DOI: 10.36558/rsc.v12i2.7705

Análise de Desempenho em Algoritmos de Aprendizagem de Máquina na Detecção de Intrusão Baseada em Fluxo de Rede usando o Conjunto de Dados UNSW-NB15

Performance Analysis of Machine Learning Algorithms for Network Flow-based Intrusion Detection using the UNSW-NB15 Dataset

Welton T. M. Sousa
Instituto Federal de Minas Gerais
CEP: 34590-390, Sabará,
MG, Brasil
++55 31 2102-9370
weltonthiago
@gmail.com

Carlos A. Silva
Instituto Federal de Minas Gerais
CEP: 34590-390, Sabará,
MG, Brasil
++55 31 2102-9370
carlos.silva
@ifmg.edu.br

ABSTRACT

This work aims to research and analyze network flow intrusion detection using seven machine learning algorithms. A classic offline literature database (UNSW-NB15) was used for the computer simulation. In general, the algorithms obtained satisfactory results regarding the values of the metrics used and the computational time spent, contributing to the mitigation of cyberattacks on computer networks, whose relevance is essential for the security of computer systems.

CCS Concepts

•**Networks** → **Network performance analysis;** •**Computing methodologies** → **Feature selection;** •**Security and privacy** → **Network security;**

Keywords

Machine Learning; Network Flow; Intrusion Detection System; Cyberattack; Network Security

RESUMO

Este trabalho tem como objetivo a pesquisa e análise de detecção de intrusão em fluxo de rede utilizando nove algoritmos de aprendizado de máquina. Foi utilizada uma clássica base de dados offline da literatura (UNSW-NB15) para a simulação computacional. Em geral, os algoritmos obtiveram resultados satisfatórios quanto aos valores das métricas utilizadas e tempo computacional despendido, contribuindo para a mitigação de ciberataques nas redes de computadores, cuja relevância é primordial no sentido de endossar a segurança dos sistemas computacionais.

Palavras-chave

Aprendizado de Máquina; Fluxo de Rede; Sistema de Detecção de Intrusão; Ciberataque; Segurança de Rede

1. INTRODUÇÃO

A ascensão à informação não autorizada configura um grave problema nas organizações, isso ocorre em alguns casos devido a ações de *crackers*, *malwares* ou *ransomware*, cujo problema ocasiona indisponibilidade no acesso a informação. Em um contexto competitivo em que a informação é o principal fator na tomada de decisão, seja em instituições públicas ou privadas, é importante assegurar disponibilidade, autenticidade e integridade nos acessos aos recursos computacionais, *on-premise* ou *cloud*, por meio das redes de computadores.

Segundo o relatório anual da *IBM security*[1], publicado em 24 de fevereiro de 2021, os ataques cibernéticos às redes corporativas ampliaram consideravelmente nos segmentos de saúde, manufaturas e energia em relação ao ano de 2020, decorrente da exploração de vulnerabilidades. Os principais ciberataques foram *phishing*, *ransomware* e *DDoS* com a finalidade de parar os serviços por um determinado tempo, conforme abordado em [22], por consequência, ocasionando indisponibilidade e perdas financeiras [3].

Os administradores de redes são responsáveis por atuar no gerenciamento e projeto dos recursos computacionais das organizações, garantindo o acesso ao conteúdo e minimizando risco à segurança da informação. Nesse sentido, o monitoramento de rede é indispensável para a obtenção de métricas de desempenho, ou seja, analisar o tráfego de rede é importante na identificação de comportamentos anômalos.

Segundo [25], intrusão ou ataque é caracterizado pela investida nociva em recursos computacionais com o propósito de comprometer a integridade, confidencialidade ou disponibilidade dos recursos, obtendo êxito ou não. Mediante a ne-

[1]https://www.ibm.com/blogs/ibm-comunica/ibm-security-ataques-ciberneticos/

cessidade de reconhecer e mitigar ataques, foram projetados os Sistemas de Detecção de Intrusão, ou seja, IDS (*Intrusion Detection System*) conforme abordado por [2].

Neste trabalho buscou-se avaliar a efetividade na detecção de intrusão no tráfego de rede através do fluxo fundamentado na base de dados *offline* **UNSW-NB15**, delimitando-se a classificação do fluxo de rede normal e anômalo, sem categorizar o tipo de ataque, pois em cenário *Zero Day* não é possível rotulação. Diante disso realizou-se a análise da eficácia da classificação dos algoritmos de aprendizado de máquina por intermédio da avaliação das métricas de desempenho. Espera-se, portanto, que esse instrumento de reconhecimento do comportamento incomum no tráfego de rede *offline* possa auxiliar no desenvolvimento de uma futura ferramenta na tomada de decisão pelos administradores de redes.

Desse modo, para alcançar o objetivo central desse artigo, o texto encontra-se organizado da seguinte forma: Na seção 1 é feita a introdução da temática abordada no trabalho. Na seção 2, é retratada a fundamentação teórica para compreensão deste trabalho. Na seção 3, relevantes trabalhos da literatura relacionados ao tema de pesquisa são apresentados. Na seção 4, a metodologia empregada é descrita. Na seção 5 é detalhada cada etapa do desenvolvimento realizado, desde a base de dados e o seu tratamento até o implementação e aplicação dos algoritmos propostos, bem como as análises e discussões a respeito dos resultados obtidos. E por fim, na seção 6 são apresentadas as conclusões finais e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Tráfego de Rede e Fluxo de Rede

De acordo com [26], redes de computadores podem ser compreendidas como o agrupamento de *hosts* autônomos interconectados por única tecnologia mutuamente transferindo dados. Por outro lado, [13] apresenta as características básicas referentes aos *softwares*, *hardwares* e protocolos de comunicação com relação a infraestrutura das redes em proverem serviços, isso pode ser entendido pela abordagem apresentada por [26] no modelo cliente/servidor no qual o cliente requisita um serviço e o servidor provê.

O tráfego de rede consiste na intercomunicação entre cliente/servidor através de um meio físico utilizando protocolos de comunicação, ou seja, TCP/IP. Segundo [15] o fluxo de rede é compreendido pela agregação do endereço IP de origem/destino, porta de origem/destino e protocolo.

A análise do tráfego de rede pode ser caracterizada em duas formas: *offline* no qual é capturado o tráfego e salvo em um arquivo para posterior análise; ou *online* sendo capturada e analisada simultaneamente [4]. Diante disso, a detecção de intrusão no tráfego de rede tem por objetivo classificar comportamentos destoantes que possam afetar a autenticidade, integridade e disponibilidade dos recursos computacionais [5].

2.2 Aprendizado de Máquina

[21] conceitua aprendizado de máquina como técnicas computacionais de identificação e classificação de padrões com a capacidade de aprendizado automático, onde são capazes de inferir conhecimento baseado em um conjunto de atributos a partir do treinamento do modelo. Os algoritmos de aprendizado de máquina podem ser catalogados em 3 (três) tipos:

- **Não Supervisionado**: Consiste em inferir padrões embasados nos atributos de entrada sem a rotulação dos dados para treinamento, resultando em agrupamentos ou *clusters* [17].
- **Supervisionado**: Consiste em inferir padrões fundamentados nos atributos de entrada com a rotulação dos dados para treinamento, resultando em um classificador [17].
- **Reforço**: Consistem em orientar o aprendizado com recompensa positivamente quando ocorre o acerto e penalização quando ocorre erro, na categorização de um problema alvo [24].

2.3 Métricas de Classificação

Os critérios da avaliação dos algoritmos de aprendizado de máquina implementados nesse trabalho foram utilizados em duas circunstâncias: análise da classificação das melhores *features* e treinamento/predição dos modelos computacionais. A seguir são descritas as métricas de classificação utilizadas.

- κ **(coeficiente kappa de Cohen)**: consiste em uma medida estatística de confiabilidade entre dois avaliadores (juízes) na concordância de um ponto específico, em outras palavras, no contexto desse trabalho avalia o grau de credibilidade da seleção das *features*, cujo valor máximo é 100% [14].
- **Acurácia**: corresponde dentre todas as classificações preditas que o modelo realizou, quantas classificou corretamente, cujo valor máximo é 100% [1].
- **Precisão**: compreende dentre todas as classificações da classe positiva que o modelo realizou, quantas estão corretas, cujo valor máximo é 100% [1].
- **Sensibilidade**: representa a porcentagem da eficácia do modelo treinado em predizer a classe positiva, ou melhor, dadas todas as observações positivas da base de dados UNSW-NB15 *Training* e *Testing* quantas discerniu como positiva, cujo valor máximo é 100% [1].
- **F1-score**: constitui-se como medida harmônica entre **Precisão** e **Sensibilidade**, aplicado em bases de dados cujas classes estão desbalanceadas, com valor máximo de 100% [1].
- **AUC** (*Area Under the ROC Curve*) : em tradução literal "área sobre a curva", baseada na curva ROC (*Receiver Operating Characteristic Curve*) é uma métrica utilizada na classificação onde as classes estão desbalanceadas, das quais as previsões aleatórias representam o valor 0,5. Quanto mais próximo do valor 1, indica que o modelo treinado está predizendo de forma correta [6].
- **FAR** (*False Acceptance Rate*): Reflete a taxa de falsa aceitação, no qual é calculado pela razão entre o número de conexões normais que são classificadas incorretamente e o total de conexões normais. Portanto, quanto menor o valor de FAR, melhor o resultado. Valores abaixo de 10% são considerados resultados promissores [7].

3. TRABALHOS RELACIONADOS

Nessa seção são apresentados os trabalhos da literatura, cujos autores aplicaram aprendizado de máquina para classificação do fluxo de rede na base de dados **UNSW-NB15**, com o propósito de diferenciar fluxo normal e anômalo.

Em seu estudo [18] os autores utilizaram a técnica de seleção de *features*, *Association Rule Mining*, cujo método consiste na avaliação de dois ou mais *features* da base de dados, agrupando as melhores na etapa de pré-processamento, diminuindo o número de *features*. Após a seleção das melhores *features* conduziu-se o treinamento do modelo com o algoritmo *Naive Bayes* cuja métrica de desempenho obtida foram acurácia 37,5% e FAR 62,6%. Utilizando o algoritmo *Expectation–Maximization* foi obtida a acurácia 23,8% e FAR 75,8% para classificação binária da base de dados **UNSW-NB15** *Testing*, ou seja, indicando fluxo normal e anômalo sem levar em consideração os tipos de ciberataque.

Segundo [20] em seu trabalho, para a seleção dos atributos na base de dados **UNSW-NB15** foram utilizadas técnicas de análise estatística e correlação. Posteriormente realizou-se o treinamento com os algoritmos de aprendizado de máquina *Decision Tree* com respectiva acurácia de 85,5% e FAR 15,8%, *Linear Regression* com acurácia 83,1% e FAR 18,5%, *Naive Bayes* com acurácia 82,1% e FAR 18,5%, *Artificial Neural Network* com acurácia 81,3% e FAR 21,1% e *Expectation–Maximization* com acurácia 78,5% e FAR 23,8%.

Em [9] é proposto a utilização do WEKA, no qual vários métodos e algoritmos são implementados, utilizando especificamente os métodos: *CfsSubsetEval*, *GreedyStepwise*, *InfoGainAttributeEval* e *Ranker* em conjunto com o algoritmo *Random Forest* para seleção de atributos na base de dados **UNSW-NB15**. As métricas para avaliação da seleção de atributos foram acurácia "Instância Classificada Corretamente" com 75,7% e κ (coeficiente *kappa* de Cohen) com 82,9% utilizada para mensurar a concordância entre a categorização predita e a esperada na base de dados *Training* e acurácia "Instância Classificada Corretamente" com 76,4% e κ com 81,6% na base de dados *Testing*.

[10] discorre sobre a classificação binária do conjuntos de dados **UNSW-NB15**, ou seja, a classificação do tráfego baseada em fluxos de rede normais e anômalos implementada na linguagem Java, realizando a transformação dos atributos fundamentada na escala logarítmica, na etapa de pré-processamento. Realizou-se a implementação do algoritmo *Support Vector Machine* com o parâmetro de *kernel RBF* e utilizou-se a validação cruzada para treinamento do modelo, alcançando os seguintes resultados na base de dados *Testing*, acurácia 85,9% e FAR 15,3% .

Conforme [16], apresentou a abordagem de eliminação da *feature service* na base de dados **UNSW-NB15**, posteriormente utilizado o algoritmo *Random Forest* com *10-Fold Cross Validation* para seleção das 5 (cinco) principais *features* no procedimento de pré-processamento. Sucessivamente conduziu-se na implementação do *Support Vector Machine* cuja métrica obtida para classificação binária foi acurácia 82,11% na base de dados *Testing*.

O trabalho de [8], propôs a exclusão de vários atributos dentre eles: *proto*, *service*, *attack_cat*, *stime*, *ltime* e utilização do método *ensemble Extreme Gradient Boosting - XGBoost* na seleção de *features* baseado na base de dados **UNSW-NB15** no pré-processamento. Foram catalogadas 23 (vinte e três) atributos com os melhores resultados pelo algoritmo *XGBoost*. Posteriormente foi realizado o treinamento do modelo com *XGBoost* e aplicação da função *train_test_split*, com respectivamente 70% de treinamento e 30% de avaliação, obtendo a acurácia 75,88% na base de dados *Testing* para classificação multivariada, no qual o modelo também consegue identificar o tipo de ataque.

Por outro lado o [11], propôs a eliminação dos atributos: *ltime*, *stime*, *sport* e utilização dos algoritmos *Recursive Feature Elimination* com o algoritmo *Random Forest*, ou seja, RFE/RF na seleção de atributos baseado no *dataset* **UNSW-NB15**. Através do modelo proposto foram obtidas 4 (quatro) *features* com acurácia 98%. Em seguida dirigiu-se para implementação da *Artificial Neural Network - ANN*, para classificação do *dataset* **UNSW-NB15**. A categorização do fluxo normal e anômalo, ou seja, binário, baseado na base de dados *Training* culminou nas seguintes métricas: acurácia 96%, precisão 97%, sensibilidade 96%, *F1-score* 97% e AUC 99% e base de dados *Testing* nas seguintes métricas: acurácia 89%, precisão 99%, sensibilidade 85%, *F1-score* 91% e AUC 98%.

[23] utilizou a abordagem no pré-processamento de eliminação dos atributos identificadores do fluxo de rede como IP de origem, IP de destino, *sttl*, *dttl* e *ct_state_ttl* na base de dados **UNSW-NB15**. Aplicou-se a técnica de transformação *Min-Max Scaling* para dimensionar os atributos da base de dados. Por fim, conduziu-se na utilização do algoritmo *Ensemble Extra Trees Classifier* constituído de 50 estimadores para criação das árvores de decisão. A classificação do fluxo binário, alicerçado na base de dados *Testing* sucedeu nas seguintes métricas de desempenho acurácia 99,2%, AUC 95,4%, *F1-score* 92,0%, DR 91,2% e FAR 0,3%.

Em seu estudo [12], utilizou a abordagem de seleção de *features* implementado pelo algoritmo *Extreme Gradient Boosting - XGBoost* na base de dados **UNSW-NB15**, com desfecho de 19 (dezenove) atributos selecionados, cuja pontuação está relacionada à importância das características. Conduziu-se na aplicação do método de normalização *min-max scaling* e implementação dos modelos preditores para classificação binária. Os resultados dos algoritmos na base de dados *Testing* respectivamente são: *Artificial Neural Networks* acurácia 84,39%, precisão 78,56%, sensibilidade 98,53%, *F1-score* 87,42%, *Linear Regression* acurácia 77,64%, precisão 73,18%, sensibilidade 93,74%, *F1-score* 82,20%, *K Nearest Neighbors* acurácia 84,46%, precisão 80,31%, sensibilidade 95,09%, *F1-score* 87,08%, *Support Vector Machine* acurácia 60,89%, precisão 58,89%, sensibilidade 95,88%, *F1-score* 72,97%, Decision Trees acurácia 90,85%, precisão 80,33%, sensibilidade 98,38%, *F1-score* 88,45%.

Embora a utilização dos algoritmos de aprendizado de máquina possam apresentar falsos positivos, o presente artigo propõe a utilização de técnicas de pré-processamento com a seleção de atributos e transformação nos dados, realizando assim, o treinamento do modelo, levando em consideração as métricas de classificação, objetivando a minimizar FAR e maximizar as métricas acurácia, precisão, sensibilidade, *F1-score* e AUC.

4. METODOLOGIA

O presente trabalho caracteriza-se como pesquisa aplicada de caráter descritivo, que visa estudar e analisar a eficácia de algoritmos de aprendizado de máquina supervisionados com a implementação dos seguintes métodos: *Recursive Feature Elimination* - RFE com *LinearSVC*, *f_classif*, *chi2* e *Random Forest* - RF, para a realização da seleção de *features*, ou seja,

minimizar o número de atributos no treinamento do modelo supervisionado. Foram implementados os seguintes algoritmos supervisionados: *K Nearest Neighbor* - KNN, *Logistic Regression* - LR, *Support Vector Machine* - SVM, *Naive Bayes* - NB, *Neural Network Multi-Layer Perceptron* - MLP, *AdaBoost* - ADA, *Decision Tree* - DT, *Random Forest* - RF e *Gradient Boosting* - GB para a classificação do tráfego de rede com identificação de intrusão através do fluxo de rede.

Nesse sentido conduziu-se utilizando o método hipotético dedutivo, com levantamento dos dados secundários e revisão bibliográfica. A base de dados utilizada foi desenvolvida pela Universidade de Nova Gales do Sul em Sydney [19], considerando o tráfego de rede normal e anômalo. Portanto, a apresentação dos resultados é quali-quantitativa mediante análise dos resultados pelas métricas: acurácia, precisão, sensibilidade, *F1-score*, AUC e FAR, observado o contexto e objetivos deste trabalho.

5. DESENVOLVIMENTO

Para o desenvolvimento do trabalho foram utilizadas as seguintes ferramentas: *Google Colaboratory* (ambiente de programação), *Google Drive* (armazenamento da instância), linguagem *Python* v3.7, além das bibliotecas *pandas*, *scikit-learn*, *numpy* e *matplotlib*.

O modelo de classificação *offline* proposto neste trabalho, pode ser resumido pela Figura 1.

Posteriormente a importação das bases de dados (*Training* e *Testing*) no *Google Colab*, foram realizadas as seguintes etapas: análise exploratória de dados, exclusão de *features*, transformação dos dados com a utilização do *One Hot Enconding* (OHE), seleção de *features*, normalização dos dados, realização do treinamento e avaliação dos modelos mediante os algoritmos implementados, finalizando a predição do modelo treinado na base de dados *Testing* e obtenção das métricas de desempenho da classificação.

5.1 Base de dados

A base de dados **UNSW-NB15**[2] foi desenvolvida com o objetivo de reproduzir o cenário atual das redes, tendo em vista que as bases de dados disponíveis na literatura foram geradas a cerca de uma década, na qual o comportamento da rede, seja tráfego normal ou anômalo é diferente.

A base de dados é representativa no cenário de redes corporativas onde os serviços estão segmentados na rede Lan. Várias empresas de pequeno e médio porte não utilizam a infraestrutura em nuvem devido ao investimento elevado, mantendo os serviços *on-premise* dentre eles servidor de arquivos, servidor ERP, servidor de banco de dados e outros.

Para simular o tráfego de rede normal e anômalo, o Centro Australiano de Segurança Cibernética desenvolveu a ferramenta *IXIA PerfectStorm no Cyber Range Lab* para criar de forma sintética, atividades de comportamento normal e de nove formas de ciberataque obtidas através do site *Common Vulnerabilities and Exposures*[3], o qual funciona como uma base de dados referente às vulnerabilidades encontradas e exposições relacionadas à segurança da informação. O procedimento de captura bruta de pacotes e armazenamento foi realizado pela ferramenta de análise e captura de tráfego de rede *Tcpdump*[4] exportando a captura no arquivo *PCAP*, para desenvolvimento da base de dados **UNSW-NB15**.

Após a geração do arquivo *PCAP*, foi realizado o procedimento de categorização pelas ferramentas *Bro-IDS*[5], o qual é um *Network Intrusion Detection System*, responsável pela análise do tráfego de rede com identificação de ciberataques, e por último o *Argus*[6] incumbido de gerar os fluxos de rede linha a linha com as respectivas categorias anteriores no formato CSV para utilização no *Python*, finalizando assim a base de dados **UNSW-NB15**.

A **UNSW-NB15** disponibilizou duas bases de dados, o **UNSW_NB15_training-set.csv** para treinamento e avaliação e o **UNSW_NB15_testing-set.csv** para teste, ambos com 45 *features* contendo fluxo de rede catalogados como normal e anômalo.

5.2 Análise e Tratamento dos Dados

A análise exploratória dos dados consiste na identificação do conteúdo dos dados, auxiliando no reconhecimento da dispersão, desvio padrão, variáveis categóricas, correlação entre *features*, identificação de valores faltantes, dentre outras técnicas, por consequência melhorar a tomada de decisão na modelagem do problema. Mediante a análise exploratória, dirigiu-se na idealização da estratégia para o tratamento dos dados, ou seja, o pré-processamento.

O tratamento dos dados (pré-processamento) implica na manipulação, estruturação e organização, que precede a realização das predições, sendo importante, pois impacta diretamente na qualidade final da análise. No presente trabalho foram destacadas e realizadas três ações específicas nas bases de dados (*Training* e *Testing*):

- Limpeza dos dados com exclusão da *feature* '**id**'.
- Limpeza dos dados com exclusão da *feature* '**attack_cat**', pois ao manter o tipo de ataque nos *datasets* (*Training* e *Testing*) ocorrerá o sobreajuste por consequência do vazamento, em virtude de que os ataques no mundo real não estarão catalogados.
- Utilização do **One Hot Enconding** - OHE nas *features*: '**proto**', '**service**', '**state**', por ser um atributo categórico, no qual é necessário converter os dados sem afetar a segmentação equivalente. É criado um *array* com o valor 1 para a *feature* boleana referente a categoria e 0 na *feature* que não existe na categoria, isso em cada linha do fluxo.

Foi realizada a instalação do pacote **category_encoders** e utilização do método OHE, possibilitando a categorização dos atributos mediante a criação do sufixo *underscore*, após o nome de cada *feature*: '**proto_**', '**service_**', '**state_**', para identificar cada valor e a sua respectiva *feature* de origem. Após o procedimento do OHE, o número de *features* nas bases de dados (*Training* e *Testing*) diferem entre si. Em seguida dirigiu-se à verificação dos nomes das *features* e a criação das inexistentes nas respectivas bases de dados, além da inserção do valor **int** "0" para povoar os conjuntos de dados.

[2]https://research.unsw.edu.au/projects/unsw-nb15-dataset
[3]https://cve.mitre.org/cve/search_cve_list.html
[4]https://www.tcpdump.org/manpages/tcpdump.1.html
[5]https://bricata.com/blog/what-is-bro-ids/
[6]https://openargus.org/

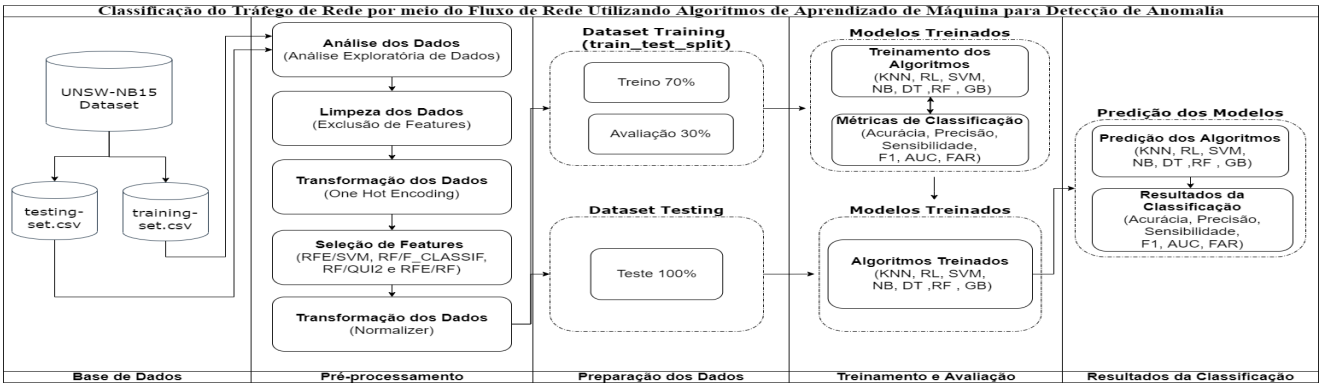

Figura 1: Modelo de Classificação *offline*.

5.3 Seleção de Features

Nesta etapa foram realizadas quatro simulações por intermédio dos algoritmos implementados de seleção de *features*, utilizando a base de dados *Training* com o propósito de reduzir o número de 197 *features* após categorização pelo OHE. o número elevado de atributos se deve a estratégia de pré-processamento implementada, nesse sentido, buscase refinar o treinamento e predição do modelo.

O número de 10 *features* foi definido conforme avaliação de três características principais: menor número de *features*, visto que impacta diretamente no tempo computacional; maior valor das métricas acurácia e; κ (coeficiente *kappa* de Cohen) que estão diretamente relacionadas com o quão acurado o valor obtido pelo modelo está do real.

Nas simulações foram utilizados os algoritmos RFE/RF, com método *SelectKbest*, além dos métodos estatísticos *f_classif* e *Chi2* para a seleção das melhores *features*. As métricas de desempenho utilizadas, foram: menor número de *features* maximizando o valor da acurácia e κ.

Em todas as simulações ocorreram duas etapas: a primeira etapa incide sobre o procedimento de separação do treinamento e avaliação utilizando os seguintes parâmetros: *stratify*: esse parâmetro aloca de forma proporcional as classes 0 e 1 no treino e avaliação; *test_size*: 70% treino e 30% avaliação; e *random_state*: com valor numérico 78, para controlar a aleatoriedade.

A segunda etapa das simulações consiste na seleção das *features* utilizando métodos computacionais como *LinearSVC*, *SelectKbest* e *Random Forest Classifier*. Para todas as simulações foram obtidas as 10 melhores *features* conforme apresentado na Tabela 1, sendo que para a segunda simulação, implementando o algoritmo RFE e as duas últimas simulações implementando o algoritmo RF adotou-se o parâmetro para criação de 100 árvores na floresta.

Utilizou-se todos os processadores disponíveis em paralelo e considerou o valor numérico 78, para controlar a aleatoriedade das amostras na construção das árvores e a manutenção de sua reprodutibilidade.

A Tabela 1 descreve sucintamente os resultados das simulações e *features* obtidas.

Nota-se que a acurácia em todas as simulações apresentam valor superior a 80%, com destaque para a quarta simulação com 97% de acurácia. Os algoritmos utilizados *RFE/RF* obtiveram os melhores desempenhos na seleção das 10 principais *features* ('dpkts', 'sbytes', 'dbytes', 'rate', 'dttl', 'ackdat', 'ct_srv_src', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd'), consistindo no modelo de seleção escolhido entre as simulações propostas para utilização no treinamento e análise de desempenho.

Tabela 1: Resultados das simulações.

Simul.	Sel. melhores *features*	Teste estatístico	*features*	Acu.	κ
RFE	LinearSVC		'ct_state_ttl', 'label', 'proto_udp', 'proto_arp', 'service_pop3', 'service_ssl', 'service_ssh', 'state_INT, 'state_FIN', 'state_ACC'.	0,82	0,64
RF	SelectKbest	f_classif	'rate' 'sttl', 'swin', 'dwin', 'ct_state_ttl', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'proto_tcp', 'service_dns' 'state_INT'.	0,85	0,69
RF	SelectKbest	Chi2	'sbytes', 'dbytes', 'rate', 'sload', 'dload', 'sinpkt', 'sjit', 'stcpb', 'dtcpb', 'response_body_len'.	0,93	0,87
RFE	RF		'dpkts', 'sbytes', 'dbytes', 'rate', 'dttl', 'ackdat', 'ct_srv_src', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd'.	0,97	0,94

5.4 Modelos, Treinamento e Análise de Desempenho

Nesta seção são apresentados os modelos de predição com os algoritmos de aprendizado de máquina propostos para treinamento, além dos resultados mediante cinco medidas de desempenho na base de dados *Testing*.

Devido a similaridade na etapa de predição pelos algoritmos de aprendizado de máquina, conduziu-se exibindo somente o primeiro modelo. Os demais seguem o idêntico procedimento, alterando basicamente o algoritmo e seus respectivos parâmetros e atributos. Para a construção dos modelos de predição são listados cinco passos a saber:

1. Divisão das variáveis preditoras (seleção das *features* obtidas pelo algoritmo *RFE/RF* na etapa anterior) e variável alvo.

2. Procedimento de separação do treino e avaliação pelo *train_test_split*.

3. Normalização das variáveis preditoras.

4. Instanciação do modelo e treinamento com algoritmo.

5. Predição do modelo treinado com o *dataset Testing* e exibição das métricas de desempenho.

O **passo 1** procede com a divisão das variáveis preditoras: ('dpkts', 'sbytes', 'dbytes', 'rate', 'dttl', 'ackdat', 'ct_srv_src', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd') e o alvo: (*'label'*), atribuídas respectivamente nas variáveis X e y. No **passo 2** ocorre a separação do treino e a avaliação pela função *train_test_split*, utilizando respectivamente 70% e 30%. Esta função, presente no *scikit-learn*, divide os dados em conjuntos de treinamento e avaliação. Em seguida conduziu-se realizando a normalização das variáveis preditoras, configurando assim o **passo 3**. Dirigiu-se na utilização em todos os modelos o método de normalização: *Normalizer*. No **passo 4** dirigiu-se a instanciação do modelo e treinamento com algoritmo. As relações entre os modelos podem ser visualizadas na Tabela 2, sendo que para determinados algoritmos foram utilizados parâmetros específicos, como: **KNN** ($n\_neighbors = 5$), **SVC** ($probability = True$), **MLP** ($hidden\_layer\_sizes = (150, 200, 250), max\_iter = 500, activation =' relu', solver =' adam', random\_state = 78$), **DT** ($random\_state = 78$), **ADA** ($n\_estimators = 1000, learning\_rate = 1, random\_state = 78$), **RF** ($n\_estimators =$ 1000, n_jobs = -1, random_state = 78), **GB** ($n\_estimators = 1000, random\_state = 78$). A predição dos modelos treinados com a base de dados *Testing* e a exibição das métricas de desempenho objetivam o **passo 5**.

A Tabela 2 apresenta o desempenho de cada modelo associado ao seu respectivo algoritmo de aprendizado de máquina, *K-Nearest Neighbors* (KNN), *Logistic Regression* (LR), *Support Vector Machine* (SVM), *Naive Bayes* (NB), *Neural Network Multi-Layer Perceptron* (MLP), *AdaBoost* (ADA), *Decision Tree* (DT), *Random Forest* (RF) e *Gradient Boosting* (GB). Foram consideradas as métricas: acurácia (A), precisão (P), sensibilidade (S), *F1-score* (F1), AUC e FAR, além do tempo tempo em segundos despedidos pelos algoritmos.

Tabela 2: Desempenho dos modelos treinados.

Modelos	Algor.	Medidas de desempenho (%)						
		A	P	S	F1	AUC	FAR	Tempo(s)
1	KNN	90,0	97,0	88,0	92,0	96,0	9,0	11,36
2	LR	71,0	83,0	71,0	77,0	80,3	30,0	2,29
3	SVM	80,0	93,0	76,0	84,0	91,0	18,0	1951,70
4	NB	62,0	70,0	78,0	74,0	72,0	47,0	1,64
5	MLP	90,0	95,0	90,0	92,0	96,0	9,0	1859,21
6	ADA	89,0	97,0	87,0	92,0	97,0	9,0	137,75
7	DT	90,0	97,0	88,0	92,0	91,0	9,0	1,88
8	RF	91,0	97,0	89,0	93,0	98,0	8,0	122,16
9	GB	91,0	98,0	89,0	93,0	98,0	8,0	154,42

Os modelos utilizando os algoritmos: KNN, MLP, DT, RF e GB obtiveram resultados satisfatórios mediante as métricas acurácia, precisão, *F1-score* e AUC, com valores iguais ou superiores a 90%, sensibilidade superior a 85% e apresentaram um valor de FAR abaixo de 10%. O modelo DT apresenta menor tempo de execução dentre os melhores modelos de classificação propostos em contra partida o modelo GB obtêm as melhores métricas de classificação dentre os modelos utilizados.

Pode-se inferir que o modelo utilizando o algoritmo *Gradient Boosting* pode ser utilizado como alternativa factível para classificação binária do fluxo de rede para identificação de intrusão.

6. CONCLUSÃO

Conforme apresentado ao longo do artigo, a partir da pesquisa e análise de classificação do tráfego de rede por meio do fluxo utilizando os algoritmos de aprendizado de máquina para classificação binária, ou seja, fluxo de rede normal e anômalo na base de dados *offline* **UNSW-NB15**, pode-se, então, ratificar sua relevância no reconhecimento de fluxo de rede destoante.

A classificação do fluxo de rede é crucial como métrica de desempenho para monitoramento da rede, servindo de subsídio na tomada de decisão pelos administradores de redes. Portanto, dentre os algoritmos de aprendizado de máquina propostos: KNN, ADA, MLP, RL, NB, SVM, DT, RF e GB para classificação do fluxo, os modelos implementados *Random Forest* e *Gradient Boosting*, obtiveram os melhores resultados sejam em métricas de classificação e tempo de execução computacional, cuja métrica *F1-score* alcançou valor superior aos comparados da literatura e AUC idêntico ao melhor modelo *Ensemble Extra Tree Classifier - ETTC* de [23] para identificação de intrusão.

Portanto, mediante os resultados obtidos, os algoritmos de aprendizado de máquina emergem como alternativas no estudo e implantação de futuras ferramentas para categorização do tráfego de rede por meio do fluxo no reconhecimento de intrusão, seja, *offline* ou *online* e vulnerabilidade *Zero Day*.

Diante do trabalho realizado, recomenda-se para trabalhos futuros a implementação de outros algoritmos de aprendizado de máquina na seleção de atributos e aplicação dos modelos cujas métricas de desempenho foram melhores na categorização para utilização em ambiente real, ou seja, *online* para análise do fluxo de rede na detecção de intrusão e também para discernir o tipo de ataque.

7. AGRADECIMENTOS

Agradecemos aos professores Glauco Douglas Moreira - (Chefe do Setor de Tecnologia da Informação do IFMG-Sabará) e Jean Nunes Ribeiro Araújo (Pesquisador do ORC-SLab@UFMG) pela assistência e comentários que aprimoraram o manuscrito.

8. REFERÊNCIAS

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150, 2021.

[2] E. D. S. Bentes, Y. F. C. de Figueiredo, and L. M. de Campos. Aplicação de algoritmos de aprendizado de máquina para detecção de intrusão. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 209–216. SBC, 2021.

[3] J. Chigada and R. Madzinga. Cyberattacks and threats during covid-19: A systematic literature review. *South African Journal of Information Management*, 23(1):1–11, 2021.

[4] L. C. de Brito Guimarães, G. A. F. Rebello, F. S. Fernandes, G. F. Camilo, L. A. C. de Souza, D. C. dos Santos, L. G. C. M. de Oliveira, and O. C. M. B. Duarte. Temia-nt: Monitoramento e análise inteligente

de ameaças de tráfego de rede. In *Anais Estendidos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 57–64. SBC, 2020.

[5] A. C. A. de Oliveira and M. A. Spohn. Escalonamento de máquinas virtuais baseado em custo e tolerante a anomalias de tráfego de rede para dados-como-serviço. *Revista Brasileira de Computação Aplicada*, 12(3):85–96, 2020.

[6] P. R. d. Franceschi. Modelagens preditivas de Churn: o caso do Banco do Brasil. Dissertação de mestrado. Programa de Pós-Gradução em Gestão e Negócios, Universidade do Vale do Rio dos Sinos, 2019.

[7] M. S. Hoque, M. Mukit, M. Bikas, A. Naser, et al. An implementation of intrusion detection system using genetic algorithm. *arXiv preprint arXiv:1204.1336*, 2012.

[8] A. Husain, A. Salem, C. Jim, and G. Dimitoglou. Development of an efficient network intrusion detection model using extreme gradient boosting (xgboost) on the unsw-nb15 dataset. In *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 1–7. IEEE, 2019.

[9] T. Janarthanan and S. Zargari. Feature selection in unsw-nb15 and kddcup'99 datasets. In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*, pages 1881–1886. IEEE, 2017.

[10] D. Jing and H.-B. Chen. Svm based network intrusion detection for the unsw-nb15 dataset. In *2019 IEEE 13th international conference on ASIC (ASICON)*, pages 1–4. IEEE, 2019.

[11] V. Kanimozhi and P. Jacob. Unsw-nb15 dataset feature selection and network intrusion detection using deep learning. *International Journal of Recent Technology and Engineering*, 7(5S2):443–446, 2019.

[12] S. M. Kasongo and Y. Sun. Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *Journal of Big Data*, 7(1):1–20, 2020.

[13] J. F. Kurose and K. W. Ross. *Redes de Computadores e a Internet*. Person, São Paulo, 2006.

[14] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[15] D. S. Medeiros, H. N. Neto, M. A. Lopez, L. C. S. Magalhaes, E. F. Silva, A. B. Vieira, N. C. Fernandes, and D. M. Mattos. Análise de dados em redes sem fio de grande porte: Processamento em fluxo em tempo real, tendências e desafios. *Sociedade Brasileira de Computação*, 2019.

[16] S. Meftah, T. Rachidi, and N. Assem. Network based intrusion detection using the unsw-nb15 dataset. *International Journal of Computing and Digital Systems*, 8(5):478–487, 2019.

[17] M. C. Monard and J. A. Baranauskas. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32, 2003.

[18] N. Moustafa and J. Slay. The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*, pages 25–31. IEEE, 2015.

[19] N. Moustafa and J. Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[20] N. Moustafa and J. Slay. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. *Information Security Journal: A Global Perspective*, 25(1-3):18–31, 2016.

[21] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials*, 21(2):1988–2014, 2018.

[22] B. Pranggono and A. Arabo. Covid-19 pandemic cybersecurity issues. *Internet Technology Letters*, 4(2):e247, 2021.

[23] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann. Netflow datasets for machine learning-based network intrusion detection systems. In *Big Data Technologies and Applications*, pages 117–135. Springer, 2020.

[24] R. d. S. Silva. Detecção de intrusão usando aprendizagem por reforço. Technical report, Universidade Federal do Amazonas, 2013.

[25] M. Souza. Readaptação do modelo acme para detecção de novas técnicas de intrusão. *Monografia de Graduação. UNESP–Departamento de Ciência da Computação e Estatística, São José do Rio Preto-SP*, 2002.

[26] A. S. Tanenbaum. *Redes de computadores*. Editora Campus, Rio de Janeiro, 2003.