Uma Proposta de Detecção de Ataques Cibernéticos em Sistemas de Controle Industrial (ICS)

Ivo G. A. Nicolaio[1], **Anelise Munaretto**[1], **Mauro Fonseca**[1]

[1]Programa de Pós Graduação em Engenharia Elétrica e Informática Industrial (CPGEI)
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brasil

ivogabriel@alunos.utfpr.edu.br, {anelise,maurofonseca}@utfpr.edu.br

Abstract. *Cyberattacks on industrial networks are not always limited to malware infection on personal computers victims of phishing tactics. Despite peculiarities of Industrial Control Systems (ICS), successful attacks can lead to devastating impacts whose trails can bypass conventional networks oriented security. Therefore, Intrusion Detection Systems have been employed to monitor ICS based network traffic in order to mitigate this risk, focusing on Machine Learning, Deep Learning or Graph related techniques. This work evaluated the performance gain in the joint use of supervised and unsupervised classifiers, based on the recall metric. We used the Hardware-in-the-loop Augmented Industrial Control System Dataset (HAI), which is focused on simulating a Cyber-Physical System of an industrial environment, specifically an electric power generation plant composed of steam-based thermoelectric and pumped-storage hydroelectric plant. The framework used is Scikit-Learn and it was possible to obtain results of 0.995 with the proposed parameters. The joint use of classifiers allowed an average gain of 24% in the recall in contrast to a single-step approach.*

Resumo. *Ataques cibernéticos em redes de ambiente industrial nem sempre limitam-se à instalação de malwares em máquinas de funcionários vítimas de phishing. Apesar das peculiaridades dos Sistemas de Controle Industrial (ICS), ataques bem sucedidos podem causar impactos devastadores que os mecanismos de segurança, orientados a uma rede convencional, podem não detectar. Portanto, o uso de Sistemas de Detecção de Intrusão monitorando os dados trafegados pelo ICS vêm sendo utilizados para contrapor esse risco, com foco no uso de técnicas baseadas em Machine Learning, Deep Learning ou Teoria dos Grafos. Este trabalho avaliou o ganho de desempenho no uso conjunto de classificadores supervisionados e não-supervisionados, com base na métrica recall. Utilizou o dataset Hardware-in-the-loop Augmented Industrial Control System Dataset (HAI), que é focado em simular um Sistema Ciber-Físico de ambiente industrial, em específico de uma usina de geração de energia elétrica composta de termoelétrica baseada em vapor e de hidroelétrica reversível. O framework utilizado é o Scikit-Learn e foi possível obter resultados de 0,995 com os parâmetros propostos. O uso conjunto de classificadores permitiu um ganho médio de 24% em recall comparados à classificação simples.*

1. Introdução

No Brasil, infraestruturas críticas são formadas por um complexo conjunto de instalações, bens, serviços e sistemas cuja interrupção ou destruição, total ou parcial, causa sérios impactos de ordem econômica, social, ambiental, ou à segurança do Estado e da sociedade, inclusive ferindo a imagem da Soberania Nacional no cenário internacional [Brasil 2018]. Dentre esses sistemas, as classes de água e energia são algumas cujos sistemas computacionais são alvos de ataques cibernéticos sofisticados, seja pelos danos que causam seja pelas características que esses ambientes incorporam.

Tais ambientes integram dispositivos diversos de uma rede convencional, como sensores e atuadores, controladores lógicos programáveis (PLC), sistemas supervisórios de controle e aquisição de dados (SCADA), interfaces homem-máquina (HMI), compondo o sistema de controle industrial (ICS) [Alanazi et al. 2023].

A adoção de complexos protocolos proprietários e de políticas orientadas à atuação presencial das equipes de produção e controle proporcionam melhor nível de segurança, ficando vulneráveis, via de regra, a sabotagens internas. Por outro lado, diante da necessidade de obtenção de informações em tempo real da linha de produção pelas redes remotas corporativas e conjunta à adoção de protocolos abertos, incorpora-se vulnerabilidades a serem exploradas por atores externos ao ICS [de Azambuja and Almeida 2021]

A integração dos ICS como parte responsável por serviços críticos como abastecimento de energia e água, alerta para a possibilidade de falhas de operação, que não apenas trazem prejuízos de ordem material como podem resultar em danos à população na medida em que outros serviços críticos dependam dos recursos negados. Como exemplo, pode-se citar o caso da empresa de fornecimento de energia elétrica ucraniana *Prykarpattyaoblenergo*, que reportou a interrupção do serviço aos seus clientes, causada, em um primeiro momento, pelo acesso ilegal de uma empresa parceira em seus sistemas SCADA. As investigações mostraram que o incidente se tratou de um ataque cibernético de grandes proporções, afetando cerca de 225 mil clientes em várias regiões da Ucrânia, devido à invasão no sistema SCADA da empresa e desconexão de várias subestações [Lee et al. 2016].

O último relatório de detecções da Kaspersky [CERT 2023] reportou que 40,6% dos computadores em ICS protegidos pela empresa, globalmente, tiveram atividade maliciosa detectada, sendo que as principais fontes de ameaça são provenientes da Internet. Reportaram, ainda, um aumento de 1 ponto percentual no segundo semestre no setor de energia, que alcançou 34,5% de computadores com objetos maliciosos bloqueados pela plataforma de segurança.

Uma das principais diferenças entre sistemas industriais e ambientes padrão de TI reside no uso de sistemas legados e desatualizados, cuja atualização ou remoção é dificultada pela incompatibilidade de componentes industriais com versões mais modernas, bem como pela complexidade de implementar as atualizações, que podem ser restringidas no caso da inviabilidade de se pausar ou transferir uma linha de produção [Huda et al. 2017]. Por vezes, só o planejamento de uma atualização de algum componente, do qual foi identificada uma vulnerabilidade, pode levar tempo relevante e suficiente para que tal vulnerabilidade seja explorada por *hackers*.

Para fazer frente a essas ameaças, técnicas diferenciadas de segurança cibernética podem ser empregadas, como o uso de *honeynets*, *appliances* específicos para ICS e virtualização com monitoramento de *hosts*. A detecção dos efeitos de uma invasão ou de *malware* por meio de *Machine Learning* (ML), em especial nos sistemas de detecção de intrusão (IDS), tem se mostrado eficaz e eficiente, com diversos trabalhos publicados [Buczak and Guven 2016]. Contudo, as técnicas de invasão continuam a ser desenvolvidas orientadas à complexidade e sofisticação, buscando contornar os esforços de proteção cibernética [Huda et al. 2017].

Os IDS podem operar de três modos: baseado em assinaturas, focados em detecção de novidades e com técnicas híbridas dos dois primeiros [Buczak and Guven 2016]. A detecção baseada em assinatura é muito eficaz para detectar indícios de invasão, contudo depende do aprendizado de assinaturas disponíveis e, portanto, de frequentes atualizações da base de dados. [Viegas and Santin 2020] verificaram que, após um ano utilizando um *dataset* de tráfego de rede real, sem atualizações para novos treinos pelos classificadores, houve uma queda de até 23% no desempenho da detecção.

Na detecção focada em anomalias o aprendizado vale-se dos registros normais da rede e qualquer registro diferente tende a ser interpretado como anomalia. Eficaz para novos ataques, essa abordagem traz o problema de acusar um elevado número de falsos alarmes, dado que, via de regra, não é possível aprender todo o conjunto de possíveis comportamentos legítimos na rede.

Com isso, a proposta deste trabalho é verificar como o uso de classificadores supervisionados e não-supervisionados em duas etapas pode contribuir para o aumento da detecção de anomalias em dados de componentes de um ICS, tais como leituras de sensores de temperatura e pressão e variáveis de controle de um sistema SCADA ou de parâmetros de ajustes em interfaces homem-máquina, contribuindo no esforço de trabalho de detecção iniciado por [Shin et al. 2020].

Espera-se que tais medidas possam apresentar, em comparação a abordagem supervisionada, melhores resultados para as métricas de detecção baseadas em *machine learning*, dado que a análise supervisionada, apesar de proporcionar bons resultados no esforço da detecção, é dependente de informações prévias que, em caso de ataques inéditos, podem não estar disponíveis para aprendizado do classificador. Neste caso a classificação não supervisionada pode vir a complementar esse esforço, na medida que se adapte e solucione problemas de detecção de anomalias.

Nos trabalhos verificados que utilizam a plataforma de [Shin et al. 2022], é perceptível a inclinação pelo uso de *Deep Learning*, principalmente de *autoencoders* e técnicas para predição de valor segundo uma série temporal. Este trabalho buscou utilizar o reconhecimento de padrões para contribuir no esforço da pesquisa sobre os dados extraídos da plataforma, que diferem dos dados de redes TCP/IP usuais e que já possuem trabalhos demonstrando o amplo uso da detecção de anomalias por meio de *Machine Learning*.

2. Trabalhos anteriores

Algumas das técnicas para desenvolver sistemas de detecção de intrusão envolvem diferentes áreas da computação que, apesar de relacionadas, possuem seu próprio nicho de

conhecimento. Nos trabalhos verificados notamos uma concentração em detecções baseadas em Teoria dos Grafos, *Machine Learning* e *Deep Learning*.

A revisão da literatura concentrou na busca por invasões em sistemas industriais, segurança de Sistemas de Controle Industrial, segurança de *Smart Grid*, Sistemas de Detecção de Intrusão e detecção de anomalias. Também focamos em trabalhos que contribuíram para a pesquisa de [Shin et al. 2020], tendo em vista os resultados de sua plataforma de testes serem mais próximos de um sistema real por utilizar componentes reais na simulação.

Segundo [Buczak and Guven 2016], o objetivo do uso de *Machine Learning* é a classificação ou predição de um dado e para tanto três abordagens são possíveis: supervisionada, semi-supervisionada e não-supervisionada. Nesta, não se conhece a classe dos dados trabalhados e busca-se encontrar estruturas ou padrões que possam agregar os registros e assim obter algum conhecimento. Quando são rotulados uma fração dos dados, durante a coleta ou após inspeção humana, o problema é dito semi-supervisionado. No supervisionado as classes são conhecidas e os dados são rotulados e geralmente busca-se encontrar uma função que modele o problema.

A tarefa de classificação de dados segue uma metodologia com diversas variáveis a cada etapa, passando, por exemplo, pela seleção de atributos e redução de dimensionalidade, conforme a necessidade (custos computacionais) ou características dos dados. Utilizar modelagem descritiva de classificação pode ser útil para depreender informações relevantes de determinados valores de atributos, como, por exemplo, ao observar a construção de uma árvore de decisão. Por outro lado, a modelagem preditiva foca em melhorar o resultado da classificação, preterindo a compreensão do modelo, o que pode ser mais adequado conforme o experimento [Witten et al. 2017].

Saber *a priori* qual dos modelos é mais adequado para a classificação em um conjunto de dados não é uma tarefa trivial e nos trabalhos observados não é incomum realizar os experimentos com diversos modelos e anotar os que possuem melhor resultado para os dados trabalhados [Huda et al. 2017]. Para comparação de modelos, em problemas binários (i.e. que apresentam duas possibilidades de classificação), é comum observar a Matriz de Confusão, como apresenta a Tabela 1:

- *Verdadeiros Positivos* (VP): são os registros preditos como anomalias e que assim são classificados na realidade.
- *Falsos Positivos* (FP): são registros que foram preditos como anormalidade mas que deveriam ter sido classificados como normais.
- *Falsos Negativos* (FN): são registros que foram preditos como de comportamento normal e que deveriam ter sido classificados como anomalias.
- *Verdadeiros Negativos* (VN): são registros preditos corretamente como normais.

Há variadas formas de comparar os modelos com base nos valores obtidos da matriz de confusão. Os trabalhos que se utilizam de *Machine Learning* costumam apresentar o desempenho da classificação de problemas binários segundo alguma métrica, como veremos adiante. [Buczak and Guven 2016] apontam como algumas das métricas mais utilizadas em trabalhos dessa natureza:

- Acurácia: $(VP+VN)/(VP+VN+FP+FN)$. Útil para problemas de classes balanceadas.

Tabela 1. Matriz de Confusão

		Classe real		
		Anomalia	*Normalidade*	Total Predito
Classe predita	*Anomalia*	VP	FP	$VP + FP$
	Normalidade	FN	VN	$FN + VN$
	Total real	$VP + FN$	$FP + VN$	

- Precisão: $VP/(VP + FP)$. Traduz a capacidade da modelagem em não rotular como anormalidade um registro de comportamento normal.
- *Recall* ou Sensitividade ou Taxa de detecção: $VP/(VP + FN)$. Reflete a habilidade do modelo em encontrar todas as anomalias.
- F1-Score: $2VP/(2VP + FP + FN)$. Média harmônica entre a Precisão e o *Recall*.

[Buczak and Guven 2016] verificaram em seu *survey* que os trabalhos de detecção de intrusão valeram-se de *datasets* baseados em capturas de pacotes de rede, *NetFlow* ou demais dados de redes. Dentre os mais conhecidos destacam o KDD-99 [1], que compreende milhões de registros de tráfego de rede, com diferentes categorias de ataques tanto no conjunto de treino quanto de teste. Apesar desses dados não serem provenientes de leituras ou capturas de dispositivos de campo típicos de um ambiente industrial, eles compartilham semelhanças com o tráfego de uma rede padrão no nível corporativo, que pode ser vetor de entrada para as atividades ilícitas, não podendo ser ignoradas [Ani et al. 2016].

[Feng et al. 2017] propuseram um esquema de dupla detecção de anomalias visando reconhecer ataques de negação de serviço (DoS) em um sistema SCADA de gasoduto. O primeiro detector, baseado em classificação semi-supervisionada com clusterização por *K-means*, inspeciona pacotes de rede característicos de ICS em busca de assinaturas presentes em sua base de dados, previamente construída com operação normal e ataques simulados. Se o padrão do pacote não é reconhecido, é dito anômalo. Se é reconhecido como de comportamento normal, é encaminhado para o segundo detector, baseado em *Long-short Term Memory* (LSTM), que compara os dados do pacote com uma predição baseada em leituras anteriores. O resultado alcançado foi de 92% de acurácia, porém foi alertado para o custo temporal para treinamento do LSTM.

Utilizando um *dataset* gerado por simulador digital de tempo real para um cenário de usina de geração de energia elétricas, [Wilson et al. 2018] propuseram um detector de anomalias baseado em redes neurais por meio de *Stacked Auto-Encoder* (SAE). Na primeira fase há o treino do modelo com dados históricos gerados para esse fim, tendo por resultado *features* de alto nível, utilizadas na segunda fase. Nessa, essas estruturas são acopladas a uma camada para a classificação supervisionada e ocorre o treino com retropropagação. A aplicação do modelo alcançou 96% de acurácia em amostras balanceadas, marginalmente melhor que o modelo de base comparado.

Em [Sanz and Duarte 2019], teoria dos grafos foi utilizada para o enriquecimento de dados e, em conjunto com *Machine Learning* para a descoberta de padrões em ataques distribuídos em uma rede IP convencional. Os autores dispuseram de informações

[1] Disponível em: http://kdd.ics.uci.edu/databases/kddcup99/kddcp99.html

como campos dos protocolos TCP/IP, em capturas de pacotes em janelas de tempo, e organizaram os dados em vértices e arestas para construção do grafo e extração de suas métricas. As métricas extraídas do grafo compuseram os dados de cada captura e, após a tarefa de classificação, obtiveram ganhos de acurácia de até 15,7% quando comparados à classificação dos dados sem o enriquecimento. Utilizaram os classificadores supervisionados *Decision Tree*, *Naive Bayes* e *Multilayer Perceptron*.

Diante da dificuldade em se obter dados em cenários de ataques em uma rede industrial, [Shin et al. 2020] propuseram uma plataforma para simular um sistema ciberfísico de uma usina de geração de energia elétrica, composta de uma hidroelétrica reversível e de uma termoelétrica baseada em vapor. Tal plataforma mescla componentes reais, como uma turbina GE Mark-Vle, com uma simulação em Hardware-In-The-Loop (HIL) que atuou combinando os modelos de geração na *Power Grid*. Esta plataforma de testes deu origem ao *HIL-based Augmented ICS Security Dataset* (HAI), cujos dados vieram a proporcionar base para pesquisas em segurança cibernética complementarmente aos trabalhos que focam em tráfego usual de rede e terminologias da tecnologia IP.

Desenvolvido com objetivo na pesquisa em detecção de anomalias em Sistemas Ciberfísicos, o HAI traz quatro processos que trabalham de forma interconectada: um processo para a caldeira (P1), um para a turbina a vapor (P2), um para a hidroelétrica reversível (P3) e um para o HIL, que faz a integração dos outros três e simula os modelos de geração de energia e balanço da demanda de carga (P4).

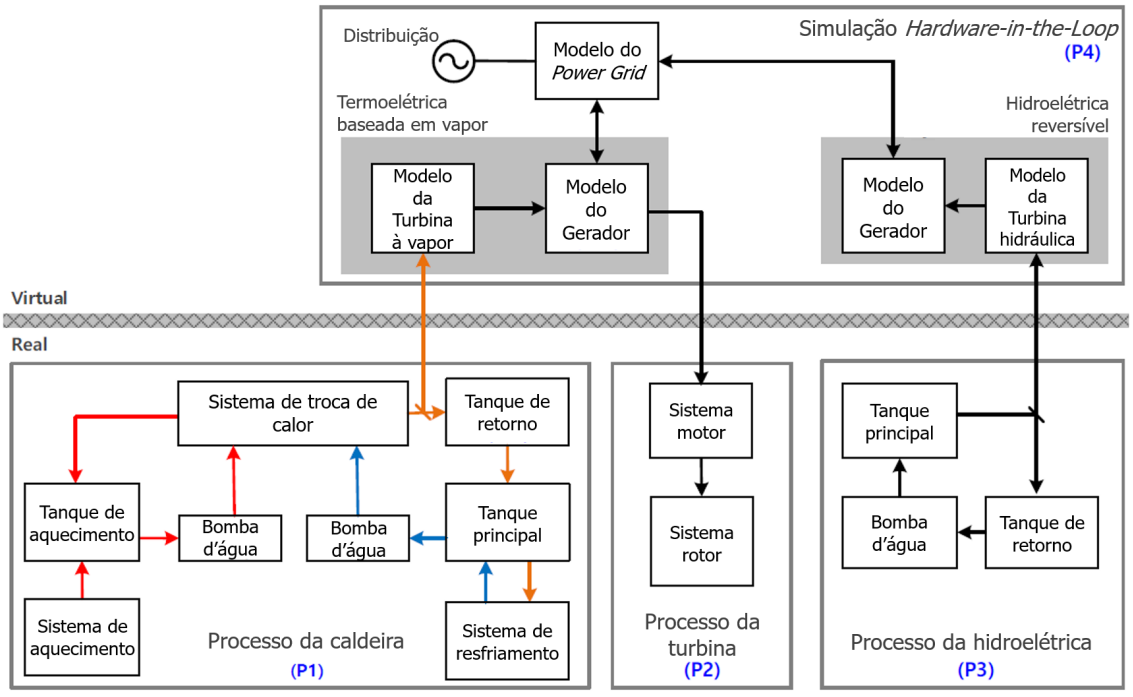

Figura 1. Esquema da plataforma HAI (Extraído de [Shin et al. 2022])

A Fig.1 ilustra a construção da plataforma de teste. O HAI já possui três versões: 20.07, 21.03 e 22.04, cada uma referente ao ano/mês em que os dados foram coletados. A cada versão houve melhorias, adição e subtração de *features* e de cenários de ataque. Os ataques são voltados para os processos emulados (P1, P2 e P3) e ocorrem ou em ajustes

de configuração do sistema, ou em variáveis de processo ou controle, ou em parâmetros de controle. Não foi foco de [Shin et al. 2020] a forma do ataque e sim observar os efeitos nas medições dos diversos pontos de coleta de dados e proporcionar a geração do *dataset* para posterior pesquisa de detecção de anomalias.

No trabalho de [Xingchao 2020] foi empregada classificação não supervisionada com redes neurais para análise de frequência em séries temporais. Chamado *Stacked Gated Recurrent Unit - Infrequent Residual Analysis*, o *framework* proposto foi testado com HAI-20 comparando os resultados com os obtidos pelo LSTM. Os dados foram particionados em treino, validação e teste, foram pré-processados por normalização e redução de ruídos e então aplicados no modelo proposto. A métrica utilizada foi *F1-score* para avaliar o desempenho, que variou de 0,811 a 0,977.

Em [Mokhtari et al. 2021] foi proposta uma abordagem de classificação supervisionada do HAI-20 a qual chamaram *"Measurement Data Instrusion System"* e que utiliza os classificadores *K-Nearest Neighbors* (KNN), *Random Forest* e *Decision Tree*. Para a seleção de *features* utilizaram a Correlação de Pearson, reduzindo de 59 para 17 atributos. Dado que o conjunto é bastante desbalanceado, aplicaram o método *Synthetic Minority Over-sampling Technique* (SMOTE), que gera novas instâncias da classe de interesse baseada nos registros existentes [Chawla et al. 2002]. Com isso, de menos de 4% dos registros associados a ataques foram gerados outros 46% para o balanceamento das classes. Foi realizado o pré-processamento dos dados com o *MinMax Scaler* e o conjunto foi dividido em proporção de 0,7 para o treino e 0,3 para os teste. Concluíram que o melhor classificador foi o *Random Forest*, que entregou 99,76% de acurácia no menor tempo de execução.

Nos trabalhos verificados que buscaram a detecção de anomalias no HAI, observamos uma preferência pelo uso de *Deep Learning* e recuperação de informação em séries temporais. Sem considerar o HAI, os trabalhos de detecção em captura de pacotes de redes TCP/IP apresentou maior variedade de técnicas, passando tanto por *Machine Learning* quanto por Teoria dos Grafos. Buscamos utilizar, portanto, algumas das técnicas de *Machine Learning* no HAI, com a finalidade de verificar como tais técnicas resolvem o problema da detecção de anomalias, seus potenciais e fragilidades, proporcionando meios de comparação para técnicas mais sofisticadas.

3. Proposta

Este trabalho propõe o uso de técnicas de *Machine Learning* para a detecção de anomalias em dados de uma rede industrial. Para tanto, utilizaremos o HAI na versão 22.04, por ser a mais recente e possuir mais atributos que as versões anteriores [Shin et al. 2022]. Em vez de optar pela abordagem supervisionada ou não-supervisionada da tarefa de classificação, implementamos uma detecção em duas etapas, a fim de verificar qual par de classificadores supervisionados e não supervisionados fornecem ganhos de desempenho na métrica *recall* ao comparar a predição com o conjunto verdade.

O uso do HAI vem em complemento às pesquisas de detecção de invasão nas redes corporativas, como verificado em alguns dos trabalhos anteriores, uma vez que os dados coletados na plataforma do HAI se aproximam do observado em conexões nos ICS entre sensores, PLCs e sistemas SCADA. Dados como temperatura e pressão são numéricos e prontos para serem utilizados. Ademais, os mecanismos de segurança embutidos contra

super-aquecimento, quando presentes, ajudam a filtrar valores alterados bruscamente ou que ultrapassem as margens de segurança. Desse modo, ataques sutis passam a ser o foco da detecção e vários cenários de ataques foram empregados na construção do *dataset* [Shin et al. 2022].

Das três versões do HAI, as versões de 2020 e 2021 contêm, além da classe "attack", três classes binárias adicionais especificando o processo alvo do ataque (P1, P2 ou P3). Na versão de 2022 não há essa divisão no *dataset*, de modo que inspeções neste sentido necessitam de inserção manual dessa informação, presente na documentação que acompanha o *dataset*. Neste trabalho, tendo em vista que estamos interessados na detecção sistêmica desses processos, escolhemos como classe o atributo "attack", independentemente do processo associado. Entende-se que alguns ataques podem ocasionar efeitos nos demais processos e espera-se que os classificadores sejam capazes de detectar em qualquer caso.

Todo trabalho de aprendizado e reconhecimento de padrão é feito no contexto do conjunto, dado que alguns atributos foram alterados segundo a versão do *dataset*. De uma inspeção em cada coluna, retiramos da análise as que não apresentaram variação (isto é, desvio-padrão nulo). Para enriquecer os dados foram construídas quatro *features* por atributo, por período associado: máximo e mínimo dos últimos n registros, média móvel simples e média móvel exponencial dos últimos n registros. Para o valor de n serão observados os últimos 5 e últimos 10 registros, lembrando que cada registro corresponde a uma leitura em todo sistema em um intervalo de um segundo. Tendo em vista que o menor intervalo de cenário de ataque no HAI-22 é de 38 segundos, valores de n maiores podem acabar por diluir os impactos dos ataques nas leituras efeituadas. Valores muito pequenos, por outro lado, ajudam a observar mudanças mais bruscas no comportamento de uma sequência.

Propõe-se utilizar os conjuntos de teste do HAI-22, compostos de quatro arquivos no formato CSV. Os quatro arquivos contêm 100,3 horas de registros da operação da plataforma de teste, ou pouco mais de 360 mil instâncias de dados entre atividades normais e ataques. A Fig.2 ilustra o sistema proposto neste trabalho.

Para a seleção de *features* propomos o uso da função *SelectFromModel* do próprio Scikit-Learn [Pedregosa et al. 2011], com uso do classificador *RandomForest*. Dentre os testes prévios com o HAI, o *RandomForest* obteve, em geral, os melhores resultados para a tarefa de classificação. A outra vantagem do uso da função *SelectFromModel*, em vez de *RecursiveFeatureSelection-Cross Validation*, está no tempo computacional. Enquanto esta demandou 17 horas em testes com 60 *features*, a *SelectFromModel* apresentou seu resultado em questão de minutos.

Para determinar o tamanho da amostragem n baseada na estimativa da média populacional nos valemos da Equação 1, com valor de margem de erro E de 0,01, grau de confiança bi-caudal de 90% e grau de liberdade tendendo ao infinito, o que resulta, no valor da distribuição *TStudent* $Z\alpha/2$ em 1.645.

$$n = \left(\frac{Z\alpha}{2} \times \frac{\sigma}{E}\right)^2 \tag{1}$$

Para determinar o valor de σ, normalizamos os dados e determinamos o desvio-

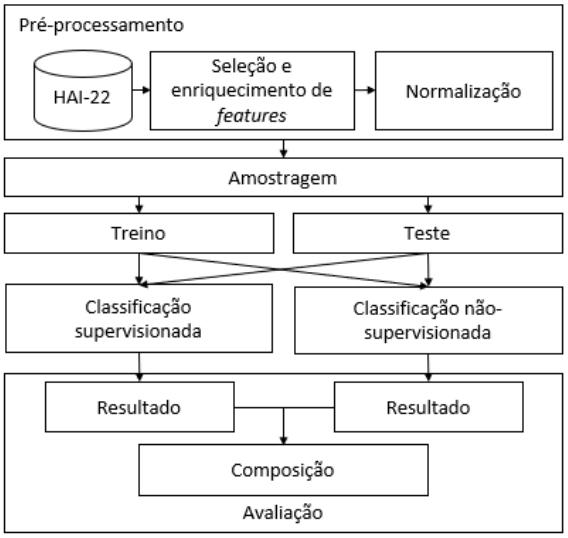

Figura 2. Sistema proposto

padrão de cada coluna, escolhendo o maior valor como o representativo da dispersão do universo de amostras. Tendo determinado n, criamos a cada iteração, com estado de aleatorização controlado, conjuntos de treino e teste segundo parâmetros de execução como proporção de anomalias e relação do tamanho treino/teste.

Os classificadores a serem empregados constam da tabela 2, fazendo parte do Scikit-Learn e são de uso amplamente aceito no meio científico [Pedregosa et al. 2011]. Há variados parâmetros de execução para cada classificador, porém em um primeiro momento optamos por deixar os valores *default*, ou seja, nas configurações padrão sendo indicado na própria tabela quando houver opções, como no caso do *Bagging*, dos baseados em *Support Vector Machine*, KNN e *Local Outlier Factor*. Foi fixado o estado aleatório para permitir a reprodutibilidade dos resultados.

Para avaliar o desempenho dos classificadores propomos o uso da métrica *recall* no lugar da amplamente utilizada métrica acurácia [Buczak and Guven 2016]. Como para este cenário um ataque não detectado impacta mais que um falso positivo, a métrica *recall* é adequada pois traduz a habilidade do classificador em encontrar as anomalias [Pedregosa et al. 2011]. A acurácia é amplamente utilizada porém não é adequada em problemas de classes desbalanceadas que, via de regra, são os cenários reais de ataques.

A proposta de detecção em duas etapas, indicada pela caixa "Composição" da etapa de Avaliação, na Fig.2, não necessita que os classificadores processem de forma serial. Neste trabalho apresentamos os resultados de uma das estratégias, que foi somar o resultado para cada registro e, caso qualquer um dos dois classificadores identifique a amostra como um ataque, ela será considerada como tal.

Uma vez que a composição seja avaliada, comparamos seu desempenho com o desempenho dos classificadores isolados para então concluir sobre a possibilidade de adoção da estratégia em casos reais.

Tabela 2. Classificadores empregados

Tipos	
Supervisionados	Não-supervisionados
Ada Boost	*One Class SVM - linear*
Bagging (Decision Tree)	*One Class SVM - radial basis function*
Bagging (Extra Trees)	*Local Outlier Factor - Novelty*
Bagging (Random Forest)	*Local Outlier Factor*
Decision Tree	*Isolation Forest*
Extra Trees	
Gaussian Naive Bayes	
Gaussian Process	
K-Nearest Neighbors (K=2)	
K-Nearest Neighbors (K=5)	
Logistic Regression	
Multilayer Perceptron	
Perceptron	
Quadratic Discriminant Analysis	
Random Forest	
Stochastic Gradient Descent	
Support Vector - linear	
Support Vector - radial basis function	

4. Resultados

Com a remoção prévia e o enriquecimento de dados, inicialmente com 86 colunas, o *dataset* resultou em 488 colunas. A seleção de *features* pela ferramenta resultou em 129 atributos dentre os 448 passados. Já em relação à amostragem, determinada pela Equação 1, o tamanho calculado foi de 3.929 amostras. Realizamos o teste com a proporção de anomalias em 50% e 500 iterações.

A Fig.3 apresenta o diagrama de caixa da distribuição de escores obtidos pela métrica *recall* das detecções realizadas pelos classificadores propostos. Os valores máximos foram obtidos pelos classificadores *Bagging* com *Extra Trees*, *Extra Trees* e *Random Forest*, com medianas das distribuições em 0,98. Dentre os classificadores não-supervisionados o *One Class SVM* com *kernel radial basis function* obteve o melhor resultado, com mediana em 0,69.

Os classificadores que entregaram a maior dispersão de escores foram o *Perceptron*, com variações entre 0,1 e 1,0 e mediana 0,71, e o *Stochastic Gradient Descent*, com variação semelhante porém mediana em 0,82. As variações dos demais classificadores foram bem contidas, a despeito do número de iterações, ficando na média de 6,88E-2 entre valores máximos e mínimos.

Em relação à detecção em duas etapas, o comportamento geral observado foi uma melhora na distribuição das métricas de *recall* quando são utilizados os classificadores *One Class SVM*, com destaque para o *kernel* RBF. Os classificadores *Local Outlier Factor* e *Isolation Forest* em geral não alteraram o desempenho obtido pelos classificadores supervisionados. A Fig.4 apresenta a melhora obtida nas medianas com a detecção em duas etapas para cada classificador supervisionado.

Também para os melhores classificadores, que obtiveram na métrica *recall* a medi-

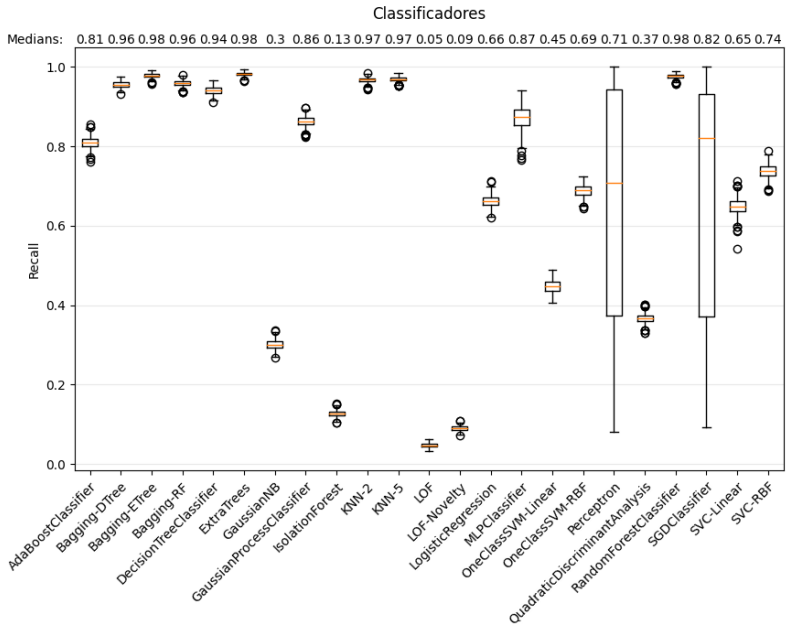

Figura 3. Distribuição de escores *recall* por classificador

ana da distribuição em 0,98, com a segunda etapa alcançaram medianas em 0,99. A maior diferença foi observada para o classificador *Gaussian Naive Bayes*, que obteve mediana em 0,30 e na segunda etapa, com o *One Class SVM* alcançou 0,75. Naturalmente, classificadores cujas distribuições apresentaram medianas mais próximas a 1 tiveram pouco espaço para melhoria. Se considerarmos apenas as medianas superiores a 0,9, o ganho médio, comparado à detecção em uma etapa, foi de 1,72%.

Foi verificado que o aumento na inclusão de exemplos de ataque no conjunto de treino e teste apresentou melhores resultados tanto para os classificadores supervisionados quanto para os não-supervisionados. Enquanto que os supervisionados não obtêm exemplos suficientes de assinaturas para classificar corretamente quando do ataque, os não-supervisionados são prejudicados pela construção da métrica *recall*, dado que uma diminuição no número de anomalias, e, portanto verdadeiros positivos, influencia para baixo este valor.

A etapa de enriquecimento de *features* ocorre antes que a seleção pela ferramenta e, em testes realizados invertendo essa ordem, a melhora observada em ganhos de métrica *recall* foi menor, ficando em 18,9% na média, contra 24,1% pela forma descrita. Já sem o enriquecimento de *features*, o ganho médio foi de 21,7%, marginalmente acima do observado ao realizar a seleção antes do enriquecimento, contudo inferior ao observado realizando a seleção após o enriquecimento. De fato, dentre as 129 *features*, a grande maioria é resultante do enriquecimento e não foi incomum observar atributos que não foram selecionados mas suas derivações sim.

Como continuação do trabalho pretendemos utilizar o sistema proposto em classificações baseadas em fluxo de dados, combinada com *drifted detection*, já inclinando para uma aplicação prática da solução do problema de detecção dos ataques cibernéticos.

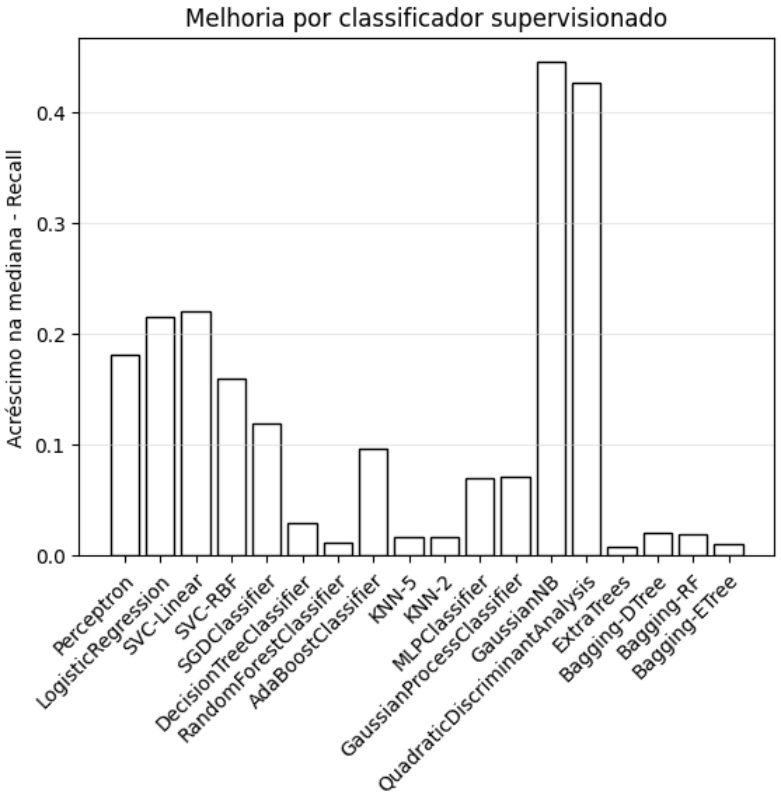

Figura 4. Melhoria absoluta com a dupla detecção

5. Conclusão

Os ambientes de redes industriais vêm se tornando alvo de cibercriminosos com a adoção de protocolos abertos e mudança de cultura organizacional para execução remota de rotinas operacionais, incorporando vulnerabilidades a serem exploradas e que podem resultar em impactos a nível nacional.

Diversas abordagens são empregadas para identificar invasões tanto na rede corporativa quando na comunicação entre os dispositivos de campo, seja com auxílio de *Machine Learning* ou *Deep Learning*. Tais técnicas exigem dados representativos do problema estudado, o que nem sempre é viável pelas características do ambiente. Ademais, a obtenção de assinaturas de atividades maliciosas, sejam simuladas ou adquiridas de situações reais, não cobrem todas as possibilidades de ataques, demandando constância no monitoramento e na proteção cibernética.

Neste trabalho buscamos contribuir com a pesquisa de detecção de anomalias no *dataset* HAI, que simula uma rede industrial de uma estação conjunta de energia elétrica baseada em termo-usina e hidroelétrica reversível, pelo uso de *Machine Learning*, propondo um sistema com uma abordagem de duas etapas. Propomos um sistema para detecção sistêmica, com amostragem dos registros para o treinamento, construção e seleção de atributos e classificação. A avaliação foi feita em Python com uso da biblioteca Scikit-Learn.

Nos trabalhos verificados não foi observado um padrão nas técnicas e métricas empregadas, conferindo amplo espaço para pesquisa e colaboração.

Os resultados obtidos apontaram para uma melhora do desempenho avaliado pela métrica *recall* na adoção conjunta do par de classificadores supervisionados e não-supervisionados, em média com ganhos de 24%, superior aos testes com classificador isolado ou sem o enriquecimento dos dados.

A melhora do desempenho obtido com a amostragem dos dados encaminha para uma detecção em fluxo de dados, que vem ao encontro da aplicação prática da solução do problema, bem como para a elaboração de estratégias para a composição dos resultados supervisionados e não-supervisionados.

Tendo em vista que os *datasets* fornecidos possuem centenas de milhares de registros, ainda que o treino dos classificadores seja realizado de forma *off-line*, o custo temporal e computacional é elevado, não apenas pelo volume de registros, mas também pela dimensionalidade do *dataset* após o enriquecimento. Com isso, pretendemos prosseguir na pesquisa de detecção valendo-se dos resultados obtidos neste trabalhos em conjunto com a detecção em fluxo, que reduz o custo para o treinamento ao se ajustar a janela de dados para os mais recentes, segundo critérios a serem definidos.

Referências

Alanazi, M., Mahmood, A., and Chowdhury, M. J. M. (2023). SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues. *Computers & Security*, 125:103028.

Ani, U. P. D., He, H. M., and Tiwari, A. (2016). Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective. *Journal of Cyber Security Technology*, 1(1):32–74.

Brasil (2018). Decreto nº 9.573, de 22 de novembro de 2018. Política Nacional de Segurança de Infraestruturas Críticas. Diário Oficial da União. Imprensa Nacional.

Buczak, A. L. and Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176.

CERT, K. I. (2023). Threat landscape for industrial automation systems for h2 2022. Technical report, AO Kaspersky Lab.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321 – 357.

de Azambuja, A. J. G. and Almeida, V. R. (2021). Um estudo bibliométrico das publicações sobre Segurança Cibernética na Indústria 4.0. *Research, Society and Development*, 10(3):4210312937e.

Feng, C., Li, T., and Chana, D. (2017). Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE.

Huda, S., Miah, S., Hassan, M. M., Islam, R., Yearwood, J., Alrubaian, M., and Almogren, A. (2017). Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data. *Information Sciences*, 379:211–228.

Lee, R. M., Assante, M. J., and Conway, T. (2016). Analysis of the Cyber attack on the Ukrainian power grid. Technical report, Electricity Information Sharing and Analysis Center (E-ISAC), SANS.

Mokhtari, S., Abbaspour, A., Yen, K. K., and Sargolzaei, A. (2021). A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics*, 10(4):407.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sanz, I. and Duarte, O. (2019). Graph-based feature enrichment for online intrusion detection in virtual networks. In *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 129–136, Porto Alegre, RS, Brasil. SBC.

Shin, H.-K., Lee, W., Yun, J.-H., and Kim, H. (2020). *HAI 1.0: HIL-Based Augmented ICS Security Dataset*. USENIX Association, USA.

Shin, H.-K., Lee, W., Yun, J.-H., and Min, B.-G. (2022). HAI - HIL-Based Augmented ICS Security Dataset Version 22.04.

Viegas, E. K. and Santin, A. O. (2020). Towards reliable intrusion detection in high speed networks. In *Anais Estendidos do XX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg Estendido 2020)*. Sociedade Brasileira de Computação - SBC.

Wilson, D., Tang, Y., Yan, J., and Lu, Z. (2018). Deep learning-aided cyber-attack detection in power transmission systems. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE.

Witten, I., Frank, E., Hall, M. A., and Pal, C. J. (2017). *Data Mining*. Elsevier LTD, Oxford, 4th edition.

Xingchao, B. (2020). Detecting anomalies in time-series data using unsupervised learning and analysis on infrequent signatures. *Journal of IKEEE*, 24(4):1011–1016.