REVISTA DE ciência e INOVAÇÃO DO IFFAR

V. 09 / 2023
ISSN 2448-4091
DOI 10.26669/2448-4091.2023.388

ARTIGO ORIGINAL

Sistema de detecção de intrusão utilizando métodos de aprendizagem de máquina em redes de computadores

Intrusion detection system using machine learning methods in computer networks

Sistema de detección de intrusión utilizando métodos de aprendizaje automático en redes de computadoras

Matheus Santos Andrade [1]
Jean Santos [2]
Jonathas Freitas [3]

RECEBIDO EM 13/12/2022
ACEITO EM 05/04/2023

RESUMO

Nos últimos anos, houve um grande aumento de serviços baseados na internet, que levanta grandes disrupções à segurança da informação. E com a colossal gama de tráfego de rede de dados que são gerados diariamente, isto com a sua alta velocidade, faz as ameaças à segurança serem cada vez mais enigmáticas. Nesse sentido, o presente artigo apresenta uma abordagem baseada em métodos de aprendizagem de máquinas aplicadas à busca de ameaças em redes de computadores, no objetivo de tentar detectar intrusão e, portanto, ajudar a prevenir a ocorrência de ataques. Com isso, foram testados algoritmos para classificação de ataques na rede, com três métodos distintos: Árvore de Decisão, Tabelas de Decisão e *Naive Bayes*. A eficácia de cada técnica é avaliada por meio de experimentos usando a base de dados KDD'99, e se baseia na matriz de confusão, a qual obteve, por meio de uma pequena porção (cerca de 10%)

[1] Instituto Federal de Educação, Ciência e Tecnologia de Sergipe, Lagarto, SE, Brasil.
matheusbsi1992@gmail.com - https://orcid.org/0000-0001-7274-9633

[2] Instituto Federal de Educação, Ciência e Tecnologia de Sergipe, Lagarto, SE, Brasil
jeanamorim08@gmail.com - https://orcid.org/0009-0009-9223-9425

[3] Universidade Tiradentes, Aracaju, SE, Brasil
jonathas200@gmail.com - https://orcid.org/0009-0008-3268-6090

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

da base de dados, uma acurácia, precisão e *recall* acima de (89%) sobre os classificadores analisados, afirmando a viabilidade de máquinas de aprendizagem em busca de classificação de anomalias em redes de computadores.

PALAVRAS-CHAVE: ataque; métodos de aprendizagem de máquina; matriz de confusão.

ABSTRACT

In recent years, there has been a large increase in internet-based services, that raises major disruptions to information security. And with the colossal range of data network traffic that is generated daily, this with its high speed, makes security threats increasingly enigmatic. In this sense, this article presents an approach based on machine learning methods applied to the search for threats in computer networks, in order to try to detect intrusion and, therefore, help to prevent attacks from occurring. Thus, algorithms were tested for classification of attacks on the network, with three different methods: Decision Tree, Decision Tables and Naive Bayes. The effectiveness of each technique is evaluated through experiments using the KDD'99 database, and is based on the confusion matrix which obtained, throug a small portion (about 10%) of the database obtained accuracy, precision and recall. above (89%) on the analyzed classifiers, affirming the feasibility of learning machines in search of classification of anomalies in computer networks

KEYWORDS: attack; machine learning methods; confusion matrix.

RESUMEN

En los últimos años, se ha producido un enorme aumento de los servicios basados en internet, lo que plantea importantes disrupciones en la seguridad de la información. Y con la colosal gama de tráfico de red de datos que se genera a diario, esto con su alta velocidad, hace que las amenazas a la seguridad sean cada vez más enigmáticas. En este sentido, este artículo presenta un enfoque basado en métodos de aprendizaje automático aplicados a la búsqueda de amenazas en redes de computadoras, con el objetivo de tratar de detectar intrusiones y, por lo tanto, ayudar a prevenir la ocurrencia de ataques. Así, se probaron algoritmos de clasificación de ataques a la red, con tres métodos diferentes: Árbol de Decisión, Tablas de Decisión y Naive Bayes. La efectividad de cada técnica se evalúa a través de experimentos utilizando la base de datos KDD'99, y se basa en la matriz de confusión, que obtuvo, a través de una pequeña porción (alrededor del 10%) de la base de datos, una exactitud, precisión y recuperación superior (89%) sobre los

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

clasificadores analizados, afirmando la viabilidad de máquinas de aprendizaje em busca de clasificación de anomalías en redes de computadoras.

PALABRAS CLAVE: ataque; métodos de aprendizaje automático; matriz de confusión.

1 Introdução

Há uma crescente e maciça produção de dados nos sistemas de informação em todo o mundo, como apontado por (Manyika et al., 2011). É sabido que o amplo uso dos sistemas de informação, da *Internet* e das redes de computadores, em geral, traz grandes benefícios para a população e corporações, porém, é preciso destacar a importância de proteger os dados e informações, uma vez que é cada vez mais corriqueira a ocorrência de ataques nos meios eletrônicos (Andrade, 2017).

De acordo com Leu *et al*. (2017), diversos levantamentos estatísticos demonstram nos últimos anos um número alarmante de invasões reportadas ao *Symantec Global Internet Security Threat Report*. Todavia, é necessário investir em Sistemas de Detecção de Intrusão (*IDS - Intrusion Detection System*) eficientes, vislumbrando a uma operação que busca, analisa e verifica as anomalias, afim de manter uma rede de computadores operando com baixo nível de incidentes de segurança (Barford et al., 2002).

Depois do surgimento do primeiro IDS (Denning, 1987), milhares de estudos foram publicados nesta área. Ainda assim, continua a ser um problema não resolvido completamente, em razão do constante surgimento e aperfeiçoamento de tipos de ameaças (Sommer; Paxson, 2010).

Métodos de aprendizagem de máquina vêm sendo aplicados para detectar e classificar anomalias no tráfego na rede (Fung; Boutaba, 2013; Utimura; Costa, 2018). Para isso, neste artigo, o objetivo é apresentar uma avaliação de métodos de aprendizagem de máquina para classificar disfunções referentes a possíveis

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

ataques. As análises foram realizadas através do conjunto de dados KDD'99 (Stolfo et al., 2000), além disso, demonstraram boas taxas de acurácia, precisão e *recall*, todos acima de 89% para os 03 (três) métodos de classificação avaliados. Entretanto, cabe destacar que houve um baixo número de falso positivo para os algoritmos de Árvore de Decisão e Tabelas de Decisão.

Nas próximas sessões, apresentar-se-á: uma breve fundamentação sobre Sistemas de Detecção de Intrusão e Aprendizagem de Máquina; a demonstração de uma Revisão Empírica; os materiais e métodos utilizados para realizar os experimentos – com a apresentação dos resultados numéricos obtidos na aplicação do algoritmo classificatório para a resolução deste tipo de problema; e, por fim, as conclusões encontradas.

2 Fundamentação

Esta seção apresenta uma breve fundamentação contendo Sistema de Detecção de Intrusão e Máquina de Aprendizagem.

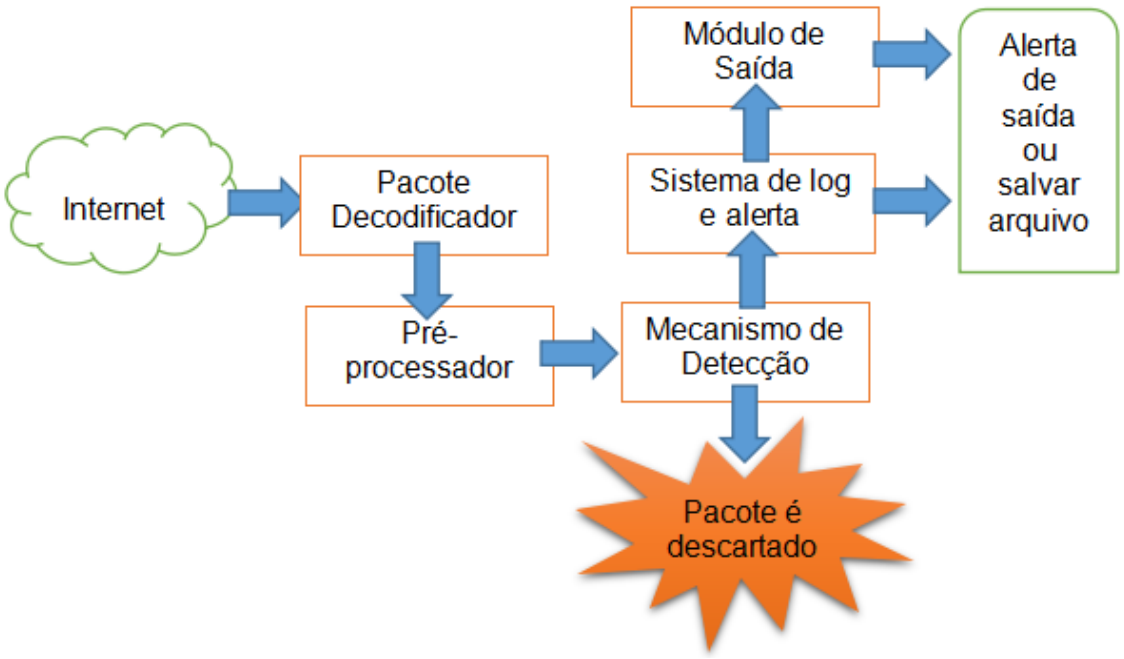
2.1 Sistema de Detecção de Intrusão

O Sistema de Detecção de Intrusão (IDS) consiste em um mecanismo responsável pelo processo de monitoramento/detecção de eventos, no intuito de identificar comportamentos alheios (Andrade, 2017). Na indicação de existência de tráfego intrusivo, o Sistema de Detecção de Intrusão Baseado em Rede (NIDS) utiliza os procedimentos necessários, a fim de que os eventos: (i) capturados, (ii) decodificados, (iii) analisados, (iv) relacionados com assinaturas armazenadas em uma base de dados ou por tráfego de rede, e, porventura, (v) seja detectado alguma anomalia, (vi) realizar o registro do evento e (vii) alertar ao administrador do sistema, conforme Figura 1.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

O modo pelo qual alguns NIDSs analisam os processos é representado previamente por informações contidas no tráfego de rede e comportamentos reconhecidos pelas assinaturas, que estão integrados em alguns tipos de ataques. Identificando que uma anomalia é algo diferente, anormal, peculiar ou que não é facilmente classificado. Na segurança de computadores, uma anomalia é definida como ações ou dados que não são considerados normais por um determinado sistema, usuário ou rede. Anomalias podem ser causadas por ataques cibernéticos, erros humanos ou até mesmo mudanças naturais no ambiente. Ao identificar e responder a anomalias, os profissionais de segurança podem ajudar a proteger seus sistemas contra ataques e outros riscos (Pinheiro, 2017).

FIGURA 1 – Sistema de detecção de intrusão baseado em redes de computadores.



Fonte: Andrade (2017).

Comparando o NIDS por assinatura e anomalia, podemos verificar que as anomalias são identificadas como intrusões sem assinaturas previstas, fornecendo além de conhecimento para ser aplicado futuramente na detecção por assinatura, a segurança para o NIDS. Verificou-se, então, que a assinatura pode gerar

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

um grande número ou resultados de falsos positivos, diferente de detecção por anomalias, em que os números de alarmes falsos são pequenos, e isso tornará o método mais rápido na detecção (Santos, 2010).

Portanto, um sistema NIDS funciona comparando o tráfego de rede a uma lista de ataques conhecidos. Se o tráfego corresponder a um padrão de ataque conhecido, o IDS gerará um alerta, ou seja, que estão incluídos no conjunto de ataques que o NIDS possui. Por isso necessita de constante atualização diante da rapidez com que novos ataques surgem. Isso pode ser minimizado com um método para o reconhecimento de intrusão eficaz, a exemplo do método de aprendizagem de máquina.

2.2 Aprendizagem de Máquina

O termo aprendizagem de máquina (AM) surgiu no século XX com o justo aprimoramento das atividades através de experiência, com a relação da tarefa e a medida do desempenho (Mitchell, 1997). A área de aprendizagem de máquina estuda algoritmos, métodos e ferramentas (*i.e.*, são desenvolvidos) para mecanismos de aprendizagem. Esse âmbito integra a inteligência artificial (IA) que, segundo Bellman (1978); Norvig e Russel (2013) propicia a automatização de atividades associadas ao pensamento humano, como a tomada de decisões, a resolução de problemas, o aprendizado etc., aparentando o funcionamento do cérebro humano.

Embora existam características sobre o comportamento do cérebro humano, a AM necessita de métodos de aprendizagem: Pré-processamento, mineração de dados e pós-processamento de acordo com Smola e Schölkopf (2001) e Le, Smola e Vishwanathan (2007). Os procedimentos precisam de abrangência, de forma que os dados sejam extraídos e as suas características validadas por meio de métodos de estatísticas e outros métodos.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

O pré-processamento é relevante, visto que são aplicados métodos para coleta dos dados, organização na forma de conjuntos e preparação dos dados. Fundamentando a necessidade do processo de classificação, ele compreende desde a análise dos dados até a formatação e a normalização. No entanto, um dos problemas dessa fase são os dados em pequena quantidade e a má qualidade desses dados (Hussain et al., 2000). Todavia, vale salientar que a obtenção de vários problemas consiste em ruídos sobre dados reais originados da má representação, disseminação de dados inexatos nos experimentos ou até mesmo erro de digitação etc. Por isso, a utilização de algoritmos ou métodos estatísticos é imprescindível para que haja a remoção dos ruídos, trazendo a confiabilidade e a segurança das informações sobre o sistema (Gurek, 2001; Libralon, 2007).

Logo após o recebimento do conjunto de dados através do pré-processamento, para prosseguir com a fase de mineração de dados, é necessária a utilização de possíveis ferramentas, a serem empregadas nas formas das validações e de padrões (Larrañaga et al., 2006).

Contudo, a fase de pós-processamento consiste em transformar os dados obtidos da fase de aplicação ou mineração de dados de forma bem simples que os tornem observáveis. De uma forma que a seleção de padrões vem a identificar quais padrões são realmente interessantes para o domínio de dados. Isso pode ser feito considerando uma variedade de fatores, como a relevância do padrão para o domínio de dados, a significância estatística do padrão e a relevância do padrão para o objetivo da análise. Dessa forma, os dados podem ser interpretados, a partir da utilização de gráficos, dados estatísticos, entre outros recursos (Carvalho, 2014).

3 Revisão Empírica

A utilização de um NIDS para identificação, processamento e classificação de dados de intrusão e normais é uma tarefa contínua. À medida que

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

as ameaças à segurança da rede evoluem, as organizações precisam estar constantemente atualizadas para proteger suas redes de computadores. Isso porque se deve buscar, com primazia, a eficiência, para obter resultados com um grande número de verdadeiro positivo e um pequeno (ou até zero) número de falso positivo. Entretanto, esta é a dificuldade em desenvolver um NIDS. Para tanto, históricos de estudos em NIDS podem ser analisados, a fim de alavancar a estrutura do projeto.

Alsubhi, Zhani e Boutaba (2012) propuseram a criação de um sistema de detecção de intrusão e prevenção sobre intrusões em ataques diversificados por *hackers* (IDPS). Entretanto, verificou-se que o estudo realizado por eles pode afetar o desempenho da rede em termos de atraso de pacotes ponta a ponta e até mesmo na perda de pacotes. Isso propõe um modelo de fila analítica, que usa a cadeia de *Markov* incorporada e pode analisar o desempenho do IDPS. Logo, entende-se que essa abordagem supera outras práticas tradicionais e pode obter segurança junto com a qualidade de serviço da rede.

Em seguida, Fung e Boutaba (2013) propuseram um sistema de detecção de intrusão distribuída através da estatística *bayesiana* mencionando os dois fatores adicionais à pesquisa. Esse modelo pode identificar tanto ataques externos à rede quanto ataques internos, escalonáveis em densidade de *Dirichlet*, para garantir a alta confiança ao estimar a confiabilidade do IDS.

Outra referência são Kumar e Reddy (2014). Eles realizaram um estudo com o objetivo de criar um agente que coleta dados da rede e aplica a um sistema artificial imune a intrusões. Esse método faz e/ou torna o tráfego de dados mais seguro em um determinado canal. Constatou-se que o sistema funciona normalmente sem qualquer incapacidade, mostrando-se mais eficiente do que qualquer outro caminho propenso a ataques, assim como fornece resultados promissores.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Kim, Lee e Kim (2014) propuseram um sistema híbrido de detecção de intrusão integrado, condizente com o modelo de detecção de intrusão e uso indevido. Ele altera os dados normais em subconjuntos menores, usando detecção de uso indevido. Posteriormente, os autores utilizaram o classificador *Support Vector Machine* (SVM) para ter um comportamento preciso do perfil normal. Para testar o sistema, foi utilizada a base de dados NSL-KDD, amplamente testada e conhecida pela comunidade. Porém, o modelo supera métodos convencionais em termos de taxas de detecção de intrusão e falsos positivos. Além disso, o tempo de treinamento e de teste são considerados entre 50% e 60% mais eficazes em comparação aos métodos tradicionais.

Baseado no estudo, Utimura e Costa (2018) usaram 10% do conjunto de dados ISCXIDS2012 para avaliar seus algoritmos. Mesmo que os dados tenham sido separados em partes (*i.e.*, treinamentos e teste de porções), subdividiram-se em duas fases: a primeira concentrou-se na análise comparativa do desempenho dos classificadores *Multi-Layer Perceptron* (Redes Neurais de Múltiplas Camadas) e o *Optimum-Path Forest* (OPF); e o segundo faz-se da validação da extensão SDI *Snort*++ – aqui cabe mencionar que a precisão e o tempo de execução são essenciais como métricas de desempenho.

Na sequência, Rama Devi e Abualkibash (2019) apresentaram um mecanismo de comparação em aprendizagem supervisionado, não supervisionado e semi-supervisionado. Ele foi projetado e implementado para um sistema IDS, na utilização do conjunto de dados KDD-99 CUP e o também conhecido pela comunidade: o NSL-KDD. Baseando-se na utilização de redes neurais para o pré-processamento dos conjuntos em si supracitados, para recursos nominais, com valores distintos, utilizando-se pontos flutuantes em redes neurais, o UDP obteve de [0,1], o ICMP = [1,0] e o TCP = [-1,1], em um conjunto de teste sendo rotulado com +1 dados normais e -1 anomalias. Empregado neste estudo de detecção de intrusão, os testes nos algoritmos de Regressão

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Logística, Árvore de Decisão, KNN, SVM, Floresta Randômica, *Adaboost*, *Multi-Layer Perceptron* e *Naive Bayes*, obteve-se acurácia no conjunto de dados KDD-99 CUP variando de 79.7% até 94.17%. Já com a utilização dos algoritmos Regressão Logística, Floresta Randômica, Descida do Gradiente Estocástico, *Naive Bayes, Adaboost e Multi-Layer Perceptron*, constatou-se a variação de 88.9% até 99.7% sobre o conjunto de dados NSL-KDD.

Analisando o descrito, vale destacar que a busca por várias soluções é possível com uma abordagem em classificadores. Isso inclui os aspectos que influenciam diretamente em uma tomada de decisão, a qual pode ser avaliada por meio de aprendizagem de máquina com um conjunto de dados.

4 Material e Métodos

4.1 Conjunto de Dados

O KDD'99 é um dos conjuntos de dados mais amplamente utilizado para avaliação dos métodos de detecção de intrusões. Além disso, é usado em 149 artigos, sendo 65 revistas produzidas com o fator *Science Citation Index* (Özgür; Erdem, 2016). Essa base de dados foi preparada por Stolfo et al. (2000) e construída com base nas informações capturadas pela DARPA'98 IDS (*Defense Advanced Research Projects Agency* - https://www.darpa.mil/). Contudo, essa base de dados foi reunida em 41 atributos pré-processados, utilizados para detectar ataques mais frequentes: (DoS U2R, R2L e *Probing Attack*), no total de 22 tipos de ataques na base de treinamento e 17 tipos de ataques extras na base de teste (Stolfo et al., 2000), além de variedades de intrusões mais apresentados pela base de dados KDD'99:

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

• **DENIAL OF SERVICE (DoS):** O atacante utiliza-se de certas ferramentas, para deixar os serviços ou até mesmo sites fora de operação, não podendo responder às requisições legítimas, mas sim invalidando informações por sobrecarga (*i.e.*, vários pacotes de dados sendo enviados em descomedimento pela rede atacante). Estes são alguns exemplos de ataque: *Apache2, Back, Land, Mail Bomb, SynFlood (Neptune), Ping of Death, Process Table, Smurf, Syslog D, Tear Drop, UDP Storm.*

• **USER TO ROOT ATTACK (U2R):** O atacante lida de alguns *exploits*[1], e como usuário padrão, buscando total privilégio sobre o sistema. Então, explorando as suas vulnerabilidades sobre algum serviço. Exemplos de ataque: *Eject, Ffbconfig, Fdformat, Loadmodule, Perl, PS, Xterm*.

• **REMOTE TO LOCAL ATTACK (R2L):** Para obter acesso a alguma máquina, o atacante envia pacotes para a rede mantenedora, na tentativa de explorar vulnerabilidades como se fosse um usuário local do sistema. Exemplos de ataque: *Imap, Guest, Xsmap, Xlock, Named, Dictionary*.

• **PROBING ATTACK:** Consiste na varredura sobre portas de serviços na rede de computadores e analisa quais *exploits* serão utilizados para explorar as vulnerabilidades. Exemplos de ataque: *Nmap, Saint, MScan, Ipswap*.

A. Classificação de Aprendizagem de Máquina

Devido ao perfil da base de dados KDD'99, o treinamento realizado neste artigo foi o supervisionado. O treinamento ocorreu utilizando métodos: Árvores de Decisão, *Naive Bayes* e Tabelas de Decisão. Classificadores presentes na ferramenta WEKA (*do Inglês - Weikato Environment for Knowledge Analysis*) que é um software de código aberto, composto de algoritmos de AM, com a sua arquitetura modular e extensível permitiu que diversos métodos de AM fossem aplicados em conjunto de dados ou até mesmo associado ao *JAVA*[2] , para pré-processamento,

[1] Os *exploits* são ferramentas maliciosas desenvolvidos/utilizadas por pessoas especializadas através de técnicas e testes, em busca de vulnerabilidades nos sistemas e programas.
[2] A linguagem de programação *Java* é orientada a objetos, com o intuito de ser executada em qualquer plataforma ou até mesmo dispositivos.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

regressão, classificação, agrupamento, mineração de dados, clusterização, seleção de recursos e pós-processamento da aprendizagem (Witten et al., 2016), e em geral, os princípios funcionais são uma boa escolha para o desenvolvimento de IDS (Scarfone; Mell, 2007).

1) Árvores de Decisão: Método de aprendizagem de máquina supervisionado que utiliza diagramas com uma sequência de decisões inter-relacionadas, e os seus resultados são esperados de acordo com a natureza da alternativa escolhida. Através de procedimentos, o algoritmo aprende sobre funções de classificação que decide por meio de um conjunto de variáveis o valor de um atributo dependente, mas considera os valores de atributos independentes de entrada na Árvore de Decisão, isto conforme Bhargava et al. (2013). Pela WEKA, a Árvore de Decisão J48 é a implementação do algoritmo ID3 (*Iterative Dichotomiser 3*). Detalhes deste algoritmo podem ser encontrados a partir de Quinlan (2014).

2) *Naive Bayes*: Classificador probabilístico baseado na aplicação do teorema de *Bayes* (*i.e.* das estatísticas bayesianas) com fortes suposições de independência sobre os seus atributos. Sendo altamente escalonáveis, exigindo um alto número de variáveis lineares (preditores) em um problema de aprendizagem. Calculando a probabilidade condicional de cada atributo, seguido de uma aplicação contida do teorema de Bayes para determinar a probabilidade relativa sobre as características dos atributos, no sentido da previsão do resultado, conforme (Aggarwal, 2014). Mais detalhes da implementação do algoritmo *Naive Bayes* em WEKA pode ser encontrado em John e Langley (1995).

3) *Tabelas de Decisão*: o método Tabelas de Decisão faz representação visual concisa para especificar quais ações serão executadas. Determinando condições apresentadas após seguir uma série de decisões relacionadas, para uma tomada de decisão. Detalhes sobre Tabelas de Decisão podem ser encontrados no trabalho de Kohavi (1995) implementado em *WEKA*.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Inicialmente, os dados são analisados como arquivos independentes (i.e., treinamento e teste). Esses dados contêm diversas características básicas das conexões TCP individuais, recursos de conteúdo de conexão sugeridos pelo conhecimento do domínio e características de tráfego usando um *buffer* de dois segundos. Entretanto, o algoritmo de classificação irá processar o fluxo de dados, indicando se a informação é caracterizada uma intrusão ou não. Por último, os resultados obtidos por cada um dos classificadores são visualizados para uma devida interpretação, ver Quadro 1.

QUADRO 1 – Algoritmo de classificação baseado na detecção de intrusão na rede.

Linha	Passo	Explicação
1	Início	Começa o algoritmo.
2	Instância de classe de treino e Instância de classe de teste.	Lê os dados de treino e teste.
3	Identifique algoritmo proposto (NaiveBayes, Árvore de Decisão, Tabelas de Decisão).	Identifica o algoritmo de classificação a ser usado.
4	predicao ← null, atual ← null, vp ← 0, vn ← 0, fp ← 0, fn ← 0, total_instancias ← 0, valorpredicao ← 0.	Inicializa as variáveis.
5	Se algoritmo proposto <> null então	Se um algoritmo foi especificado, então
6	buildClassifier(classe de treino)	Constrói o classificador.
7	Fim-se	Se não, então
8	Para i = 1 → n faça	Para cada instância de teste:
9	Se algoritmo proposto <> null então	Se um algoritmo foi especificado, então
10	valorpredicao = classifyInstance (classe de teste)	Classifica a instância de teste.
11	Fim-se	Se não, então
12	atual = instância de teste classificada	Armazena o rótulo da instância de teste.
13	predicao = instância de teste(valorpredicao)	Armazena a previsão do classificador.
14	Se atual <> “normal” e predicao <> “normal”	Se a instância de teste não for normal e a previsão do classificador também não for normal:

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

15	então	
16	vp ← vp +1	Incrementa o contador de verdadeiros positivos.
17	Fim-se	Se não, então
18	Se atual = “normal” e predicao = “normal” então	Se a instância de teste for normal e a previsão do classificador também for normal:
19	vn ← vn + 1	Incrementa o contador de verdadeiros negativos.
20	Fim-se	Se não, então
21	Se atual = “normal” e predicao <> “normal” então	Se a instância de teste for normal e a previsão do classificador não for normal:
22	fp ← fp + 1	Incrementa o contador de falsos positivos.
23	Fim-se	Se não, então
24	Se atual <> “normal” e predicao = “normal” então	Se a instância de teste não for normal e a previsão do classificador for normal:
25	fn ← fn + 1	Incrementa o contador de falsos negativos.
26	Fim-se	Se não, então
27	total_instancias←total_instancias+1	Incrementa o contador de instâncias.
28	Fim-para	Termina o laço.
29	acuracia ← (vp+vn)/(vp+vn+fp+fn)	Calcula a acurácia.
30	precisao ← vp/(vp+fp)	Calcula a precisão.
31	recall ← vp/(vp + fn)	Calcula o recall.
32	Fim.	Termina o algoritmo.

Fonte: Os autores (2022).

A função *buildClassifier* foi utilizada para classificação com base nas instâncias de treinamento fornecidas e geradas dos classificadores: Árvores de Decisão, *Naive Bayes* e Tabelas de Decisão, sendo necessário inicializar todos os campos do classificador que não estão sendo definidos por meio de opções (*i.e.*, várias chamadas de *buildClassifier* sempre devem levar ao mesmo resultado), não devendo alterar o conjunto de dados de alguma forma.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

A função *classifyInstance* foi utilizada para que realize uma instância de teste e retorne vários valores de classe previstos através do uso do conjunto de dados teste KDD'99 para a instância que é fornecido.

Neste projeto, o objetivo é a prova de conceito da caracterização do fluxo de dados anômalos em aprendizagem de máquina para as redes de computadores. Assim, não é preciso a utilização de assinaturas, pois os algoritmos supracitados são utilizados para fins de classificação na obtenção de detecção de anomalias. Não necessitando, ainda, de blocos de construção para sistemas reais e de qualquer outra realimentação constante sobre a base de dados.

5 Resultados Numéricos

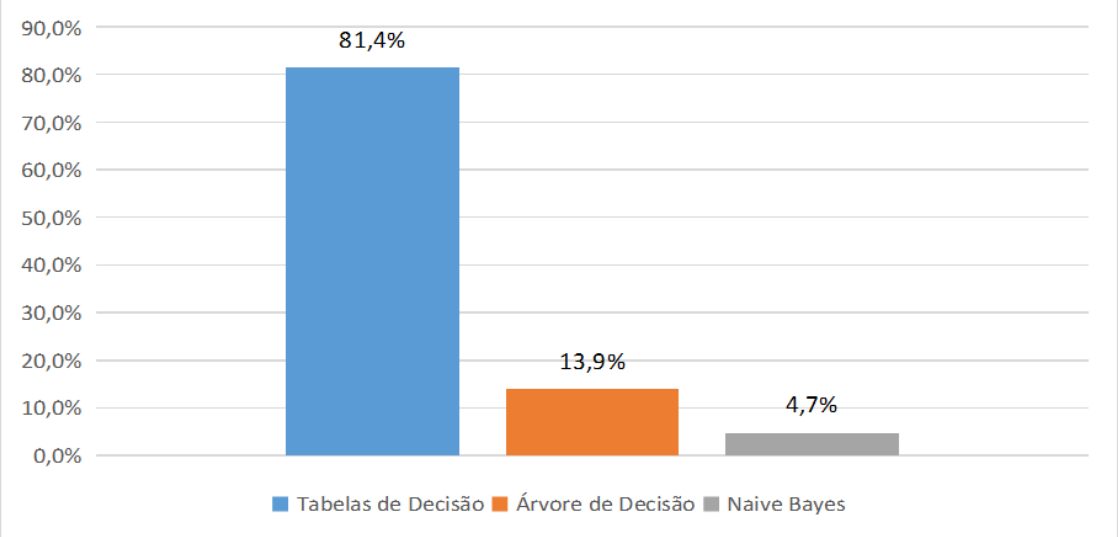
O desempenho do sistema pode ser avaliado utilizando métricas, em *micro average* (Abracadabra, 2018) acurácia, precisão e *recall* (Tarca et al., 2007), e taxas de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos são métricas importantes para avaliar o desempenho de cada um modelo identificado no estudo. Essas métricas vem a ser usadas para estudar a matriz de confusão, que mostra a distribuição das previsões e rótulos reais de cada classe. Também indica que as medidas aqui tratadas foram geradas através de um único teste aplicado sobre o conjunto de dados KDD'99, utilizando de um processador Intel(R) Xeon(R) CPU E5-2650L v2 @ 1.70GHz, 1701 *Mhz*, 10 Núcleo(s), 20 Processador(es) Lógico(s), além de 16 (dezesseis) GB's de memória ram a 1333 *MHz* e o sistema operacional *Ubuntu* 22.04. Já o ambiente de desenvolvimento de *software* foi utilizado o A*pache NetBeans* 12.0, com a casualização da interface de programação de aplicação (do Inglês *Application Programming Interface* - API) em *Java*.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

5.1 Tempo da Análise dos Métodos Classificatórios Citados ao Estudo

O tempo da análise é verificado com detalhes sobre o conjunto de dados de ataques e a relação de cálculo de tempo total computacional em percentual (%). Na Figura 2, é fomentado o tempo de execução sobre o algoritmo adotado. Constatou-se que o método classificatório Árvore de Decisão apresentou 13,9%; já o *Naive Bayes* informou um detalhamento de apenas 4,7%; sendo a menor margem de poder computacional em custo computacional. Enquanto o método Tabelas de Decisão apresentou o maior valor, o que é admirável, pois o valor total apresentado foi de 81,4%

FIGURA 2 – Tempo de custo computacional de *ms* em % sobre os algoritmos classificatórios, Tabelas de Decisão, Árvore de Decisão e *Naive Bayes*.



Fonte: Os autores (2022).

5.2 Análise de Métricas de Desempenho sobre a Conjuntura da Base de Dados KDD'99

O Quadro 1 demonstra os resultados sobre a busca pela detecção de dados anômalos ao estudo na obtenção de valores cruciais para falsos positivos, falsos

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

negativos, verdadeiros positivos e verdadeiros negativos. Ele mostra a comparação dos distintos classificadores pela unidade de ataque ou não. As colunas do Quadro 2 mostram os muitos números de ataques classificados corretos (dados classificados como ataques, todavia sendo ataques diferentes e não classificados como normal são contabilizados), o número de verdadeiro negativo (prevê uma resposta do tipo não e está condizente com o dado observado na conexão), o número de falso positivo (prevê na conexão o tipo do dado anômalo como sim, mas deveria ser não) e o falso negativo (prevê uma conexão anômala como não, mas ela deveria ser sim).

QUADRO 2 – Taxas de valores apresentados pelos classificadores em: VP, VN, FP E FN.

Classificador Proposto	VP	VN	FP	FN
Árvore de Decisão	396.723	97.271	6	20
Naive Bayes	396.542	53.074	44.203	201
Tabelas de Decisão	396.509	97.202	75	234
Total de Amostras: 494.020				

Legenda: VP = Verdadeiro positivo; VN = Verdadeiro negativo; FP = Falso positivo; FN = Falso negativo; Fonte: Os autores (2022).

Os resultados constantes no Quadro 2 implicam dizer que o classificador Árvore de Decisão demonstrou o seu arquivamento de verdadeiro positivo com 396.723 números, seguido do *Naive Bayes* e seus 396.542 números, enquanto o método Tabelas de Decisão obteve o menor número: 396.509. Apesar disso, em se tratando de valores de números de verdadeiros negativos, o classificador Árvore de Decisão e Tabelas de Decisão obtiveram valores aproximados (92.271 e 97.202), já o *Naive Bayes* teve o menor valor, os seus exatos 53.074 números.

Quanto aos falsos positivos aplicados ao contexto de IDS, a Árvore de Decisão apresentou a menor taxa com 6, seguido da Tabelas de Decisão com 75 números, e a maior taxa entre todos os classificadores foi o *Naive Bayes*, exibindo 44.203 números.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Considerando a classificação deste estudo, podemos analisar que a taxa de falsos negativos foi a menor com o classificador Árvore de Decisão, com 20 números, seguido de perto pelo classificador *Naive Bayes* e pelas Tabelas de Decisão, apresentando, respectivamente, 201 e 234 números.

Posto isto, disponibilizados pelo conjunto de dados reais e condizentes com 494.020 amostras, os testes dos números apresentam, no Quadro 2, as distintas comparações dos algoritmos de classificação em termos de acurácia, precisão e *recall*. Observou-se que mesmo com uma gama pequena de amostras, em apenas 10% do total sobre a base de treinamento de cada um dos três classificadores foram obtidos valores satisfatórios.

O *recall* é a relação do classificador que pôde reconhecer os números de ataques positivos; enquanto a acurácia avalia o quanto o classificador pode reconhecer a relação entre o número de ataques classificados corretamente pelo número total de amostras. Em seguida, a precisão avalia a quantidade de classificações positivas que estão condizentes ao conjunto de dados.

QUADRO 3 – Taxas de Acurácia, Precisão e Recall entre os classificadores propostos.

Algoritmo Classificador	Acurácia (%)	Precisão (%)	Recall (%)
Árvore de Decisão	99,99	99,99	99,99
Naive Bayes	91,01	89,97	99,94
Tabelas de Decisão	99,93	99,98	99,94

Fonte: Os autores (2022).

É possível analisar no Quadro 3 que a acurácia do classificador Árvore de Decisão obteve 99,99%, com a aproximação das Tabelas de Decisão, que obtiveram taxa de 99,93%, e o menor valor apresentado foi do *Naive Bayes*, com 91,01%. Mesmo assim, a precisão contida da Árvore de Decisão foi a maior de todas, com exato 99,99%, que foi próxima à das Tabelas de Decisão, com valor de 99,98%. E, carecido de muitos falsos positivos, o classificador *Naive Bayes* teve a taxa menor, com exatos 89,97%.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Sobre uma análise de reconhecimento de ataques positivos, os classificadores *Naive Bayes* e Tabelas de Decisão obtiveram valores similares (99,94%), já a Árvore de Decisão alcançou a maior taxa (99,99%).

6 Conclusão

Este estudo propôs um sistema de detecção de intrusão baseado em redes, com uma boa base de conjunto de dados e o Quadro 1 que trouxe a exploracidade, validações, bem como apresentou métodos de classificação *Naive Bayes*, Tabelas de Decisão e Árvore de Decisão. Mesmo que o algoritmo classificatório *Naive Bayes* apresente uma margem elevada de falsos positivos, os resultados apresentados contemplaram uma média acima de 89% em acurácia, precisão e recall.

Cabe ressaltar que, as métricas alcançadas foram acima da média, o que é comparável e observado em estudos existentes. Isso sugere que o modelo é robusto e pode ser usado para resolver uma variedade de problemas. Além disso, operou-se com mais detalhes sobre os métodos classificatórios citados. Nesse sentido, constatou-se que o sistema apresenta viabilidade, desempenho e precisão, possibilitando, portanto, uma solução para a definição de estratégias sobre a detecção de ameaças.

Como sugestões para trabalhos futuros, a partir dos resultados obtidos, pode-se acrescentar métodos para a diminuição das taxas de atributos na obtenção de melhores desempenhos e otimização sobre a veracidade dos algoritmos de aprendizagem de máquina. Além disso, conforme Sousa et al. (2022), é possível alterar o Quadro 1, para classificar as anomalias em tempo real, por meio da distribuição do cálculo gaussiano dos dados pelo tráfego obtido na rede, permitindo a detecção de ataques desconhecidos em ambientes de produção.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

Referências

ABRACADABRA. **Micro-and Macro-average of Precision, Recall and F-Score**. 2018. Disponível em: https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly--explained-b603420b292f Acesso em: 27 dez. 2022.

AGGARWAL, C. C. **Data Classification**: Algorithms and Applications. New York: CRC Press, 2014. *E-book*.

ALSUBHI, K.; ZHANI, M. F.; BOUTAVA, R. Embedded Markov process based model for performance analysis of Intrusion Detection and Prevention Systems. *In*: GLOBAL COMMUNICATIONS CONFERENCE, 2012, Anaheim. **Proceedings**... Anaheim: Globecom, 2012. Disponível em: https://www.researchgate.net/publication/261046303_Embedded_Markov_Process_based-Model_for_Performance_Analysis_of_Intrusion_Detection_and_Prevention_Systems. Acesso em: 2 ago 2023.

ANDRADE, M. S. **Monitoramento integrado de desempenho e segurança dos ativos de redes de computadores**. 2017. Trabalho de Conclusão de Curso (Sistemas de Informação) - Instituto Federal de Educação Ciência e Tecnologia de Sergipe – Lagarto, 2017. Disponível em: https://zabbixbrasil.org//files/TCC_MATHEUS_SANTOS_ANDRADE%20_IFS_SISTEMAS_DE_INFORMACAO_2018_FINAL.pdf. Acesso em: 20 set. 2022.

BARFORD, P. et al. A signal analysis of network traffic anomalies. *In*: PROCEEDINGS OF THE 2ND ACM SIGCOMM WORKSHOP ON INTERNET MEASURMENT, 2., 2002, Marseille.**Proceedings**... Marseille: ACM, 2002. Disponível em: https://www.researchgate.net/publication/2861593_A_Signal_Analysis_of_Network_Traffic_Anomalies/link/0deec517630d327262000000/download. Acesso em: 02 ago. 2023.

BELLMAN, E. R. **An introduction to artificial intelligence**: can computers think? Boston: Boyd & Fraser Pub. Co., 1978.

BHARGAVA, N. et al. Decision tree analysis on j48 algorithm for data mining. **International Journal of Advanced Research in Computer Science and Software Engineering**, v. 3, n. 6, p. 45-98, 2013.

CARVALHO, H. M. **Aprendizado de máquina voltado para mineração de dados**: árvores de decisão. 2014. Monografia (Engenharia de Software) – Universidade de Brasília – Brasília, 2014.

DENNING, D. E. An intrusion-detection model. **IEEE Transactions on Software Engineering**, v. 13, n. 2, p. 222-232, 1987.

FUNG, C. J.; BOUTABA, R. Design and management of collaborative intrusion detection networks. *In*: INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 13., 2013, **Proceedings**...IFIP/IEEE, 2013.

GUREK, E. L. **Aquisição de conhecimento em bancos de dados.** 2001. Trabalho de Conclusão de Curso (Tecnólogo em Processamento de Dados) – Universidade Tuiuti do Paraná, Santo Inácio, 2001.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

HUSSAIN, F. *et al*. Exception rule mining with a relative interestingness measure. *In*: HUSSAIN, F. *et al*. **Knowledge discovery and data mining**. Berlin: Springer Berlin Heidelberg, 2000. p. 86-97.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. *In*: PROCEEDINGS OF THE ELEVENTH CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 11., 1995, Montréal. **Proceedings**...Montréal: Morgan Kaufmann Publishers, 1995.

KIM, G.; LEE, S.; KIM, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. **Expert Systems with Applications**, v. 41, n. 4, p.1690-1700, 2014.

KOHAVI, Ron. The power of decision tables. *In:* KOHAVI, Ron. **Lecture Notes in Computer Science**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995. p. 174-189.

KUMAR, G. V.; REDDY, D. K. An agent based intrusion detection system for wireless network with artificial immune system (AIS) and negative clone selection. *In*: INTERNATIONAL CONFERENCE ON ELECTRONIC SYSTEMS, 2014, Nagpur. **Proceedings**...Nagpur: IEEE, 2014.

LARRAÑAGA, P. *et al*. Machine learning in bioinformatics. **Brief Bioinformatics**, v. 7, n. 1, p. 86-112, 2006.

LE, Q.; SMOLA, A.; VISHWANATHAN, S. Bundle methods for machine learning. **Advances in neural information processing systems**, v. 20, 2007.

LEU F. Y. *et al*. An internal intrusion detection and protection system by using data mining and forensic techniques. **IEEE Systems Journal**, v. 11, n. 2, p. 1-12, 2017.

LIBRALON, G. L. **Investigação de combinações de técnicas de detecção de ruído para dados de expressão gênica**. 2007. Dissertação (Mestrado em Ciências Matemáticas e de Computação) - Universidade de São Paulo, São Carlos, 2007.

MANYIKA, J. *et al*. **Big data**: the next frontier for innovation, competition, and productivity. [*s.l.*]: McKinsey Global Institute, 2011.

MITCHELL, T. **Machine learning**. New York: McGraw-Hill, 1997.

NORVIG, P.; RUSSEL, S. **Inteligência artificial**. Rio de Janeiro: Guanabara Koogan, 2013.

ÖZGÜR, A.; ERDEM, H. A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. **PeerJ Preprints**, 4:e1954v1, 2016.

PINHEIRO, J. M. dos S. Ameaças e Ataques aos Sistemas de Informação: prevenir e antecipar. **Cadernos UniFOA**, v. 3, n. 5, p. 11–21, 2017.

QUINLAN, J.R. **C4.5**: programs for machine learning. [*s.l.*]: Elsevier Science & Technology Books, 2014.

RAMA DEVI R. R., ABUALKIBASH, M. Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets - a review paper. **International Journal of Computer Science & Information Technology**, v. 11, n. 3, 2019.

DOI 10.26669/2448-4091.2023.388
V. 09 / 2023

SANTOS, V. Sistemas de Detecção de Intrusões usando unicamente softwares open source. **Sistema de Informação**, v. 10, 2010.

SCARFONE, K.; MELL, P. **Guide to intrusion detection and prevention systems (IDPS)**. Gaithersburg: NIST, 2007.

SMOLA, A. J.; SCHÖLKOPF, B. **Learning with Kernels**: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge: The MIT Press, 2001.

SOMMER, R.; PAXSON, V. Outside the closed world: On using machine learning for network intrusion detection. *In*: SYMPOSIUM ON SECURITY AND PRIVACY, 10., 2010, Washington. **Proceedings**... Washington: IEEE Computer Society, 2010.

SOUSA, R. M. *et al*. A New Approach for Including Social Conventions Into Social Robots Navigation by Using Polygonal Triangulation and Group Asymmetric Gaussian Functions. **Sensors**, v. 22, n. 12, p. 4602, 2022.

STOLFO J. *et al.*, Cost-based modeling for fraud and intrusion detection: Results from the JAM project. *In*: DARPA INFORMATION SURVIVABILITY CONFERENCE AND EXPOSITION, 2., 2000, Hilton Head. **Proceedings**... Hilton Head: IEEE Computer Society, 2000.

TARCA, A. L. *et al.* Machine learning and its applications to biology. **PLoS Computational Biology,** v. 3, n. 6, p. 953-963, 2007.

UTIMURA, L. N.; COSTA, K. A. Aplicação e análise comparativa do desempenho de classificadores de padrões para o sistema de detecção de intrusão Snort. *In:* SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 36., 2018. Porto Alegre. **Anais**… Porto Alegre: SBC, 2018.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C.J. **Data Mining**: practical machine learning tools and techniques. Chennai: Morgan Kaufmann, 2016.