Detecção de Intrusão e Análise Cyberfísica em Redes Industriais

Wagner Carlos Mariani[1,2], Anelise Munaretto[2], Mauro Fonseca[2], Heitor Lopes[2], Thiago H Silva[2]

[1]Instituto Federal de Educação, Ciência e Tecnologia Catarinense (IFC)
Videira, SC

[2]PPG em Engenharia Elétrica e Informática Industrial (CPGEI)
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, PR

wagner.mariani@ifc.edu.br,

{anelise,maurofonseca,hslopes,thiagoh}@utfpr.edu.br

Abstract. *This paper investigates cyber-physical security in Industrial Control Systems (ICS) facing emerging cybernetic risks. We have developed an anomaly-detecting system featuring a two-step classification: the first distinguishes between normal and anomalous operations, and then identifies the type of attack. SWaT dataset, a water treatment simulator, has been used, and techniques, such as SMOTE, have been applied to balance the data. Various machine learning algorithms have been tested, highlighting Random Forest due to its recall. Results show the proposed system can classify operations according to the state and type of attack.*

Resumo. *Este trabalho investiga a cibersegurança em Sistemas de Controle Industrial (ICS) diante de riscos cibernéticos emergentes. Desenvolvemos um sistema de detecção de anomalias com uma abordagem de classificação em duas etapas: distinguindo entre operações normais e anômalas, e identificando o tipo específico de ataque. Utilizou-se o dataset SWaT, um simulador de tratamento de água, e técnicas, como SMOTE, foram aplicadas para balancear os dados. Vários algoritmos foram testados, com destaque para o Random Forest pela sua capacidade de identificar os incidentes sem incorrer em falsos negativos (recall). Os resultados mostram que o sistema proposto pode classificar as operações de acordo com seu estado e tipo de ataque.*

1. Introdução

Sistemas de Controle Industrial (ICSs) são essenciais em diversos setores, incluindo frigoríficos, metalúrgicas e infraestruturas críticas como usinas hidroelétricas. Esses sistemas, que incluem tecnologias como SCADA, DCS, e CLPs, historicamente isolados, agora incorporam avanços da TI como computação em nuvem e inteligência artificial, elevando os desafios de segurança.A modernização dos ICSs traz a necessidade de novas abordagens de segurança para lidar com ameaças cibernéticas cada vez mais sofisticadas. Enquanto métodos de detecção de anomalias podem ser baseados em regras ou em modelos de aprendizado de máquina, este trabalho adota uma metodologia que se concentra no

comportamento físico do sistema para detectar anormalidades operacionais ou ataques cibernéticos. Ao aliar a informática à realidade operacional, este estudo avança na proteção dos sistemas no espaço ciberfísico, onde a integração dos componentes cibernéticos e físicos apresenta tanto elevado valor econômico quanto aumentados riscos de segurança.

2. Trabalhos Relacionados

O crescente interesse na aplicação de algoritmos de *Machine Learning* (ML) para a detecção de ataques em Sistemas de Controle Industrial (ICSs) tem levado a estudos diversificados. Por exemplo, pesquisadores na Universidade do Mississippi empregaram classificadores como J48 e *Naive Bayes* (NB) em *datasets* de gasodutos, explorando métricas de precisão, *recall* e F1-*score* [Morris et al. 2011, MSU Critical Infrastructure Protection Center 2013]. Outras investigações neste *dataset* aplicaram *Random Forest* (RF), *Support Vector Machine* (SVM) e NB, ilustrando a variedade de métodos aplicáveis [Beaver et al. 2013]. Além disso, combinações de técnicas como J48, NB, RF, *Logistic Regression* (LR), e *Multi Layer Perceptron* (MLP) foram testadas no *dataset* WUSTL-IIOT-2018 para simular operações de ICS sobre ModBUS TCP/IP [Jones et al. 2014, Teixeira et al. 2020]. Pesquisas no campo do *Industrial Internet of Things* (IIoT) destacam a importância do balanceamento de *datasets* e pré-processamento, onde técnicas como RF se mostraram particularmente eficazes quando aplicadas a *datasets* enriquecidos com amostras sintéticas [Eid et al. 2024]. Especificamente tratando-se do *dataset* SWaT, que será usado no desenvolvimento desta proposta, o trabalho de [Kravchik and Shabtai 2018] usa redes neurais convolucionais (CNN) para diferenciar estados normais e de ataque, e seus melhores resultados dentro das métricas utilizadas foram de 0,968 para precisão, 0,791 para *recall* e 0,871 para F1-*Score*. Por outro lado, o trabalho de [Inoue et al. 2017] compara o uso de SVM com o uso de redes neurais profundas (DNN) para detectar diretamente os tipos de ataques. Este trabalho usou precisão (0,982 para DNN e 0,925 para SVM), *recall* (0,678 para DNN e 0,699 para SVM) e F1-*Score* (0,802 para DNN e 0,796 para SVM) como métricas de resultados.

3. O *Dataset* SWaT

O *dataset* utilizado neste trabalho traz dados uma instalação laboratorial em escala reduzida de uma planta de tratamento de água, totalmente funcional. Esta instalação é composta por CLPs, Interfaces Homem-Máquina (HMIs) e um SCADA. A planta industrial está organizada em seis grandes áreas, cada uma equipada com diversos sensores e atuadores, totalizando 51 componentes.

Os dispositivos na planta operam utilizando o protocolo industrial Modbus sobre redes IP, com todos os dados sendo capturados por um *sniffer* na rede local Ethernet e armazenados em um servidor. O período de coleta totaliza 11 dias, divididos em uma **Fase Inicial** de 7 dias de *ramp-up*, durante a qual nem todos os sensores e atuadores estavam ativos, e uma **Fase de Ataque** de 4 dias, na qual ataques foram simulados intercalados com operação normal para estabilização do sistema.

Os dados estão disponíveis em duas formas para os interessados: diretamente dos arquivos do *sniffer* Ethernet, que estão divididos em vários arquivos, ou em planilhas eletrônicas do Excel, onde o estado de cada sensor ou atuador é registrado a cada segundo. As planilhas estão divididas em dois arquivos, um para a fase inicial e outro para a fase de

ataques. Cada planilha contém uma coluna com o carimbo de data/hora exato, 51 colunas com dados dos sensores e atuadores, e uma coluna com o rótulo da classe: "*Normal*" ou "*Attack*". Para este trabalho, optou-se pela utilização dos dados das planilhas físicas, que foram revisados e enriquecidos, como detalhado na próxima subseção. Interessantemente, novos experimentos com ataques inéditos também foram realizados posteriormente e são oferecidos separadamente, permitindo que futuros trabalhos utilizem estes dados para analisar comportamentos não previamente treinados pelos modelos. Este e outros *datasets* são mantidos pela *Singapore University of Technology and Design*, e a autorização para seu uso e detalhes sobre eles podem ser obtidos através do repositório *online* disponível em [Singapore University of Technology and Design 2024].

3.1. Análise inicial dos dados

Inicialmente foram identificados 41 ataques diferentes. Os ataques 5, 9, 15 e 18 foram descritos como ineficazes pelos mantenedores do *dataset* e foram excluídos da análise. O ataque 26 também foi informado como contendo valores não realistas devido a uma falha técnica. Os ataques 4, 14, 30 e 31 apresentaram comportamentos distintos dos descritos e foram excluídos, resultando em 32 ataques efetivamente disponíveis para estudo. Os períodos de normalidade foram rotulados como "Normal", e os períodos de ataque receberam o acrônimo "ATQ" seguido do número correspondente ao tipo do ataque. Com isto, as planilhas passaram a ter 54 colunas, sendo: uma para o *timestamp*, 51 para cada sensor ou atuador, uma para a classificação binária e uma nova para o tipo de ataque.

Os primeiros 300.000 registros da planilha inicial foram removidos, e incluiam 100.000 correspondente ao período de *ramp-up* e o restante excluídos para equilibrar a quantidade de dados classificados como normais entre a primeira planilha, que representa os primeiros sete dias a partir do início, e a segunda planilha, onde períodos de normalidade e ataque se alternam. Para a classificação Normal/Ataque o dataset final continha 615.512 (97,14%) e 18.114 (2,86%) segundos, respectivamente. A análise da duração dos diferentes tipos de ataques revela um forte desbalanceamento do *dataset*. O maior ataque (ATQ13) tem duração de 2254 segundos, enquanto o menor ataque (ATQ34) tem duração de 99 segundos, sugerindo um problema de natureza complexa.

3.2. Normalização dos dados

Nesta etapa, considerou-se que os dados das diferentes colunas de sensores e atuadores no *dataset* podem ser de duas naturezas. Primeiramente, valores categóricos, que são representados como 0, 1 ou 2, correspondendo a "sem sinal", "desligado" e "ligado", respectivamente. Esses valores indicam o estado de motores, válvulas abertas ou fechadas, e outros componentes. Em segundo lugar, valores contínuos, que são representados em ponto flutuante e refletem as leituras reais dos sensores, como níveis de tanques, temperaturas, e pressões.

Para garantir uma análise mais eficaz, é crucial normalizar os valores contínuos devido às diferenças de escala, que podem prejudicar a interpretação dos fenômenos monitorados. A normalização dos valores categóricos, no entanto, poderia distorcer a interpretação do estado dos equipamentos ou sensores. Portanto, apenas os valores contínuos foram normalizados, utilizando-se o método *Min-Max Scaling*, que ajusta os dados a uma escala comum sem distorcer as diferenças nos intervalos de valores.

4. Metodologia de tratamento de dados

Embora a normalização seja uma etapa importante de pré-processamento, ela não é, por si só, suficiente para preparar os dados para os algoritmos de classificação. Assim, foram realizadas também outras tarefas, como a extração de características (*features*) das séries temporais, a redução de dimensionalidade, a produção de amostras sintéticas para equilibrar o *dataset* e a divisão dos dados em diferentes blocos de treinamento e teste para as distintas fases a que serão submetidos.

4.1. Extração de características e redução de dimensionalidade

Para otimizar a classificação no *dataset* SWaT, a biblioteca TSFEL foi empregada para extrair características temporais e estatísticas de janelas de tempo, excluindo as características espectrais consideradas menos relevantes para o contexto. Este processo gerou 60 características por janela, ajustando o tamanho da janela para capturar nuances significativas do tratamento de água e assegurar a representatividade das amostras.

Devido ao desbalanceamento do *dataset*, a técnica de sobreposição foi adotada, utilizando janelas de 30 segundos com sobreposição de 6 segundos, para suavizar as transições entre janelas e melhor representar todos os tipos de ataques. O uso da TSFEL reduziu o número de amostras de 633.626 para 24.370, enquanto o número de características aumentou para 2.655, intensificando o desafio do *"curse of dimensionality"*.

Para enfrentar esse aumento de dimensionalidade, aplicaram-se métodos de redução como o *Fast Correlation-Based Filter* (FCBF) e o ReliefF. Esses métodos selecionaram características relevantes, eliminando redundâncias e fortalecendo a capacidade do modelo de distinguir entre classes normais e de ataque. A aplicação dessas técnicas reduziu o número de características para 600, incluindo 3 informativas (sendo *timestamp*, classes e tipo de ataque) e 597 características de sensor/atuador, o que reflete uma média de 11,7 características por sensor/atuador original. Este refinamento das características conduziu a um conjunto de dados mais gerenciável e eficaz, permitindo análises mais precisas e facilitando a diferenciação entre os estados normais e de ataque, bem como a identificação específica dos tipos de ataque.

4.2. Balanceamento dos dados e separação do *dataset* em treinamento e teste

Devido ao significativo desbalanceamento do *dataset*, a técnica SMOTE (*Synthetic Minority Over-sampling Technique*), conforme proposto por [Chawla et al. 2002], foi utilizada para mitigar este problema. O SMOTE cria amostras sintéticas através da interpolação entre amostras vizinhas da mesma classe. Este processo não apenas duplica amostras existentes, mas principalmente introduz variações sutis nas novas amostras, efetivamente adicionando um ruído controlado. Essas variações são cruciais para enriquecer o conjunto de dados, proporcionando um treinamento mais robusto ao modelo ao diversificar as características dentro de cada classe, o que é vital para uma generalização eficaz dos classificadores.

Na fase inicial de classificação, que distingue apenas entre estados "Normal" e "Ataque", o balanceamento foi ajustado especificamente para cada um dos 32 tipos de ataque. Assim, cada tipo, tanto ataque como normal, foram representados por 2.752 amostras, distribuídas em 86 amostras por tipo dentro da categoria "Ataque". Este balanceamento cuidadoso assegura uma representação equitativa de cada tipo de ataque durante o treinamento do modelo.

Após o balanceamento, o *dataset* foi dividido em 70% para treinamento e 30% para teste de maneira estratificada e aleatória. Este conjunto de treinamento serve de base para a segunda fase de classificação, que se concentra especificamente nos diferentes tipos de ataques, após a remoção das amostras normais para afinar a precisão na identificação dos tipos de ataques.

Os resultados da classificação binária da primeira fase estabelecem o *dataset* para a segunda fase, utilizando as instâncias identificadas como ataques para a subsequente classificação detalhada de tipos de ataques. Ambos os conjuntos de dados mantêm duas colunas de classes, sendo utilizadas de forma distinta em cada fase para adaptar os dados às exigências específicas dos modelos de aprendizado de máquina, facilitando assim a detecção e mitigação eficaz de ameaças cibernéticas.

5. Métodos de Classificação

Os modelos de classificação detalhados nesta Seção são fundamentais para a análise deste estudo e são comumente empregados em pesquisas sobre classificação de dados, como indicado na Seção 2. A escolha desses métodos busca resolver o problema proposto e facilitar a comparação dos resultados com estudos similares.

O processo de classificação será realizado em duas fases distintas, aplicando os métodos *Naive Bayes* (NB), *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Logistic Regression* (LR), *Decision Tree* (DT) e *Random Forest* (RF) em ambas as etapas.

Para avaliar o desempenho na primeira fase, foram utilizadas as métricas de *Recall*, Precisão, *F1-score* e a curva ROC-AUC. O *Recall* é especialmente importante, pois indica a proporção de ataques que foram corretamente identificados. Para a segunda fase, devido às classificações multiclasses, foi utilizado o MCC no lugar da curva ROC-AUC, mantendo as demais métricas para uma avaliação consistente. Além disto, o tempo de processamento será monitorado para cada método de classificação, permitindo uma análise detalhada da eficiência operacional.

6. Implementação e Análise dos Resultados

Após a preparação e divisão dos *datasets*, detalhada na Seção 4, e com os métodos de classificação e métricas de avaliação definidos na Seção 5, procedeu-se a implementação prática. Os resultados da primeira fase, a classificação binária diferenciando estados "normais" e "ataques", são apresentados na Tabela 1. Estes resultados são obtidos aplicando os modelos, previamente treinados com o conjunto de treino, ao conjunto de **teste**, que era inédito para os modelos. A coluna de tempo na tabela representa o tempo total necessário para treinar e posteriormente classificar o modelo com o conjunto de teste. O mesmo se aplica para a Tabela 2.

Classificador	F1	Precisão	Recall	ROC	Tempo (s)
Naive Bayes	0.5185	0.9455	0.3571	0.6683	0.090
SVM	0.7693	0.8561	0.6985	0.7906	4.555
KNN	0.9545	0.9319	0.9782	0.9534	0.031
Logistic Regression	0.9509	0.9515	0.9504	0.9510	67.720
Decision Tree	0.9779	0.9658	0.9903	0.9776	1.814
Random Forest	0.9940	0.9880	1.0000	0.9939	6.693

Tabela 1. Resultados dos Métodos de Classificação na Primeira Fase

Durante a execução da primeira fase, cada modelo não só classificou as amostras, mas também criou um subconjunto do *dataset* original, especificamente com aquelas identificadas como ataques. Este processo resultou em conjuntos de dados únicos para cada modelo, que foram utilizados na segunda fase. Nesta etapa, então, cada modelo é treinado para diferenciar os tipos específicos de ataques. Os resultados destas avaliações específicas por modelo são apresentados na Tabela 2.

Classificador	F1	Precisão	Recall	MCC	Tempo (s)
Naive Bayes	0.9214	0.9609	0.9423	0.9398	0.05
SVM	0.6067	0.8428	0.6662	0.6682	0.33
KNN	0.8900	0.9351	0.9193	0.9193	0.02
Logistic Regression	0.9294	0.9593	0.9515	0.9513	78.82
Decision Tree	0.9497	0.9696	0.9658	0.9653	1.56
Random Forest	0.9823	0.9890	0.9880	0.9877	3.52

Tabela 2. Resultados dos Métodos de Classificação da Segunda Fase

Na avaliação dos modelos, o NB foi menos eficaz na primeira fase, com um *recall* baixo e alta precisão, indicando muitos falsos negativos; seu valor de ROC foi de apenas 0,6683. Melhorias foram notadas na segunda fase, alinhando-se com estudos anteriores [Morris et al. 2011], [Beaver et al. 2013]. Em contraste, o SVM teve um desempenho moderado em ambas as fases, com desafios similares observados em pesquisas como [Beaver et al. 2013], [Keliris et al. 2016], e [Inoue et al. 2017]; seu MCC de 0,6682 ainda sugere alguma eficácia.

O KNN se destacou com alto *recall*, essencial para a detecção de ataques, apesar de sua precisão reduzida afetar o *F1-Score* (0,8900). A LR variou de moderada a alta eficácia, sendo eficiente na classificação de estados normais assim como ataques na segunda fase, conforme visto em [Jones et al. 2014] e [Beaver et al. 2013]. O DT e o RF mostraram robustez, com o DT alcançando um *recall* de 0,9903 e um *F1-score* de 0,9779 na primeira fase, e o RF alcançando um *recall* perfeito de 1,000 e um *F1-score* de 0,9940, ambos demonstrando alta eficácia na identificação de ataques.

7. Conclusão e Trabalhos Futuros

Este estudo destacou o *Random Forest* (RF) como significativamente superior em termos de *recall* e precisão, usando um *dataset* enriquecido com amostras sintéticas, enquanto o *Naive Bayes* (NB) foi menos eficaz. O *Logistic Regression* (LR) e o *Decision Tree* (DT) mostraram desempenhos comparáveis, com o LR sendo o mais lento. O SVM teve um desempenho moderado e o KNN destacou-se principalmente pela sua rapidez.

As limitações do estudo incluem o uso de um conjunto de dados específico, o que pode limitar a generalização dos resultados. Expansão para diferentes conjuntos de dados e contextos, como variados tipos de ICSs além do tratamento de água, são recomendadas para validações mais abrangentes.

Para futuras pesquisas, sugerem-se várias direções entre elas podemos citar: experimentar diferentes métodos de detecção e classificação entre as fases de análise; ampliar a validação usando *datasets* de diversos contextos industriais; explorar substituições de métodos de classificação por técnicas de detecção de anomalias para uma melhor identificação de comportamentos atípicos; investigar o uso de *Graph Neural Networks*

(GNN) para modelar a planta industrial como uma rede de grafos interconectados; e ainda, aplicar múltiplos métodos de classificação em paralelo para uma validação mais robusta dos resultados.

Agradecimentos

Trabalho apoiado parcialmente pelo projeto Fapesp-SocialNet (2023/00148-0) e CNPq (314603/2023-9 e 441444/2023-7).

Referências

Beaver, J. M., Borges-Hink, R. C., and Buckner, M. A. (2013). An evaluation of machine learning methods to detect malicious scada communications. In *2013 12th International Conference on Machine Learning and Applications*, volume 2, pages 54–59.

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.

Eid, A. M., Soudan, B., Bou Nasif, A., and Injadat, M. (2024). Comparative study of ML models for IIoT intrusion detection: impact of data preprocessing and balancing. *Neural Computing and Applications*, 36(13):6955–6972.

Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C. M., and Sun, J. (2017). Anomaly detection for a water treatment system using unsupervised machine learning. In *Proc. IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1058–1065.

Jones, A., Kong, Z., and Belta, C. (2014). Anomaly detection in cyber-physical systems: A formal methods approach. In *Proc. 53rd IEEE Conference on Decision and Control*, pages 848–853.

Keliris, A., Salehghaffari, H., Cairl, B., Krishnamurthy, P., Maniatakos, M., and Khorrami, F. (2016). Machine learning-based defense against process-aware attacks on industrial control systems. In *Proc. IEEE International Test Conference (ITC)*, pages 1–10.

Kravchik, M. and Shabtai, A. (2018). Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proc. Workshop on Cyber-Physical Systems Security and Privacy*, page 72–83. ACM.

Morris, T., Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., and Reddi, R. (2011). A control system test bed to validate critical infrastructure protection concepts. *International Journal of Critical Infrastructure Protection*, 4(2):88–103.

MSU Critical Infrastructure Protection Center (2013). Home-page. `http://www.security.cse.msstate.edu/cipc/`. Acesso em: 27/04/24.

Singapore University of Technology and Design (2024). iTrust Labs Datasets. `https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/`. Acesso em: 27/04/24.

Teixeira, M., Zolanvari, M., and Jain, R. (2020). WUSTL-IIOT-2018. `https://dx.doi.org/10.21227/kzgp-7t84`. Acesso em: 27/04/24.