

Trabalho possui artefato?	Tipo do artefato:	SeloD: Artefato	SeloD: Artefato publicamente Disponível (comentários)	SeloF: Artefato	SeloF: Artefato Funcional (comentários)	SeloR: Experi	SeloR: Experimentos Reprodutíveis (comentários)	SeloS: Artefato	SeloS: Artefato Sustentável (comentários)
Sim	Software e Dados (e.g., modelo e dataset)	Não	Os códigos estão disponíveis em: https://github.com/Vuchikawa/pcapAdFilter Não apresenta documentação mínima, conforme orientações disponibilizadas em: https://doc-artefatos.github.io/sbseg2024/reinstrucoes.html	Não	Apesar de os códigos serem apresentados, não há, no repositório, um readme que auxilie em sua execução, conforme solicitado nas instruções para revisão.	Não	Problema em cascata. Há ausência de documentação que auxilie na execução do código.	Não	Mesmo problema dos demais selos, ausência de documentação.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ código python e arquivo TXT com a lista de flagged domâns disponíveis - falta README.md - falta documentação mínima	Não	- falta instruções ou arquivo de dependências do projeto - faltam informações sobre a versão do Python utilizada - faltam informações sobre o ambiente ou o código foi executado (e.g., SO) - instalação do pip leva a erros no gerenciador de pacotes Python no Ubuntu 22.04.3 LTS	Não		Não	- código com definições estáticas (e.g., 01 pcap) - código poderia ser mais intuitivo e auto-descritivo - não há documentação do código
Sim	Software	Sim	+ Disponível em um repositório estável - Sem documentação - Título - Resumo - Ainda que não exista documentação, o artefato é bastante simples (87 linhas de código em Python). - Recomendado a atribuição do selo.	Sim	+ Artefato funcional - Recomendado a atribuição do selo.	Sim	+ Experimentos reprodutíveis - Recomendado a atribuição do selo.	Sim	Ainda que o artefato seja bastante simples, seria interessante que os autores fornecessem informações adicionais ao repositório, juntamente com uma documentação mínima. Além disso, os arquivos pcap e flagged_domain usados pelo script poderiam ser passados por parâmetro para avaliar modificações no código-fonte. Não posso recomendar a atribuição do selo
Sim	Software e Dados (e.g., modelo e dataset)	Não	Apesar de o projeto estar disponível em: https://github.com/GTA-UFRJ/Woif.atecy/free/main o mesmo não atende aos requisitos necessários para receber o selo disponível, dispostos em: https://doc-artefatos.github.io/sbseg2024/reinstrucoes.html	Não	O projeto não apresenta em seu arquivo readme os requisitos mínimos listados nas orientações para revisão.	Não	O projeto não apresenta em seu arquivo readme os requisitos mínimos listados nas orientações para revisão.	Não	O projeto não apresenta em seu arquivo readme os requisitos mínimos listados nas orientações para revisão.
Sim	Software e Dados (e.g., modelo e dataset)	Não	+ Github disponível: https://github.com/GTA-UFRJ/Woif.atecy/ - Readme ou documentação mínima indisponível + Título - Resumo - Não posso recomendar a atribuição do SeloD, pois o repositório não contém documentação mínima e resumo do artigo.	Não	- Não se aplica pois não possui SeloD.	Não	- Não se aplica pois não possui SeloD. + Repositório apresenta software para coleta de dataset - Repositório não apresenta instruções para instalação e execução dos softwares. + Repositório apresenta datasets. - Repositório não apresenta instruções para processar os datasets (e gerar gráficos, por exemplo).	Não	- Não se aplica pois não possui SeloD.
Sim	Software e Dados (e.g., modelo e dataset)	Não	+ disponíveis em um repositório estável; - documentação mínima; - título; - resumo. - Não recomendo a atribuição do selo. Não há nenhuma documentação explicando o objetivo do repositório/nome do artigo. Apesar de existir um dataset (possivelmente interessante), não há explicação clara dele.	Não	- executado (código de app para celular); - informações adicionais: - lista de dependências; - lista de versões das dependências/linguagens/ambiente; - descrição do ambiente de execução; - instruções de instalação e execução; - um exemplo de execução mínima. - Não recomendo (weak reject) a atribuição do selo. Não possui conhecimento nem infraestrutura para validar o artefato. Não existem instruções para execução do código.	Não	- instrução para executar as principais reivindicações; - descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo, - descrição de detalhes técnicos específicos do ambiente (detalhes incluídos apenas no artigo). - Não recomendo (weak reject) a atribuição do selo pois não possui o ambiente adequado para validar os resultados obtidos.	Parcialmente	/ artefato esteja modularizado / organizado / intuitivo e de fácil compreensão / significa parcialmente. O código do cliente/servidor está bem separado assim como os datasets gerados. Entretanto, não tenho domínio de Kotlin para dizer se o código foi criado utilizando as convenções esperadas. Recomendo (weak accept) a atribuição do selo.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável (github) - documentação suficiente - título - resumo - código em um branch diferente do principal - Recomendado a atribuição do selo pelo fato dos autores não terem recebido as diretrizes para definição dos artefatos. Portanto, seguindo a orientação de reter a falta do título e resumo, e mesmo o código estando em outro branch que não o principal eu recomendo a atribuição do SeloD.	Sim	+ código executa gerando gráficos similares aos reportados no artigo + script para instalar as dependências + script para executar todas as etapas de análise - um pequeno erro no script para instalar as dependências O código disponível consome o dataset previamente definido e gera gráficos e análises similares às reportadas no artigo. Além disso, existem scripts complementares como a instalação de dependências e a coleta de dados. Especificamente sobre o script de instalação de dependências, eu obtive um erro ao executá-lo. Contudo, com uma pesquisa rápida no google eu resolvi o problema. Além disso, os autores deixaram um script (.sh) que instala todos os scripts python para a geração dos resultados. Eu considero isso como uma espécie de documentação sobre o script. Portanto, eu recomendo a atribuição do SeloF.	Não	Como apresentado no comentário do seloF, o código gera gráficos similares aos apresentados no artigo. Contudo, os resultados gerados não correspondem aos resultados reportados. Ao executar o script, os resultados gerados são referentes ao dia 28 de maio de 2023. Contudo, os resultados reportados no artigo são referentes ao dia 29 maio de 2023. Portanto, somente executando o código preparado pelos autores eu não conseguirei obter os mesmos resultados reportados. Aparentemente a base de dados original do dia 29 está disponível no repositório. Contudo, o/a hipótese de que é necessário pré-processar os dados para depois gerar os resultados informados no artigo. Como os dados e o script podem estar disponíveis, algum com tempo poderia até conseguir replicar os resultados apresentados. Assim, como os autores não deixaram os dados pré-processados para simplificar a replicação dos experimentos eu NÃO recomendo a atribuição do seloF.	Parcialmente	+ arquivos diferentes que contém código com objetivos únicos (módulos) - organização dos arquivos ruim + nomes de pastas e arquivos bons - pouca documentação - códigos (funções) com poucos comentários Aparentemente, todos os códigos que os autores usaram estão no repositório. Mas a organização poderia ser melhor. Encontrei códigos em lugares diferentes e com pouca documentação. O que ajuda a entender o que é cada pasta e arquivo são os nomes, pois eles são bem descritivos. A maioria do código está separado em funções com bons nomes, mas existem reseravas como o "case 1 e " case2" que dificultam o entendimento do código. Recomendo PARCIALMENTE a atribuição do selo do selo, pois as instruções não foram enviadas para os autores e muitos pontos poderiam ser melhorados. Contudo, o que mais pesa para minha decisão é que, se eu precisasse evoluir essa pesquisa, eu provavelmente conseguiria. Gostaria muito energia (tempo) avaliando os códigos. Contudo, acredito ser possível. Portanto, recomendo o SeloS PARCIAL.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	- documentação do repositório não apresenta informações mínimas do paper - README.md em formato demasiadamente textual - README.md deveria conter informações específicas sobre os experimentos do paper - README.md deveria conter um passo-a-passo de instruções de setup do ambiente e execução dos experimentos - README.md deveria conter as informações detalhadas sobre o ambiente de testes, incluindo descrição do SO, kernel, versão do Python e outras dependências - o main do repositório apenas contém um README.md que aponta para o branch develop, o que não parece ser necessário e nem fazer sentido	Não	- erro de execução no primeiro código Python (python3 data_collector.py) Traceback (most recent call last): File "/wp924/Correlation-Analysis-of-Network-Monitoring-Datas/codes/data_collector.py", line 318, in <module> request = File "/wp924/Correlation-Analysis-of-Network-Monitoring-Datas/codes/data_collector.py", line 124, in request f.close() UnboundLocalError: local variable 'referenced' referenced before assignment	Não	- falta de padrão na nomenclatura dos arquivos (alguns com "_" e outros com ".") - se os arquivos .py representam etapas do processo, a própria nomenclatura poderia ajudar a indicar isto - inglês e português misturado na nomenclatura de métodos, variáveis e mensagens - falta documentação mínima do código - listas de valores estaticamente definidas em código, como as listas de endereços "(banda(atras)).mp.br - redundância de código, como nas chamadas request() sendo que o que muda são os dois parâmetros internos (string) e os valores i e j - caminhos dos datasets definidos estaticamente em código, quando deveriam ser parâmetros de entrada ou configuração da execução - valores fixos em código e não documentados (e.g., 00.00, 20000000.0) - utilização de caminhos estáticos, fora em código, para diferentes arquivos (e.g., /lists/rp_list.pkl) - utilização de nomenclatura de variáveis inadequada (e.g., time1, time2, file1, file2, df1, df2, file1, file2, file1, file2, file_1, file_2, cont1, cont2, cont4, sum1, sum1, sum2, sum3) - busca por extensão de arquivos em diretórios, o que pode levar a problemas - diversas variáveis com valores fixos em código - um código não deve ser escrito para ser compreendido apenas pelo programador enquanto está programado - as mensagens de saída, da execução, não são nada instrutivas para o usuário, o que significa Ok Ok endereço Ok endereço endereço OK????	Não	- documentação mínima do código descrevendo arquivos, funções, funcionalidades etc - Legibilidade mínima de código + Possível que avaliadores consigam identificar a principais reivindicações do artigo no artefato Recomendo: Não recomendo a atribuição do selo. As funcionalidades (código) carecem de descrição, dificultando o entendimento do fluxo de execução das atividades. Por mais que o código esteja separado por módulos, as funções tornam-se obscuras para a avaliação. Para ser sustentável, sugiro que cada função ou grupo de funções seja descrita, tal como prevê a boa prática da linguagem utilizada.
Sim	Software	Sim	+ Repositório estável + Documentação mínima + Título + Resumo - Recomendado a atribuição do selo. Há discrepâncias no link apresentado no artigo, que funciona como uma "fora de rosta" para o repositório de um projeto. Porém não acredito que isso prejudicou o artefato, visto que não houve orientação específica de que o repositório do projeto deveria ser exclusivo para o evento e que a documentação deveria ser elaborada em "página única".	Sim	+ lista de dependências; + lista de versões das dependências/linguagens/ambiente; + descrição do ambiente de execução; + instruções de instalação e execução; - um exemplo de execução mínima. Recomendação: Recomendado a atribuição do selo. As dependências encontram-se em um arquivo padronizado de dependências na linguagem python, facilitando a criação do ambiente. Não há um exemplo de execução do script, mas existem orientações de execução que permitem a utilização do artefato.	Sim	+ Instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas). + Descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo. + Descrição de detalhes técnicos específicos do ambiente (quando aplicável). Por exemplo, incluir detalhes da infraestrutura utilizada na Amazon Cloud ou Google Cloud. Eventualmente, incluir até chaves de acesso para permitir a reprodução do experimento. Recomendação: Recomendado a atribuição do selo.	Não	
Não									
Sim	Software	Não		Não		Não		Não	
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável + documentação suficiente + título + resumo	Sim	+ lista de dependências + lista de versões das dependências/linguagens/ambiente + descrição do ambiente de execução + instruções de instalação e execução + um ou mais exemplos de execução	Sim	+ instrução para executar o(s) código(s); + descrição do processo de execução dos experimentos; + descrição de detalhes técnicos específicos do ambiente (quando aplicável)	Parcialmente	- documentação do(s) código(s) (descrevendo arquivos, funções, APIs etc); + legibilidade mínima no(s) código(s) e outros artefatos; + identificar as principais reivindicações do artigo no(s) artefato(s). ACESAR da documentação do código ser mínima. RECOMENDADO a atribuição do SeloS.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável + documentação suficiente + título + resumo - RECOMENDO a atribuição do SeloD.	Sim	+ lista de dependências + lista de versões das dependências/linguagens/ambiente + descrição do ambiente de execução + instruções de instalação e execução + um ou mais exemplos de execução - RECOMENDO a atribuição do SeloF.	Sim	Experimentos Reprodutíveis - SeloR + instrução para executar o(s) código(s); + descrição do processo de execução dos experimentos; - descrição de detalhes técnicos específicos do ambiente (quando aplicável). - Teste em 4 ambientes. 1 funcionou parcialmente. 2 deram erro com dependência com openssl. 1 com erro "obvious ownership in repository" mesmo após aplicar chmod. - NÃO RECOMENDO a atribuição do SeloR.	Parcialmente	Artefatos Sustentáveis - SeloS - documentação do(s) código(s) (descrevendo arquivos, funções, APIs etc); + legibilidade mínima no(s) código(s) e outros artefatos; + identificar as principais reivindicações do artigo no(s) artefato(s). A documentação do código existe, mas é insuficiente em alguns locais. RECOMENDO PARCIALMENTE a atribuição do SeloS.

Trabalho possui artefato?	Tipo do artefato:	Selo/D: Artefato	Selo/D: Artefato publicamente Disponível (comentários)	Selo/D: Artefato	Selo/D: Artefato Funcional (comentários)	Selo/D: Experi. Selo/D: Experimentos Reprodutíveis (comentários)	Selo/S: Artefato	Selo/S: Artefato Sustentável (comentários)	
Sim	Software	Sim	+ repositório bem organizado e bem documentado - instruções longas e detalhadas demais - deverá ter uma instrução simples e objetiva (e.g., um passo-a-passo de 5 linhas) de "instalação" e utilização utilizando Dockerfile ou imagens no dockerhub	Não	- faltaram instruções mais simples e diretas de utilização das imagens docker - deveria incluir a documentação com um exemplo bem simples de setup do ambiente com docker ou docker-compose e acesso aos recursos da aplicação, as instruções detalhadas, passo-a-passo de instalação manual, etc., deveriam ser apenas algo complementar e opcional, é importante pensar sempre no usuário final, a documentação não é para quem desenvolveu a solução e nem apenas para herdeiros ou geeks que adoram ficar digitando zifres de comandos e coisas manuais para simplesmente fazer um teste rápido de funcionamento e usabilidade de uma ferramenta ou sistema. - depois de executar o docker-compose up, o usuário visualiza as "instruções" a seguir: System check identified no issues (0 silenced). do May 25, 2024 - 18:25:01 do Django version 4.2.5, using settings 'doo.settings' do Starting development server at http://0.0.0.0:8000/ do Quit the server with CONTROL-C. - estas instruções não parecem ser muito úteis ou auto-confiadas. - instruções como esta a seguir não parecem ser muito práticas para o usuário: Pronto, basta acessar o DOO na url http://localhost:8000 e se logar utilizando o usuário e senha criados nos passos anteriores. - o usuário deverá voltar para trás e sair procurando o login e senha? - fiz um find por user e encontrei usuário e senha do BD_DB_USER=doe DOO_PASSWORD=doe123, tentei e não funciona. - procurei mais um pouco e encontrei o "SUPERUSER_NAME=admin", tentei e consegui acessar, mas deu mensagem de erro, acredito que falhou especificar também navegadores compatíveis/testados e outros eventuais recursos necessários. Forbidden (403) CSRF verification failed. Request aborted. More information is available with DEBUG=True. - ao logar no sistema utilizando outro navegador, fiquei surpreso pelo fato de não haver instruções mínimas quaisquer sobre o funcionamento e a utilização, há apenas 4 itens de menu, o doo, Solicitações, Catálogo e Repositório, não há conteúdo em qualquer dos itens, seria bem interessante e importante ter instruções básicas e também conteúdo (demo) de exemplo, fica muito difícil avaliar as funcionalidades e o funcionamento sem instruções simples e precisas. - a instrução a seguir é difícil de entender na pasta "DOO/br2024/Webservier" "Depois de fazer checkout do código vamos entrar na pasta DOO/ops/containers/Webservier" qual checkout? depois de toda o comando "sudo utilinst.sh -W" o diretório indicado não existe no repositório principal do projeto e nem no do container, é um diretório / caminho dentro de um docker instanciado? - diferentemente do indicado no manual, encontrei o arquivo "servico_web_container.py" na pasta "DOO/br2024/Webservier" - ao executar "sudo python3 servico_web_container.py" ocorrem várias excessões, aparentemente, uma delas requer o setup correto do xhost para diferentes ambientes e/ou execução do código logado como root na interface gráfica, novamente, faltam detalhes e informações mais precisas. [1]	Não	+ documentação pontual de alguns métodos e classes da app web + falta documentação global dos códigos, suas sequências, funcionalidades e finalidades - códigos com muitas definições estáticas não recomendadas - login e senha de sistemas e usuários fixos nos códigos de setup do ambiente - nomes de variáveis pouco intuitivos ou auto-confiados (e.g., -d, -k, -w, -q, -s1, -d2, -z1, -d2, Q, ...) - caminhos e repositórios definidos e fixos em código - valores fixos no código (e.g., "0.0.0.0", "3000") - arquivos .py protótipo (e.g. modelo) sem instruções adequadas - arquivos .py que implementam um único método curto e simples que é utilizado em uma única classe [2]		
Sim	Software	Sim	+ Artefato disponível em um repositório estável + Documentação vasta, com seção destinada aos revisores do WGRS24 e vídeos demonstrativos - Não apresenta título do artigo no repositório - Não apresenta resumo do artigo no repositório Recomendo a atribuição do selo.	Não	O artefato é composto de 3 cenários, cada um deles com uma rede virtual Docker. Todos os três cenários dependem de um acesso HTTP partindo de um container para outro que roda um serviço GitLab. Contudo, a conexão HTTP não é iniciada, mesmo recarregando a página web a cada 10 segundos conforme descrito na documentação. É possível pingar a máquina que roda o serviço GitLab, o que indica que não é problema de link. O primeiro pacote SYN da conexão HTTP é respondida com um RST, o que indica que o serviço GitLab pode não estar sendo executado. A saída do netstat com essa indicação: <pre>... tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:9236 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:8151 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:8150 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:8155 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:8154 0.0.0.0:* LISTEN tcp 0 0 127.0.0.1:8153 0.0.0.0:* LISTEN tcp 0 0 *:*:*:* LISTEN udp 0 0 128.10.0.241:41645 10.138.1.2:53 ESTABLISHED udp 0 0 0.0.0.0:14585 :144585 ESTABLISHED ...</pre> A saída do comando "gitlab-ct status" é a seguinte: run: ghlay: (pid 555) 1290s; run: log: (pid 576) 1288s run: gitlab-ct: (pid 779) 1287s; run: log: (pid 790) 1265s run: logrotate: (pid 516) 1302s; run: log: (pid 530) 1299s run: postgres: (pid 600) 1278s; run: log: (pid 612) 1275s run: redis: (pid 533) 1296s; run: log: (pid 543) 1292s run: sshd: (pid 52) 1317s; run: log: (pid 51) 1317s Por isso, não foi possível seguir com a execução dos testes.	Não	Artefato não funcional.	Parcialmente	Por mais que eu não tenha conseguido executar os cenários descritos no artigo devido a um comportamento não descrito na documentação, percebe-se que os autores foram cuidadosos ao descrever o passo a passo das operações.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	O projeto está disponível em: https://github.com/KiebrVienna/LTE-NS3 Apresentemente é um projeto de continuidade, pois há arquivos antigos (1 ano ou mais). No entanto, atendem às especificações para a revisão de artefatos disponíveis.	Sim	O projeto atende aos requisitos de funcionalidade, trata-se de uma implementação adicional ao NS3, um simulador de redes de código aberto.	Sim	O projeto como um todo permite reproduzir os achados dos autores. Os códigos e dados utilizados estão disponíveis.	Sim	A natureza do projeto já faz com que ele seja sustentável. É uma contribuição ao NS3.
Sim	Software	Sim	+ documentação básica boa - faltou detalhar os requisitos de hardware e software (incluindo todas as dependências e respectivas versões) é importante que isto esteja também na documentação. - faltou detalhar os ambientes (hardware + pilha de software, desde o SO) utilizados na execução dos experimentos	Não	- seria bastante útil ter um script de demo pronto, com todos os comandos para um teste funcional básico da solução - seria muito útil e produtivo disponibilizar um Dockerfile (para construção de uma imagem docker) e uma imagem pronta no Docker Hub - instalação manual e lenta de coisas complexas, como o PyTorch e pouco recomendado, nesses casos, melhor disponibilizar um Dockerfile e/ou uma imagem docker pronta, eventualmente, utilizar como base uma imagem docker do PyTorch. - comandos com problemas, por exemplo, o usuário muda de sistema para sistema, comandos como o a seguir não dão problema. sudo pip3 install --user sem pandas numpy scikit-learn sudo pip3 install --user \$USER pandas numpy scikit-learn - recomendações como "cd ~" para instalação de pacotes de software não são recomendadas nem para usuários leigos - ao invés de instruções que podem gerar dúvidas, é importante apresentar um passo-a-passo detalhado e preciso, como se fosse um script que pode ser executado automaticamente - seria bastante útil ter um script de setup (que inclui instalação de pytorch e outras coisas eventualmente necessárias) e um script de execução (ao menos uma demo simples) - não detalha a versão exata dos pacotes (dependências) a serem instaladas - execução do comando "mkdir configures" por falta de dependências, qualquer bom projeto de software deve obrigatoriamente relacionar todas as dependências e as devidas instruções de instalação. - comandos como "python3 generate-positioning-data.py" não geram informação alguma para o usuário, não indicam o que está sendo executado. - o comando "jstest-ram.sh" leva a erros. (bash: jstest-ram.sh: No such file or directory), o comando correto para ser "jstest-zram.sh", aparentemente, os autores não foram capazes de validar instruções mínimas do artefato. - a execução do comando "generate-training-data.py" leva a erros de dependências. - o comando "python3 generate-training-data.py" não gera informações úteis ao usuário. - o código não informa o que está sendo executado e nem estimativa de tempo de execução, o comando fica simplesmente parado, vai apresentar algo na tela? quanto tempo devo aguardar até sair algum print ou resultado? o comando inicia de 4 a 6 processos cc1plus, suporho que seja a compilação do NS-3 para simulação e geração do dataset. - depois de executa por algum tempo o comando "python3 generate-training-data.py", ocorrem erros de compilação. (pathwz34LTE-NS3/loran-ite-2-ite-ml-handover-simulation.cc:37:10: fatal error: ns3loran-module.h: No such file or directory 37 #include "ns3loran-module.h" ~~~~~ compilation terminated.) - ambiente onde os testes foram realizados: Ubuntu 22.04.4 LTS [3]	Não	- não contemplou o seto de funcionalidade	Não	- documentação superficial - não é explicitado quais modificações (e onde - arquivos) foram realizadas no NS-3 - os códigos python possuem vários parâmetros fixos e não documentados (e.g., número de coordenadas, raio do disco, acenter, ycenter, intervalo das coordenadas, caminhos(path), ...) - nomes de variáveis pouco auto-explicativos (e.g., h, p, pos, sub, f, con, emb, pos, x, d, crf, ...) - nomes de variáveis que podem levar a interpretações conflitantes (e.g., cotc, nCotc, crf, ...) - parâmetros de combinações fixos em códigos ao invés de arquivos de configuração externos ou parâmetros de execução. - falta de padronização mínima nas nomenclaturas de variáveis, funções e métodos no código (e.g., nCotc, nCotc, eu_data, emb, emb_pos, emb_load, ...) - diversos valores fixos no código que podem levar a diferentes problemas (e.g., train_data_loc:./3 / 1000); - valores fixos em código que deveriam ser parâmetros da execução, como os a seguir: self.layer_1 = nn.Linear(num_feature, 512) self.layer_2 = nn.Linear(512, 128) self.layer_3 = nn.Linear(128, 64) self.layer_out = nn.Linear(64, num_class) BATCH_SIZE = 100 LEARNING_RATE = 0.0001 NUM_FEATURES = len(X.columns) NUM_CLASSES = 3 batch_size=10 torch.device("cpu") nRuns = 32 [4]

Trabalho possui artefato?	Tipo do artefato:	SeioD: Artefato	SeioD: Artefato publicamente Disponível (comentários)	SeioF: Artefato	SeioF: Artefato Funcional (comentários)	SeioR: Experm SeioR: Experimentos Reprodutíveis (comentários)	SeioS: Artefato	SeioS: Artefato Sustentável (comentários)	
Sim	Software	Sim	+ disponíveis em um repositório estável (github) + documentação mínima Recomendo a atribuição do seio.	Sim	+ lista de dependências (possui, porém incompleta. Tive que instalar dependências adicionais conforme erro no terminal); + lista de versões das dependências/linguagens/ambiente; - descrição do ambiente de execução (incompleta); + instruções de instalação e execução (parcial, tive que resolver alguns problemas que não estavam descritos no README). + um exemplo de execução mínima (parcial, as simulações ítem levar mais de 24h de execução na minha máquina). Não recomendo a atribuição do seio. Eu consegui vencer alguns obstáculos mas posso contornar de Python e Linux, acho improvável que usuários mais leigos conseguissem replicar o experimento.	Parcialmente	+ instrução para executar as principais reivindicações (parcialmente, para replicar as reivindicações do artigo, eu precisaria de 24h de execução do experimento); + descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo (aparentemente sim, não consegui executar o último passo). Recomendo (weak accept) a atribuição do seio. Alé onde consegui executar (por restrição de tempo para obtenção dos resultados), me parece correto.	Não	+ modularizado + organizado (os arquivos a serem executados estão quase todos no root) - documentação mínima do código + legibilidade mínima de código (parcialmente) Não recomendo a atribuição do seio.
Não									
Não									
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável + documentação mínima + título + resumo Recomendo a atribuição do seio, atende todos os requisito do seio.	Sim	+ executado + funcionalidades + lista de dependências + lista de versões das dependências/linguagens/ambiente + instruções de instalação e execução - um exemplo de execução mínima Recomendo a atribuição do seio, atende quase todos os requisito do seio, além disso, contém um script que executa os experimentos.	Sim	+ modularizado + organizado + inteligível + fácil compreensão + existe uma documentação mínima do código (descrevendo arquivos, funções, ...) + legibilidade mínima de código; + permite que os avaliadores consigam identificar as principais reivindicações do artigo no artefato. Recomendo a atribuição do seio, atende todos os requisito do seio, bem documentado e bem comentado.	Sim	+ revisor consegue reproduzir as principais reivindicações apresentadas no artigo + instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas); + descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo; Recomendo a atribuição do seio, atende todos os requisito do seio.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável (github) + documentação suficiente - título - resumo O artefato relacionado ao artigo está disponível publicamente. O readme apresenta um título e um resumo diferente do título e do resumo do PDF. Além disso, existem dois códigos, um no branch principal e um em outro branch, que, segundo o readme, é mais atualizado que o principal. Considere o mais recente nas avaliações pois é o que parece conter os mesmos resultados reportados. Recomendo a atribuição do seio mesmo identificando essas discrepâncias.	Sim	+ código executa gerando vários gráficos + script para instalar as dependências + scripts separados para executar as etapas descritas no artigo Recomendo a atribuição do seioF.	Parcialmente	Executando um comando descrito no readme eu consegui reproduzir gráficos similares aos apresentados no artigo. Contudo, não consegui identificar os resultados apresentados na Tabela 1. Suspeito que os resultados estejam em um arquivo .JSON, mas não consigo afirmar. Recomendo a atribuição do seio PARCIALMENTE devido a não ter identificado um resultado relacionado ao erro do modelo.	Sim	+ arquivos diferentes que contém código com objetivos separados (mais ou menos modular) + organização dos arquivos + nomes de pastas e arquivos bons (similares ao apresentado no texto) + código indentado (facilita a leitura) - documentação poderia abranger os scripts, o readme possui uma descrição dos scripts, mas a documentação das funções poderia ser melhor Recomendo a atribuição do seioS
Sim	Software	Sim	+ Repositório estável + Documentação mínima + Título + Resumo Recomendação: Recomendo a atribuição do seio. Ao analisar as contribuições do trabalho, duas são identificadas, porém apenas uma enreja a criação de artefato. Dessa forma, o artefato apresentado não possui diretamente o título e resumo do artigo, mas sim um título e resumo do artefato em si. Como não houve orientação de que os artefatos incluídos nos artigos deveriam ser apresentados em um repositório exclusivo para o artigo, excepcionalmente pela falta de orientações, sugiro a atribuição do seio. Da mesma forma, o link aponta para uma apresentação de alto nível, necessitando acessar outro link de versão mais nova da ferramenta para ter acesso às informações de uso mais específicas.	Sim	+ lista de dependências; + descrição do ambiente de execução; + instruções de instalação e execução; - um exemplo de execução mínima. Recomendação: Recomendo a atribuição do seio. As dependências encontram-se em arquivo padronizado de dependências na linguagem python, facilitando o seu ajuste. Não há um exemplo de execução do script, mas os autores incluem um script para realizar a utilização da ferramenta. O link do repositório aponta para um repositório com uma apresentação de alto nível, necessitando acessar outro link de versão mais nova da ferramenta para ter acesso às informações de uso mais específicas (como dependências etc). Como não houve orientação de que os artefatos incluídos nos artigos deveriam ser apresentados em um repositório do artigo exclusivamente, excepcionalmente pela falta de orientações, sugiro a atribuição do seio.	Sim	+ instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas); + descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo; + Descrição de detalhes técnicos específicos do ambiente (quando aplicável). Por exemplo, incluir detalhes da infraestrutura utilizada no Amazon Cloud ou Google Cloud. Eventualmente, incluir até chaves de acesso para permitir a reprodução do experimento. Recomendação: Recomendo a atribuição do seio	Sim	+ Existe documentação mínima do código descrevendo arquivos, funcionalidades etc + Legibilidade mínima de código + Positivo que avaliadores consigam identificar a principais reivindicações do artigo no artefato Recomendação: Recomendo a atribuição do seio. O artefato avaliado (código) apresenta boa documentação (na minha opinião), tanto para criação do ambiente, quanto de funcionalidades. O autor foi mais a fundo e não apenas documentou funções, mas partes de funções, facilitando a avaliação. O código de maneira geral segue as boas práticas de linguagem. A única sugestão é que seja mantido um idioma padrão para a documentação. Grande parte da documentação está em português, porém algumas partes estão em inglês. Parabéns aos autores. [5]
Sim	Teoria (formulação matemática)	Sim	O código está disponível no github	Não	Não possui lista de dependências; Não possui lista de versões das dependências/linguagens/ambiente; Não possui descrição do ambiente de execução; Não traz instruções de instalação e execução.	Não	Embora o código esteja disponível, os autores não deixam claro como é possível reproduzir o KIS. Há menção ao KIS no README, mas apenas isso. O readme inclui diretório inexistente: kibgascheduler	Parcialmente	O código em si é legível e documentado, mas ao mesmo tempo não está claro como eu, como avaliador, poderia identificar as principais reivindicações dos autores.
Sim	Software	Sim	+ disponíveis em um repositório estável + documentação suficiente + título + resumo Recomendo a atribuição do seio.	Sim	+ instruções de execução + artefato pode ser executado + mais do que um exemplo de execução + o revisor consegue observar as funcionalidades - documentação das dependências Recomendo a atribuição do seio. A documentação poderia incluir as dependências, que apesar de serem poucas, requerem o Python e as suas bibliotecas matplotlib e numpy.	Parcialmente	- reproduzir exatamente os resultados + parece ser fácil de obter resultados semelhantes através de parametrização Devido à aleatoriedade na geração das matrizes e na evolução do algoritmo genético, é difícil de reproduzir exatamente os mesmos resultados do artigo. A minha sugestão neste sentido talvez seria incluir seed nos números aleatórios que permitissem gerar as mesmas matrizes e depois aplicar exatamente as mesmas mudanças, crossover, população inicial, etc. para obter exatamente o mesmo valor que está no artigo. Porém, isto parece ser uma extensão do artefato muito complicada e permitiria aos usuários obter uma execução idêntica a alguma documentação que explica os resultados encontrados (mesmo que isso talvez não se consiga) e fazer para os resultados apresentados no artigo)	Sim	+ boa documentação do código + código legível + boa parametrização do código Recomendo o seio pois é fácil de gerar outras execuções apenas modificando os parâmetros de configuração que estão documentados no código
Sim [6]	Software [7]	Sim [8]	+ descreve resumidamente o que é o KibgaScheduler + apresenta casos de teste + apresenta os arquivos existente no repositório + possui instruções de como utilizar + apresenta exemplos de resultados e parâmetros - não informa dados básicos sobre o paper - não descreve dependências - não informa os requisitos de software (e.g., versão do Python) - não informa ambiente de desenvolvimento e/ou testes (e.g., SO) - README md demasiadamente extenso - README md deveria ter as instruções mais básicas de requisitos, execução e testes, as demais informações poderiam ser arquivos .md complementares, linkados no README md - não informa os parâmetros de execução utilizados para gerar os resultados contidos no paper - não possui um script demo para uma execução rápida e testes básicos de funcionalidade - não há uma relação clara e direta entre o que é apresentado na seção 4 e os códigos que estão disponíveis no repositório [9]	Não [10]	+ há alguns arquivos python no repositório + códigos python executáveis (e.g., kibgascheduler_a.py) e geram resultados de melhor execução global - não há informações sobre os pods com Nginx - não há informações sobre o ambiente Kubernetes - não há scripts e nem parâmetros de setup do ambiente - há arquivos citados no paper que não existem no repositório (e.g., media demanda T 10.py) [11]	Não [12]	- faltam instruções - faltam parâmetros exatos das execuções das 3 versões do GA - faltam instruções sobre o ambiente de execução - faltam passo-a-passo técnicos e precisos - há códigos citados no paper que não existem no repositório [13]	Não [14]	+ código legivelmente inteligível e bem escrito + boa nomenclatura geral de variáveis e métodos - muitos comentários e coisas desnecessárias no código - diversos valores fixos em código (deveriam ser arquivos de configuração ou parâmetros de entrada) - algumas nomenclaturas duvidas e poucos intuitivas (e.g., pa1, pa2, filho1, filho2, ...) - arquivos python aparentemente funcionais, mas projeto como um todo não funcional (faltam instruções, códigos, dependências, arquivos de setup e execução automática) [15]
Sim	Software	Sim	+ disponível em um repositório estável + documentação mínima + título - resumo Recomendo a atribuição do SeioD. O artigo disponibiliza um link para um repositório do Github contendo a implementação da solução proposta, um escritório para Kubernetes baseado em algoritmo genético. O repositório contém ainda instruções detalhadas de uso dos códigos (implementados em Python).	Sim	+ lista de dependências + lista de versões das dependências/linguagens/ambiente + descrição do ambiente de execução + instruções de instalação e execução + um exemplo de execução mínima	Parcialmente	+ instruções para reproduzir resultados + descrição de detalhes técnicos específicos do ambiente (quando aplicável) Recomendo a atribuição do SeioR. Os artefatos disponibilizados permitem a reprodução de algumas das principais reivindicações apresentadas no artigo. Em especial, é possível reproduzir os resultados das tabelas 2 e 3 do artigo, as quais apresentam as aptões obtidas com a solução de GA proposta (KibGAScheduler).	Sim	+ documentação mínima do código (descrevendo arquivos, funções, etc.) + legibilidade mínima do código Recomendo a atribuição do SeioS. Os arquivos estão bem descritos e o código está bem comentado.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável + documentação mínima - título - resumo Recomendo a atribuição do seio, apesar de não conter o título e resumo, os dados estão disponíveis.	Não	- executado - funcionalidades - lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um exemplo de execução mínima Não recomendo a atribuição do seio, falta instruções de como executar o artefato e de como utilizar as funcionalidades.	Parcialmente	+ modularizado + organizado - inteligível - fácil compreensão - existe uma documentação mínima do código (descrevendo arquivos, funções, ...) + legibilidade mínima de código; + permite que os avaliadores consigam identificar as principais reivindicações do artigo no artefato. Não recomendo a atribuição do seio, pouca ou nenhuma documentação, poucos comentários, difícil entendimento sobre o que cada código faz.	Parcialmente	+ revisor consegue reproduzir as principais reivindicações apresentadas no artigo + instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas); - descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo; Recomendo a atribuição do seio, o experimento é reproduzível.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável (github) - título - resumo O artefato relacionado ao artigo está disponível publicamente. O readme apresenta um título e um resumo diferente do título e do resumo do PDF. Além disso, a relação de autores apresentados no readme também é diferente da relação de autores do PDF. Recomendo a atribuição do seioD mesmo identificando essa discrepância.	Sim	+ código executa gerando dois gráficos + script para instalar as dependências + scripts separados para executar todas as etapas da análise Recomendo a atribuição do seioF.	Parcialmente	Recomendo parcialmente a atribuição do seioR pois não encontrei o código para gerar a Figura 5. Predição realizada para os pontos PA-BA, que são os 3 resultados apresentados no artigo. Provavelmente os autores conseguiram facilmente adicionar o código faltante para que o seioR seja atribuído integralmente.	Sim	+ arquivos diferentes que contém código com objetivos únicos (código modular) + organização dos arquivos + nomes de pastas e arquivos bons + documentação poderia abranger mais scripts, porém está presente em alguns dos principais scripts e no readme possui uma descrição de cada script. + código indentado (facilita a leitura) + em alguns casos, os autores usaram o jupyter notebook, assim, separando o código em células facilita a leitura Recomendo a atribuição do seioS
Sim	Software e Dados (e.g., modelo e dataset)	Sim	Recomendo a atribuição do seio.	Sim	Recomendo a atribuição do seio.	Sim	Recomendo a atribuição do seio.	Parcialmente	Embora que eu tenha encontrado os códigos e conseguido reproduzir os experimentos, senti falta de uma clara especificação de qual código se refere a qual resultado. Isso poderia ser resolvido com frases simples no readme.

Trabalho possui artefato?	Tipo do artefato:	SeioD: Artefato	SeioD: Artefato publicamente Disponível (comentários)	SeioF: Artefato	SeioF: Artefato Funcional (comentários)	SeioR: Experm SeioR: Experimentos Reprodutíveis (comentários)	SeioS: Artefato	SeioS: Artefato Sustentável (comentários)
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ informações gerais de um projeto - informações específicas relacionadas ao paper - falta identificar claramente no repositório os códigos e dados específicos do paper - falta identificar claramente todos os requisitos (e.g., versão do python) - falta identificar informações precisas sobre o ambiente de desenvolvimento e/ou testes (e.g., SO e versão, relação com bibliotecas e dependências para além dos pacotes python) - falta um passo a passo simples para execução dos mesmos experimentos do paper, idealmente uma sequência simples e direta de comando ou um script - faltam detalhes dos parâmetros da execução que levaram aos resultados do paper	Não	+ possui notebook jupyter executáveis - não possui instruções básicas sobre os testes funcionais para reprodução dos experimentos do paper - README não descreve os diretórios e alguns arquivos, mas não o que e como deverá ser executado - a instalação de alguns dos requisitos python leva a erros no Ubuntu 22.04 LTS, como os a seguir ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior is the source of the following dependency conflicts. virtulien 20.24.2 requires platformdirs<4.0,>=3.9.1, but you have platformdirs 4.2.0 which is incompatible. prety 1.1.12 requires keyring<22.0.0,>=21.2.0; python_version >= "3.6" and python_version < "4.0", but you have keyring 23.0.0 which is incompatible. prety 1.1.12 requires packaging<21.0,>=20.4, but you have packaging 24.0 which is incompatible. - execução do forecasting-model.ipynb não gera mensagens de saída e nem resultados jupyter rconvert -to python -execute forecasting-model.ipynb python3 forecasting-model.py - o arquivo do notebook (forecasting-model.ipynb) é absurdamente grande, contendo 13MB, o que é bem incomum - inteligibilidade do notebook é inaceitável	Não	- execução do forecasting-model.ipynb não gera mensagens de saída e nem resultados jupyter rconvert -to python -execute forecasting-model.ipynb python3 forecasting-model.py - não é possível reproduzir os resultados do paper em qualquer ambiente - autores não forneceram todas as especificações técnicas do ambiente e instruções de execução	+ nomenclatura e legibilidade geral razoáveis + valores fáceis em código (e.g., base, url, target, bandwidth) - variáveis pouco intuitivas e auto-explicativas (e.g., cont, teste, f, k, v, ...) - o arquivo do notebook (forecasting-model.ipynb) é absurdamente grande, contendo 13MB, o que é bem incomum - aparentemente, um dos notebooks contém código binário embarcado - inteligibilidade (inacessível de alguns notebooks) - datasets desacompanhados de informações básicas - há diversas questões de documentação e instruções básicas do repositório, como apontado em outros seios
Sim	Software e Dados (e.g., modelo e dataset)	Sim		Não	Não recomendado atribuir o seio já que o artefato requerer a edição do jupyter notebook (falta de documentação)	Não	Não	Não
Sim	Software e Dados (e.g., modelo e dataset)	Não	+ disponíveis em um repositório estável - documentação suficiente - título - resumo	Não	- lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um ou mais exemplos de execução	Não	- lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um ou mais exemplos de execução	- documentação do(s) código(s) (descrevendo arquivos, funções, APIs etc); + legibilidade mínima no(s) código(s) e outros artefatos; - identificar as principais reivindicações do artigo no(s) artefato(s).
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponíveis em um repositório estável - documentação mínima - título - resumo Recomendo a atribuição do seio, porém, não foi colocado nenhuma informação além dos dados e do código.	Não	+ executado - funcionalidades - lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um exemplo de execução mínima NÃO recomendo a atribuição do seio, pois a execução do script que gera os resultados não funciona.	Não	+ modularizado + organizado + intuitivo + fácil compreensão - existe uma documentação mínima do código (descrevendo arquivos, funções, ...) + legibilidade mínima de código. - permite que os avaliadores consigam identificar as principais reivindicações do artigo no artefato NÃO recomendo a atribuição do seio, existe um código que não funciona, sem documentação, sem comentários.	- revisor consegue reproduzir as principais reivindicações apresentadas no artigo - instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas); - descrição de um processo de como foram executados os experimentos para chegar ao resultado do artigo; NÃO recomendo a atribuição do seio, não tem como reproduzir os resultados.
Sim	Software e Dados (e.g., modelo e dataset)	Sim	- resumo não apresenta informações básicas sobre o trabalho (e.g., título e resumo do paper, nome dos autores) - repositório não apresenta informações técnicas mínimas e necessárias, como versão do Python, relação de dependências e suas respectivas versões - repositório não apresenta detalhes do ambiente de desenvolvimento e/ou testes - nome do repositório parece ser pouco apropriado - nome dos arquivos poderia ser melhor e mais auto-explicativo - não há instruções sobre a sequência de passos e/ou execuções para reproduzir os resultados do paper	Não	- erro na execução após instalar as dependências - exemplo: AttributeError: 'numpy.ndarray' object has no attribute 'append'	Não	- faltam informações básicas sobre os experimentos - faltam códigos para geração dos gráficos - falta identificar os valores e parâmetros precisos da execução que levaram aos resultados do paper	- para ser sustentável é necessário resolver primeiro a maioria das deficiências apontadas nos demais seios
Sim	Software e Dados (e.g., modelo e dataset)	Não	Não recomendo atribuir o seio considerando a qualidade na descrição do artefato. O conteúdo encontrado no github não possui uma descrição mínima que possa auxiliar em um processo posterior de utilização do artefato.	Não		Não	Não	Não
Sim	Software e Dados (e.g., modelo e dataset)	Sim	+ disponível em um repositório estável + documentação mínima - título - resumo Recomendo a atribuição do SeioD. O artigo disponibiliza um link para um repositório do GitHub contendo os dados utilizados (arquivo CSV), um script Python e um arquivo README simples.	Não	A descrição do repositório dos artefatos (contida no arquivo README) retrata brevemente o conteúdo do arquivo de dados (em CSV) e indica o script Python que gera os resultados. No entanto, no repositório não há uma lista de dependências, lista de versões das bibliotecas utilizadas, instruções de execução, nem um exemplo de execução. Ocorre que, ao executar o script em um ambiente Python limpo, várias libs (ou módulos) não são encontradas. Ao instalá-las manualmente (mesmo sem conhecimento da versão esperada) ocorre o seguinte erro referente à linha 14 do script: "AttributeError: 'numpy.ndarray' object has no attribute 'append'". A princípio, este tipo de erro parece indicar falha na implementação. Contudo, pode ainda ser causado pelo uso de uma versão incorporeada de alguma lib específica (o que não é possível saber através da documentação disponibilizada).	Não	- instruções para reproduzir resultados - descrição de detalhes técnicos específicos do ambiente (quando aplicável) NÃO recomendo a atribuição do SeioR, pois (além da ausência dos itens acima) não foi possível executar com sucesso o script que gera os resultados. Como mencionado nos comentários sobre o SeioF, após instalar manualmente algumas libs que são requeridas, ocorre o seguinte erro: "AttributeError: 'numpy.ndarray' object has no attribute 'append'". Acredito que com a correção do problema (via modificação no código, indicação das dependências apropriadas, etc.), os experimentos passados a ser reprodutíveis. Isso porque, aparentemente, os dados necessários já estão disponibilizados.	- documentação mínima do código (descrevendo arquivos, funções, etc.) + legibilidade mínima do código Recomendo a atribuição do SeioS, mas com ressalvas. No repositório dos artefatos há uma breve descrição sobre os arquivos contidos. Além disso, o código é relativamente curto e de fácil entendimento (por ser modularizado em funções).
Sim	Software e Dados (e.g., modelo e dataset)	Não	+ disponíveis em um repositório estável + documentação suficiente - título - resumo A documentação no código é boa. No entanto, precisa melhorar no README. NÃO RECOMENDO a atribuição do SeioD.	Não	- lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um ou mais exemplos de execução NÃO RECOMENDO a atribuição do SeioF.	Não	- instrução para executar o(s) código(s); - descrição do processo de execução dos experimentos; - descrição de detalhes técnicos específicos do ambiente (não aplicável) NÃO RECOMENDO a atribuição do SeioR.	+ documentação do(s) código(s) (descrevendo arquivos, funções, APIs etc); + legibilidade mínima no(s) código(s) e outros artefatos; + identificar as principais reivindicações do artigo no(s) artefato(s). RECOMENDO a atribuição do SeioS.
Sim	Software	Sim	+ Disponíveis em um repositório estável. + Título + Abstract. - Documentação bastante escassa. Recomendo a atribuição do seio.	Sim	+ artefato em Python Notebook funcional Recomendo a atribuição do seio.	Não	O primeiro resultado gerado pelo artefato é um gráfico que depende somente do dataset (gerado por terceiros) que foi utilizado. Este gráfico mostra a quantidade de pacotes rotulados como "normais" e pacotes rotulados como "abaixo". Contudo, quando executado, o artefato gera um gráfico que não apresenta pacotes "normais", diferentemente do gráfico apresentado no repositório. Esse resultado, aparentemente, errôneo, pode ter influenciado nos demais testes. Além disso, as etapas seguintes dos testes dependem de atribuição de uma variável em código ("end" + # 5% of dataset), mas a única documentação é um comentário que não deixa claro o que essa variável deve conter. Dessa forma, os demais gráficos gerados foram bastante diferentes dos apresentados no repositório. Por isso, não posso recomendar a atribuição do seio.	- documentação do artefato é bastante excessiva. Além disso, durante a execução do código, os scripts apresentavam erros, aparentemente erros esperados, mas o artefato não documenta nada com relação a isso. Esses pontos geram uma certa desconformidade com relação ao procedimento de execução, de forma que eu não soube discernir, em diversos momentos, se o artefato estava configurado corretamente ou não. Por isso, não posso recomendar a atribuição do seio.
Sim	Software	Sim	+ disponível em um repositório estável - documentação mínima - título - resumo + abstract Recomendo a atribuição do seio SeioD. O artefato encontra-se disponível em um repositório estável, com título e abstract. No entanto, para melhorar a usabilidade e acessibilidade do artefato, seria interessante incluir uma documentação detalhada explicando o objetivo do artefato e orientações sobre como utilizá-lo.	Sim	- lista de dependências - lista de versões das dependências/linguagens/ambiente - descrição do ambiente de execução - instruções de instalação e execução - um exemplo de execução mínima Recomendo a atribuição do seio SeioF. Apesar da ausência das especificações mencionadas, a disponibilidade no formato Jupyter Notebook permitiu a utilização do artefato. Ainda assim, para uma melhor análise de funcionalidade, seria útil adicionar ao repositório informações detalhadas sobre pré-requisitos e dependências, bem como uma base de dados (ou um subconjunto da base) utilizável para exemplo de execução da seleção de características.	Sim	+ instrução para executar as principais reivindicações (e.g., resultados dos principais gráficos/tabelas) - descrição de um processo de como foram executados os experimentos para chegar até o resultado do artigo Recomendo a atribuição do seio SeioR. Embora não haja uma instrução clara de execução, o formato do notebook torna a execução dos experimentos bastante intuitiva. Seria benéfico acrescentar uma descrição detalhada nesse processo para melhorá-la e sua reprodutibilidade.	- existe uma documentação mínima do código (descrevendo arquivos, funções, ...) + legibilidade mínima de código + permite que os avaliadores consigam identificar as principais reivindicações do artigo no artefato + artefato modularizado Recomendo a atribuição do seio SeioS. Embora a documentação do código esteja ausente, a sua legibilidade e modularização permitiu identificar as principais reivindicações do artigo no artefato. Mesmo que artefato contenha alguns comentários ao longo do código, a inclusão de uma documentação descrevendo as funções do código e destacando sua principal contribuição ajudaria a aumentar a sustentabilidade do artefato.
Sim	Dados (e.g., dataset)	Sim	Os dados utilizados estão disponíveis. Apesar de não haver descrição, dá para entender do que se trata o dado em relação ao que é apresentado no artigo, relacionando os nomes dos arquivos. + disponíveis em um repositório estável + documentação suficiente - título - resumo Recomendo a atribuição do seio. [16]	Não		Não	Não	Não

- [1] Responder updated this value.
- [2] Responder updated this value.
- [3] Responder updated this value.
- [4] Responder updated this value.
- [5] Responder updated this value.
- [6] Responder updated this value.
- [7] Responder updated this value.
- [8] Responder updated this value.
- [9] Responder updated this value.
- [10] Responder updated this value.
- [11] Responder updated this value.
- [12] Responder updated this value.
- [13] Responder updated this value.
- [14] Responder updated this value.
- [15] Responder updated this value.
- [16] Responder updated this value.