Modelagem e Detecção de Ataques *Grayhole* ao Protocolo GOOSE usando o *Framework* ERENO

Jerusa C. Gonçalves[1], Silvio E. Quincozes[1,2], Vagner E. Quincozes[3] e Juliano F. Kazienko[4]

[1]Universidade Federal de Uberlândia (UFU)

[2] Universidade Federal do Pampa (UNIPAMPA)

[3]Universidade Federal Fluminense (UFF)

[4]Universidade Federal de Santa Maria (UFSM)

jesa.cg@ufu.br, silvioquincozes@unipampa.edu.br,

vequincozes@id.uff.br, kazienko@redes.ufsm.br

Abstract. *The growing need to enhance cybersecurity in critical infrastructure, specifically in electric substations that communicate via the Generic Object Oriented Substation Event (GOOSE) protocol, calls for effective threat detection and prevention techniques. This protocol, defined by the IEC-61850 standard, protects physical devices by notifying events such as electrical faults. However, its adoption opens gaps for the exploitation of vulnerabilities through attacks whose signatures need to be mapped. In particular, the literature lacks Grayhole attack signatures. This work proposes the modeling and implementation of such an attack targeted to the GOOSE protocol. Furthermore, such modeling is incorporated into ERENO, a framework for generating intrusion datasets. The effectiveness of the resulting dataset is validated through five machine learning algorithms, with the J48 algorithm standing out, achieving a 90.68% F1-Score.*

Resumo. *A crescente necessidade de reforçar a segurança cibernética na infraestrutura crítica, especificamente em subestações elétricas que se comunicam através do protocolo Generic Object Oriented Substation Event (GOOSE), requer técnicas efetivas de detecção e prevenção de ameaças. Esse protocolo é definido pelo padrão IEC-61850 e protege dispositivos físicos notificando eventos como faltas elétricas. Entretanto, a sua adoção abre brechas para a exploração de vulnerabilidades através de ataques cujas assinaturas precisam ser mapeadas. Destaca-se uma lacuna na literatura referente à falta de assinaturas do ataque Grayhole. Neste artigo, é proposta a modelagem e implementação de tal ataque ao protocolo GOOSE. Ademais, tal modelagem é incorporada ao ERENO, um framework para geração de datasets de intrusões. A eficácia do dataset resultante é validada através de cinco algoritmos de aprendizado de máquina, com destaque para o algoritmo J48 que obteve 90,68% de F1-Score.*

1. Introdução

Com a gradativa conexão de sistemas de controle industriais à Internet, cresce a preocupação com ataques cibernéticos. Um cenário de redes industriais importante consiste nas subestações elétricas digitais, as quais frequentemente operam com Dispositivos

Eletrônicos Inteligentes, do inglês *Intelligent Electronic Devices* (IEDs) a fim de possibilitar o monitoramento e controle de eventos que acontecem nesses ambientes. Particularmente, dentre os diversos protocolos existentes, o protocolo *Generic Object Oriented Substation Event* (GOOSE), definido pelo padrão IEC-61850, emergiu como um dos mais proeminentes para a comunicação entre IEDs [Quincozes 2022][Wang et al. 2022].

Ao passo que a norma IEC-61850 propõe protocolos que facilitam a comunicação entre IEDs, a adoção dos mesmos possibilita a exploração de novas vulnerabilidades nas subestações elétricas digitais. Segundo [Rajkumar et al. 2020], o padrão IEC-61850 apresenta diversas vulnerabilidades de segurança. Muitos desses problemas se devem aos requisitos rígidos de tempo das aplicações onde tal padrão é comumente adotado. O protocolo GOOSE, por exemplo, não implementa nenhum mecanismo de criptografia devido aos requisitos de tempo real impostos pelo sistema de proteção para comunicar o disparo de sinais. Além disso, as falhas e vulnerabilidades de segurança cibernética estão crescendo cada vez mais. Relatórios recentes [McLennan et al. 2022] reafirmam as pesquisas feitas no setor de que o número de ataques cibernéticos vem aumentando a cada ano. Consequentemente os ataques cibernéticos às subestações podem causar diversos cenários desastrosos como apagões ou dano à equipamentos, por exemplo. Assim, é fundamental fortalecer a segurança cibernética da subestação para aumentar a resiliência da rede [Hong and Liu 2019].

Em contraste com os sistemas de informações tradicionais, onde as propriedades de *Confidencialidade* e *Integridade* são consideradas prioritárias, nos sistemas industriais de infraestrutura crítica a *Disponibilidade* é fundamental [Hahn et al. 2016]. Nesse contexto, uma das principais ameaças consistem nos ataques de negação de serviço, do inglês *Denial of Service* (DoS). Dentre os ataques desta categoria, destaca-se o ataque de descarte seletivo de mensagens, conhecido como *Grayhole* [Quincozes et al. 2023]. Nesse contexto, o uso de Sistemas de Detecção de Intrusões, do inglês *Intrusion Detection Systems* (IDSs) se torna fundamental para a proteção das redes onde são transmitidas mensagens baseadas nesse padrão. No entanto, a obtenção de dados realistas e representativos para o treinamento de IDSs é um desafio. Até onde o conhecimento se estende, não há qualquer trabalho na literatura sobre o assunto que realize a modelagem, implementação e avaliação do ataque *Grayhole* no âmbito do protocolo GOOSE.

Este trabalho tem como objetivo modelar e implementar o ataque *Grayhole* direcionado ao protocolo GOOSE. Como prova de conceito, o ataque é incorporado ao *framework Efficacious Reproducer Engine for Network Operations* (ERENO) [Quincozes 2022], que possibilita a modelagem e simulação de ataques cibernéticos em redes de subestações elétricas. Além disso, é realizada uma avaliação da eficácia do *dataset* produzido por meio dessa modelagem, utilizando os algoritmos de aprendizado de máquina *J48*, Naive Bayes, *Random Forest*, *Random Tree* e *K-Nearest Neighbors*, com destaque para o algoritmo J48 que obteve 90,68% de F1-Score.

O restante do trabalho está organizado como segue. A Seção 2 fornece a fundamentação teórica pertinente ao protocolo GOOSE e ao framework ERENO. Na Seção 3, são discutidos trabalhos relacionados ao tema abordado. A modelagem do ataque *Grayhole* e experimentos são detalhados nas Seções 4 e 5, respectivamente. Na Seção 6, os resultados são apresentados. Por último, a Seção 8 apresenta as considerações finais.

2. Fundamentação Teórica

Nesta seção, apresenta-se o protocolo GOOSE, responsável pela troca de mensagens entre IEDs, e o *framework* ERENO, que permite a modelagem e simulação de ataques cibernéticos em redes de subestações elétricas.

2.1. Protocolo GOOSE

O protocolo GOOSE, definido pela norma IEC-61850 [Commission 2003], permite a troca de mensagens entre os IEDs para relatar eventos nas subestações, como mudanças de status, alarmes e comandos de controle. Diversos componentes enviam esses eventos por meio de mensagens GOOSE, que englobam alarmes de temperatura, estado do disjuntor, intertravamento da chave seccionadora, entre outros. Esses dados são adicionados em um campo denominado *datSet* e posteriormente transmitidos, utilizando o conceito de publicação/assinatura, para um grupo de IEDs assinantes. Cada IED pode se inscrever em um tópico específico relacionado ao seu domínio, como controle, proteção ou medição, por exemplo. Na Figura 1 é apresentado o esquema de retransmissão utilizado para a entrega confiável de mensagens GOOSE, bem como a estrutura de um quadro *ethernet* que encapsula essas mensagens.

Dentro do campo *datSet* podem conter valores *boolean* que indicam, por exemplo, se um disjuntor deve ser aberto para isolar uma linha de transmissão durante uma falta elétrica ou fechado para reestabelecer a transmissão de energia, conforme ilustrado pelos valores “Fechar” e “Abrir” nas mensagens enviadas na Figura 1.

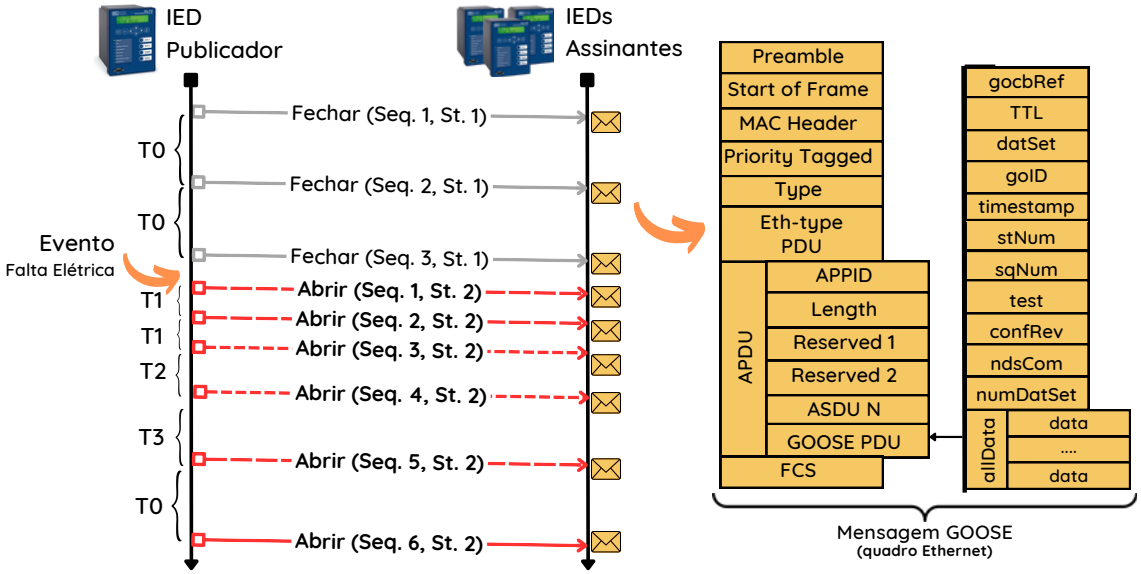


Figura 1. Esquema de retransmissão de mensagens GOOSE e detalhamento do quadro Ethernet que encapsula tais mensagens.

Quando as situações estão estáveis, não ocorrem novos eventos, portanto, não são detectadas alterações nos valores do conjunto de dados GOOSE. As mensagens GOOSE são enviadas em intervalos regulares de tempo, representados por T_0, e o número de sequência, *SqNum*, é incrementado. No início de um evento, o campo *SqNum* é definido como zero, o número de status, *StNum*, é incrementado e uma nova mensagem é enviada imediatamente. A mensagem é retransmitida em intervalos crescentes, começando

com o menor intervalo de retransmissão, (T_1), que é utilizado como separador para as três primeiras mensagens e repetido duas vezes, e a cada retransmissão o intervalo é sucessivamente aumentado (T_2, T_3, etc.), até alcançar o intervalo estável original, (T_0). Esse processo é ilustrado na Figura 1, onde os parâmetros *StNum* e *SqNum* foram abreviados para `St.` e `Seq.`, respectivamente.

A Descrição de Configuração de Subestação, em inglês *Substation Configuration Description* (SCD), é um formato de arquivo padronizado utilizado no campo da engenharia elétrica, especificamente no domínio de sistemas de energia e subestações. Um arquivo SCD fornece uma descrição abrangente e estruturada da configuração, layout e ajustes de vários componentes dentro de uma subestação. Portanto, em cenários reais de subestações, o intervalo T_1 representa o parâmetro *minTime* definido no arquivo SCD, enquanto o intervalo T_0 é definido pelo parâmetro *maxTime* [Commission 2003], nesse mesmo arquivo. Embora os valores exatos dos incrementos não sejam definidos pela norma, os IEDs tipicamente implementam cálculos tais como o da Progressão Aritmética (PA) ou Progressão Geométrica (PG) para defini-los [Hoyos et al. 2012].

Dadas essas características típicas, um IDS pode analisar várias propriedades para distinguir entre atividades legítimas e maliciosas. O registro de tempo do GOOSE pode revelar mensagens atrasadas que podem ser indicativas de um ataque DoS, pois em circunstâncias normais, o intervalo de tempo entre duas mensagens recebidas não deve exceder T_0. O *SqNum* e o *StNum* são elementos importantes, pois podem indicar a possibilidade de injeção de mensagens falsas ou ataques de repetição de mensagens [Hoyos et al. 2012, Hong et al. 2014, Ustun et al. 2019, Quincozes 2022]. Da mesma forma, os valores do *datSet* podem ser correlacionados com o *StNum* para identificar alterações indesejadas.

2.2. ERENO Framework

O ERENO é um *framework* de código aberto para gerar conjuntos de dados IEC-61850 com *features* (atributos) representativos para detectar diferentes tipos de intrusões. Ele é capaz de gerar atributos representativos, tais como os campos de mensagens GOOSE ilustrados na Seção 2.1, bem como outras informações relevantes para a detecção de intrusões através de algoritmos de aprendizado de máquina. Por ser uma ferramenta extensível, o ERENO possibilita a modelagem de novos ataques. Portanto, tal ferramenta é ideal para a implementação do ataque *Grayhole*, proposto neste trabalho. O ERENO gera um conjunto de 28 *features* para o protocolo GOOSE, que são divididas nas seguintes categorias:

- *Features de tempo*: incluem informações sobre o intervalo de tempo entre as mensagens GOOSE, a duração da mensagem GOOSE e a hora do dia em que a mensagem foi enviada;
- *Features de conteúdo*: incluem informações sobre o tipo de mensagem GOOSE, o número de sequência da mensagem, o tamanho da mensagem e o valor dos campos da mensagem;
- *Features de tráfego*: incluem informações sobre o número de mensagens GOOSE enviadas, o número de mensagens GOOSE recebidas, o número de mensagens GOOSE enviadas por segundo e o número de mensagens GOOSE recebidas por segundo.

2.3. Ataque Grayhole

O ataque *Grayhole* consiste em uma categoria específica de ataques, sendo classificado como um tipo de ataque DoS [Quincozes et al. 2023]. No *Grayhole*, o invasor obtém controle sobre um ou mais dispositivos na rede e, em seguida, descarta seletivamente os pacotes que trafegam por eles, em vez de encaminhá-los adequadamente. O objetivo do atacante é causar o máximo de danos possível, eliminando a maior quantidade de pacotes sem ser detectado, conforme mencionado em [Pal et al. 2018].

Grayhole modelado no *framework* ERENO é um tipo de ataque de negação de serviço (DoS), no qual o atacante compromete a rede de comunicação ao se fazer passar por um nó publicante legítimo. A modelagem apresentada nesta seção independe da estratégia de invasão adotada pelo atacante. No entanto, a fim de exemplificação, assume-se que ataques Grayhole podem ocorrer em uma subestação por meio do comprometimento do IED publicador para descarte seletivo de mensagens GOOSE antes da transmissão, comprometimento de um dispositivo intermediário, como um *switch* de rede, para descarte de mensagens GOOSE específicas, ou comprometimento do(s) IED(s) assinante(s) para descarte seletivo de mensagens GOOSE antes do processamento pelo receptor.

Dessa forma, o nó publicador atacante descarta deliberadamente uma parte dos pacotes de dados, sem enviá-los aos dispositivos assinantes. Como resultado, os dispositivos que assinaram o publicador não receberão todos os pacotes esperados [Quincozes et al. 2023]. Outra possibilidade consiste no comprometimento de um nó assinante, o impedindo de processar todas as mensagens recebidas.

No estudo de [Attia et al. 2015], o ataque *Grayhole* consiste na eliminação de mensagens enviadas ou recebidas entre dispositivos (ou seja, excluí-las ou não enviá-las). Esse tipo de ataque é identificado pela alteração no número de pacotes enviados. Consequentemente, a distribuição normal do número de pacotes enviados não é mais observada, diferenciando-se de situações normais.

3. Trabalhos Relacionados

Atualmente, na literatura existem ataques cibernéticos já modelados através de ferramentas de geração de *datasets*, como, por exemplo, o *framework* ERENO [Quincozes 2022]. Nesta seção serão discutidos os ataques ao protocolo GOOSE que já foram modelados e catalogados através de *datasets* publicamente disponíveis.

Em [Hoyos et al. 2012], os autores demonstram a modelagem e execução de um ataque prático (*i.e., Message Injection*) no protocolo GOOSE, demonstrando suas consequências devastadoras na infraestrutura cibernética. Em resumo, o valor dos dados *booleanos* são alterados de "*False*" para "*True*" para causar o disparo do relé em uma subestação de energia. Ao executar tal ataque, o atacante cria e envia mensagens maliciosas, tais como a injeção de comandos pela rede. Nesse modelo, os invasores podem fazer alterações aleatórias (ou seja, ignorar sua conformidade com a norma IEC-61850) ou alterações com reconhecimento de padrões (ou seja, cumprir com a norma IEC-61850).

O trabalho de [Hong et al. 2014] propõe o estudo de um IDS baseado em alguns ataques, incluindo ataques de retransmissão de mensagens, do inglês *Replay Attacks*. Esse tipo de ataque consiste na captura e retransmissão de mensagens após algum período de tempo escolhido pelo atacante. Uma das principais características desse ataque consiste

em não modificar o conteúdo original da mensagem. No entanto, esse ataque tem como principal limitação a sua facilidade de detecção, visto que o *SqNum* será idêntico ao de uma mensagem previamente transmitida na rede.

No estudo de [Abdul et al. 2014], diversos ataques são explorados no contexto de subestações digitais. No contexto do protocolo GOOSE, os autores discutem os *Replay Attacks* e a possibilidade de *malwares* automatizados executarem modificações nas mensagens antes de retransmiti-las na rede, constituindo assim o *Modification Attack*. Todavia, a simples modificação de uma mensagem evita apenas a repetição de mensagens previamente enviadas, não sendo essa uma abordagem suficientemente eficaz para enganar os mecanismos de descartes dos IEDs baseados nos valores dos campos *StNum* e *SqNum* das mensagens GOOSE.

Em [Kush et al. 2014], os autores estudam abordagens de ataques de negação de serviço do tipo *poisoning*, explando vulnerabilidades no mecanismo de descarte de mensagens antigas dos IEDs, que se baseiam nos valores de *StNum* e *SqNum*. Uma das variantes estudadas consiste no *High-Status Number Attack*, onde as mensagens GOOSE são interceptadas e retransmitidas de modo a adulterar o valor do *StNum*, sendo este maior que o *StNum* atual. Com isso, mensagens GOOSE legítimas são descartadas no receptor devido ao seu *StNum* aparentemente desatualizado. Outra variante é o *High-Rate Flooding Attack*, na qual o invasor inunda o canal multicast enviando várias mensagens GOOSE falsas em um curto intervalo (entre duas mensagens legítimas). Desse modo, cada mensagem falsa incrementa o valor de *StNum* em uma unidade e a próxima mensagem legítima é descartada pelo IED que a recebe. Em ambas as variações, as mensagens descartadas podem gerar alertas se um IDS estiver monitorando o descarte de mensagens. Em geral, ataques de envenenamento visam impedir o processamento das mensagens legítimas mas são detectáveis por mecanismos existentes na literatura [Bohara et al. 2020].

Em [Ustun et al. 2019] é proposto o ataque *Masquerade*. Esse ataque consiste em uma especialização dos ataques de injeção que aumenta significativamente a dificuldade de detecção por parte de um IDS, pois o mesmo exige que os invasores aprendam com o conteúdo das mensagens GOOSE anteriores e imitem o comportamento legítimo ao enviar novas mensagens falsas. Isso envolve a modificação inteligente (não aleatória) de campos de mensagem como *StNum* e *SqNum* e a injeção de eventos de mudança de estado maliciosos falsos para causar a mesma mudança de comportamento de uma mensagem legítima. Uma característica importante do ataque *Masquerade* consiste na reprodução do mecanismo de retransmissão usado pelos IEDs legítimos.

Por fim, destaca-se que, embora a maioria dos trabalhos relacionados apresentados não sejam recentes, a literatura atual está calcada majoritariamente nas abordagens clássicas de ataques reportados nesta seção, como é o caso do mecanismo EDA4GNeT [Elbez et al. 2022], cujo objetivo consiste em detectar ataques de *poisoning* em subestações IEC-61850. Esse e todos os ataques supracitados já foram implementados através do *framework* ERENO e avaliados em [Quincozes 2022]. No entanto, uma lacuna que foi identificada na literatura consiste na falta da implementação do ataque Grayhole considerando as particularidades e desafios específicos do protocolo GOOSE. Dessa forma, até onde sabemos, a presente proposta é a primeira a modelar o ataque Grayhole no contexto de subestações elétricas digitais.

4. Modelagem do Ataque Grayhole

Nesta seção serão abordadas as etapas de modelagem do ataque *Grayhole* no *framework* ERENO. Na Subseção 4.1, a implementação de tal ataque é apresentada. Os ataques *Grayhole* são uma forma de ataque direcionado às redes e computadores que se caracteriza pelo descarte seletivo de pacotes de dados. Neste trabalho, propõe-se a modelagem desses ataques visando o descarte seletivo de mensagens transmitidas através do protocolo GOOSE, no contexto da automação de subestações de energia. A Figura 2 ilustra a modelagem do ataque *Grayhole* em uma comunicação entre IEDs através do paradigma de publicação e assinatura.

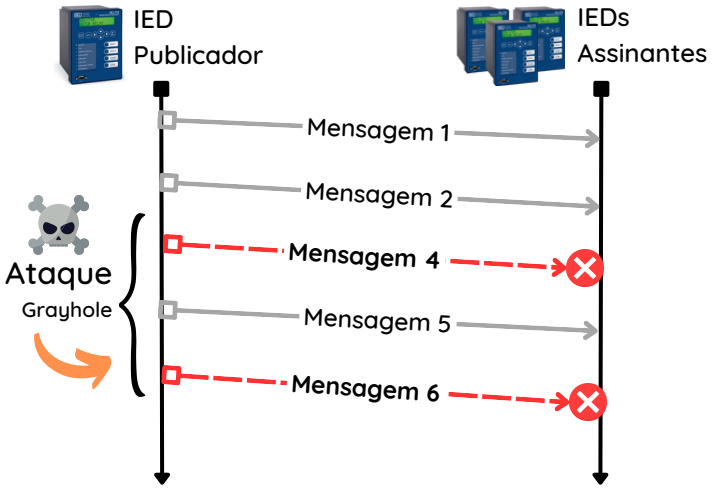


Figura 2. Modelagem do ataque Grayhole.

A modelagem de ataques *Grayhole* em um cenário de subestação GOOSE foi concebida considerando um descarte seletivo randômico de 20% das mensagens enviadas por um IED. Então, o atacante, neste cenário hipotético modelado, não tem conhecimento prévio sobre a relevância ou o conteúdo das mensagens que está descartando. Como resultado, as mensagens descartadas podem variar de simples notificações de status até comandos críticos de controle e operação. Essa porcentagem foi escolhida assumindo-se que os efeitos de um ataque *Grayhole* no protocolo GOOSE podem ser significativos com um descarte de 20% das mensagens, podendo-se gerar falhas na comunicação entre os IEDs, ocasionando possíveis interrupções na transmissão de energia, bem como outros potenciais problemas na subestação. Isso pode resultar em perda de controle operacional, aumento do tempo de inatividade e potencialmente uma paralisação total das operações da subestação. Destaca-se que, na revisão da literatura encontrou-se modelagens deste ataque para fins de geração de *datasets* para a detecção de intrusões apenas em outros contextos, tais como em redes de sensores sem fio, conforme publicado por [Almomani et al. 2016]. Portanto, este trabalho é pioneiro na modelagem e implementação do ataque Grayhole ao protocolo GOOSE.

4.1. Implementação do Ataque Gryahole

A implementação do ataque *Grayhole* proposto neste trabalho é ilustrada no pseudocódigo do Algoritmo 1. A função `GrayholeAtaque()` é a função principal responsável por realizar o ataque Grayhole, a qual chama as outras funções para obter as

mensagens (`obterMensagens()`), selecionar as mensagens a serem descartadas de maneira aleatória seguindo uma distribuição uniforme com uma taxa de descarte de 20% (`selecionarMensagensDescartadas()`), e enviar as mensagens descartadas que não foram sorteadas para o descarte para o destino original (`enviarMensagem()`).

Algoritmo 1 Lógica do *Grayhole* implementada no ERENO *framework*.

```
1:  function GRAYHOLEATAQUE
2:      mensagens ← obterMensagens()
3:      mensagensDescartadas ← selecionarMensagensDescartadas(mensagens)
4:      enviarMensagensDescartadas(mensagensDescartadas)
5:  function OBTERMENSAGENS
6:      mensagens ← []
7:      while houverMensagens() do
8:          mensagem ← receberMensagem()
9:          mensagens ← mensagens ∪ mensagem
10:     return mensagens
11: function SELECIONARMENSAGENSDESCARTADAS(mensagens)
12:     mensagensDescartadas ← []
13:     for mensagem in mensagens do
14:         if aleatorio() < 0.2 then                ▷ 20% de chance de descarte
15:             mensagensDescartadas ← mensagensDescartadas ∪ mensagem
16:     return mensagensDescartadas
17: function ENVIARMENSAGENSDESCARTADAS(mensagensDescartadas)
18:     for mensagem not in mensagensDescartadas do
19:         destino ← selecionarDestino()
20:         enviarMensagem(mensagem, destino)
```

Esse algoritmo, implementado neste trabalho, foi integrado ao *framework* ERENO [Quincozes 2022]. Conforme mencionado na Subseção 2.2, o ERENO é um *framework* extensível de código aberto que pode ser usado para gerar um conjunto de dados (*dataset*) que reproduz o comportamento de protocolos como o GOOSE, baseado na norma IEC-61850. Tal *framework* contém *features* (*i.e.,* variáveis) que foram extraídas dos protocolos de comunicação GOOSE. A Figura 3 ilustra a metodologia adotada neste trabalho ao implementar um novo modelo de ataque, estendendo o ERENO. Os componentes do lado direito da Figura, na cor marrom, representam a etapa de avaliação dos algoritmos de detecção de intrusões (na Subseção 5.1), e a extração de resultados de métricas de desempenho (abordado na Seção 6). Já na parte esquerda da Figura, são ilustrados os processos de modelagem do ataque *Grayhole* e de extração de *features*. Essas *features* são usadas para treinar algoritmos de aprendizado de máquina, por exemplo, tendo como propósito a detecção dos diferentes tipos de ataques modelados no ERENO. Como prova de conceito, o *framework* ERENO [Quincozes 2022] foi configurado para a geração de um *dataset* com 1.000 amostragens que representam correlações entre mensagens transmitidas usando os protocolos GOOSE e SV, incluindo amostras de comportamento normal e também ataques contra os IEDs. Cada amostra possui 32 *features* extraídas das mensagens GOOSE. A metodologia para a detecção de ataques *Grayhole* ao protocolo GOOSE é apresentada na Seção 5.1.

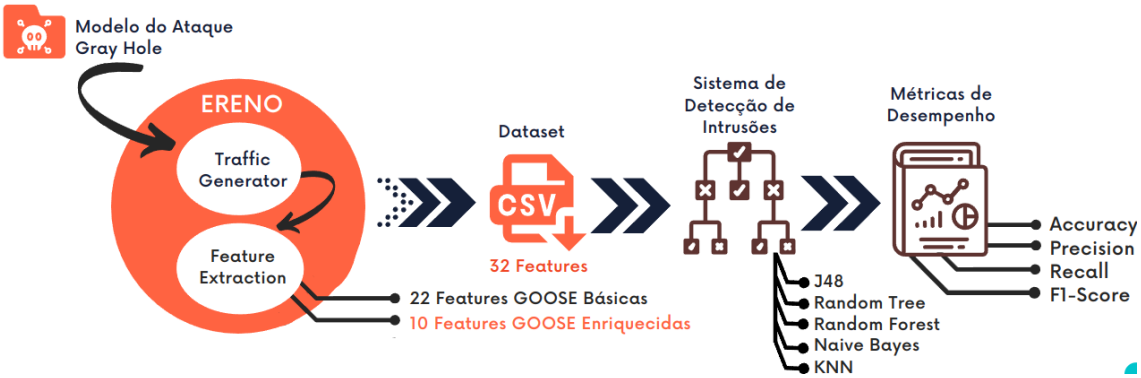


Figura 3. Figura do processo de modelagem no *framework* ERENO adaptada de [Quincozes 2022], com adição do ataque *Grayhole*.

5. Experimentos

Nesta seção são apresentados os Materiais e Métodos (Subseção 5.1) que serão utilizados no desenvolvimento dos ataques propostos e um cenário de ataque (Subseção 5.2). Os resultados obtidos a partir dos experimentos realizados são apresentados na Seção 6.

5.1. Materiais e Métodos

A detecção do ataque *Grayhole* desempenha um papel crucial na segurança de redes de subestações digitais baseadas no protocolo GOOSE. Neste contexto, a identificação desse tipo de ataque pode ser realizada através de diferentes estratégia de detecção. No estudo de [Attia et al. 2015] esse tipo de ataque é identificado pela alteração no número de pacotes enviados. Consequentemente, a distribuição normal do número de pacotes enviados não é mais observada, diferenciando-se de situações normais. No entanto, para a detecção eficiente desses ataques, é necessário o uso de algoritmos capazes de processar e analisar grandes volumes de dados e aprender dinamicamente os padrões do atacante.

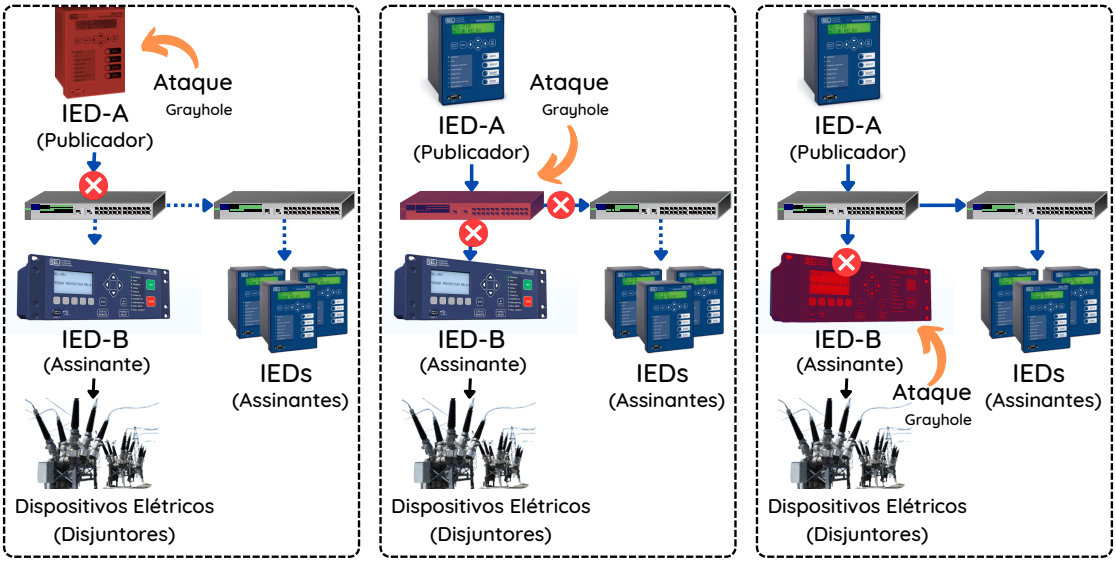
A abordagem experimentada para a detecção de ataques *Grayhole* consiste na utilização de algoritmos de mineração de dados. O WEKA [Witten and Frank 2002], uma coleção de algoritmos de mineração de dados de código aberto, desenvolvido em Java e licenciado sob a *General Public License* (GNU), é uma ferramenta amplamente utilizada para esse fim. Portanto, adotou-se tal biblioteca para avaliar a capacidade de algoritmos classificadores detectar o ataque *Grayhole*, verificando-se assim a qualidade e representatividade do *dataset* gerado através da modelagem proposta.

Especificamente, serão utilizados algoritmos de árvore de decisão, como o *Random Forest*, *Random Tree* e *J48*, bem como outros algoritmos tradicionais como o *K-Nearest Neighbors* (*KNN*) e *Naive Bayes*. Dessa forma, a utilização dos algoritmos apontados pelo WEKA desempenhará um papel fundamental na avaliação da modelagem proposta. Os resultados dos experimentos são apresentados na Seção 6.

5.2. Cenário de Ataque Grayhole

Com base na definição fornecida em [Pal et al. 2018], o ataque *Grayhole* possuí um modelo em que o invasor obtém controle sobre um ou mais dispositivos na rede e, posteriormente, descarta seletivamente os pacotes que os atravessam, em vez de encaminhá-los, visando causar o máximo de danos, descartando a maior quantidade possível de pacotes

sem ser detectado. Dessa forma, observa-se que os cenários de ataques *Grayhole* ao protocolo GOOSE podem seguir três abordagens diferentes, conforme ilustrado na Figura 4 e discutido a seguir.



(a) Origem infectada. **(b) *Switch* infectado.** **(c) Destino infectado.**

Figura 4. Modelagem do ataque Grayohole dividida em subfiguras.

- Comprometimento do IED publicador, de modo a permitir o descarte seletivo das mensagens GOOSE antes mesmo do seu envio (descarte na fonte), conforme ilustrado na Figura 4a;
- Comprometimento de um dispositivo intermediário, tal como um *switch* de rede que, por sua vez, passa a descartar determinadas mensagens GOOSE, conforme ilustrado na Figura 4b;
- Comprometimento do(s) IED(s) assinante(s), de modo a permitir o descarte seletivo das mensagens GOOSE antes de serem processadas pelo receptor (descarte no destino), conforme ilustrado na Figura 4c.

A partir da definição apresentada para o ataque *Grayhole*, foi modelada uma variação para as redes de subestações digitais baseadas no padrão IEC–61850 [Commission 2003]. Em contraste às redes de sensores sem fio, onde o *Grayhole* pode ser executado por nós sensores [Quincozes et al. 2023], em subestações digitais a o atacante deve seguir uma das alternativas apresentadas anteriormente. Essa metodologia pode explorar, por exemplo, uma vulnerabilidade na comunicação *multicast* do protocolo GOOSE que permite a exclusão da vítima (*i.e.,* dispositivo assinante) do inventário de assinantes no grupo multicast relacionado ao evento ao qual pretende-se evitar deliberadamente a chegada da mensagem. Ou então, em subestações que usam Redes Definidas por Software, do inglês *Software Defined Network* (SDN), regras de descarte de fluxos ou mensagens podem ser configuradas no *switch*. Assim, não será possível o recebimento de mensagens transmitidas pelos dispositivos publicadores que forem vítimas do ataque. Dessa forma, o atacante assume o controle das mensagens do publicador e descarta deliberadamente as mensagens GOOSE, sem enviá-las aos dispositivos assinantes. Como

resultado, nenhum dispositivo que tenha assinado o publicador receberá os pacotes de dados esperados [Quincozes et al. 2023].

6. Resultados

O gráfico da Figura 5 apresenta os resultados de diferentes algoritmos de classificação, incluindo: *J48*, *Naive Bayes*, *Random Forest*, *Random Tree* e KNN. A seguir, serão discutidos os principais achados e destacados pontos relevantes referentes à escolha do melhor algoritmo, a partir da análise dos resultados obtidos, para a detecção de ataques do tipo *Grayhole*.

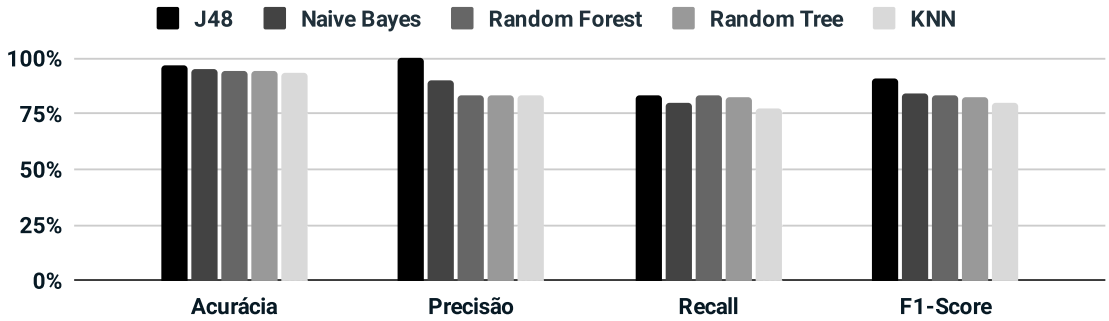


Figura 5. Detecção do ataque *Grayhole* com uma taxa de descarte de 20%.

A acurácia mede a proporção de amostras classificadas corretamente em relação ao total de amostras. Observou-se que o algoritmo *J48* obteve a maior acurácia (97,11%), seguido por *Naive Bayes* (94,80%), *Random Forest* (94,23%), *Random Tree* (94,06%) e *KNN* (93,40%). Portanto, em termos de acurácia geral, *J48* se destacou.

A precisão mede a proporção de instâncias classificadas corretamente como positivas (*Grayhole*) em relação ao total de instâncias classificadas como positivas. Nesse aspecto, o algoritmo *J48* obteve 100% de precisão, o que significa que todas as instâncias classificadas como positivas foram corretas (*i.e.,* sem apresentar falsos positivos). *KNN* (83,27%), *Random Forest* (82,87%) e *Random Tree* (82,92%) também apresentaram níveis razoáveis de precisão, enquanto *Naive Bayes* (89,81%) foi ligeiramente superior que os anteriores, ficando atrás apenas do J48. Tais resultados demonstram que mesmo os algoritmos mais simples, como o *Naive Bayes*, são capazes de processar corretamente as *features* geradas a partir da modelagem do ataque *Grayhole* apresentada neste trabalho.

O *recall* (ou recuperação) mede a proporção de instâncias positivas corretamente classificadas em relação ao total de instâncias positivas presentes. *Random Forest* obteve um *recall* de 83,55%, seguido por *J48* (82,96%), *Random Tree* (82,72%), *Naive Bayes* (80,20%) e *KNN* (77,47%). Isso indica que *Random Forest* tem um melhor desempenho em identificar amostras de ataque corretamente em relação aos demais classificadores.

O *F1-Score* é a média harmônica entre a precisão e o *recall*, fornecendo uma medida geral do desempenho do modelo. O algoritmo J48 obteve o maior *F1-Score* (90,68%), seguido por *Naive Bayes* (84,35%), *Random Forest* (83,17%), *Random Tree* (82,76%), e *KNN* (80,17%). Isso indica que J48 possui um equilíbrio entre precisão e *recall*, resultando em um desempenho geral melhor.

6.1. Tempo de Processamento

Além das métricas relacionadas ao desempenho na detecção que foram apresentadas, o tempo de execução é um fator relevante a ser considerado ao avaliar o desempenho dos algoritmos. Nos experimentos deste trabalho, observou-se uma variação significativa nos tempos de execução entre os diferentes algoritmos. A Figura 6 ilustra os tempos obtidos para o processamento das amostras produzidas pelos cinco algoritmos classificadores experimentados. Tais resultados foram obtidos a partir de um processador AMD Ryzen 7, com 16GB de memória RAM, por meio da função *nanotime()* do Java 17.0.5.

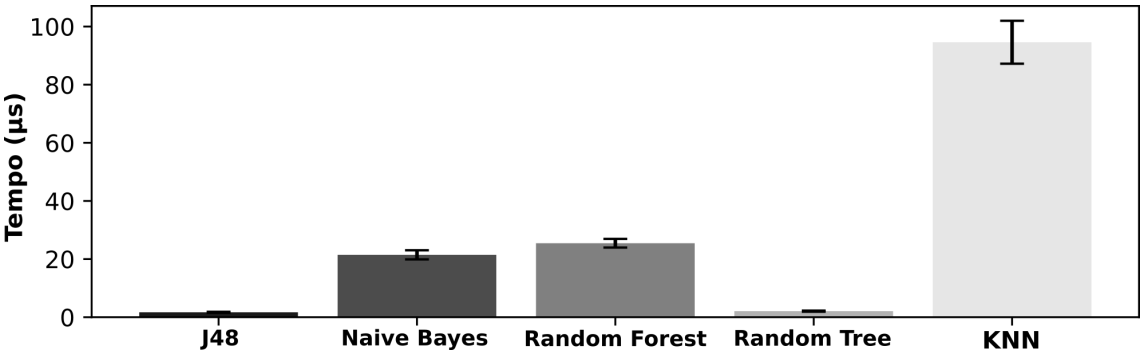


Figura 6. Tempo de processamento (Intervalo de Confiança de 95%)

O algoritmo *J48* se destaca como o mais rápido, requerendo, em média, apenas 1,68 microssegundos para concluir o processo de classificação de cada amostra. Em contraste, o algoritmo *KNN* apresenta o tempo mais longo, demandando, em média, 94,60 microssegundos para cada classificação. Essa diferença considerável no tempo de execução pode ter implicações práticas em situações onde a rapidez é uma exigência, como em sistemas de tempo real ou aplicações sensíveis à latência. Tais resultados estão de acordo com o que se espera, visto que o KNN é um algoritmo *Lazy* (preguiçoso). Isso significa que ele não mantém um modelo simplificado (como a árvore de decisão que o J48 monta) e, portanto, precisa computar os K vizinhos mais próximos sempre que há uma nova instância a ser classificada [Witten and Frank 2002].

6.2. Discussão

O *dataset* gerado através da modelagem proposta mostrou-se útil para todos os classificadores experimentados, especialmente para o algoritmo classificador *J48*. Os resultados demonstraram que o *J48* obteve um desempenho geral destacado, com alta acurácia, precisão, *recall* e *F1-Score*. Além disso, o *J48* apresentou um maior número de verdadeiros positivos e um menor número de falsos negativos em comparação com os outros algoritmos. Por fim, o J48 foi o algoritmo que apresentou resultados mais satisfatórios no tempo de processamento. Os resultados obtidos evidenciam a qualidade do *dataset* gerado através da modelagem proposta do ataque *Grayhole* no *framework* ERENO, que forneceu informações precisas e representativas para a detecção eficiente desse ataque por meio de diferentes classificadores.

7. Código-fonte e Dataset

O código-fonte que modela o ataque proposto neste trabalho está publicamente disponível no GitHub[1]. O conjunto de dados gerados como prova de conceito, que foram usados nos

[1]Código-fonte: `https://github.com/sequincozes/ereno/tree/uc08_grayhole`

experimentos realizados neste trabalho, estão disponíveis na plataforma Kaggle[2].

8. Conclusão

O protocolo GOOSE, utilizado em sistemas de energia elétrica, é um protocolo que possui brechas de segurança que permitem a exploração de vulnerabilidades que não possuem assinaturas catalogadas. Portanto, neste trabalho, desenvolveu-se uma nova modelagem de ataque que ainda não foi catalogada para preencher essa lacuna. O ataque modelado consiste no *Grayhole*, que é categorizado como uma variação de ataque DoS. Tal ataque modelado foi implementado e simulado usando um *framework* extensível e de código aberto capaz de gerar conjuntos de dados IEC-61850, denominado ERENO. Além disso, foi avaliado através de diversos algoritmos de aprendizado de máquina disponíveis na biblioteca WEKA, a saber: *J48*, *Naive Bayes*, *Random Forest*, *Random Tree* e KNN.

Os resultados indicam que o algoritmo J48 obteve maior acurácia (97,11%) e precisão (100%), enquanto que o algoritmo *Random Forest* apresentou maior *recall* (83,55%). Ademais, J48 se destacou obtendo 90,68% na métrica F1-Score, resultando em um desempenho geral melhor. Por fim, J48 foi o algoritmo mais rápido em questões de tempo de processamento, requerendo, em média, apenas 1,68 microssegundos para concluir o processo de classificação de cada amostra.

Como trabalhos futuros, pretende-se explorar outros ataques que ainda não foram catalogados, de forma a produzir assinaturas para o treinamento de IDSs, tais como os ataques *blackhole*, *stealthy false data injection* e *reflection/amplification DoS*. Ademais, futuros trabalhos poderão explorar mais profundamente a influência da variação do percentual de descarte, a fim de entender como essa alteração afeta os resultados obtidos, ampliando assim o entendimento da metodologia aplicada. Por fim, a análise da inteligência ou estratégia do atacante, aplicando a teoria dos jogos, permitirá investigar as táticas entre ataque e defesa em cenários complexos, incluindo situações de "*concept-drift*", onde acontecem mudanças súbitas nos padrões de ataque.

Referências

Abdul, R. M. T., Salman, Y., Yunus, Y., and Roslan, I. (2014). A review of security attacks on IEC61850 substation automation system network. In *6th International Conference on Information Technology and Multimedia*, pages 5–10. IEEE.

Almomani, I., Al-Kasasbeh, B., and Al-Akhras, M. (2016). Wsn-ds: A dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors*, 2016.

Attia, M., Sedjelmaci, H., Senouci, S. M., and Aglzim, E.-H. (2015). A new intrusion detection approach against lethal attacks in the smart grid:temporal and spatial based detections. In *2015 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–3, Guadalajara, Mexico.

Bohara, A., Ros-Giralt, J., Elbez, G., Valdes, A., Nahrstedt, K., and Sanders, W. H. (2020). Ed4gap: Efficient detection for goose-based poisoning attacks on iec 61850 substations. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE.

[2]*Dataset* (prova de conceito): `https://www.kaggle.com/datasets/sequincozes/gray-hole-attacks-in-power-substations-iec61850`

Commission, I. E. (2003). *Communication networks and systems in substations - ALL PARTS*. IET.

Elbez, G., Nahrstedt, K., and Hagenmeyer, V. (2022). Early Detection of GOOSE Denial of Service (DoS) Attacks in IEC 61850 Substations. In *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 367–373.

Hahn, A., Sun, C.-C., and Liu, C.-C. (2016). *Cybersecurity of SCADA within Substations*.

Hong, J., Liu, C., and Govindarasu, M. (2014). Detection of Cyber Intrusions Using Network-Based Multicast Messages for Substation Automation. In *Innovative Smart Grid Technologies (ISGT)*, pages 1–5. IEEE.

Hong, J. and Liu, C.-C. (2019). Intelligent electronic devices with collaborative intrusion detection systems. *IEEE Transactions on Smart Grid*, 10(1):271–281.

Hoyos, J., Dehus, M., and Brown, T. X. (2012). Exploiting the goose protocol: A practical attack on cyber-infrastructure. In *2012 IEEE Globecom Workshops*, pages 1508–1513.

Kush, N., Branagan, M., Foo, E., and Ahmed, E. (2014). Poisoned goose: exploiting the goose protocol. In *Proceedings of the Twelfth Australasian Information Security Conference (AISC 2014)*, pages 17–22. Australian Computer Society, Inc.

McLennan, M., Group, S., and Group, Z. I. (2022). The global risks report 2022 17th edition. Disponível em: `https://www.weforum.org/reports/global-risks-report-2022/`.

Pal, S., Sikdar, B., and Chow, J. H. (2018). An online mechanism for detection of gray-hole attacks on pmu data. *IEEE Transactions on Smart Grid*, 9(4):2498–2507.

Quincozes, S. (2022). *ERENO: An Extensible Tool for Generating Realistic IEC–61850 Intrusion Detection Datasets*. PhD thesis, Fluminense Federal University.

Quincozes, S. E., Kazienko, J. F., and Quincozes, V. E. (2023). An extended evaluation on machine learning techniques for Denial-of-Service detection in Wireless Sensor Networks. *Internet of Things*, 22:100684.

Rajkumar, V. S., Tealane, M., Ştefanov, A., and Palensky, P. (2020). Cyber attacks on protective relays in digital substations and impact analysis. In *2020 8th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, pages 1–6. IEEE.

Ustun, T. S., Farooq, S. M., and Hussain, S. S. (2019). A Novel Approach for Mitigation of Replay and Masquerade Attacks in Smartgrids Using IEC 61850 Standard. *IEEE Access*, 7:156044–156053.

Wang, X., Fidge, C., Nourbakhsh, G., Foo, E., Jadidi, Z., and Li, C. (2022). Anomaly detection for insider attacks from untrusted intelligent electronic devices in substation automation systems. *IEEE Access*, 10:6629–6649.

Witten, I. H. and Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *ACM Sigmod Record*, 31(1):76–77.