SISTEMA DE DETECÇÃO DE ATAQUE EM REDES INDUSTRIAIS QUE UTILIZAM PROTOCOLO PROFINET

Oswaldo Felipe da Silva Costa e Wesley Candido da Silva*

RESUMO

Este artigo busca detalhar o desenvolvimento, aplicabilidade e análise de um sistema de detecção de ataques cibernéticos em redes industriais utilizando o protocolo *PROFINET*. Ao longo do estudo, foi simulado um ambiente industrial utilizando CLP, IHM e uma bancada experimental, configurados a partir do software TIA Portal. Foi realizada coleta de tráfegos de dados com o *sniffer* de rede *Wireshark*, analisando as condições normais e anômalas, gerado por um disparador de ataque do tipo *ARP Spoofing*, implementado por meio da ferramenta *Ettercap* operando em plataforma Kali Linux. A partir da coleta, foi elaborado um *Dataset, posteriormente* submetido a um pré-processamento, normalização de dados e aplicado a um modelo de *Machine Learning* por meio do algoritmo *Random Forest*, desenvolvido no ambiente *MATLAB*, gerando métricas para avaliar o desempenho nos sistemas desenvolvidos. O resultado apresentado foi satisfatório, pois todas as métricas alcançaram altos valores de assertividade, evidenciando um modelo de classificação robusto e equilibrado, atendendo à expectativa de eficácia do sistema.

Palavras-chave: Detecção de Ataques; Indústria 4.0; Machine Learning; Profinet.

ATTACK DETECTION SYSTEM ON INDUSTRIAL NETWORKS USING PROFINET PROTOCOL

ABSTRACT

This article seeks to detail the development, applicability and analysis of a system for detecting cyberattacks in industriais networks using the *PROFINET* protocol. Throughout the study, an industrial environment was simulated using PLC, HMI and an experimental bench, configured from the TIA Portal software. Data traffic was collected with the *Wireshark* network *sniffer*, analyzing the normal and anomalous conditions, generated by an ARP *Spoofing* attack trigger, implemented through the *Ettercap tool* operating on Kali Linux platform. From the collection, a Dataset was elaborated, *later* submitted to pre-processing, data normalization and applied to a *Machine Learning* model through the *Random Forest* algorithm, developed in the *MATLAB* environment, gene rating metrics to evaluate the performance in the developed systems. The result presented was satisfactory, as all metrics achieved high values of assertiveness, evidencing a robust and balanced classification model, meeting the expectation of effectiveness of the system.

Key words: Attack Detection; Industry 4.0; Machine Learning; Profinet.

* Autor correspondente (e-mail): wesley.candido@sistemafiep.org.br

1 INTRODUÇÃO

A indústria 4.0 é uma expansão na comunicação, automação moderna, controle, monitoramento de máquinas, sistemas em processos produtivos e manufatura (Turcato, 2020). Seu conceito teve início em 2011 após um evento em Hannover, na Alemanha, no qual houve uma proposta tecnológica que busca conciliar produtividade e competitividade. A finalidade do evento foi integrar as tecnologias avançadas, tais como inteligência artificial, Big Data, robótica e sensores inteligentes, conectados por meio da internet, criando uma cadeia de valor conectada (Mota, *et al.,* 2022). Diante desse contexto, compreende-se a necessidade de atender às demandas atuais, principalmente em redes industriais.

Redes industriais são definidas como sistemas de comunicação desenvolvidos em ambientes industriais com a finalidade de integrar os dispositivos da IoT (Internet das Coisas), possibilitando a coleta, o processamento e a análise de dados em tempo real, o que possibilita aprimorar a produção, diminuir custos e fortalecer a competitividade (Amo; Roepke, 2024). Assim, a rede PROFINET se destaca pela sua comunicação em tempo real e pela facilidade de identificar os problemas nas redes (Demétrio; Junior; Psiciotta, 2024).

Com a expansão na comunicação industrial, há uma vulnerabilidade e exposição de dados e informações que necessita de modelos de negócios voltados à proteção de computadores, dispositivos industriais e servidores contra os ataques cibernéticos (Mota, *et al.,* 2022). Vale destacar o mais recente relatório de detecções da *Kaspersky* (CERT 2023), que apresentou que 40,6% dos computadores em Sistemas de Controle Industrial (ICS) protegidos pela empresa foram alvo de atividades maliciosas (Nicolaio; Muranetto; Fonseca, 2023).

De acordo com a FIESP (Federação das Indústrias do Estado de São Paulo), 30% das indústrias já foram alvos de ataques cibernéticos (Melo, 2025). A partir deste cenário, a pesquisa teve como motivação responder ao seguinte questionamento: Como a rede PROFINET, utilizada em ambientes industriais, pode ser protegida contra os ataques cibernéticos?

Para responder ao questionamento, o objetivo geral do estudo foi analisar o fluxo de dados na rede, identificar e classificar os ataques e inconsistências em redes industriais com protocolo PROFINET, garantindo maior segurança para os sistemas produtivos. Os objetivos específicos foram: implementar um ambiente experimental para coleta de dados sobre tentativas de ataques cibernéticos em redes PROFINET; aplicar técnicas de aprendizado de máquina, como o *Random Forest*, para a detecção de ataques em redes industriais; e avaliar a eficácia dos classificadores de aprendizado de máquina no contexto de segurança de redes industriais.

Sendo assim, para atingir os objetivos, foi proposta uma implementação de uma simulação industrial, utilizando o TIA Portal para configuração de uma rede PROFINET com CLP, IHM e esteira simulada. A coleta de dados foi realizada por meio do *Wireshark*, lendo os pacotes sem e com os ataques gerados pelo *Ettercap* à rede. Na etapa de análise, foi aplicada a técnica de aprendizado de máquina Random Forest, utilizando o ambiente MATLAB para treinamento, teste e avaliação do modelo. Dessa forma, o estudo buscou garantir soluções para a cibersegurança industrial, fortalecendo os processos produtivos da Indústria 4.0.

2 FUNDAMENTAÇÃO TEÓRICA

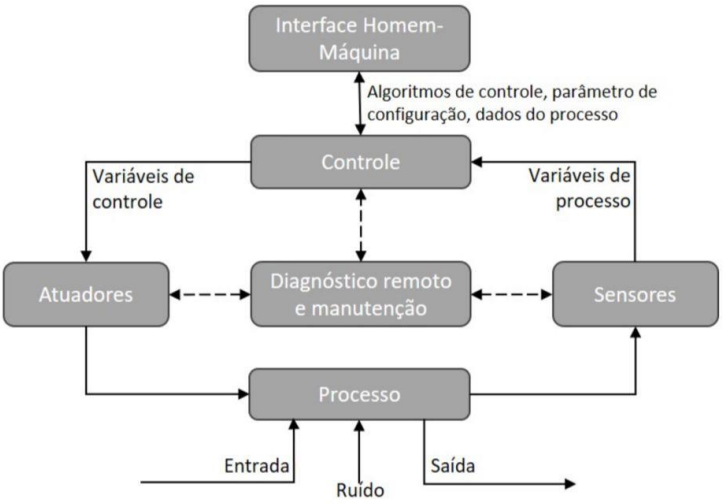
2.1 Redes industriais e os protocolos de rede

As redes industriais são essenciais para a indústria 4.0, conectando os dispositivos, sensores, máquinas de controle e possibilitando a troca de informação em tempo real e o monitoramento remoto (Amo; Roepke, 2024). Impulsionadas pela integração entre Tecnologia da Informação (TI) e Tecnologia Operacional (OT), elas são complexas com atuadores, controladores lógicos programáveis (PLCs), unidades terminais remotas (RTUs) e sistemas SCADA (*Supervisory Control and Data Acquisition)* (Nicolaio, 2024).

A arquitetura PERA (*Purdue Enterprise Reference Architecture*) organiza os sistemas industriais com sensores e atuadores, responsáveis pela interação com o

processo físico. No nível acima, observam-se os CLPs e RTUs, os que realizam a coleta e o controle de dados. Em outra camada, se encontram as IHMs, aquelas que visualizam e comandam as operações. Nos níveis quatro e cinco, é a rede corporativa que gerencia as informações empresariais e dá o suporte às operações. Esse exemplo está ilustrado na **Figura 1**:

Figura 1 - Fluxo típico de uma rede OT



Fonte: Adaptado de Stouffer (2023 *apud.* Nicolaio)

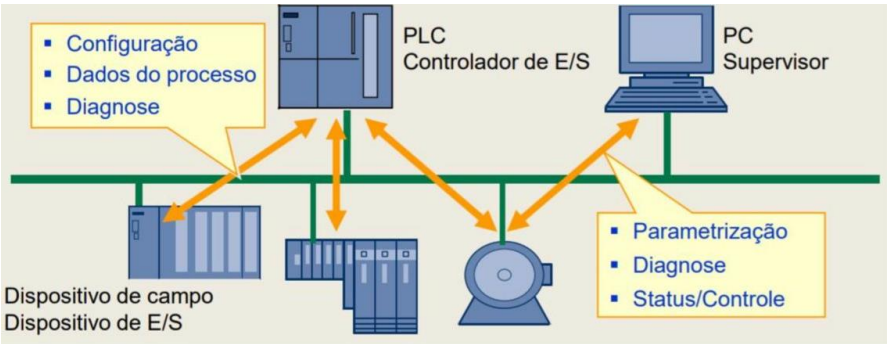
Conforme foi apresentado na **Figura 1**, evidencia-se a importância dos protocolos de rede para atender às comunicações. Turcato (2020) apontou que o protocolo *Profibus* e *Profinet*, garantem que a comunicação seja confiável e rápida em transmitir dados entre os dispositivos. Além deles, foram encontrados na literatura os protocolos Ethernet/IP, *Modbus TCP, EtherCAT, DeviceNet* e *MQTT* (Amo; Roepke, 2024; Demétrio; Junior; Psiciotta, 2024).

Entre eles, destaca-se o Profinet, uma extensão do *Profibus* desenvolvida pela *Profibus International* (PI) em 2002. Baseado na Ethernet industrial (IEEE 802.3) e nos padrões IEC 61158 e IEC 61784, o protocolo garante alta performance, confiabilidade e precisão em ambientes automatizados (Turcato, 2020). De acordo com Demétrio, Junior e Psicotta (2024) há funcionalidades avançadas no diagnóstico

remoto, rápida identificação de falhas, programação e atualização em tempo real sem interromper os processos produtivos.

Os Controladores E/S (também conhecidos como Controladores Lógicos Programáveis – PLC) e os Dispositivo E/S (sensores, atuadores e módulos de entrada e saída) são os principais componentes do protocolo Profinet. Também se destaca o Supervisor E/S (PC Supervisor), um software que gerencia e realiza o diagnóstico de parâmetros das unidades, conforme apresentado na Figura 2:

Figura 2 - Principais tipos de componentes do protocolo PROFINET.



Fonte: Ilário (2020).

Nesse sentido, o seu funcionamento opera com agilidade junto com outras redes industriais, com alta velocidade de transmissão e com flexibilidade (Demétrio; Junior; Psciotta, 2024). Há diversos tipos, tais como anel, estrela, árvore e linha. A sua comunicação ocorre de forma simultânea por um único cabo em serviços de TI e em tempo real por meio de gateways, convertendo dados de um protocolo para outro, sendo ajustado conforme a necessidade da aplicação. Suas tarefas exigem alto desempenho, com comunicação TCP/IP, RT (*Real Time*) e IRT (*Isochronous Real Time*), oferecendo velocidades de até 100 Mbps em um único barramento (Ilário, 2020).

O protocolo adota o modelo de comunicação *Provider/Consumer*, onde o *Provider* transmite os dados de um ou mais dispositivos, enquanto o *Consumer* recebe essas informações e controla as entradas e saídas. (Turcato, 2020). Os

autores Amo e Roepke (2024) afirmaram que esse tipo de comunicação é fundamental em ambientes industriais, pois exige alta precisão, controle de movimento, sistemas robóticos e soluções de segurança.

2.2 Cibersegurança

De acordo com Pontes (2024), a segurança é um dos maiores desafios aos desenvolvedores, diante do crescente número de ataques em ambientes industriais. O Brasil apresenta deficiência em cibersegurança, ocupando a 73ª posição no *National Cyber Security Index* (NCSI) de 2021, refletindo vulnerabilidades na prevenção e resposta a incidentes cibernéticos (Mota *et al.,* 2021).

Sendo assim, as medidas de segurança são estratégicas do modelo de negócio, que protegem os dados corporativos e dispositivos industriais e garantem a continuidade dos processos produtivos (Mota *et al.,* 2022). São necessárias atualizações constantes nas abordagens de segurança, especialmente com a introdução de tecnologias como 5G e a computação em borda. Garantir a proteção nesses novos contextos exige soluções inovadoras e adaptadas às especificidades dos sistemas industriais (Pontes, 2024).

Nesse contexto, entende-se a necessidade de garantir o desempenho da rede, considerando fatores como interferência eletromagnética, aterramento, distâncias máximas entre dispositivos e volume de dados. Porém, esse crescimento da adoção de redes industriais baseadas em Ethernet deixa vulneráveis a riscos cibernéticos com a necessidade de medidas de segurança contra acessos não autorizados, malwares e outras ameaças (Turcato, 2020). Isso ocorre por conta da ausência de camadas de proteção adequadas, como antivírus, firewalls e a dependência de sistemas legados.

Turcato (2020) destaca que uma anomalia em redes de comunicação é qualquer desvio do comportamento normal, muitas vezes utilizado como meio para comprometer o funcionamento do sistema. Tais anomalias podem indicar tentativas de acesso não autorizado ou a presença de malware.

Desse modo, há diversos tipos de ataques, tais como: *Denial of Service (DoS)* e sua versão distribuída (DDoS), *Sniffing, Man-in-the-Middle (MITM)* e *ARP Spoofing* (Adari; Alla, 2024). O ataque ARP *Spoofing* permite ao invasor coletar dados sigilosos e enviar respostas falsas, fazendo com que seu endereço MAC seja associado ao IP legítimo. Assim, todo o tráfego destinado ao gateway ou a outro dispositivo passa a ser redirecionado ao invasor, que pode monitorar, alterar ou bloquear as informações transmitidas (Teixeira; Clarim, 2017).

Mota *et al.* (2021) destacam a possibilidade de minimizar os riscos com o uso de roteadores com *firewall* e VPN, os quais ajudam a impedir interceptações e acessos não autorizados. Os modelos baseados em detecção de anomalias apoiados por aprendizado de máquina e Redes Neurais Profundas são importantes para detectar os comportamentos inesperados (Adari; Alla, 2024).

Nesse contexto, Adari e Alla (2024) defendem que a detecção de anomalias baseada em aprendizado de máquina é uma abordagem eficiente para identificar padrões de comportamento suspeitos em ambientes industriais. Como as comunicações em redes como PROFINET tendem a ser regulares e previsíveis, desvios abruptos podem sinalizar tentativas de intrusão ou falhas. O uso de algoritmos de aprendizado profundo permite automatizar esse processo e aplicar respostas rápidas e eficazes, garantindo a integridade, disponibilidade e segurança dos processos industriais.

De acordo com Turcato (2020), os sistemas de detecção baseados em anomalias enfrentam desafios como a necessidade de grandes volumes de dados históricos, alto custo computacional e constante atualização dos modelos. Além disso, a natureza polimórfica de muitos ataques e a existência de vulnerabilidades ainda não mapeadas dificultam a eficácia dos sistemas tradicionais baseados em assinaturas.

2.3 Aplicação de *Machine Learning*

O uso de aprendizagem de máquina aprimora o IDS, contribuindo para identificar as ameaças e tomadas de decisões automatizadas. Assim, é preciso obter os dados rotulados das características (*features*) que serão utilizadas no treinamento.

Essa ação visa melhorar a performance e reduzir a complexidade computacional ao operarem em tempo real (Nicolaio, 2024).

Os métodos de aprendizado de máquina incluem classificação supervisionada, não supervisionada e semi-supervisionada. O foco neste estudo é o aprendizado supervisionado, o qual usa dados previamente rotulados para treinar um modelo e identificar padrões anômalos na rede (Nicolaio, 2024).

Além das abordagens tradicionais de aprendizado de máquina, as Redes Neurais Profundas (*Deep Learning*) são eficazes na detecção de anomalias, principalmente quando aplicadas a grandes volumes de dados com características temporais ou estruturais complexas (Chollet, 2021). Nesse sentido, eliminam a necessidade de extração manual de atributos, aprendendo representações diretamente dos dados brutos, o que resulta em uma elevada capacidade preditiva.

No entanto, seu desempenho está relacionado à qualidade do pré-processamento dos dados, sendo fundamentais etapas como normalização, balanceamento e limpeza para garantir a eficácia na detecção de padrões anômalos. Sua resposta é rápida aos incidentes, assegurando a continuidade dos processos operacionais (Chollet, 2021).

2.4 Classificadores

Segundo Chollet (2021), classificadores podem ser vistos como transformações geométricas progressivas de dados, construídas por camadas sucessivas de redes neurais. Assim, permite que modelos aprendam representações complexas e não-lineares, superando os limites de classificadores tradicionais, principalmente em contextos como a detecção de anomalias em redes industriais.

Dessa forma, vale ressaltar os classificadores utilizados na detecção de anomalias em redes industriais: *Perceptron*, Regressão Logística, Máquinas de Vetores de Suporte (SVM) e Árvores de Decisão. Porém, o Random Forest é adotado neste estudo como principal classificador diante de sua capacidade. Assim,

apresentam-se breves considerações sobre os outros classificadores (Raschka e Mirjalili, 2017).

O Random Forest se destaca na detecção de anomalias em redes complexas, pois, no contexto da segurança cibernética, é preciso ser capaz de modelar padrões triviais, considerando que diversos ataques se manifestam por meio de alterações sutis no comportamento de tráfego. Nesse sentido, Chollet (2021) destaca que o treinamento de redes profundas com dados capturados por sniffers possibilita identificar desvios em fluxos aparentemente legítimos.

O Random Forest é um método de aprendizado de máquina baseado em conjunto (ensemble) de árvores de decisão, onde cada árvore é treinada com subconjuntos aleatórios de dados e atributos (Raschka; Mirjalili, 2017). Destaca-se que a diversidade de árvores garante a validação *out-of-bag* e contribui para minimizar o *overfitting,* possibilitando maior estabilidade e acurácia. Além disso, possibilita também identificar a importância de cada atributo no processo de classificação. (Pontes, 2024).

A avaliação do desempenho de classificadores aplicados à detecção de intrusões em redes industriais valida sua eficácia e confiabilidade. Nesse contexto, é utilizada a Matriz de Confusão, a qual tabula os resultados de classificação em quatro categorias: Verdadeiros Positivos (VP), Falsos Positivos (FP), Falsos Negativos (FN) e Verdadeiros Negativos (VN) (Nicolaio; Muranetto; Fonseca, 2023).

2.5 Sniiffer de rede

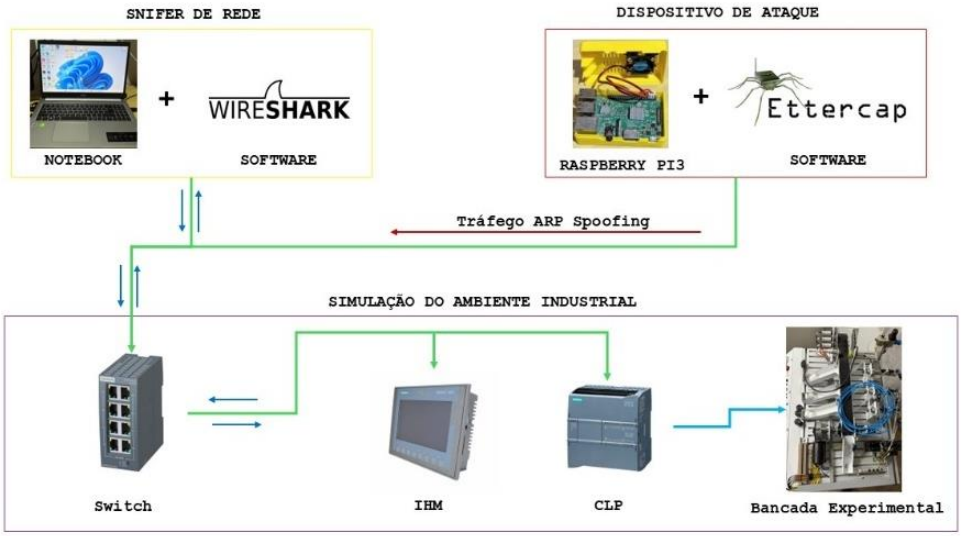
Os *sniffers de rede* são ferramentas indispensáveis para o monitoramento, diagnóstico e instrumentos de ataques. É um *software* ou *hardware* utilizado para capturar e registrar pacotes de dados trafegando em uma rede (Raschka; Mirjalili, 2017). Eles operam em modo promíscuo, permitindo que a interface de rede monitore todos os pacotes transmitidos no segmento, independentemente do destino. Assim, é explorado pelos profissionais para monitorar a integridade das comunicações e pelos atacantes para capturar informações (Adari;Alla, 2024).

Nesse contexto, o *Wireshark* permite uma captura em tempo real, análise minuciosa dos pacotes coletados, utilizando filtros por protocolos, portas ou endereços IP. Identificam-se os padrões anômalos, sendo precisa e previsível. Os dados são capturados para construção de datasets em pesquisas acadêmicas e aplicações práticas de aprendizado de máquina. Esses *datasets,* como o KDD Cup 1999 ou o HAI-22.04, são essenciais para treinar classificadores como o *Random Forest*, permitindo a identificação automatizada de padrões maliciosos com base em históricos de tráfego (Adari; Alla, 2024).

3 METODOLOGIA

O estudo experimental é de abordagem quantitativa e aplicada com a finalidade de simular um ambiente industrial com o protocolo de rede Profinet. Assim, foram analisados o tráfego da rede, gerados cenários de ataque e avaliada a eficácia de um modelo de detecção de intrusões e anomalias.

Figura 3 – Topologia Elaborada



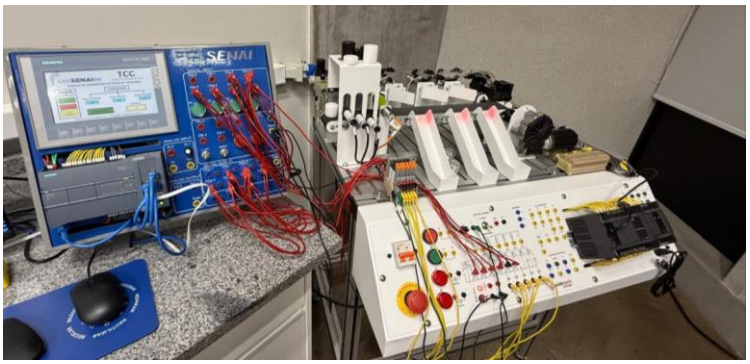
Fonte: Do Autor (2025)

A coleta de dados foi realizada por meio de registros numéricos, possibilitando uma análise estatística e a aplicação de algoritmos de aprendizado de máquina.

3.1 Simulação do ambiente industrial

A simulação foi construída com uma bancada de testes representando uma linha de produção industrial, composta por uma esteira separadora de peças, programada para contar e classificar objetos de tamanhos distintos (pequeno, médio e grande).

Figura 4 – Simulação Industrial

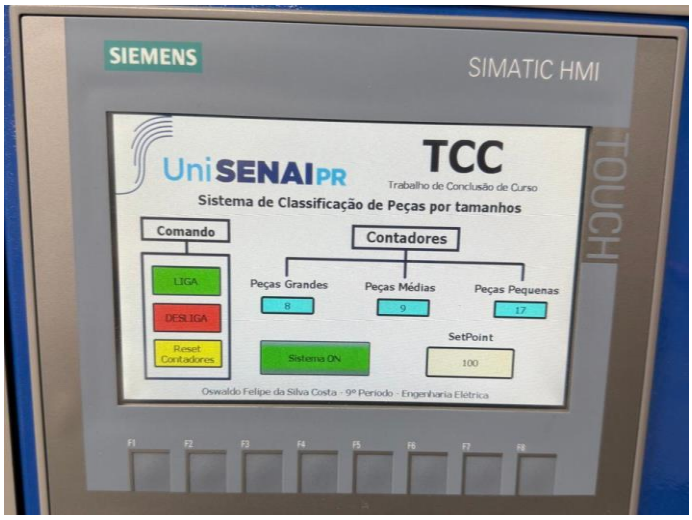


Fonte: Arquivo pessoal (2025)

O sistema de automação foi implementado utilizando um Controlador Lógico Programável (CLP) Siemens modelo *S7-1200* e uma Interface Homem-Máquina (IHM) Siemens modelo *KTP700 Basic*. A **Figura 5** demonstra a interface desenvolvida, na qual é possível visualizar os comandos do sistema, status de operação, contadores de peças classificados por tamanhos e valor de referência.

Para o desenvolvimento das lógicas de controle e criação da tela de monitoramento e ajustes da bancada experimental, utilizou-se o software TIA Portal com diagrama na linguagem *Ladder,* que controla sensores e atuadores, possibilitando ao operador monitorar, comandar o sistema e as variáveis em tempo real.

Figura 5 – IHM em pleno funcionamento

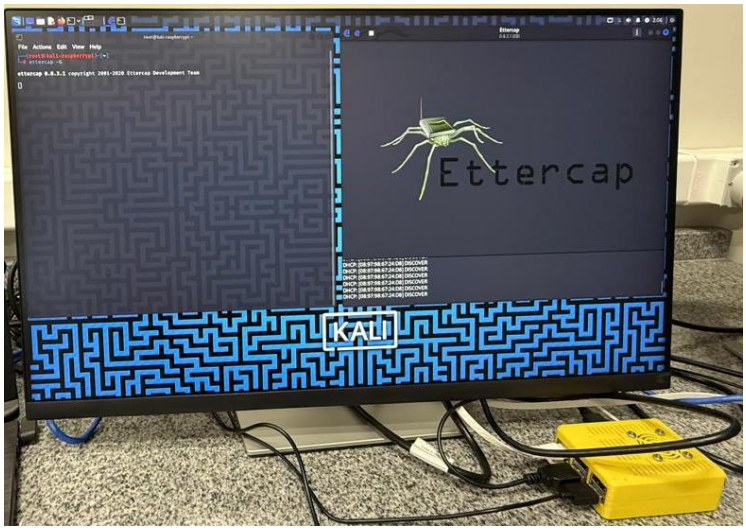


Fonte: Arquivo pessoal (2025)

3.2 Disparador de ataques

Na elaboração do dispositivo, foi utilizada uma *Raspberry Pi 3,* operando com sistema operacional *Kali Linux,* que fornece uma arquitetura simples e especializada em testes de intrusão e análise de segurança, pois possui nativamente ferramentas de ataque e detecção, conforme ilustra a Figura 6.

Figura 6 – Execução de ferramenta *Ettercap*



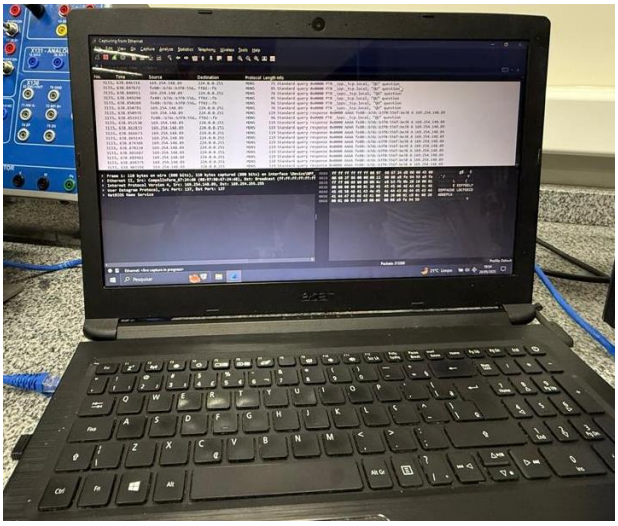
Fonte: Arquivo pessoal (2025)

Entre essas possibilidades, foi selecionado o *Ettercap para* geração dos fluxos maliciosos do tipo *ARP Spoofing*, capaz de interceptar e manipular os pacotes de dados trafegados na rede entre os componentes interligados, gerando diversos tipos de falhas, como congelamentos de telas, erros de leituras, lentidão e até travamento dos comandos.

3.3 Aquisição de dados

A coleta de dados foi realizada utilizando um computador com sistema operacional Windows, com o software *Wireshark*, que atuou como *sniffer* de rede.

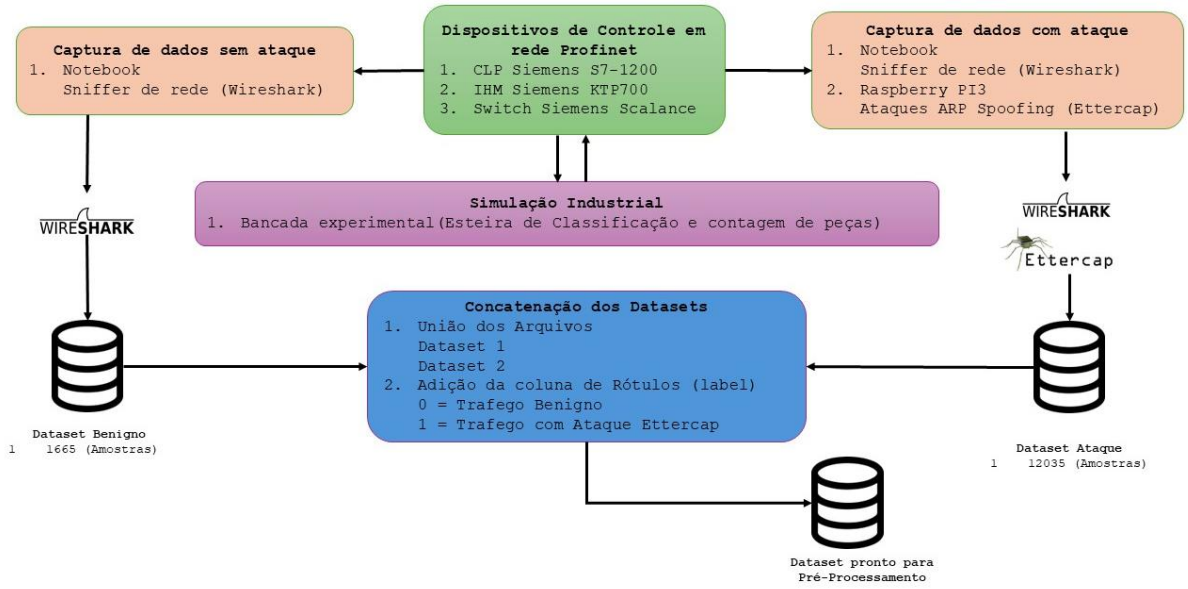
Figura 7 – Execução da ferramenta *Wireshark*



Fonte: Arquivo pessoal (2025)

O *Sniffer* de rede foi configurado para capturar o tráfego da rede em dois cenários, onde no primeiro as operações de rede ocorrem sem interferências externas, tendo apenas o fluxo de dados entre CLP e a IHM, devido à simulação normal da bancada experimental de seleção e contagem de peças. Já no segundo cenário, a captura foi realizada com a aplicação dos fluxos maliciosos. Ambos os dados foram inicialmente armazenados separados em arquivos do tipo “.CSV”.

Figura 8 – Fluxograma da estrutura para captura do tráfego de dados



Fonte: Do Autor (2025)

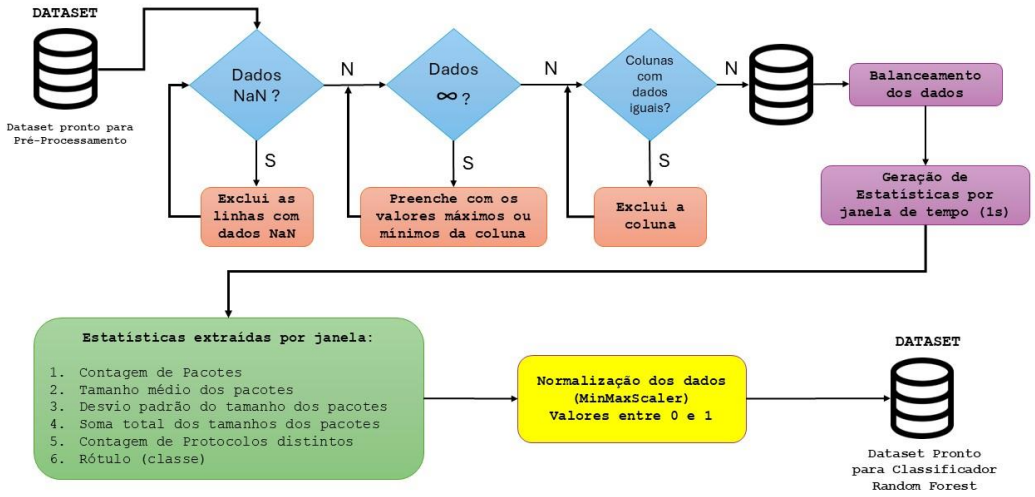
Como demonstra o fluxograma (**Figura 8**), no próximo passo foi realizada a concatenação entre os dois arquivos gerados, ou seja, a junção dos dois arquivos em um só. Além disso, foi criada uma coluna para sinalizar a classe da amostra (“*label*”), onde foi rotulado o valor ‘0’ para os tráfegos benignos, e ‘1’ para os tráfegos com ataque, tornando o *Dataset* apto para o pré-processamento.

3.4 Pré-processamento e *DataSet*

O pré-processamento dos dados se inicia a partir do *Dataset* “bruto”, que consiste em um processo de limpeza e verificação da integridade das amostras, desenvolvido em três etapas antes de serem aplicadas ao classificador Random Forest. Na primeira etapa, ocorre a remoção de amostras que contenham valores ausentes (*NaN*), pois esses dados podem comprometer o desempenho do modelo. Na segunda etapa, são identificados valores infinitos (∞), que não são excluídos, mas sim substituídos pelo valor máximo ou mínimo da respectiva coluna, sem comprometer a estrutura da base. Na terceira etapa, são eliminadas colunas com

valores idênticos em todas as linhas, pois a ausência de variação não contribui para o aprendizado do modelo e apenas aumenta o volume de dados sem relevância. Nessa ação, possibilita-se que os dados estejam limpos e prontos para o treinamento e validação do classificador.

Figura 9 – Fluxograma da estrutura para pré-processamento de dados



Fonte: Do Autor (2025)

Com o processo de limpeza finalizado, é realizado o balanceamento dos dados, processo que ajuda a garantir que o classificador aprenderá a distinguir as situações anômalas e benignas, pois este passo equilibra a quantidade de amostra entre as duas classes. Seguindo o fluxograma (**Figura 9**), a sequência se trata da geração de estatísticas por janela de tempo, neste caso, foram seguidos intervalos fixos de 1 segundo. Desse modo, no **Quadro 1**, é listado e representado cada dado extraído:

Quadro 1 - Dados extraídos

Estatísticas gerada pela janela de tempo		
Nome da Coluna	**Significado**	**Descrição**
packet_count	Contagem de pacotes	Número de pacotes capturados durante a janela de tempo.

mean_packet_size	Tamanho médio dos pacotes	Média dos tamanhos dos pacotes trocados na janela.
std_packet_size	Desvio padrão do tamanho dos pacotes	Variação dos tamanhos dos pacotes. Um desvio elevado pode indicar comportamento irregular ou ataque.
total_packet_size	Soma total dos tamanhos dos pacotes	Quantidade total de bytes transmitidos na janela.
protocol_count	Contagem de protocolos distintos	Número de diferentes protocolos (como TCP, UDP, ARP) presentes no tráfego da janela.
label	Rótulo (classe)	Identifica se o tráfego é benigno = 0 ou ataque = 1.

Fonte: Do Autor (2025)

Sendo assim, todas as colunas são submetidas ao processo de normalização, onde é utilizada a técnica *MinMaxScaler,* que ajusta todos os dados em valores entre 0 e 1, fazendo com que todas as amostras tenham o mesmo peso, contribuindo positivamente para o treinamento do classificador.

Fórmula *MinMaxScaler:*

$$Xnormalizado = \frac{X - Xmin}{Xmax - Xmin}$$

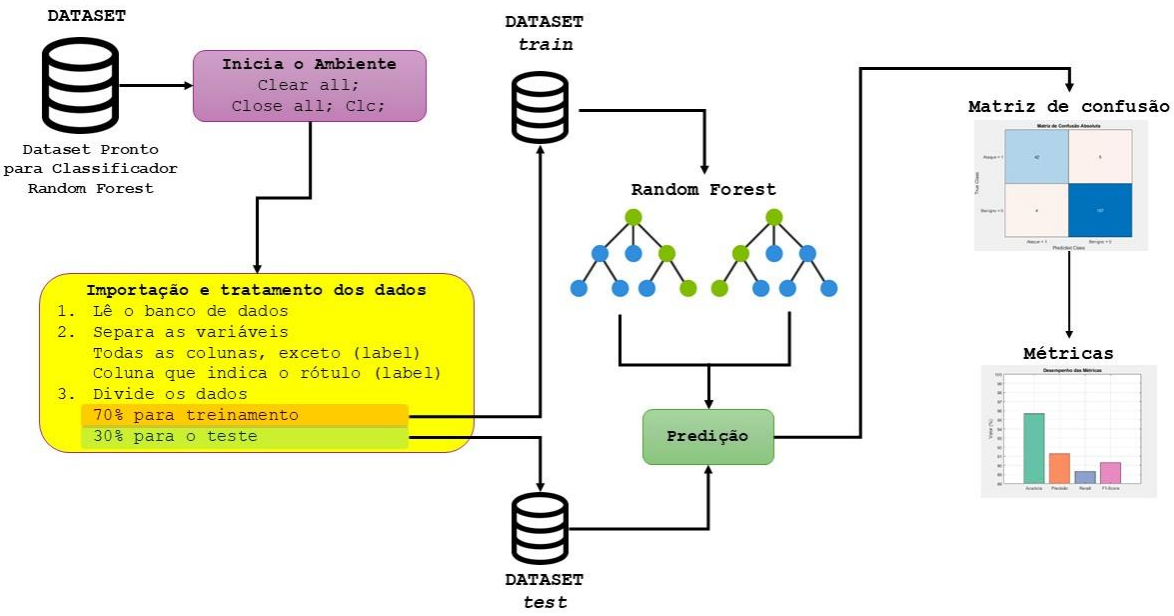
Após todos esses procedimentos de limpeza, balanceamento, extração de dados e normalização, obteve-se um *Dataset pré-processado*, ou seja, o banco de dados está apto para ser utilizado pelo classificador *Random Forest.*

3.5 Aplicação do Algoritmo de *Machine Learning utilizando MatLab*

Com o modelo construído, o classificador *Random Forest* foi praticado na plataforma *MatLab*, devido à sua capacidade de lidar com dados variados e resistência a sobreajuste *(overfitting).* Sendo assim, no fluxograma **(Figura 10),** o script inicializa garantindo o fechamento de guias anteriores e exclusão de dados,

garantindo nenhuma interferência com ambiente limpo e pronto para execução, para realizar a importação do *Dataset* já pré-processado.

Figura 10 – Fluxograma de execução do classificador *Random Forest* no MatLab



Fonte: Do Autor (2025)

Após a leitura do banco de dados, é realizada a separação das variáveis, dividindo a coluna que rotula as classes (*label*) em uma tabela Y, enquanto as demais colunas ficam armazenadas em X. Em seguida, são separados os dados de treinamento e teste. No caso deste modelo, as amostras foram divididas da seguinte forma:

- 70% para treinamento.
- 30% para teste.

Os dados separados para treinamento são alimentados pelo classificador, que executa a função de *Machine Learning.* Nesta ação, são estudados os padrões do tráfego de rede. Esse modelo de aprendizado é realizado por um conjunto de 100 árvores de decisão, construídas a partir de amostras diferentes. As árvores

determinam de forma independente e a classificação final é definida pela maioria das árvores.

Finalizando o treinamento, são utilizados os dados destinados aos testes, que são as amostras que não foram vistas. Nessa ação, é realizada a predição na finalidade de comparar os rótulos reais, separados no início do modelo. Então, esse comparativo evidencia de forma realista e eficaz a validação do desempenho do classificador. A partir dessa comparação, é gerada a matriz de confusão, da qual são extraídas as principais métricas de avaliação.

3.6 Avaliação do Modelo

A avaliação do desempenho de classificadores aplicados à detecção de intrusões em redes industriais é essencial para validar sua eficácia e confiabilidade. Entre os principais instrumentos utilizados nessa avaliação está a matriz de confusão, que fornece uma representação tabular dos resultados de classificação em quatro categorias: Verdadeiros Positivos (VP), Falsos Positivos (FP), Falsos Negativos (FN) e Verdadeiros Negativos (VN).

Quadro 2 - Matriz de Confusão

Matriz de Confusão	
Verdadeiros Positivos (VP)	Casos corretamente classificados como anomalias
Falsos Positivos (FP)	Casos erroneamente identificados como anomalias, quando na verdade são normais
Falsos Negativos (FN)	Casos incorretamente classificados como normais, quando na realidade são anomalias
Verdadeiros Negativos (VN)	Casos corretamente identificados como normais

Fonte: Do autor (2025)

Para avaliar o modelo, seguiram-se as métricas: Acurácia, Recall, *Precision* e *F1-Score*, onde ambos são calculados a partir da matriz de confusão, conforme fórmulas abaixo:

$$Acurácia = (VP + VN) / (VP + VN + FP + FN)$$

A acurácia é uma métrica simples que representa a proporção de previsões corretas em relação ao total de casos avaliados, sendo útil especialmente em conjuntos de dados balanceados. No entanto, em contextos com desequilíbrio entre classes, pode gerar interpretações equivocadas.

$$Precisão = VP / (VP + FP)$$

A precisão, por sua vez, mede a proporção de acertos entre os casos classificados como positivos, é importante para avaliar o quanto o modelo consegue identificar corretamente as anomalias sem gerar muitos falsos positivos. Um modelo com alta precisão tende a ser mais confiável para acionar alertas.

$$Recall = VP — (VP + FN)$$

O *Recall*, também conhecido como sensibilidade, mede a capacidade do classificador em identificar corretamente todos os casos positivos reais. Um alto valor de recall indica que o modelo está sendo eficaz em detectar a maioria das ameaças. No entanto, se o recall for muito baixo, o sistema pode falhar ao deixar de detectar ataques reais, comprometendo a segurança da rede.

$$F1 - Score = 2VP / (2VP + FP + FN).$$

O *F1-Score* é a média harmônica entre precisão e recall, sendo uma métrica de equilíbrio útil quando há necessidade de ponderar os dois aspectos. Essa métrica é particularmente recomendada em situações de desequilíbrio entre classes, pois evita que o desempenho do modelo seja superestimado com base em apenas uma das medidas.

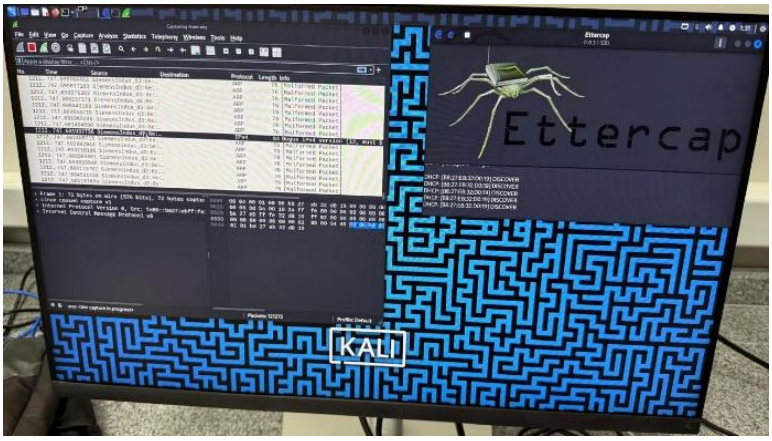
Nesse sentido, o uso das métricas possibilita entender sobre a capacidade do modelo de classificar corretamente os dados de rede, sendo imprescindível para o desenvolvimento de soluções robustas de detecção de intrusões em ambientes industriais complexos. Apresentando de forma mais clara, as principais métricas são importantes na avaliação do modelo no contexto de segurança cibernética.

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1 Dificuldades encontradas

Em busca de atingir os objetivos mencionados, foi construída a simulação do ambiente industrial em bancada acadêmica, utilizando os itens descritos no tópico anterior. A princípio, o simulador de ataques (*Ettercap*) e o *Sniffer* de rede (*Wireshark*), seriam executados simultaneamente pelo hardware *Raspbarry PI3.*

Figura 11 – *Ettercap* e *Wireshark* em execução



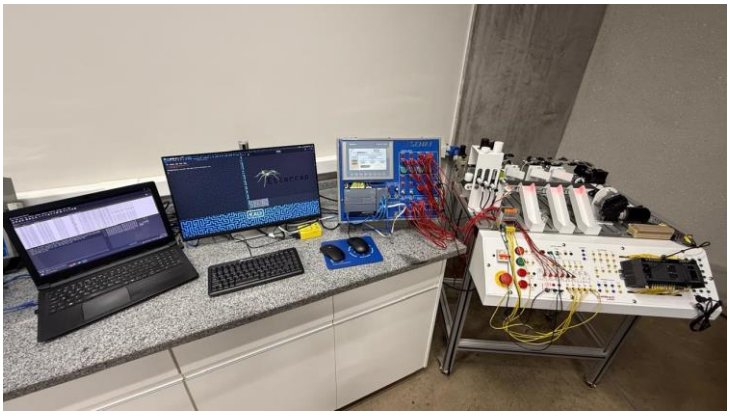
Fonte: Arquivo pessoal (2025)

Durante os testes no ambiente experimental, devido a limitações de desempenho do hardware, ocorreram diversas dificuldades, como travamentos e reinicializações. O caso foi solucionado com a adição de um notebook, para execução do *software Wireshark*, mantendo apenas o *sniffer* de rede na *Raspberry.*

4.2 Simulação e coleta de dados

Após os ajustes, todos os ambientes foram testados, ficando aptos para as simulações e coletas.

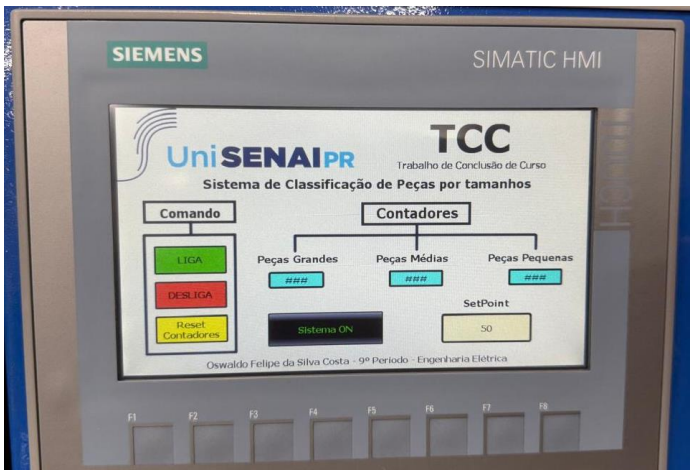
Figura 12 – Ambiente controlado para teste



Fonte: Arquivo pessoal (2025)

Foi realizada aproximadamente 30 minutos de simulação do processo de seleção das peças, em modo normal, onde foram coletados tráfegos cíclicos e acíclicos da rede entre os dispositivos industriais. Em seguida, foi inicializado o segundo teste com duração similar à anterior, porém, neste caso, executando o *Ettercap*, atacando a rede, onde nos primeiros instantes notou-se lentidão na execução das tarefas e falhas na apresentação das contagens na IHM.

Figura 13 – IHM com erros na indicação dos contadores



Fonte: Arquivo pessoal (2025)

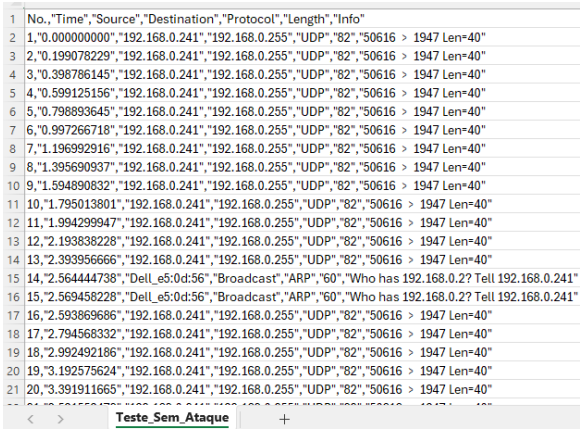
Ao decorrer da simulação, outros comportamentos anômalos foram identificados, como a paralisação dos botões de comandos (Liga, Desliga e Reset), além da impossibilidade funcional do campo de "*setpoint*", implementado para o operador definir o valor de contagem das peças.

4.3 Pré-processamento do Dataset

Coletados os dados, eles passaram pela concatenação, onde foram agrupados os arquivos e criados rótulos com a intenção de distinguir as classes das amostras.

Figura 14 – Amostras coletadas (a) sem ataque (b) com ataque

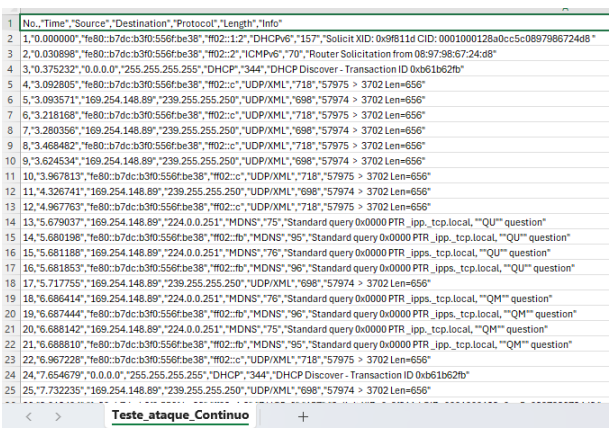
(a)



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
2	0.199078229	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
3	0.398786145	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
4	0.599125156	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
5	0.798893645	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
6	0.997266718	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
7	1.196992916	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
8	1.395690937	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
9	1.594890832	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
10	1.795013801	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
11	1.994299947	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
12	2.193838228	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
13	2.393956666	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
14	2.564444738	Dell_e5:0d:56	Broadcast	ARP	60	Who has 192.168.0.2? Tell 192.168.0.241
15	2.569458228	Dell_e5:0d:56	Broadcast	ARP	60	Who has 192.168.0.2? Tell 192.168.0.241
16	2.593869686	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
17	2.794568332	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
18	2.992492186	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
19	3.192575624	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40
20	3.391911665	192.168.0.241	192.168.0.255	UDP	82	50616 > 1947 Len=40

Teste_Sem_Ataque

(b)



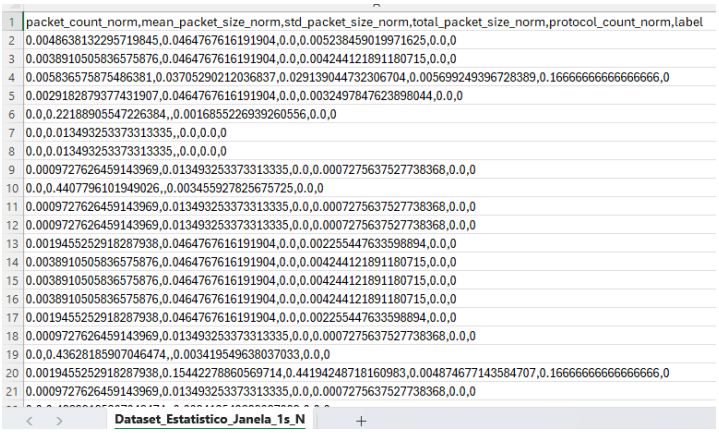
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::b7dc:b3f0:556f:be38	ff02::1:2	DHCPv6	157	Solicit XID: 0x9f811d CID: 0001000128a0cc5c0897986724d8
2	0.030898	fe80::b7dc:b3f0:556f:be38	ff02::2	ICMPv6	70	Router Solicitation from 08:97:98:67:24:d8
3	0.375232	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xb61b62fb
4	3.092805	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
5	3.093571	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656
6	3.218168	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
7	3.280356	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656
8	3.468482	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
9	3.624534	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656
10	3.967813	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
11	4.326741	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656
12	4.967763	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
13	5.679037	169.254.148.89	224.0.0.251	MDNS	75	Standard query 0x0000 PTR _ipp._tcp.local, "QU" question
14	5.680198	fe80::b7dc:b3f0:556f:be38	ff02::fb	MDNS	95	Standard query 0x0000 PTR _ipp._tcp.local, "QU" question
15	5.681188	169.254.148.89	224.0.0.251	MDNS	76	Standard query 0x0000 PTR _ipps._tcp.local, "QU" question
16	5.681853	fe80::b7dc:b3f0:556f:be38	ff02::fb	MDNS	96	Standard query 0x0000 PTR _ipps._tcp.local, "QU" question
17	5.717755	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656
18	6.686414	169.254.148.89	224.0.0.251	MDNS	76	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question
19	6.687444	fe80::b7dc:b3f0:556f:be38	ff02::fb	MDNS	96	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question
20	6.688142	169.254.148.89	224.0.0.251	MDNS	75	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question
21	6.688810	fe80::b7dc:b3f0:556f:be38	ff02::fb	MDNS	95	Standard query 0x0000 PTR _ipp._tcp.local, "QM" question
22	6.967228	fe80::b7dc:b3f0:556f:be38	ff02::c	UDP/XML	718	57975 > 3702 Len=656
23	7.654679	0.0.0.0	255.255.255.255	DHCP	344	DHCP Discover - Transaction ID 0xb61b62fb
24	7.732235	169.254.148.89	239.255.255.250	UDP/XML	698	57974 > 3702 Len=656

Teste_ataque_Continuo

Fonte: Do autor (2025)

Feito o tratamento inicial, realizou-se o pré-processamento dos dados que resultou em um *Dataset* organizado, limpo e normalizado, contendo apenas atributos relevantes das amostras, como demonstra a Figura 15.

Figura 15 – Dataset pronto



	packet_count_norm,mean_packet_size_norm,std_packet_size_norm,total_packet_size_norm,protocol_count_norm,label
2	0.004863813229571984 5,0.0464767616191904,0.0,0.005238459019971625,0.0,0
3	0.003891050583657587 6,0.0464767616191904,0.0,0.004244121891180715,0.0,0
4	0.005836575875486381,0.03705290212036837,0.029139044732306704,0.005699249396728389,0.16666666666666666,0
5	0.002918287937743190 7,0.0464767616191904,0.0,0.003249784762389804 4,0.0,0
6	0.0,0.22188905547226384,,0.0016855226939260556,0.0,0
7	0.0,0.013493253373313335,,0.0,0.0,0
8	0.0,0.013493253373313335,,0.0,0.0,0
9	0.000972762645914396 9,0.013493253373313335,0.0,0.000727563752773836 8,0.0,0
10	0.0,0.4407796101949026,,0.003455927825675725,0.0,0
11	0.000972762645914396 9,0.013493253373313335,0.0,0.000727563752773836 8,0.0,0
12	0.000972762645914396 9,0.013493253373313335,0.0,0.000727563752773836 8,0.0,0
13	0.001945552918287938,0.0464767616191904,0.0,0.002255447633598894,0.0,0
14	0.003891050583657587 6,0.0464767616191904,0.0,0.004244121891180715,0.0,0
15	0.003891050583657587 6,0.0464767616191904,0.0,0.004244121891180715,0.0,0
16	0.003891050583657587 6,0.0464767616191904,0.0,0.004244121891180715,0.0,0
17	0.001945552918287938,0.0464767616191904,0.0,0.002255447633598894,0.0,0
18	0.000972762645914396 9,0.013493253373313335,0.0,0.000727563752773836 8,0.0,0
19	0.0,0.43628185907046474,,0.003419549638037033,0.0,0
20	0.001945552918287938,0.15442278860569714,0.44194248718160983,0.004874677143584707,0.16666666666666666,0
21	0.000972762645914396 9,0.013493253373313335,0.0,0.000727563752773836 8,0.0,0

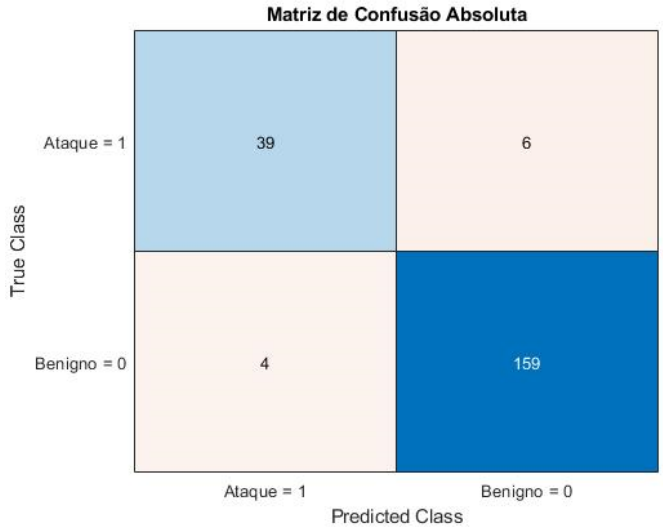
Dataset_Estatistico_Janela_1s_N

Fonte: Do Autor (2025)

4.4 Aplicação do Classificador

Com *Dataset* concluído e apto para ser aplicado ao modelo de classificador escolhido, foi inicializado, conforme *script* criado pelo *software Matlab*, o processo de *Machine learning* (treinamento). Como previsto, foi utilizado 70% do total de amostra do banco, gerando as predições, que posteriormente foram comparadas com os dados separados para testes e validações (30% das amostras do banco), finalizando o processo de classificação e obtendo o resultado apresentado pela figura abaixo:

Figura 16 – Matriz de confusão

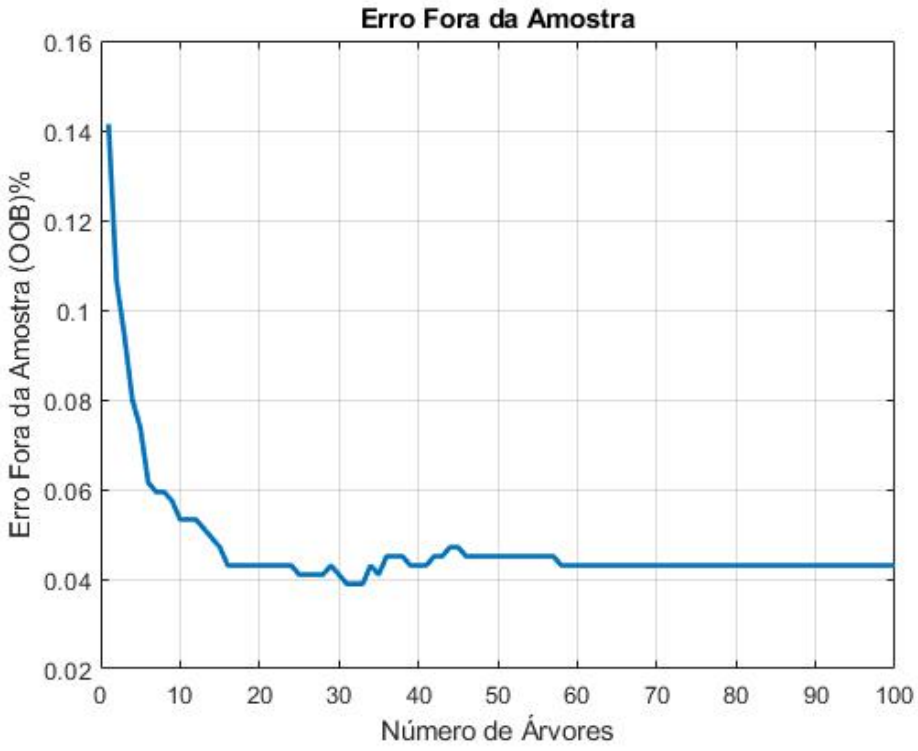


Fonte: Do Autor (2025)

Conforme resultado da matriz de confusão, observa-se que, considerando as amostras que realmente eram ataques, o modelo acertou em classificar 39 dessas amostras como anômalas (Verdadeiros Positivos) e errou classificando 6 delas como benignas (Falsos Negativos). É observado também que, considerando amostras que realmente eram benignas, as predições acertaram em 159 (cento e cinquenta e nove) casos (Verdadeiros Negativos), e errou em classificar 4 (quatro) amostras como ataques (Falsos Positivos). Contendo esses resultados, é possível gerar as métricas para melhor análise.

Para complementar a análise dos resultados do modelo de classificação, é necessário avaliar também a taxa de erro calculada sobre os dados não utilizados no treinamento de cada árvore (OOB).

Figura 17 – Resultado do OOB em porcentagem (%)

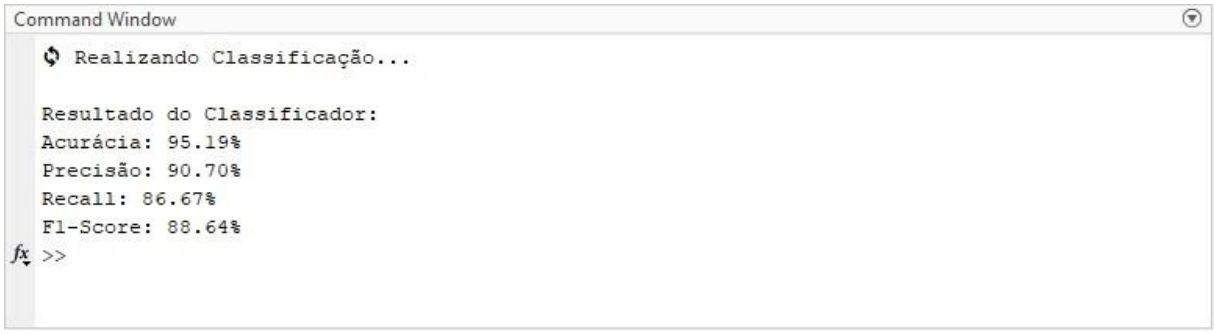


Fonte: Do Autor (2025)

Os dados apresentados evidenciam uma queda na taxa de erro no modelo aplicado, conforme o aumento do número de árvores, onde em cerca de 60 (sessenta) árvores o OOB se estabiliza na casa de 4%, o que comprova uma boa capacidade de predição, com baixo risco de *overfitting,* que seria um treinamento com foco excessivo em padrões irrelevantes e ruídos, gerando baixo desempenho.

Visto que os resultados até então foram satisfatórios, consideraram-se os dados gerados pela matriz de confusão, para elaboração das principais métricas conforme métodos já apresentados.

Figura 18 – Resultados das métricas (MatLab)



Fonte: Do Autor (2025)

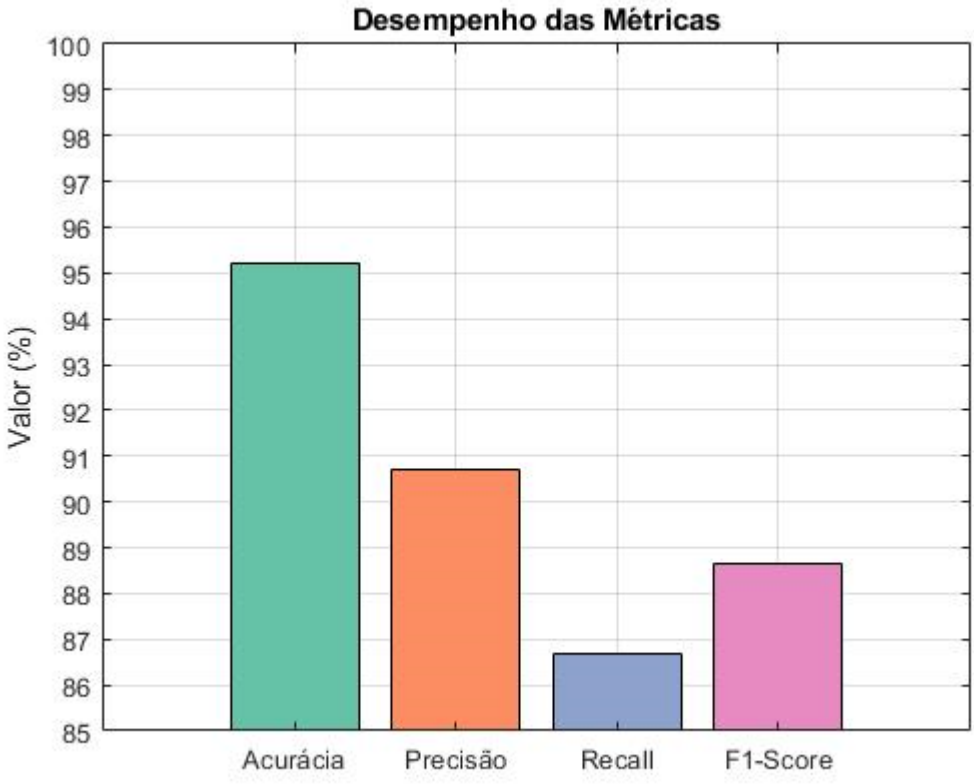
Analisando os resultados, identifica-se que a acurácia, que representa a proporção de classificações corretas, teve um alto desempenho geral, ultrapassando o índice de 95% de acerto nas predições.

No *Recall,* o resultado também foi positivo, chegando próximo de 87%, tendo uma boa detecção de anomalias. Visto que a métrica aponta a capacidade do modelo de identificar corretamente os exemplos positivos, onde no contexto atual de detecção de ataques cibernéticos é essencial a alta eficácia, pois deixar de identificar um ataque é mais perigoso do que um falso alarme.

Ao analisar a precisão, o resultado de 90,7% foi eficaz, pois o valor aponta que a maioria dos alertas dos modelos estava correta, ou seja, poucos falsos positivos. E, por fim, o resultado do *F1-Score*, que avalia a média harmônica entre precisão e

Recall, foi de 88,6%, que se comportou bem balanceado, obtendo resultado positivo, assim como as demais métricas.

Figura 19 – Comparativo das métricas



Fonte: Do Autor (2025)

Perante a análise geral das métricas, fica evidenciado que o modelo Random Forest apresentou desempenho robusto, equilibrado e com baixa tendência a *overfitting*, resultado esperado em contexto como o do sistema desenvolvido, que exige tanto confiabilidade quanto precisão nas predições, pois os resultados demonstraram a eficácia em detectar corretamente eventos relevantes e em evitar erros de classificação.

5 CONSIDERAÇÕES FINAIS

Conclui-se que, mesmo com as dificuldades encontradas, a aplicação do algoritmo de aprendizado de máquina nos dados coletados, alinhada a um bom pré-processamento dos dados e a um ambiente experimental bem estruturado, utilizando ferramentas seguras e eficazes, representa uma solução viável, eficiente e altamente precisa para detecção de anomalias e ataques em redes industriais baseadas em protocolo *Profinet.* O trabalho desenvolvido contribui visivelmente para o fortalecimento da segurança cibernética no contexto da Indústria 4.0, trazendo uma metodologia econômica e aplicável, não só em ambientes acadêmicos, mas também em ambientes industriais.

Como proposta para trabalhos futuros, recomenda-se a ampliação do estudo para outros tipos de ataques, como, por exemplo: *Replay Attack*, *DoS, MAC*, que também são tipos de ataques em redes *Profinet,* ou outros protocolos industriais, como *Modbus, Profbus,* entre outros. Além da possibilidade de se aprofundar mais em técnicas ainda mais avançadas de aprendizado, como *Deep Learning.*

REFERÊNCIAS

ADARI, Suman Kalyan; ALLA, Sridhar. **Beginning anomaly detection using Python-based deep learning: implement anomaly detection applications with Keras and PyTorch**. 2. ed. [S.l.]: Apress, 2024.

AMO, Gustavo Garcia de; ROEPKE, Hermano. As IOT NETWORKS AND TECHNOLOGICAL INNOVATION IN INDUSTRY: Industrials network. **Revista e-TECH: Tecnologias para Competitividade Industrial-ISSN-1983-1838**, v. 17, n. 1, 2024.

CHOLLET, François. **Deep learning with Python.** 2. ed. Shelter Island, NY: Manning Publications, 2021.

CERT.br. Centro de Estudos, **Resposta e Tratamento de Incidentes de Segurança no Brasil. Relatório de Atividades 2023**. São Paulo: NIC.br, 2023. Disponível em: https://www.cert.br. Acesso em: 25 mai. 2025.

DEMETRIO, Danilo; JUNIOR, Orlando Rosa; PISCIOTTA, Alex. COMPARATIVO DE REDES DE COMUNICAÇÃO DE ROBÔS INDUSTRIAIS: Resumo. **REVISTA DE ENGENHARIA E TECNOLOGIA**, v. 16, n. 1, 2024.

ILÁRIO, Lucas Rodrigues. **Contribuições para integração da indústria 4.0 nas redes elétricas inteligentes**. Dissertação (Mestrado). Engenharia Elétrica. Programa de Pós-Graduação em Engenharia Elétrica, Interunidades, entre o Instituto de Ciência e Tecnologia de Sorocaba e o Campus de São João da Boa Vista da Universidade Estadual Paulista “Júlio de Mesquita Filho”. Sorocaba. 2020.103p.

MÊLO, Thiago. ***FIESP diz que 30% das indústrias já foram alvo de ataques cibernéticos.*** Metrópoles, 17 fev. 2025. Disponível em: https://www.metropoles.com/negocios/industria/fiesp-diz-que-30-das-industrias-ja-foram-alvo-de-ataques-ciberneticos. Acesso em: 3 mar. 2025.

MOTA, Rubens Nascimento et al. Rastreabilidade no processo industrial baseado em conceitos da Indústria 4.0. **Revista Brasileira de Mecatrônica**, v. 5, n. 1, p. 39-62, 2022

NICOLAIO, Ivo GA; MUNARETTO, Anelise; FONSECA, Mauro. Uma proposta de detecção de ataques cibernéticos em sistemas de controle industrial (ics). In: **Workshop de Gerência e Operação de Redes e Serviços (WGRS)**. SBC, 2023. p. 153-166

NICOLAIO, Ivo Gabriel de Abreu et al. **Detecção de ataques em duas fases usando aprendizado de máquinas em sistemas de controle industrial de infraestruturas crítica**. 2024. Dissertação de Mestrado. Universidade Tecnológica Federal do Paraná.

PONTES, Francisco Jeferson da Silveira. Avaliação do desempenho de técnicas de aprendizado de máquina na detecção de malware em tráfego de Redes IoT. 2024. 59 f. TCC (Graduação em Engenharia de Computação) – Campus de Sobral, Universidade Federal do Ceará, Sobral, 2024.

RASCHKA, S.; MIRJALILI, V. **Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow**. 2. ed. Birmingham: Packt Publishing, 2017.

TEIXEIRA, Catharina Daher; CLARIM, Mariana de Lacerda. **Estudo das vulnerabilidades de tecnologias sem fio utilizadas em ambientes IoT.** Brasília,

2017. Trabalho de Graduação (Engenharia de Redes de Comunicação) – Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, 2017.

TURCATO, Afonso Celso. **Desenvolvimento de método para detecção de intrusão em redes PROFINET baseado em técnicas de Aprendizado de Máquina**. 2020. Tese de Doutorado. Universidade de São Paulo.

[Recebido/Received: Dezembro 18 2024; Aceito/Accepted: Janeiro 29, 2025]