom, 2021) para visualizações.

O ajuste de hiperparâmetros foi realizado com *GridSearchCV*, mecanismo disponibilizado pela Scikit-learn que recebe como entrada um conjunto de valores de parâmetros, treina os modelos com as combinações destes valores e avalia o modelo para cada

combinação, usando o método de validação cruzada. Ao fim deste processo, o mecanismo *GridSearchCV* identificou quais valores de parâmetros obtiveram o melhor desempenho para cada modelo de aprendizado de máquina considerado, descritos a seguir.

A arquitetura do modelo RF, implementada neste trabalho, é composta por um conjunto de 200 árvores de decisão, treinadas de forma independente e combinadas para realizar a classificação. O treinamento foi realizado utilizando-se a classe *RandomForestClassifier* da Scikit-learn, adotando-se o parâmetro *class_weight* como *balanced_subsample* para ajustar automaticamente o peso das classes com base na distribuição, observada em cada amostra de treinamento. O parâmetro de profundidade máxima de cada árvore foi limitado a 15 níveis (*max_depth*=15), o que contribui para a generalização do modelo. Além disso, foram adotados os parâmetros *min_samples_split*=10 e *min_samples_leaf*=5, restringindo a divisão de nós e exigindo um número mínimo de amostras por folha, tornando a estrutura das árvores menos sensível a ruídos. O uso de um valor fixo para *random_state* garante a reprodutibilidade dos resultados, sendo utilizado o valor 0.

A arquitetura do modelo MLP foi desenvolvida com o uso das bibliotecas Keras e TensorFlow. O modelo é composto por duas camadas ocultas, densamente conectadas, cada uma contendo 41 neurônios, correspondendo à quantidade de atributos utilizados do conjunto de dados. Estas camadas utilizam a função de ativação *tanh*, que favorece a aprendizagem de representações não lineares. A camada de saída possui 2 neurônios e utiliza a função de ativação *softmax*. O treinamento do modelo foi realizado com o otimizador *Adam*, reconhecido por sua eficiência em problemas com grandes volumes de dados e com a função de perda *sparse categorical crossentropy*. O processo de ajuste dos pesos foi conduzido ao longo de 100 épocas, com *batch_size* de 32 e utilizando 20% dos dados de treinamento para validação.

O modelo LightGBM foi inicialmente configurado com os parâmetros padrão da biblioteca LightGBM (2025) e posteriormente passou por um processo de otimização de hiperparâmetros utilizando o *GridSearchCV*. Foram exploradas diferentes combinações de parâmetros como *num_leaves*, *max_depth*, *learning_rate*, *n_estimators* e *scale_pos_weight*, este último com o objetivo de ajustar o peso das classes desbalanceadas. Para a avaliação dos modelos durante o processo de busca, foi utilizada a métrica *F1-score*, que considera o desempenho equilibrado entre as classes. Após a otimização, o melhor estimador foi

selecionado e avaliado sobre o conjunto de validação. O *GridSearchCV* identificou como melhores parâmetros os seguintes valores: *learning_rate* de 0.05; *max_depth* de 10; *n_estimators* igual a 500; *num_leaves* de 31; e *scale_pos_weight* de 1.

A explicabilidade é investigada, neste trabalho, por meio de uma camada de XAI aplicada sobre cada modelo preditivo. Esta camada utiliza o método *Shapley Additive Explanations* (SHAP) para explicar as decisões destes modelos. O SHAP utiliza os valores de Shapley, que representam a contribuição de cada atributo na predição de uma classe específica (Lundberg e Lee, 2017). Valores SHAP positivos indicam que o atributo aumentou a probabilidade de predição da classe de interesse, enquanto valores negativos indicam uma redução nessa probabilidade. A magnitude do valor reflete o grau de influência do atributo na decisão (Awan, 2023).

A biblioteca SHAP de Lundberg (2018) foi adotada na implementação da camada de explicabilidade. Essa biblioteca oferece implementações do método SHAP, aplicáveis em modelos baseados em árvores e baseados em redes neurais. No modelo RF e no LightGBM, foi utilizada a implementação *TreeExplainer*, que se beneficia da estrutura em árvore dos modelos para calcular os valores SHAP eficientemente. Após o treinamento, os valores SHAP foram calculados para estes modelos, utilizando o conjunto de teste. Já no modelo MLP, foi utilizada a implementação *KernelExplainer* para calcular os valores SHAP, com base em amostras aproximadas. Foram utilizadas 100 amostras do conjunto de treino para inicializar o explicador e 1.000 instâncias de teste para o cálculo dos valores SHAP. A estrutura dos valores SHAP foi separada por classe e posteriormente utilizada para gerar gráficos de dispersão do tipo *beeswarm* para evidenciar a contribuição dos atributos nas predições.

A consistência, nas explicações fornecidas pelos diferentes modelos, foi avaliada através da correlação de Spearman entre as ordens de importância dos atributos, obtidos por meio dos valores SHAP. A correlação de Spearman é uma medida estatística que avalia o quanto dois conjuntos possuem uma ordem (ranking) semelhante (Scipy, 2024). Ela é computada comparando a posição relativa dos elementos nos diferentes conjuntos, permitindo verificar a similaridade de ordenação entre os elementos. Uma correlação de Spearman igual a 1.0 significa que a ordem dos elementos é a mesma nos dois conjuntos. Por outro lado, uma correlação igual a -1.0 indica que a ordem dos elementos de um conjunto é inversa do outro. A

Revista brasileira de iniciação científica
ISSN: 2359-232X

análise da correlação de Spearman, para comparar as importâncias SHAP, permite identificar a similaridade entre as explicações, proporcionadas pela camada XAI, em cada modelo. A próxima seção apresenta e discute os resultados de desempenho dos modelos preditivos considerados, bem como as explicações produzidas pelo método SHAP.

Análise dos Dados e Resultados

A avaliação dos modelos foi realizada por meio de métricas clássicas de desempenho em tarefas de classificação: acurácia, precisão, *recall* e *F1-score*. A Tabela 2 apresenta as métricas de desempenho obtidas por cada modelo no conjunto de testes. Estes resultados evidenciam o desempenho positivo dos modelos de aprendizado considerados.

Tabela 2 - Métricas de desempenho dos modelos

MODELO	PRECISÃO	*RECALL*	*F1-SCORE*	ACURÁCIA
RF	1,00	1,00	1,00	1,00
MLP	0,99	0,99	0,99	0,99
LightGBM	1,00	0,99	1,00	1,00

Fonte: Elaborado pelos autores (2025)

Cabe esclarecer que não é objetivo deste estudo comparar o desempenho e a generalização dos modelos, mas sim, sua explicabilidade através do método SHAP. Sendo assim, as métricas apresentadas na Tabela 2 têm caráter informativo, visando evidenciar a habilidade preditiva dos modelos, nos quais os valores SHAP foram calculados, e fortalecer a transparência do estudo e sua reprodutibilidade.

O método SHAP explica as decisões dos modelos preditivos através da importância dos atributos utilizados nas predições. Gráficos do tipo *beeswarm* são adotados para apresentar a importância das variáveis utilizadas. O uso deste tipo de gráfico é consolidado, na literatura, para apresentar os valores SHAP dos atributos (Lundberg et al., 2020). No gráfico *beeswarm,* o eixo das ordenadas apresenta os atributos, dispostos pela relevância (de cima para baixo). Já o eixo das abscissas apresenta o valor SHAP do atributo, ou seja, sua contribuição para a classificação do modelo. Valores SHAP positivos indicam que o atributo contribuiu para predição da classe *ataque*, enquanto valores negativos indicam que contribuiu para predição da

Revista brasileira de iniciação científica
ISSN: 2359-232X

classe *normal*. A cor de cada ponto no gráfico representa o valor do atributo correspondente, com vermelho, indicando valores altos para o atributo na instância, e com azul, indicando valores baixos.

A partir do gráfico beeswarm pode-se identificar quais atributos possuem maior intervalo de alcance de valores SHAP, sendo estes os atributos que mais contribuem para (e explicam) a decisão do modelo. Além disso, atributos que apresentam uma divisão mais clara de cores (vermelho e azul em lados opostos) são bons previsores: a alteração de seus valores influenciam (e explicam) a decisão do modelo. O Gráfico 1 apresenta os atributos mais relevantes identificados nos modelos investigados. A seguir, são apresentadas as análises por modelo.

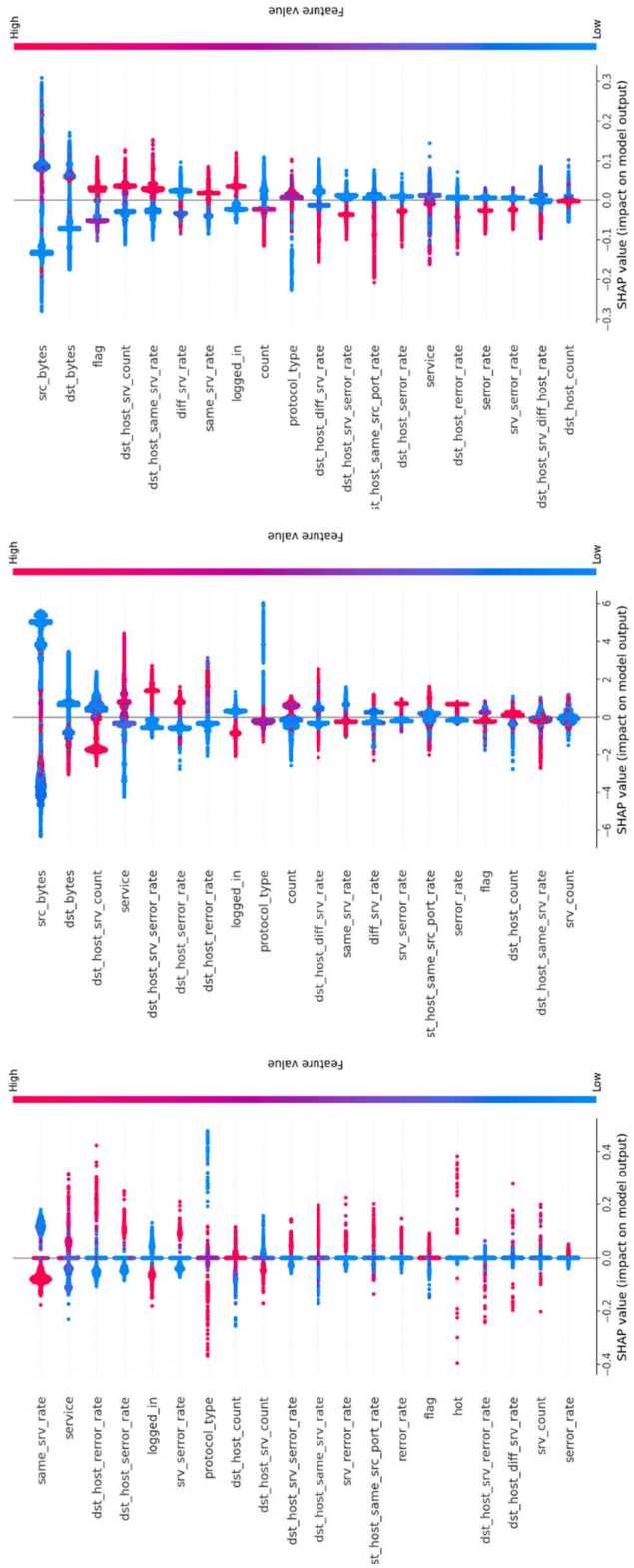
No modelo RF – Gráfico 1 (a) – verifica-se que *src_bytes, dst_bytes, flag, dst_host_srv_count, dst_host_same_srv_rate, diff_srv_rate, same_srv_rate, logged_in, count* e *protocol_type* foram os dez atributos mais relevantes para as predições.

Os maiores intervalos de alcance de valores SHAP são observados nos atributos *src_bytes* e *dst_bytes*, indicando que contribuem tanto para predição da classe *ataque,* quanto para a classe *normal*. Isso evidencia que a quantidade de informação, enviada da origem e recebida no destino, tem grande impacto nas predições. No entanto, a ausência de uma clara separação de cores vermelho e azul indica que a predição do modelo é afetada por outros atributos, pois valores altos (e baixos) de *src_bytes* e *dst_bytes* contribuem tanto para predição da classe *ataque* quanto da *normal*.

Nos atributos *flag*, *dst_host_srv_count* e *dst_host_same_srv_rate* observa-se uma separação de cores mais bem definida, indicando que valores altos desses atributos influenciam majoritariamente a predição da classe *ataque*, enquanto valores baixos influenciam a predição da classe *normal*. No caso do atributo *flag* (que representa se a conexão está normal ou se houve erro), valores que indicam presença de erro contribuem significativamente para a predição de *ataque*. No caso do atributo de *diff_srv_rate*, verifica-se que quanto mais baixo seu valor, ou seja, quanto mais conexões para os mesmos serviços e mesmo *host* de destino, maior a influência desse atributo para predição da classe *ataque*. Por fim, considerando o atributo *logged_in*, verifica-se que usuários autenticados (*logged_in* = 0) influenciam a predição da classe *ataque* do modelo.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Gráfico 1 - Valores SHAP para os modelos investigados



(a) Modelo RF

(b) Modelo LightGBM

(c) Modelo MLP

Fonte: Elaborado pelos autores (2025)

Revista brasileira de iniciação científica
ISSN: 2359-232X

Já no modelo LightGBM – Gráfico 1 (b) – observa-se que os dez atributos mais relevantes foram: *src_bytes, dst_bytes, dst_host_srv_count, service, dst_host_srv_serror_rate, dst_host_serror_rate, dst_host_rerror_rate, logged_in, protocol_type* e *count*. Da mesma forma que no modelo RF, o atributo *src_bytes* também é o que apresentou maior alcance de valores SHAP, indicando que, no LightGBM este atributo também contribui tanto para predições da classe *ataque,* quanto da classe *normal*. Porém, em relação ao atributo *dst_bytes*, verifica-se uma melhor separação das cores vermelho e azul, indicando que valores altos deste atributo contribuem para a predição da classe *normal* e valores baixos, para a classe *ataque*. Em relação ao atributo *dst_host_srv_count,* verifica-se separação de cores mais bem definida, indicando que valores baixos deste atributo contribuem majoritariamente para predições da classe *ataque*, enquanto valores altos, ou seja, um alto número de conexões do dispositivo com o mesmo número de porta, contribuem para predições da classe *normal*.

Um atributo importante no modelo LightGBM foi o *service*. Valores altos, neste atributo, correspondem a tráfegos de rede com serviços dos tipos *whois* (consulta de informações sobre domínios e IPs), *vmnet* (conexões de redes virtuais) *uucp_path* e *uucp* (associados à transferência de arquivos e mensagens entre sistemas Unix). Estes valores altos exercem influência relevante na predição da classe *ataque*.

Ainda, é possível notar uma separação clara no atributo *dst_host_srv_serror_rate*, que corresponde à porcentagem de conexões que ativaram certas *flags* de conexão. Valores altos, ou seja, alta taxa de conexões que ativaram essas *flags*, influenciaram as predições do modelo para a classe *ataque*.

Quanto às diferenças entre os modelos RF e LightGBM, cabe destacar o atributo *logged_in*. No RF, *logged_in* com valor baixo (ou seja, zero, indicando usuário não autenticado), influencia a predição da classe *normal*, enquanto, no LightGBM, a predição da classe *normal* é influenciada por *logged_in* com valor alto (ou seja, um, indicando usuário autenticado). Já o atributo *count* (que representa o número de conexões com o mesmo *host* de destino nos últimos dois segundos), quando com valores altos, indicando alto número de conexões, influenciou predições do modelo para a classe *ataque*, enquanto valores baixos contribuíram, de uma forma sutil, para predições da classe *normal*. Esse comportamento sugere que um volume elevado de conexões, em um curto intervalo de tempo, pode ser interpretado

Revista brasileira de iniciação científica
ISSN: 2359-232X

como um possível indicativo de ataque, como em varreduras de porta ou tentativas de negação de serviço. A sensibilidade a esse atributo evidencia sua importância na detecção de padrões de tráfego automatizado.

Por fim, o modelo MLP – Gráfico 1 (c) – apresentou como atributos mais relevantes: *same_srv_rate, service, dst_host_rerror_rate, dst_host_serror_rate, logged_in, srv_serror_rate, protocol_type, dst_host_count, dst_host_srv_count* e *dst_host_srv_serror_rate*.

Diferentemente dos modelos RF e LightGBM, o MLP apresentou, em suas explicações, o atributo *same_srv_rate* (taxa de conexões para um mesmo serviço) como sendo o que mais influencia as predições. Valores altos deste atributo contribuíram para predição da classe *normal*, enquanto valores baixos contribuíram para predição da classe *ataque*.

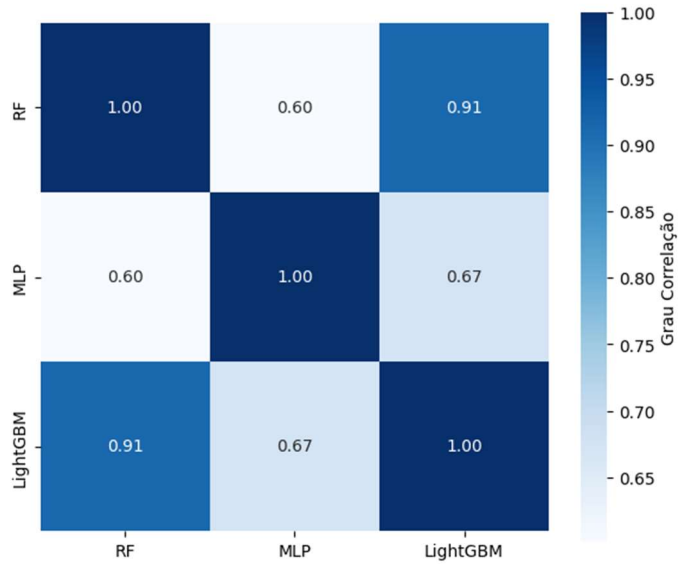
Da mesma forma que no modelo LightGBM, o atributo *service*, quando possui valores altos, influencia na predição da classe *ataque*. O atributo *logged_in* novamente apareceu entre os atributos mais importantes para as predições do modelo. Em concordância com o modelo LightGBM, nas explicações do modelo MLP, é possível verificar que, quando este atributo possui valor baixo (usuário não autenticado), ele influencia a predição da classe *ataque*.

No caso do atributo *dst_host_count* (conexões com o mesmo endereço IP do host de destino), observa-se que valores altos influenciam a predição da classe *ataque*. Contudo, na grande maioria das vezes, este atributo não impactou o modelo, conforme é observado pela alta concentração de valores SHAP em zero.

A correlação de Spearman, entre as ordens de importância consideradas pelos modelos, é apresentada no Gráfico 2. Observa-se alta correlação (0,91) entre os modelos RF e LightGBM, o que indica que eles apresentam forte similaridade na ordem de importância dos atributos. Esse resultado é esperado, pois estes modelos pertencem à família de métodos baseados em árvores de decisão. Isso faz com que eles capturem relações estruturais semelhantes no conjunto de dados, resultando em grande semelhança na ordem de importância dos atributos

Revista brasileira de iniciação científica
ISSN: 2359-232X

Gráfico 2 - Correlação de Spearman entre as ordens de importância dos atributos



Fonte: Elaborado pelos autores (2025)

Por outro lado, as correlações envolvendo o MLP são moderadas (0,60 com RF e 0,67 com LightGBM). Isso evidencia que, embora exista alguma concordância entre os modelos acerca dos atributos importantes para a decisão, o MLP tende a ordená-los de maneira diferente. Isso pode ser efeito da natureza distinta da arquitetura do modelo MLP. Por ser baseado em uma rede neural com múltiplas camadas ocultas, os padrões e relações entre os atributos são aprendidos de forma diferente das árvores de decisão.

Do ponto de vista de interpretabilidade, a forte correlação entre RF e LightGBM reforça a robustez das explicações, geradas por métodos baseados em árvores. Já as divergências, observadas no MLP, evidenciam que diferentes paradigmas de aprendizado podem produzir perspectivas explicativas complementares sobre o mesmo domínio de dados.

É importante esclarecer que os elevados valores da correlação de Spearman observados não dependem apenas da arquitetura dos modelos. Há influência da estratégia de particionamento utilizada no processo de treinamento e teste, e há influência também das características do conjunto de dados.

Conforme previamente descrito, na seção de metodologia, o presente estudo adotou a técnica *hold-out* para o treinamento e teste dos modelos de aprendizado. Para mitigar o efeito dessa técnica nos resultados, a separação em conjuntos de treinamento e teste manteve a

estratificação dos dados. Isto garantiu, nesses conjuntos, a mesma proporção de dados para os diferentes tipos de ataque, minimizando o efeito da super ou subadaptação dos modelos.

O conjunto de dados NSL-KDD contém atributos classicamente muito relevantes para detecção de intrusão, tais como duração da conexão, tipo de serviço, autenticação e estatísticas de tráfego de rede. Estes atributos possuem forte capacidade de separação entre tráfego normal e ataques. Além disso, a literatura aponta que este conjunto de dados possui padrões relativamente estruturados de ataque e comportamento de rede (Aggarwal e Sharma, 2015). Consequentemente, diferentes modelos de aprendizado podem identificar os mesmos atributos como relevantes, levando a uma convergência na ordem de importância revelada pelo método SHAP, o que aumenta a correlação de Spearman.

Considerações Finais

Este trabalho teve como diretriz a seguinte pergunta de pesquisa: como a inteligência artificial explicável (XAI) explica intrusões detectadas com aprendizado de máquina? A partir dessa pergunta, o objetivo do estudo foi investigar o uso de XAI em modelos de aprendizado de máquina treinados para identificar tráfego anômalo (intrusões) em rede IoT. O objetivo foi alcançado através da aplicação do método *Shapley Additive Explanations* (SHAP) em três modelos de aprendizado supervisionado: *Random Forest* (RF), *Multilayer Perceptron* (MLP) e LightGBM. Estes modelos de aprendizado foram treinados no conjunto de dados NSL-KDD, obtendo métricas de acurácia, precisão, *recall* e *F1-score* superiores a 0.99. Após treinados, o método SHAP foi aplicado para revelar os atributos que cada modelo considerou relevantes para as predições de tráfego *normal* ou *ataque*.

Os resultados evidenciaram que os modelos baseados em árvores (RF e LightGBM) apresentaram explicações mais estáveis e consistentes entre si, com alta correlação de Spearman entre a ordem dos atributos relevantes. Já o MLP, embora eficaz em desempenho preditivo, diverge do RF e LightGBM quanto aos atributos relevantes às predições. Os resultados obtidos conduzem para a seguinte resposta à questão de pesquisa: nos modelos RF e LightGBM os atributos relacionados ao volume de dados, têm grande influência nas predições, enquanto, no MLP, há maior relevância dos atributos que descrevem o padrão dos serviços.

O estudo apresenta limitações que afetam a generalização dos resultados. O uso unicamente do conjunto de dados NSL-KDD, embora relevante como *benchmark* na literatura, não representa, por completo, cenários atuais de redes IoT. A utilização da validação cruzada *hold-out,* em uma única execução do processo de treinamento e teste, pode limitar a robustez dos resultados. Por fim, as correlações de Spearman reveladas podem ter sido influenciadas pelas características do conjunto de dados. Neste sentido, a interpretação dos resultados deve se limitar ao procedimento e conjunto de dados utilizado no trabalho. Estudos adicionais precisam ser conduzidos para evidenciar generalização ampla dos resultados.

Com base nos resultados obtidos, algumas direções podem ser consideradas em trabalhos futuros. Uma delas é a avaliação da camada de explicabilidade, em um ambiente real de produção, com o objetivo de analisar seu desempenho em situações concretas e identificar possíveis ajustes necessários. Essa etapa é fundamental para validar a efetividade do sistema fora do ambiente controlado de testes.

Outra possibilidade é o desenvolvimento de uma interface interpretativa voltada para operadores de segurança, capaz de traduzir as saídas geradas pelo SHAP em textos claros e acessíveis. Essa interface teria como objetivo facilitar a compreensão por parte de analistas e profissionais de segurança da informação, mesmo por parte daqueles que não possuem conhecimento técnico aprofundado em modelos de aprendizado de máquina e métodos de explicabilidade.

Por fim, destaca-se a importância de realizar uma validação com profissionais da área de segurança da informação. Uma avaliação qualitativa, com especialistas, que atuam diretamente com redes IoT, pode ajudar a verificar se as explicações fornecidas pelo modelo contribuem para a rotina de análise, tomada de decisão e resposta a incidentes. Essa visão de especialistas é essencial para garantir a aplicabilidade prática de XAI.

Agradecimentos

Fernando Santos agradece a Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) pelo apoio financeiro recebido (TO2023TR246).

Revista brasileira de iniciação científica
ISSN: 2359-232X

Referências

AGGARWAL, P.; SHARMA, S. K. Analysis of KDD Dataset Attributes – Class wise for Intrusion Detection. **Procedia Computer Science**, Amsterdã, v. 57, p. 842–851, 2015.

ALI, Sajid et al. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. **Information fusion**, Amsterdã, v. 99, p. 101805, 2023.

ARRECHE, Osvaldo; GUNTUR, Tanish; ABDALLAH, Mustafa. Xai-ids: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. **Applied Sciences**, Basel, Switzerland, v. 14, n. 10, p. 4170, 2024.

AWAN, Abid Ali. **An Introduction to SHAP Values and Machine Learning Interpretability**. 2023. Disponível em: https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability. Acesso em: 23 dez. 2025.

BAND, Shahab S. et al. Application of explainable artificial intelligence in medical health: A systematic review of interpretability methods. **Informatics in Medicine Unlocked**, Amsterdã, v. 40, p. 101286, 2023.

BREIMAN, Leo. Random forests. **Machine learning**, Berlim, v. 45, n. 1, p. 5–32, 2001.

ČERNEVIČIENĖ, Jurgita; KABAŠINSKAS, Audrius. Explainable artificial intelligence (XAI) in finance: a systematic literature review. **Artificial Intelligence Review**, Cham, v. 57, n. 8, p. 216, 2024.

CLOUDFLARE. **O que é um ataque de negação de serviço (DoS)?**. 2024. Disponível em: https://www.cloudflare.com/pt-br/learning/ddos/glossary/denial-of-service. Acesso em: 23 dez. 2025.

CYNET. **Understanding Privilege Escalation and 5 Common Attack Techniques**. 2025. Disponível em: https://www.cynet.com/network-attacks/privilege-escalation. Acesso em: 23 dez. 2025.

DOSHI-VELEZ, Finale; KIM, Been. Towards a rigorous science of interpretable machine learning. **arXiv preprint**, New York, n. 1702.08608, 2017. Disponível em: https://arxiv.org/abs/1702.08608. Acesso em: 07 jan. 2026.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge: MIT Press, 2016.

GUNNING, David et al. XAI - Explainable artificial intelligence. **Science robotics**, Washington, v. 4, n. 37, p. eaay7120, 2019.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, Piscataway, v. 9, n. 3, p. 90–95, 2007.

KE, Guolin et al. Lightgbm: a highly efficient gradient boosting decision tree. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31, 2017, Red Hook. **NEURIPS**. 2017. p. 3149–3157.

KERAS. Keras: **Deep Learning for humans**. 2025. Disponível em: https://keras.io/. Acesso em: 23 dez. 2025.

KHAMPHAKDEE, Nattawat; BENJAMAS, Nunnapus; SAIYOD, Saiyan. Improving intrusion detection system based on snort rules for network probe attack detection. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY, 2, 2014, Bandung. **ICOICT**. 2014, p. 69-74.

LIGHTGBM. **LightGBM 4.6.0 documentation**. 2025. Disponível em: https://lightgbm.readthedocs.io/en/stable/. Acesso em: 23 dez. 2025.

LUNDBERG, Scott M.; LEE, Su-In. A unified approach to interpreting model predictions. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31, 2017, Red Hook. **NEURIPS**. 2017. p. 4768–4777.

LUNDBERG, Scott M. et al. From local explanations to global understanding with explainable AI for trees. **Nature machine intelligence**, Basingstoke, UK, v. 2, n. 1, p. 56-67, 2020.

MANKODIYA, Harsh et al. OD-XAI: Explainable ai-based semantic object detection for autonomous vehicles. **Applied Sciences**, Basel, v. 12, n. 11, p. 5310, 2022.

LUNDBERG, Scott. **Welcome to the SHAP documentation**. 2018. Disponível em: https://shap.readthedocs.io/en/latest/index.html. Acesso em: 23 dez. 2025.

NUMPY. **NumPy**. 2025. Disponível em: https://numpy.org/. Acesso em: 23 dez. 2025.

PANDAS. Pandas **- Python Data Analysis Library**. 2025. Disponível em: https://pandas.pydata.org/. Acesso em: 23 dez. 2025.

SCIKIT-LEARN. **Scikit-learn: machine learning in Python — scikit-learn 1.7.0 documentation**. 2025. Disponível em: https://scikit-learn.org/stable/. Acesso em: 23 dez. 2025.

SCIPY. **Spearman - Scipy**. 2024. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html. Acesso em: 07 jan 2026.

SOUZA, Cristiano Antônio de. **Abordagem para detecção e prevenção de intrusão em computação de nevoeiro e IoT**. Tese (Doutorado em Ciência da Computação). Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2023.

STATISTA**. IoT connections and applications.** In: Internet of Things (IoT) - statistics & facts. 2024. Disponível em: https://www.statista.com/topics/2637/internet-of-things/#topicOverview. Acesso em: 23 dez. 2025.

TAVALLAEE, Mahbod et al. **A detailed analysis of the kdd cup 99 data set**. In: IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR SECURITY AND DEFENSE APPLICATIONS, 2, 2009, Ottawa. CISDA. 2009. p. 1–6.

Revista brasileira de iniciação científica
ISSN: 2359-232X

TENSORFLOW. **TensorFlow**. 2025. Disponível em: https://www.tensorflow.org. Acesso em: 23 dez. 2025.

UNB, University of New Brunswick. **ISCX NSL-KDD dataset 2009**. 2009. Disponível em: https://www.unb.ca/cic/datasets/nsl.html. Acesso em: 23 dez. 2025.

WASKOM, Michael L. Seaborn: statistical data visualization. **Journal of Open Source Software**, Portland, v. 6, n. 60, p. 3021, 2021.

WAZLAWICK, Raul S. **Metodologia de Pesquisa para Ciência da Computação**. 3. ed. Rio de Janeiro: GEN LTC, 2020.

Revista brasileira de iniciação científica
ISSN: 2359-232X

Anexo I

O conjunto de dados NSL-KDD é apresentado na obra de Tavallaee et al. (2009) e disponibilizado em Unb (2009). A **Erro! Fonte de referência não encontrada.** apresenta os atributos deste conjunto de dados.

Tabela 3. Atributos do conjunto de dados NSL-KDD

Atributo	Tipo	Descrição
duration	Numérico	Duração da conexão (segundos)
protocol_type	Categórico	Tipo de protocolo (tcp, udp, icmp)
service	Categórico	Serviço de rede solicitado (http, telnet, ftp, etc.)
flag	Categórico	Estado da conexão TCP
src_bytes	Numérico	Bytes enviados da origem para o destino
dst_bytes	Numérico	Bytes enviados do destino para a origem
land	Binário	1 se origem e destino são iguais; 0 caso contrário.
wrong_fragment	Numérico	Número de fragmentos incorretos
urgent	Numérico	Pacotes com o bit urgente ativado
hot	Numérico	Número de comandos críticos
num_failed_logins	Numérico	Tentativas de login falhas
logged_in	Binário	1 se login foi bem-sucedido; 0 caso contrário
num_compromised	Numérico	Número de condições comprometidas
root_shell	Binário	1 se shell root foi obtido; 0 caso contrário
su_attempted	Binário	1 se houve tentativa de comando “su root”; 0 caso contrário
num_root	Numérico	Número de acessos root
num_file_creations	Numérico	Número de arquivos criados
num_shells	Numérico	Número de shells abertos
num_access_files	Numérico	Número de acessos a arquivos críticos
num_outbound_cmds	Numérico	Número de comandos outbound (sempre 0 no dataset)
is_host_login	Binário	1 se login foi feito como host
is_guest_login	Binário	1 se login foi feito como convidado
count	Numérico	Número de conexões para o mesmo host nos últimos 2 seg
srv_count	Numérico	Número de conexões para o mesmo serviço nos últimos 2 seg
serror_rate	Numérico	Taxa de erros SYN
srv_serror_rate	Numérico	Taxa de erros SYN por serviço
rerror_rate	Numérico	Taxa de erros REJ
srv_rerror_rate	Numérico	Taxa de erros REJ por serviço
same_srv_rate	Numérico	Taxa de conexões para o mesmo serviço
diff_srv_rate	Numérico	Taxa de conexões para serviços diferentes
srv_diff_host_rate	Numérico	Taxa de conexões para hosts diferentes
dst_host_count	Numérico	Número de conexões para o mesmo destino
dst_host_srv_count	Numérico	Número de conexões para o mesmo serviço no destino
dst_host_same_srv_rate	Numérico	Taxa de conexões para o mesmo serviço no destino
dst_host_diff_srv_rate	Numérico	Taxa de conexões para serviços diferentes no destino
dst_host_same_src_port_rate	Numérico	Taxa de conexões para a mesma porta de origem
dst_host_srv_diff_host_rate	Numérico	Taxa de conexões para diferentes hosts no mesmo serviço
dst_host_serror_rate	Numérico	Taxa de erros SYN no destino
dst_host_srv_serror_rate	Numérico	Taxa de erros SYN por serviço no destino
dst_host_rerror_rate	Numérico	Taxa de erros REJ no destino
dst_host_srv_rerror_rate	Numérico	Taxa de erros REJ por serviço no destino
class	Categórico	Classificação do tráfego (ataque ou normal)
difficulty_level	Discreto	Nível de dificuldade de detecção do evento

Fonte: adaptado de Tavallaee et al. (2009) e Unb (2009).