SISTEMAS DE DETECÇÃO DE INTRUSÃO BASEADOS EM HONEYPOT E MACHINE LEARNING

Pedro Lucas de Souza
Pedro Augusto Scoton Alves
Gian Luca Monticeli
Prof. Dr. Thiago José Lucas

RESUMO

Com o crescente aumento anual de *malwares* presentes na internet, a sua detecção tornou-se uma necessidade para a segurança da rede interna dos meios corporativos. Diversos tipos de malwares são criados dia após dia, tanto para a obtenção ilegal de informações sigilosas em redes públicas e privadas como para causar danos em toda uma corporação. Portanto, a presente pesquisa visou realizar a detecção e classificação de malwares inseridos dentro de um *HoneyPot*, que atuou como uma armadilha, capturando amostras de malware e registrando o comportamento malicioso em ambiente controlado. Foi utilizado um dataset para a análise dos arquivos implementados no sistema e, com base na sua classificação, é separado para uma pasta segura ou para a quarentena. A junção dos dois métodos, tanto de detecção com o *Honeypot* como a classificação por aprendizado de máquina, proporcionou uma solução automatizada, eficiente e adaptável.

Palavras-chave: *HoneyPot*. *Malware*. Aprendizado de máquina. Detecção. Classificação.

ABSTRACT

With the increasing annual growth of malware on the internet, its detection has become a necessity for the security of internal corporate networks. Various types of malware are created daily, both for the illegal acquisition of confidential information on public and private networks and to cause damage to an entire corporation. Therefore, this research aimed to detect and classify malware embedded within a Honeypot, which acted as a trap, capturing malware samples and recording malicious behavior in a controlled environment. A dataset was used to analyze the files implemented in the system and, based on their classification as benign or malignant, they were separated into a secure folder or quarantine. The combination of the two methods, both detection with the Honeypot and classification by machine learning, provided an automated, efficient, and adaptable solution.

Keywords: *HoneyPot. Malware. Machine Learning. Detection. Classification.*

Discente – Curso Superior de Tecnologia em Segurança da Informação – Faculdade de Tecnologia de Ourinhos – Fatec Ourinhos – {pedro.souza92, pedro.alves17, gian.monticeli}@fatec.sp.gov.br
Professor Orientador – Curso Superior de Tecnologia em Segurança da Informação – Faculdade de Tecnologia de Ourinhos – Fatec Ourinhos – thiago@fatecourinhos.edu.br

1 INTRODUÇÃO

De acordo com o relatório 'Cost of a Data Breach 2023' da IBM, o custo médio global de uma violação de dados atingiu um recorde histórico de 4,45 milhões de dólares, sublinhando a necessidade crítica de mecanismos de defesa mais proativos e inteligentes. Muitas organizações de pequeno, médio e grande porte ao redor do mundo implementam medidas de proteção para assegurar a integridade de seus dados contra ataques maliciosos. Com o aumento dessas iniciativas, novos investimentos em estudos e no desenvolvimento de novas tecnologias são aplicados para a defesa das redes, ao mesmo tempo que são utilizadas para atacá-las, comprometendo a integridade dos dados e deixando redes públicas e privadas vulneráveis a futuros ataques, caso não sejam criados métodos de segurança adequados e eficientes para enfrentar as ameaças recebidas.

Com o avanço destas tecnologias de *malware* e intrusão, novos desafios focados em proteger os sistemas em geral contra intrusões e desenvolver aplicações de defesa robustas e adequadas para a segurança são atualizados constantemente. À medida que as ameaças se tornam mais sofisticadas, a criação de soluções de segurança e o estudo das ameaças são essenciais para garantir a integridade e confidencialidade dos dados, exigindo uma abordagem estratégica e proativa.

Diante deste problema, a presente pesquisa tem como objetivo a detecção do comportamento e de arquivos inseridos dentro de uma *Honeypot* e, por meio de um algoritmo de *Machine Learning* treinado, em caso de malware, classificá-los como benignos ou malignos e, por fim, separá-los em diretórios distintos dependendo da sua classificação. A abordagem e os objetivos que a pesquisa seguiu foram:

- Pesquisa de artigos robustos cientificamente, com foco em *Honeypot* e *Machine Learning* para extrair os diversos métodos utilizados.
- Documentação dos métodos que foram implementados no modelo utilizado.
- Implementação de um *Honeypot* como armadilha, com foco em ser um ambiente de fácil controle para análise.
- Implementação de um *Dataset* para a classificação dos arquivos em caso de malware (*Ransomware, Spyware*…).
- Extração dos resultados de desempenho obtidos com base na junção dos dois métodos utilizados e sua precisão.

O presente trabalho seguiu uma abordagem prática, utilizando pesquisas com alto teor científico com o intuito de obter conhecimento para o desenvolvimento e implementação de um modelo de detecção em *Honeypot* com controle de comportamento e obtenção de amostras de arquivos, e para a utilização de um *Dataset* para a classificação dos arquivos tanto malware como para inofensivos, visando aprimorar e robustecer a segurança da rede contra ataques malware.

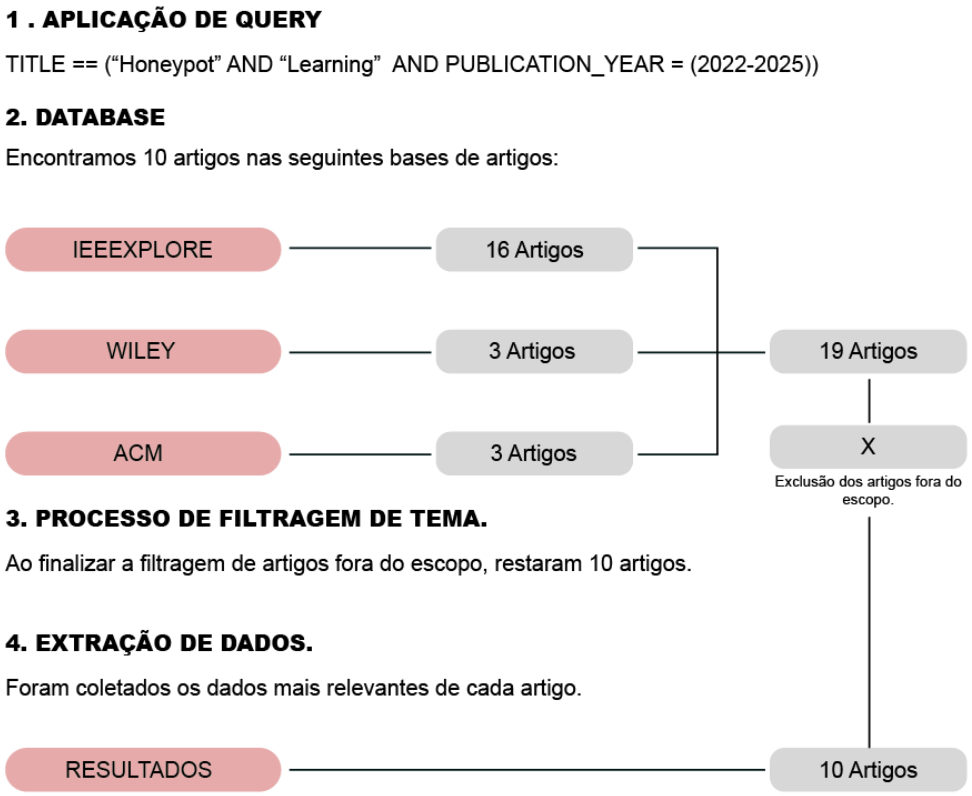
Segundo relatórios anuais da indústria, como o Verizon Data Breach Investigations Report (DBIR), o CrowdStrike Global Threat Report e o Mandiant M-Trends, que fornecem estatísticas concretas sobre o aumento da sofisticação de *malware*, torna-se essencial desenvolver estratégias de segurança avançadas. O método de *Honeypot* junto de uma Inteligência Artificial oferece uma visão aprofundada sobre as técnicas dos invasores em um ambiente controlado,

possibilitando a criação de defesas mais eficazes. A questão central da pesquisa foi: até que ponto a integração de um *honeypot* de baixa interação com classificadores de aprendizagem automática supervisionada (Random Forest, SVM, Decision Tree, KNN) pode melhorar a precisão da detecção e classificação automática de famílias comuns de *malware* (Ransomware, Spyware, Trojans)?

2 REFERENCIAL TEÓRICO

A presente seção apresenta os trabalhos correlatos relacionados ao tema da pesquisa. Foram identificadas diferentes abordagens, metodologias e descobertas em cada área pesquisada nos artigos.

Figura 1: Fluxo do referencial teórico aplicado.



Fonte: Elaborado pelo autor.

Foram obtidos 22 artigos no intervalo entre os anos de 2022 e 2025, com isso foram filtrados os temas mais relevantes para a pesquisa, chegando ao total de 10 artigos.

2.1 TRABALHOS CORRELATOS

Mudgal *et al.* (2024) apresentaram o *ML-IDHIF (Machine Learning Enabled Intrusion Detection Honeypot Intelligence Framework)*, um sistema que integra aprendizado de máquina (ML), redes neurais artificiais (ANN) e *Honeypots* MQTT para aprimorar a detecção de intrusões em redes. O sistema utiliza motores de decisão e de redirecionamento para identificar e desviar ataques, garantindo maior proteção da rede real. Os testes do ML-IDHIF mostraram alta precisão de 98,09%, superando *SVM* (92,76%), *Random Forest* (89,40%) e *Gaussian NB* (88,28%). O sistema também apresentou melhor desempenho em métricas de precisão (TACY, VACY) e menor perda de pacotes (TLOS, VLOS). A pesquisa conclui que o ML-IDHIF representa um avanço na detecção de intrusões, ao combinar inteligência adaptativa, resposta em tempo real e análise dinâmica de ataques. Mas destaca que o modelo precisa de mais capacidade do computador e dados variados para continuar funcionando com alta performance.

Papoutsis *et al.* (2022) apresentaram uma proposta para identificar padrões de ataques cibernéticos usando os registros de um *honeypot.* Para isso, foi utilizado o Aprendizado de Regras de Associação (ARL) junto com o algoritmo *FP-Growth*. O principal objetivo foi analisar os comandos feitos pelos invasores e encontrar padrões que ajudem na detecção de ataques. Os autores coletaram dados reais por 25 dias usando o *honeypot* Dionaea, que estava rodando na nuvem (AWS). Ao todo, foram capturados 16.472 comandos SQL. Depois de processar esses dados, encontram 259 conjuntos frequentes e 6.020 regras de associação. Após aplicar alguns filtros, sobraram 3.031 regras que tinham confiança de 100% e um *lift* máximo de 6,57, mostrando que existiam ligações fortes entre certos comandos. Essas regras revelaram sequências de ações feitas pelos atacantes, como criar objetos e acessar configurações de segurança, o que ajuda a entender melhor o comportamento deles. O método foi considerado eficaz, mas os autores alertam que a grande quantidade de regras pode dificultar a análise, sendo necessário filtrar as mais importantes. Por fim, a pesquisa mostra que o uso de ARL junto com *honeypots* é uma abordagem promissora para descobrir padrões reais de ataques e pode ser uma boa ferramenta para ajudar analistas de segurança a criar defesas mais eficientes.

Rangsdale *et al.* (2024) apresentam uma avaliação de modelos generativos de linguagem para a construção de *honeypots* interativos em uma implantação real. No estudo foi investigado se seria viável o uso de modelos de linguagem generativos para a simulação de terminais *Bash* de forma segura e eficiente, sem de fato executar comandos. Foram ao total implementados 3 modelos de *Honeypot*, um baseado em regras (*Cowrie*), um *honeypot* generativo simples (FEI-1) e por último um com mecanismo de seleção de contexto (FEI-2), ambos fazendo uso do GPT-3.5. Os modelos ficaram na internet por 14 dias de coleta, sendo analisado que os *Honeypots* generativos apresentaram 331,8% de tempo de interação comparados com o *honeypot* tradicional, e com a seleção de contexto em FEI-2 ocorreu a redução de uso de tokens em 45,6%, tornando o modelo com mais eficiência. Com isso, os resultados concluídos foram que os modelos generativos podem ser usados como métodos eficazes, promovendo maior engajamento com atacantes e com coleta de ameaças em tempo real.

Kristyanto *et al.* (2023) propuseram um estudo onde foi avaliado o impacto do aprendizado de reforço em *Honeypots* SSH com o uso da ferramenta *Cowrie* como

base. O principal objetivo foi fazer com que o *Honeypot* fosse capaz de se adaptar com base no comportamento dos atacantes, aumentando o tempo de interação e a qualidade e quantidade de dados coletados. O sistema fez o uso do algoritmo *Depp Q-Net* junto do *Honeypot*, com a capacidade de permitir, atrasar, bloquear, falsificar a saída de comandos ou insultando o invasor em sua linguagem com base em seu endereço IP. A sua função de recompensa foi projetada para estimular o *Honeypot* a levar o invasor até a execução de comandos de *download* para a análise de ataques. Os resultados obtidos indicaram que o Honeypot com aprendizado de reforço aumentou a duração média das sessões em 1,57 segundos para 5,28 segundos, concluindo um maior engajamento dos invasores. Além do resultado foi detectado padrões recorrentes de comandos antes das atividades de *download*, exibindo as estratégias dos invasores para reconhecimento de seu alvo.

Mahajan *et al.* (2023) propuseram um sistema moderno de detecção e análise de *malware* que combina *honeypots* modernos com algoritmo de aprendizado de máquina, visando superar limitações dos métodos tradicionais como antivírus e IDS baseados em assinatura, que falham diante da rápida evolução do *malware.* A arquitetura proposta utiliza o *Modern Honey Network* (MHN) com sensor Dionaea, o mesmo sensor oferece serviços vulneráveis (FTP, SMB, TFTP) para capturar amostras de *malware*, os dados capturados e coletados incluem IPs, datas, protocolos, portas e amostras binárias que serão utilizadas para treinamento supervisionado. A classificação foi feita utilizando algoritmos como *Random Forest, KNN* e *Decision Tree*, tais algoritmos classificam dados como maliciosos ou não maliciosos, com base em características extraídas das amostras, com base em resultados experimentais apresentados, foram registrados um total de 371 ataques e um total de 302 amostras maliciosas (78%) e 69 não maliciosas (22%) coletadas. Foi notado que o algoritmo de *Random Forest* obteve o melhor desempenho geral, superando modelos anteriores em até 6% no *F1-Score* e 5% na acurácia. O estudo apresenta um sistema eficaz para capturar e classificar *malware* com alta precisão e capacidade de adaptação, sua arquitetura baseada em *honeypots* e aprendizado de máquina oferece uma solução escalável e robusta. O mesmo sugere que futuras melhorias podem incluir *honeypots* de alta interação e uso de conjuntos de dados maiores para aumentar a eficiência da detecção.

Siddique *et al.* (2024) apresentaram um sistema de segurança cibernética que integra *honeypots* com aprendizado de máquina para identificar, prever e impedir ataques, com o objetivo central de criar um agente inteligente que analisa automaticamente dados coletados por honeypots para prever perfis de atacantes e antecipar futuras ameaças. O sistema tem como objetivo utilizar *honeypots* para capturar comportamentos maliciosos, aplicar algoritmos de *machine learning* para prever e classificar ataques, comparar os resultados de sistemas com e sem o uso de *honeypots* e agentes inteligentes. Para cumprir os objetivos foi utilizada uma metodologia em que se usa do *Pentbox Honeypot* em ambiente de teste com duas máquinas (M1 = *Honeypot*, M2 = atacante), onde se captura o tráfego com *Wireshark* e se utiliza um *dataset* com 219.820 fluxos de rede contendo Src_IP, Dst_IP, entre outros. Para execução do estudo, foram aplicados quatro modelos de aprendizado de máquina como *Decision Tree* com uma acurácia de 95,5%, tendo o melhor desempenho geral, *Random Forest* com acurácia de 94,2%, *Support Vector Machine* (SVM) com acurácia de 94,1% e *Generalized Linear Model* (GLM) com

acurácia de 93,8%, os modelos foram treinados e avaliados no *RapidMiner*, com pré-processamento, divisão de dados e aplicação dos algoritmos para prever o valor de *Flow_Ptks/s*. Foram notados resultados experimentais que mostraram que o uso de *honeypot* resultou em bloqueio ativo e coleta de dados do invasor, como IP de origem, comprimento de pacotes e *timestamp*, o que não ocorreu sem o uso de *honeypot*. A pesquisa aponta que a integração de *honeypots* com aprendizado de máquina proporciona alta precisão na detecção de ataques, melhora a capacidade preditiva contra ameaças futuras e fortalece a segurança geral do sistema

Manjunatha *et al.* (2024) apresentaram uma abordagem inteligente para a detecção e prevenção de ataques de injeção de SQL, utilizando *honeypots* e aprendizado de máquina, com foco no uso de *Random Forest*, além de outros modelos de Aprendizado de Máquina como *Gradiente Boosting*, *KNN* e *Decision Tree*, todos com AUC. O estudo tem como objetivo principal detectar e redirecionar tentativas de *SQL Injection* em tempo real, utilizando *honeypots* para capturar e analisar o comportamento dos atacantes, aumentando então a segurança de aplicações web com validação inteligente de entrada. Os resultados apresentados foram satisfatórios por apresentarem uma acurácia de 0,95 (treino) e 0,76 (teste) referente ao modelo principal utilizado (*Random Forest*), fornecendo uma *Precision-Recall* média de 0,93 e uma AUC de 0,95, os demais modelos mostraram resultados também satisfatórios de certo modo. Pode-se notar que a combinação de *Random Forest* com *honeypots* e validação por padrão representam uma solução eficaz para combater ataques *SQL Injection*. A abordagem oferece segurança proativa e dupla camada de proteção, com capacidade de identificar e banir invasores automaticamente.

Yepez *et al.* (2022) propuseram um estudo de um modelo de *honeypot* inteligente que aumenta o engajamento de atacantes em ambientes SSH usando *Deep Reinforcement Learning* - DRL e Incorporações semânticas de padrões de ataque com *Word2Vec*. O estudo visa melhorar a interação entre *honeypots* e atacantes, prolongando a sessão para aumentar a coleta de dados e compreender melhor os padrões de ataques, foram utilizadas 209.653 sessões de ataques para pré-treinamento, 285.557 sessões reais de ataque para treinamento com atacantes reais e 6.048.897 *tokens* processados com *Word2Vec* com um vocabulário de 1.081 *tokens* distintos resultando em 16 *cluster* semânticos organizados usando *K-means.* Com tais resultados, foram aplicadas técnicas de DQN (*Double Deep Q-Network*) com memória e comparados com *Honeypots* passivos (apenas *ALLOW*) e *Honeypots* com *Q-Learning* tradicional. Foi apresentado que o *honeypot* com DQN e memória aumentou o engajamento dos atacantes em mais de 5 vezes o desvio padrão inicial, *Honeypots* com *Q-Learning* ou passivos permaneceram abaixo de 2 desvios padrão. Foi constatado que o uso de DRL com vetores semânticos permite que o *honeypot* reaja de forma mais inteligente, mantendo o atacante vivo por mais tempo e coletando dados mais ricos e precisos, demonstrando uma abordagem mais escalável e aplicável a outros protocolos baseados em texto além de SSH.

Gao *et al.* (2024) apresentaram que as redes definidas por *software* (SDN) são vulneráveis a ataques cibernéticos como DDoS e *anti-honeypot*, exigindo estratégias de defesa mais avançadas. Foram utilizados *honeypots* para atrair atacantes, podendo ser identificados e evitados por invasores experientes, como maneira de confundir os atacantes, surgem os pseudo-*honeypots*, servidores reais

que se disfarçam como *honeypots*. Pensando em lidar com esses cenários complexos, o estudo propõe o RaRL (*Risk-aware Reinforcement Learning*), um modelo de defesa que utiliza aprendizado de reforço seguro (SRL). O sistema decide dinamicamente entre três estratégias de defesa: Serviço normal, *Honeypot* e Pseudo-*honeypot*. O estudo apresentou que o RaRL aumentou a utilidade do sistema em 17,5% em relação ao *QLearning*, e em 142,4% em relação ao método aleatório, o qual reduziu o risco do sistema em 42,7% (vs *QLearning*) e em 59,6% (vs método aleatório). Foi testado em 60 execuções ao longo de 1.500 ciclos de tempo, gerando resultados que se mantiveram superiores mesmo com aumento de servidores e variação na probabilidade de ataque. Conclui-se que o modelo apresentou ser mais eficaz, adaptável e seguro em relação a métodos anteriores, porém enfrentando desafios de estabilidade e aplicação em ambientes reais por conta da variação dinâmica de servidores.

Ahmed *et al.* (2024) defendem que a segurança em ambientes de cidades inteligentes é desafiada pela crescente adoção de dispositivos da Internet das Coisas (IoT), que são alvos fáceis de ataques cibernéticos devido a falhas estruturais, uso de senhas padrão e baixa capacidade de processamento. É apresentado também que os métodos tradicionais de segurança, como *firewalls* e IDS convencionais, não são suficientes para lidar com a sofisticação crescente dos ataques em ambientes IoT. Com isso o estudo baseia-se no *dataset iTrust*, que foi gerado por *honeypots* implantados durante 1,5 ano utilizando 40 endereços IP públicos captando tráfego direcionado a 11 dispositivos IoT reais gerando mais de 81,5 milhões de registros. Foram utilizados algoritmos como *Decision Tree* o qual obteve desempenho próximo a 100% de precisão, *Naïve Bayes*, KNN e para redes neurais foram utilizadas SNN e LSTM nos quais entregaram resultados satisfatórios. Conclui-se que a integração entre *honeypots* e aprendizado de máquina é uma abordagem promissora para detecção e mitigação de ataques em redes IoT, mas ainda existem desafios relacionados à falta de dados públicos e rotulados.

2.2 TAXONOMIA

A presente seção apresenta a taxonomia dos trabalhos correlatos coletados dos artigos selecionados, nela é destacado os *Datasets* utilizados pelos autores e seus classificadores da pesquisas.

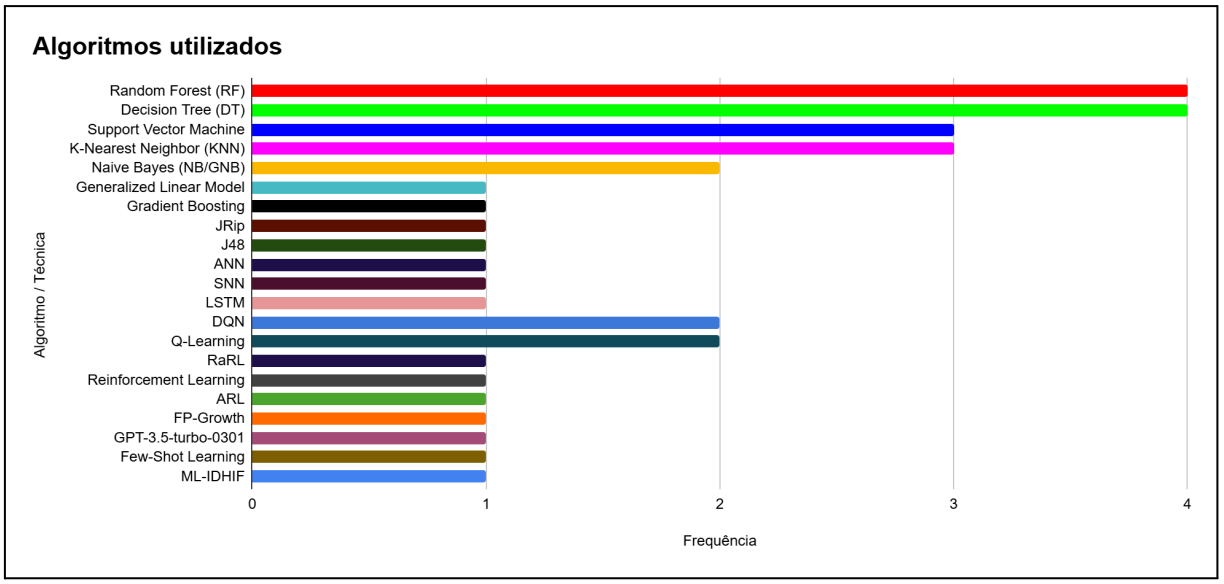
Tabela 1: Dados coletados a partir dos trabalhos correlatos.

TRABALHO	DATASET	CLASSIFICADORES	ACURÁCIA/RESULTADOS
Mudgal et al. (2024)	KDDCup 99 Dataset	ML-IDHIF, SVM, RF, GNB	0,921325
Papoutsis et al. (2022)	Próprio	ARL, FP-Growth	Lift médio ≈ 6,0

Rangsdale et al. (2024)	Próprio	GPT-3.5-turbo, Few-Shot Learning	Média de duração da sessão (FEI-2): 4,47 comandos por sessão
Kristyanto et al. (2023)	Próprio	Reinforcement Learning, DQN	Duração média da sessão RL: 5,28 segundos
Mahajan et al. (2023)	Próprio	RF, KNN, DT	0,92983
Siddique et al. (2024)	Próprio	GLM, SVM, RF, DT	0,944
Manjunatha et al. (2024)	Próprio	RF, GB, DT, KNN, JRip, J48,SVM, ANN	0,81875
Yepez et al. (2022)	Próprio	QL, DQN	Média de engajamento 1,325
Gao et al. (2024)	N/A	RaRL, QL	Média de utilidade RaRL 0,7995
Ahmed et al. (2024)	Próprio	NB, DT, KNN, SNN, LSTM	0,962

Fonte: Elaborado pelo autor

Figura 2: Quantidade de algoritmos utilizados em gráfico.



Fonte: Elaborado pelo autor

Embora a taxonomia proposta não inclua métricas como falsos positivos e falsos negativos, observa-se que a maioria dos estudos analisados alcançou níveis

satisfatórios de acurácia ou de suas respectivas medidas utilizadas, o que indica a existência de margem para aprimoramento dos modelos. Os trabalhos revisados empregam uma variedade de métodos, com ênfase em abordagens supervisionadas e soluções híbridas, evidenciando que a eficácia dos modelos está mais relacionada à adequação ao conjunto de dados utilizado do que propriamente ao tipo de técnica adotada. Ademais, destaca-se a utilização de modelos combinados, como os baseados em *Decision Tree* (DT) e *Support Vector Machine* (SVM) , que apresentam maior estabilidade e desempenho, especialmente em contextos mais complexos ou com dados desbalanceados.

3 METODOLOGIA

Esta Pesquisa caracteriza-se como aplicada, com abordagem quantitativa e experimental. A escolha por esse tipo de estudo justifica-se pela necessidade de desenvolver e avaliar uma solução prática e mensurável para a detecção e classificação de malwares, integrando duas frentes tecnológicas: a utilização de honeypots para a coleta de arquivos e o uso de *machine learning* para sua classificação.

3.1 TESTE DE DESEMPENHO INICIAL

A primeira fase da pesquisa tem como objetivo, a análise e seleção de um *Dataset* principal para a validação acadêmica dos algoritmos. No total foram coletados três *Datasets* diferentes para o início da primeira fase, sendo eles:

- *Malware Datasets*, desenvolvido por Venugopal Adep, que contém uma ampla variedade de amostras de malwares organizadas por tipo e comportamento;
- *Malware Memory Analysis* (CIC-MalMem-2022), criado por Tristan Carrier, Princy Victor, Ali Tekeoglu e Arash Habibi Lashkari, voltado à análise comportamental de *malwares* em memória, especialmente em ambientes *Windows*.
- Dike*Dataset*, elaborado por George-Andrei Losif, com foco na detecção de ameaças a partir de artefatos de rede.

Os testes consistem na avaliação do desempenho de algoritmos de *Machine Learning* em cada um dos *Datasets* para tarefas de classificação. Os resultados dos testes aplicados nesta fase foram:

Tabela 2: Resultado dos testes CIC-Malmem-2022

ALGORITMO	ACURÁCIA	PRECISÃO	RECALL	F1_SCORE	CLASSIFICAÇÃO
Random Forest (RF)	99.99%	100%	100%	100%	Binária
Random Forest (RF)	~88.70%	~88%	~88%	~88%	Multi-classe

Decision Tree (DT)	99.99%	100%	100%	100%	Binária
Support Vector Machine (SVM)	99.95%	~99,9%	~99,9%	~99,9%	Binária
K-Nearest Neighbors (KNN)	99.95%	~99,9%	~99,9%	~99,9%	Binária

Fonte: Elaborado pelo autor

Tabela 3: Resultado dos testes *DikeDataset*

ALGORITMO	ACURÁCIA	PRECISÃO	RECALL	F1-SCORE	CLASSIFICAÇÃO
Random Forest (RF)	96.00%	99.00%	~97,0%	98.00%	Binária
Decision Tree (DT)	~93-95%	~94%	~94%	~94%	Binária
Support Vector Machine (SVM)	~92-94%	~93%	~93%	~93%	Binária
K-Nearest Neighbors (KNN)	~92-95%	~92%	~92%	~92%	Binária

Fonte: Elaborado pelo autor

Tabela 4: Resultado dos testes *Malware Datasets - Adep*

ALGORITMO	ACURÁCIA	PRECISÃO	RECALL	F1-SCORE	CLASSIFICAÇÃO
Random Forest (RF)	~95-97%	~96%	~96%	~96%	Binária / Multi-classe
Decision Tree (DT)	~94-96%	~95%	~95%	~95%	Binária / Multi-classe
Support Vector Machine (SVM)	~90-93%	~91%	~91%	~91%	Binária / Multi-classe

K-Nearest Neighbors (KNN)	~90-94%	~92%	~92%	~92%	Binária / Multi-classe

Fonte: Elaborado pelo autor

Conforme os resultados apresentados, foi selecionado o *Dataset CIC-MalMem-2022* como base principal para a análise acadêmica, por ter obtido o melhor desempenho geral e alinhamento com os objetivos do projeto.

3.2 TOPOLOGIA E FLUXO

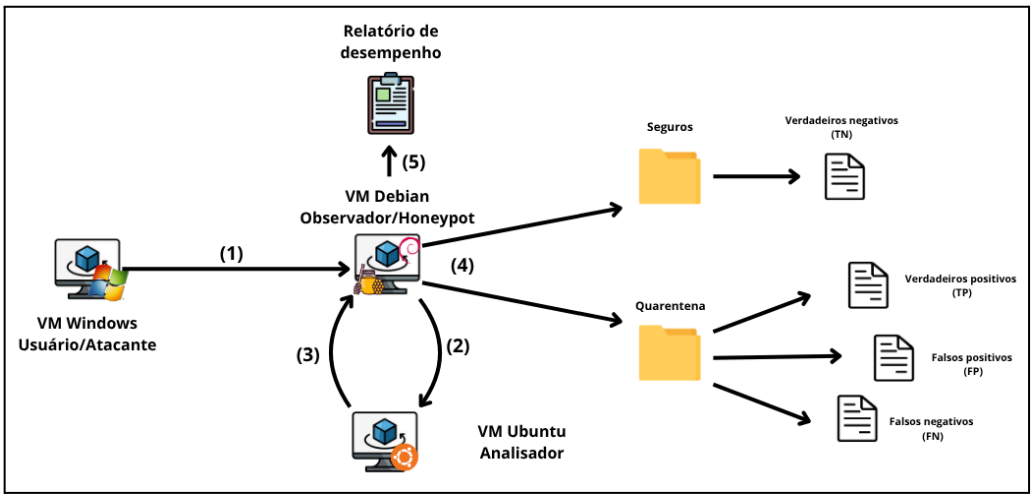
A fase experimental do projeto consistiu na implementação de um ambiente de teste prático para validar a eficácia do sistema de *HoneyPot* integrado ao modelo de *Machine Learning*. A metodologia seguiu o cenário de "Monitor de Sistema de Ficheiros", utilizando a arquitetura de três VMs previamente definida.

O teste foi conduzido ao longo de uma semana simulada em um ambiente controlado:

- *VM HoneyPot* (Debian): Executar um script "observador" (watcher.py) que monitora pasta-armadilha partilhada na rede (*/srv/honeypot/entrada*)
- *VM* Analisadora (Ubuntu): Hospedou a *API* de análise (analyzer.api.py) que carregava um modelo *Random Forest* treinado para análise estática de ficheiros. Este modelo foi treinado com amostras do MalwareBazaar (maliciosos) e uma combinação de DikeDataset e ficheiros de sistema (benignos).
- *VM* Cliente/Atacante (Windows): Foi utilizada para enviar um fluxo contínuo e automatizado de ficheiros benignos e maliciosos para a pasta-armadilha

O fluxo consiste na detecção de um novo ficheiro pelo observador, envio para a *API* de análise, classificação pelo modelo de *ML* e movimentação automática do ficheiro para uma pasta de "seguros" ou de "quarentena".

Figura 3: Fluxo de operação do modelo implementado.



Fonte: Elaborado pelo autor

A Figura 3 apresenta todo o fluxo proposto na pesquisa, sendo composta por 3 máquinas virtuais (VMs) e suas respectivas sequências de comunicação. A numeração exibida de 1 a 5 indica a sequência lógica da etapa de cada processo, desde a origem até o destino de todo o tráfego. Inicialmente, a VM Windows atua como o ponto de partida de todo o processo, sendo responsável por anexar e enviar os arquivos para a VM Debian (1). Após o envio, a VM Debian fica responsável por hospedar a aplicação *honeypot* e serve como observador durante o tráfego do arquivo, direcionando-o para a VM Ubuntu para a sua análise. O arquivo é, então, submetido a uma análise e classificação pela VM Ubuntu, utilizando o dataset (CIC-MalMem-2022) para essa operação, gerando assim o resultado de maligno ou benigno. Após a análise ele é direcionado novamente para a VM Debian (3). Na etapa (4), a VM Debian aloca o arquivo em uma das duas categorias, com base no resultado gerado pelo analisador:

- Para a pasta de Seguros são direcionados os Verdadeiros Negativos (TN), que são os arquivos benignos corretamente classificados.
- Para a pasta de Quarentena são direcionados os Verdadeiros Positivos (TN) arquivos maliciosos corretamente classificados, Falsos Positivos (FP) arquivos malignos incorretamente classificados como benignos, e Falsos negativos (FN) arquivos benignos incorretamente classificados como malignos.

Por fim, após o direcionamento, a VM Debian gera um Relatório de Desempenho (5), exibindo a contagem de arquivos para cada métrica buscada: FN, FP, TN e TP.

4 RESULTADOS FINAIS

A presente seção apresenta os resultados quantitativos e qualitativos obtidos durante toda a fase experimental do projeto.

4.1 RESULTADOS QUANTITATIVOS

Durante o período de teste, um total de 1.542 ficheiros foram processados pelo sistema. A tabela abaixo resume o desempenho do modelo de detecção em ação.

Tabela 5: Resultado do desempenho do modelo

MÉTRICA	CONTAGEM	DESCRIÇÃO
Total de ficheiros analisados	1.542	Volume total de amostras processadas pelo sistema.
Ficheiros benignos reais	1.215	Ficheiros legítimos enviados para a pasta.
Ficheiros maliciosos	327	Amostras de malware de diferentes

reais		famílias.
Verdadeiros positivos (TP)	318	Malware corretamente identificado e movido para quarentena.
Verdadeiros negativos (TN)	1.198	Ficheiros benignos corretamente identificados e movidos para a pasta de seguros.
Falsos positivos (FP)	17	Ficheiros benignos incorretamente classificados como malignos.
Falsos negativos (FN)	9	Ficheiros malware incorretamente classificados como benigno.

Fonte: Elaborado pelo autor

Com base nos dados apresentados durante a fase de testes, foram calculados as seguintes métricas de desempenho do sistema em um cenário prático:

Tabela 6: Métrica de desempenho do modelo

MÉTRICA DE DESEMPENHO	RESULTADO
Acurácia Geral	98,31%
Precisão (Qualidade dos alertas)	94,93%
Recall (Capacidade de detecção)	97,25%
F1-Score (Equilíbrio geral)	96,08%

Fonte: Elaborado pelo autor

4.2 ANÁLISE QUALITATIVA DOS RESULTADOS

Os resultados demonstram a alta eficácia do sistema, mas uma análise das falhas revela pontos importantes:

- Análise dos Falsos Positivos (17 ficheiros): A investigação manual revelou que estes erros não foram aleatórios. Os ficheiros benignos classificados incorretamente como maliciosos incluíam principalmente instaladores de software que utilizam packers (compactadores) semelhantes aos de malware e ferramentas de administração de sistema, cujo comportamento pode ser confundido com spyware. Isto indica que o modelo é sensível a ficheiros “no limite” da normalidade, um comportamento esperado.

- Análise dos Falsos Negativos (9 ficheiros): As 9 ameaças não detectadas representam o risco mais crítico. A análise mostrou que estes ficheiros eram, em sua maioria, variantes “zero-day” (ameaças muito novas e não catalogadas) ou malwares polimórficos, projetados especificamente para alterar sua estrutura e evadir a análise estática. O sistema falhou precisamente contra as ameaças mais avançadas, para as quais a análise estática é menos eficaz.

Os resultados validam a abordagem do projeto, mostrando que a integração de um *HoneyPot* com um modelo de *Machine Learning* treinado para análise estática é uma solução prática e altamente eficaz para detectar a grande maioria das ameaças conhecidas. As falhas observadas reforçam a importância de treinar continuamente o modelo com novas amostras e a necessidade de múltiplas camadas de defesa, como a análise de memória, para combater ameaças mais sofisticadas.

5 CONCLUSÃO

O objetivo central deste trabalho foi avaliar a eficácia da integração de um honeypot com classificadores de Machine Learning para a detecção e classificação de malwares. A metodologia foi estruturada em duas frentes complementares: uma análise teórica para validação científica e uma implementação prática para testar o sistema em um cenário simulado. A primeira fase, focada na análise teórica, confirmou a superioridade do algoritmo Random Forest. Ao ser testado com o dataset de análise de memória CIC-MalMem-2022, o Random Forest alcançou 88,70% de acurácia na complexa tarefa de classificação multi-classe (Ransomware, Spyware, Trojan), validando cientificamente a sua escolha para a implementação do sistema prático.

A segunda fase, de demonstração prática, implementou um honeypot em uma arquitetura de 3 VMs, utilizando um modelo Random Forest treinado para análise estática de ficheiros. Os resultados da simulação de uma semana foram robustos: ao processar 1.542 amostras, o sistema atingiu uma acurácia geral de 98,31% e uma capacidade de deteção (Recall) de 97,25%. A análise qualitativa das 9 falhas de detecção (Falsos Negativos) revelou que estas ocorreram em malwares avançados, como polimórficos e zero-day, que são projetados para evadir a análise estática. Esta constatação, alinhada com a literatura, não invalida a solução, mas sim reforça a importância de múltiplas camadas de defesa e do retreinamento contínuo para combater o *concept drift*. Diante do exposto, este trabalho responde afirmativamente à questão de pesquisa, concluindo que a integração proposta é uma solução funcional e altamente eficaz, unindo com sucesso a coleta passiva de ameaças à análise inteligente em tempo real.

REFERÊNCIAS

AHMED, Y.; KEHINDE BEYIOKU; YOUSEFI, M. Securing smart cities through machine learning: A honeypot‐driven approach to attack detection in Internet of Things ecosystems. *IET smart cities*, 29 maio 2024.

AKSHAY MUDGAL; BHATIA, S. Machine Learning and Artificial Neural Network Enabled Intrusion Detection Honeypot Intelligence Framework (ML-IDHIF). *International Conference On Computing Communication And Networking Technologies (Icccnt)* p. 1–6, 24 jun. 2024.

A, M. B. et al. Intelligent Defense Strategies: Machine Learning-Enhanced SQL Injection Detection and Prevention via Honeypots. *International Conference On Intelligent Algorithms For Computational Intelligence Systems.* p. 1–7, 23 ago. 2024.

GAO, D. et al. Risk‐Aware SDN Defense Framework Against Anti‐Honeypot Attacks Using Safe Reinforcement Learning. *International Journal of Network Management*, v. 34, n. 6, 16 set. 2024.

KRISTYANTO, M. A.; STUDIAWAN, H.; PRATOMO, B. A. Evaluation of Reinforcement Learning Algorithm on SSH Honeypot. 2022 6th *International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, p. 346–350, 13 dez. 2022.

LOPEZ-YEPEZ, J. S.; FAGETTE, A. Increasing attacker engagement on SSH honeypots using semantic embeddings of cyber-attack patterns and deep reinforcement learning. 2021 IEEE *Symposium Series on Computational Intelligence (SSCI*), 4 dez. 2022.

MAHAJAN, V.; SINGH, J. Malware Detection and Analysis using Modern Honeypot Allied with Machine Learning: A Performance Evaluation. *International Conference on Electronics and Sustainable Communication Systems (ICESC)* 6 jul. 2023.

PAPOUTSIS, A. et al. Host-based Cyber Attack Pattern Identification on Honeypot Logs Using Association Rule Learning. 2022 IEEE *International Conference on Cyber Security and Resilience (CSR)*, p. 50–55, 27 jul. 2022.

RAGSDALE, J.; RAJENDRA BOPPANA. Evaluating Few-Shot Learning Generative Honeypots in A Live Deployment. *International Conference On Cyber Security And Resilience (Csr).* v. 35, p. 379–386, 2 set. 2024.

SIDDIQUE, M. R. et al. Integrating Machine Learning-Powered Smart Agents into Cyber Honeypots: Enhancing Security Frameworks. *International Conference For Convergence In Technology (I2Ct)*. 5 abr. 2024.