UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Henrique Corrêa De Oliveira

# Detecção Interpretável de Intrusões em Subestações Elétricas Inteligentes por Meio da Inteligência Artificial Explicável

**Uberlândia, Brasil**

**2024**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Henrique Corrêa De Oliveira

# Detecção Interpretável de Intrusões em Subestações Elétricas Inteligentes por Meio da Inteligência Artificial Explicável

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rodrigo Sanches Miani

Coorientador: Prof. Dr. Silvio Ereno Quincozes

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

# Agradecimentos

Meus sinceros agradecimentos vão para a minha família e amigos, especialmente para os que estiveram comigo durante os desafios desta jornada acadêmica. Não posso deixar de agradecer a Liliane e Lúcia, que estiveram ao meu lado durante todo esse processo de elaboração deste trabalho, mesmo nos momentos mais desafiadores, obrigado pelo apoio incondicional e amor inabalável. Agradeço a todos que tiveram palavras de motivação comigo e me fizeram acreditar no meu potencial, serei sempre grato por todos que de certa forma me encorajaram. Não menos importante, agradeço imensamente o incentivo de professores da Faculdade de Computação (FACOM) que foram empenhados em ensinar e me despertaram um grande amor pela tecnologia, principalmente aos meus Orientadores Prof. Dr. Silvio Quincozes e Prof. Dr. Rodrigo Miani. A todos que me apoiaram de alguma forma neste trabalho, meus sinceros agradecimentos, sua colaboração e incentivo me ajudaram a concluir este trabalho.

# Resumo

Este trabalho propõe uma solução para aprimorar um Sistema de Detecção de Intrusões (IDS) no contexto das subestações elétricas. O principal objetivo é integrar capacidades de explicabilidade utilizando *Explainable Artificial Intelligence* (XAI) no IDS, especialmente para lidar com os desafios de segurança na comunicação das redes das subestações elétricas, que são baseadas na norma IEC-61850. Além disso, como segunda contribuição, foi implementado extração de novas *features* com técnicas de enriquecimento temporal, adicionado a um pré-processamento robusto. A melhora obtida com isso na classificação também foi possível ser observada pelas técnicas XAI. Os resultados foram bastante promissores, o aprimoramento do IDS tornou ele menos tendencioso para alguns ataques e com uma melhor interpretação de ataques complexos, como o *Masquerade*. Essa abordagem não apenas fortaleceu a detecção de ameaças, mas também tornou o IDS mais confiável, fácil de interpretar para os operadores e simplificou a análise de correções ou implementações de novas funcionalidades, resultando em tomadas de decisões com o XAI muito mais favorável. Essa integração de XAI no IDS representou um avanço para a segurança cibernética em subestações elétricas, fornecendo uma defesa mais sólida e transparente contra ameaças emergentes em ambientes altamente críticos.

**Palavras-chave**: XAI, IDS, SHAP, Substações Elétricas, *Feature Engineering*, *Masquerade Atack*, IEC-61850.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| XAI | Explainable Artificial Intelligence |
| IDS | Intrusion Detection System |
| SHAP | SHapley Additive exPlanations |
| SAS | Sistema de Automação de Subestação |
| SCADA | Supervisory Control and Data Acquisition |
| ERENO | Efficacious Reproducer Engine for Network Operations |
| AM | Aprendizado de Máquina |
| IA | Inteligência Artificial |
| LIME | Local Interpretable Model-agnostic Explanations |
| IED | Dispositivos Eletrônicos Inteligentes |
| GOOSE | Generic Object Oriented System Event |
| SV | Sampled Values |
| APDU | Application Protocol Data Unit |
| ASDU | Application Service Data Unit |
| DoS | Denial of Service |
| IEC | International Electrotechnical Commission's |

# Sumário

# 1 Introdução

Na época atual, a tecnologia vem se tornando algo mais presente e conectado em todos âmbitos de nossas vidas e de diferentes maneiras. Uma dessas tecnologias é a Inteligência Artificial (IA), que de acordo com Davenport (2018) pode ser capaz de se integrar a vários sistemas de informações existentes para aumentar as competências analíticas humanas. Isso quer dizer que, alguns processos que acompanham a sociedade no dia a dia, podem ser integrados com IA para aumentar a precisão dessas tarefas, seja para análise de tráfego, recursos, mitigar erros de diagnósticos na saúde e sobretudo até mesmo ser de grande ajuda na distribuição segura da energia.

Nesse contexto, uma parte importante do setor elétrico, as subestações elétricas, que distribuem energia para várias regiões e são vitais para o funcionamento de outros setores críticos, vem buscando se aprimorar com o uso da IA. Uma usabilidade crescente dela no setor, está na segurança cibernética, tal como em Youssef et al. (2016) foi apontado as principais preocupações relacionadas às possibilidades de evoluções de ameaças e ataques cibernéticos aos sistemas de redes desse meio. Com isso, nos últimos anos houveram inúmeras modelagens e implementações de Sistemas de Detecção de Intrusão (IDS) baseados em diferentes técnicas de Aprendizado de Máquina (AM) que de acordo com Quincozes et al. (2021) já conseguem lidar bem com alguns dos ataques.

Porém, mesmo com IDSs baseados em IA já existentes para subestações, surge uma preocupação significativa em relação a transparência deles. De acordo com Molnar (2022), alguns modelos são como uma "caixa preta", recebem recursos como entrada e geram um resultado, sem muita explicação do motivo da escolha ou o que mais afetou nesse resultado. No setor elétrico, qualquer erro é fatal, isso destaca o quanto é uma área crítica, onde é preciso dar uma margem de confiança muito alta para tais soluções que procuram abordagens com IA. Para esse fim, a investigação e aplicação de técnicas de *Explainable Artificial Intelligence* (XAI) para tornar o modelo explicável, com um resultado interpretável é crucial.

Portanto, além da aplicação de técnicas de IA na detecção de intrusões nesses domínios, se faz necessário o entendimento das decisões tomadas. Tendo em vista que os IDSs baseados em inteligência artificial frequentemente operam como "caixas pretas", a adição de técnicas de extração de Explicabilidade usando XAI surge como um requisito fundamental. Isso permite não apenas explicar e justificar as ações tomadas por IDS baseados em AM, mas também ajuda na tomada de decisões em implementações de novas defesas contra os ataques, já que a interpretação do modelo se torna simplificada.

## 1.1 Objetivos

Este trabalho tem como objetivo principal, demonstrar que o uso de XAI em IDSs de Subestações Elétricas é crucial para torná-lo mais robusto, interpretável, transparente e com a capacidade de obter melhores aprimoramentos, guiados por uma melhor interpretação das previsões. Portanto, se faz necessário, examinar, esquematizar e aplicar técnicas de geração de explicabilidade. Além disso, é fundamental mostrar na prática um aprimoramento do IDS em Subestações Elétricas, guiado por informações obtidas com XAI, e assim aplicar técnicas que ajudem a cumprir o objetivo principal.

Com isso, para aplicar essas técnicas, é essencial utilizar conjuntos de dados que contenham assinaturas de ataques representativos. Será feito uso dos *datasets* gerados pelo *Efficacious Reproducer Engine for Network Operations* (ERENO), criado como um *framework* gerador de *datasets* customizáveis para Subestações Elétricas (QUINCOZES et al., 2023). Estes são compostos por informações trafegadas nesse ambiente, usando exclusivamente protocolos de comunicação empregados pelos Dispositivos Eletrônicos Inteligentes (IED) das subestações, como o *Generic Object Oriented Substation Event* (GOOSE) e *Sampled Values* (SV).

Portanto, o objetivo deste trabalho é tornar o IDS explicável. Para tanto, foram empregadas etapas específicas, descritas abaixo na lista 1.2. Levando em consideração a aplicação do XAI ao setor foco, como a maior contribuição e o aprimoramento conduzido pelo XAI para a detecção de ameaças, sendo a contribuição secundária do trabalho. Vários objetivos foram designados visando alcançar esse intuito maior de demonstrar que, com o XAI, os resultados do IDS se tornam interpretáveis e as tomadas de decisões para aplicar qualquer melhoria ou ajuste são asseguradas, quando se observam as apurações junto com as explicações geradas por técnicas de XAI.

## 1.2 Objetivos Específicos

- Realizar experimentos com os *datasets* gerados pelo *Framework* ERENO, existente para subestações elétricas.
- Implementar o carregamento desses dados e realizar uma análise da natureza dos dados nos *datasets* gerados, inclusive examinar os ataques contidos neles.
- Aplicar técnicas de *Feature Engineering* ao *dataset*, buscando normalizar, codificar e enriquecer os dados para tornar possível a implementação mais inerente aos classificadores.

- Treinar o modelo de aprendizado para vários classificadores de AM, avaliar as métricas de desempenho desses classificadores e selecionar os com melhor resultado, além do tempo de execução.
- Aplicar as técnicas de XAI em cada modelo treinado, em seguida extrair e separar as principais explicações geradas para cada ataque e para cada modelo de classificador.
- Aprofundar uma análise acerca das explicações geradas através dos ataques contidos e quais informações de possíveis avanços eles trouxeram para aprimorar o IDS ou constatar que ele está coerente na identificação do foco de cada ataque.

## 1.3 Justificativa

De acordo com Zetter (2017), a subestação de energia da Pivnichna foi alvo de ataques cibernéticos por invasores que utilizavam de *Malwares* para permitir o controle direto de disjuntores e interruptores. Consequentemente, afetando 230 mil pessoas na cidade mais próxima, que era Kiev a capital da Ucrânia. Em Kshetri e Voas (2017) foi observado muitas companhias de energia dos *United States of América* (EUA) sendo alvos de ataques cibernéticos. Inclusive, foi o caso de uma das maiores empresas de distribuição de energia da Índia, responsável pelo fornecimento de grande parte do leste de Bengala, alvo do *Ransomware Wannacry* (TIMES, 2017), além de empresas do Brasil produtoras de energia tal como a Petrobras (ILASCU, 2021).

Dito isso, o interesse dos invasores cibernéticos em prejudicarem de alguma forma os setores de energia elétrica é alarmante. Pois, tomam como alvo as subestações elétricas, principalmente tentando danificar os disjuntores ou de alguma forma tentar inativar algum dispositivo. Portanto, o IDS precisa constantemente ser melhorado e isso pode ser altamente benéfico e possível se utilizarmos técnicas de *XAI* para gerar modelos mais explicativos. De acordo com (MOLNAR, 2022) os computadores não explicam as suas soluções, a única coisa que nos garante isso é apenas as previsões serem aceitas por uma análise e de alguma forma conferida.

Consequentemente, não basta apenas confiar na eficácia funcional de um IDS. É necessário, de acordo com (DOSHI-VELEZ; KIM, 2017) utilizar de sistemas de explicação para quando há uma "incompletude" na tentativa de explicar completamente o problema ou resultado, o que dificulta muito a melhoria e avaliação de IDS. Neste caso, é necessário evitar especulações e quantificações de incertezas em IDS aplicados em sistemas de subestações digitais. Para isso, é importante sempre apresentar algum resultado que mostre uma relevância

para o usuário, sendo necessário ter isso como requisito ao projeto desse tipo de aplicação, minimizando ao máximo as ambiguidades ou incongruências.

Assim sendo, percebe-se a importância de ter uma melhoria constante nos IDS de subestações elétricas. Por serem ambientes bem complexos, que carecem dessa alta confiabilidade para qualquer manutenção. Para isso, o conjunto de dado do ERENO é escolhido visando a simulação mais fidedigna de eventos legítimos e anormais nesses ambientes. Tendo em vista que precisa conter ataques complexos tal como o *Masquerade* e ataques *DoS*.

Portanto, embora alguns IDS existentes voltados para o setor consigam ter uma alta performance e acurácia, nem sempre conseguiram explicar com exatidão como os classificadores AM, chegaram nos resultados. Contudo, existem diversas técnicas XAI que podem ser aplicados a esse conjunto de dados e ao IDS, de forma atrelada ao resultado. Desta forma, foi possível extrair novas informações relevantes para os analistas, maximizando o uso e as possibilidades de aprimoramentos do IDS com o entendimento de seus resultados.

## 1.4 Organização da Monografia

O restante deste documento tem o seguinte modelo de organização. No Capítulo 2, apresenta-se os fundamentos teóricos do que consistem um IDS, *XAI*, Substações Elétricas principalmente o que é o padrão IEC–61850 e os dispositivos que utilizam ele para criarem seus protocolos de comunicação, a engenharia de dados aplicadas ao ERENO e por fim explorou-se um apanhado de trabalhos relacionados que abordam as mesma técnicas em conjuntos de dados diferentes para detecções em ambientes diferentes. No Capítulo 3, descreve-se os métodos utilizados e a investigação do problema de pesquisa, observando os ataques aos protocolos IEC–61850 presentes no conjunto de dados, juntamente com fatores do ERENO que mais agradam cada ataque e por fim o funcionamento do ERENO. Em seguida, declara-se a hipótese e a solução proposta que inclui a aplicação do XAI justamente para explicar essas preferência desses ataques e se o IDS acertou ou não. No Capítulo 4, apresenta-se os experimentos realizados, os resultados e discussão. Por último, no Capítulo 5 apresenta-se a conclusão, apontando as contribuições.

# 2 Fundamentação Teórica

Este capítulo tem como principal objetivo prover a base necessária para entender os conceitos fundamentais que o trabalho aborda. Para isso, primeiro foi explicado para entendimento de forma elementar o que é uma Subestação Elétrica, o modelo de comunicação, IDS, AM e XAI. Em seguida, uma breve descrição dos métodos usados no processo de pré-processamento das informações que serão utilizadas na implementação do XAI ao *dataset*. Em desfecho, foi discutido os trabalhos analisados que são correlatos.

## 2.1 Sistemas de Detecção de Intrusão

De acordo com Biermann, Cloete e Venter (2001) a detecção de anomalias é baseada na ideia de que ataques a sistemas de informação sejam visivelmente diferentes de uma atividade normal de algum usuário padrão deste sistema. Então de certa forma, estes IDS são criados para informar alguma parte do sistema de que o padrão determinado como normal, foi alterado, ou seja, uma intrusão ocorre. Definida muito bem pelo autor (BACE; MELL et al., 2001) como sendo, "a tentativa de comprometer a confidencialidade, integridade, disponibilidade ou burlar os mecanismos de segurança de um computador ou rede". Mas isso se torna ineficiente quando se tem um IDS que não sabe diferenciar realmente o que são comportamentos anômalos de normais.

Por conta disso, tal como em Lee, Stolfo e Mok (2000) começaram a criar o processo do *software* em conjunto com AM e inteligência artificial para melhorar o monitoramento no tráfego de rede maximizando a identificação desses ataques. Foi mapeado diversos problemas a partir disso com o tempo, mas de forma elementar um Sistema de Detecção de intrusão baseados em AM, deve somente identificar estes comportamentos maliciosos na rede ou em sistemas de forma mais eficiente possível, sem deixar que algum padrão anômalo mas não suficiente para ser identificado como ameaça, seja classificado como uma ameaça.

O processo de um IDS começa através da coleta e armazenamento de dados dos usuários, assim, entra a AM e os algoritmos classificadores para catalogar cada comportamento a partir da análise de uma variedade de elementos, identificando padrões comportamentais. Essa informação combinada ao conjunto de dados colabora para criar um conhecimento prévio sobre as possibilidades de ataque ou reconhecimento de um evento malicioso ou não futuramente.

## 2.2 Subestações Elétricas

As subestações elétricas são consideradas o coração das *Smart Grids* no momento presente, tal como McDonald (2003) explica, as principais funções delas consistem na geração, transmissão, distribuição de energia para a alimentação das unidades consumidoras, sejam elas uma indústria em grande escala ou até mesmo uma cidade. Essas subestações são conectadas fisicamente e cada uma tem um papel na etapa de processamento, tais como, transformações de tensão, chaveamentos ou controle dos circuitos elétricos. Cada etapa do processo elétrico hoje possui níveis de comunicação, alguns não serão muito abordados no trabalho, mas para entendimento geral pode-se verificar na Figura 1.

Os principais equipamentos que fazem parte dos circuitos nessas Subestações são os disjuntores, chaves selecionadoras, equipamentos transformadores de corrente ou tensão, para-raios e os relés de produção ou como são conhecidos modernamente como IED. De acordo com Gaushell e Darlington (1987) os relés de proteção eram eletromecânicos ou estáticos em subestações convencionais, mas após a introdução de microprocessadores a eles, tornou-se muito mais fácil a introdução de algoritmos e a capacidade de operar de forma automatizada. Logo, passou a ser possível ter dispositivos multifunções, principalmente com a função de comunicação via *Ethernet*, ou seja, rede via cabo, possibilitando comunicação rápida com a interface entre o Sistema de Automação de Subestação (SAS), seguindo para um centro de monitoramento *Supervisory Control and Data Acquisition* (SCADA).


Figura 1 – Nível de comunicação do Sistema Elétrico

Fonte: Adaptado de (MENDES; BORTOLI; COSTA, 2021)

### 2.2.1 Norma IEC-61850

A norma IEC-61850 (IEC, 2003) foi criada pra solucionar um problema muito comum no setor elétrico, que era a falta de interoperabilidade dos dispositivos. Isso significa que, cada fabricante dos dispositivos desse meio usava seus próprios protocolos de comunicação e muitas das vezes era em formatos diferentes como ondas de rádio privadas, ou em outros formatos mais propensos a carregarem limitações de banda, interferências ou segurança.

Segundo (BAIGENT et al., 2004), a norma IEC-61850 para Redes e Sistemas de Comunicação em Subestações foi criada pela *International Electrotechnical Commission* (IEC). O trabalho foi realizado pelos Grupos de Trabalho 10, 11 e 12, parte do Technical Committee 57 (TC57), que resultou na Norma Internacional IEC-61850 de Redes e Sistemas de Comunicação em Subestações.

Eles usaram como referência o modelo conceitual que descreve como funciona os protocolos OSI(*Open Systems Interconnection*). O qual é definido por (TANENBAUM, 2003), como sendo uma estrutura conceitual que descreve as funcionalidades de um sistema de comunicação em uma rede de computadores com o objetivo de padronizar e facilitar a comunicação entre dispositivos da mesma.

Portanto, é possível notar bastante semelhança entre os dois modelos, uma delas é que a norma especifica protocolos de comunicação para a camada física das subestações (dispositivos de campo), tendo como referência a comunicação de computadores via *Ethernet* e TCP/IP para garantir entrega confiável dos dados, entre outras semelhanças. A norma gerou vários protocolos, porém os principais que mais foram abordados no trabalho foram o GOOSE e o SV.

### 2.2.2 Protocolo GOOSE

Um protocolo destinado aos Evento de Subestação Orientado a Objetos Genéricos, como é definido por (IEC, 2003), é destinado para troca rápida de informações de status e controle entre os IEDs na subestação. As mensagens GOOSE são usadas para transmitir mudanças de status no sistema de energia, como status do disjuntor, posição do comutador ou condição de falha, entre todos os IEDs dentro de uma subestação.

O protocolo GOOSE da norma IEC 61850 possui uma estrutura que inclui doze campos na unidade de dados do protocolo (PDU). De acordo com a Figura 2, os primeiros campos correspondem aos campos iniciais de um quadro *Ethernet* padrão, com o destino e origem sendo um endereço *multicast MAC Ethernet* e *unicast MAC* respectivamente. O quarto octeto do endereço define o tipo de mensagem.

De acordo com Hoyos, Dehus e Brown (2012) os campos *Reserved1* e *Reserved2* são reservados para futuros aplicativos padronizados e são definidos como 0 por padrão. Os últimos campos incluem o comprimento do APDU (Unidade de Dados de Protocolo de Aplicação) e a sequência de soma de verificação do quadro para garantir a integridade dos dados transmitidos.

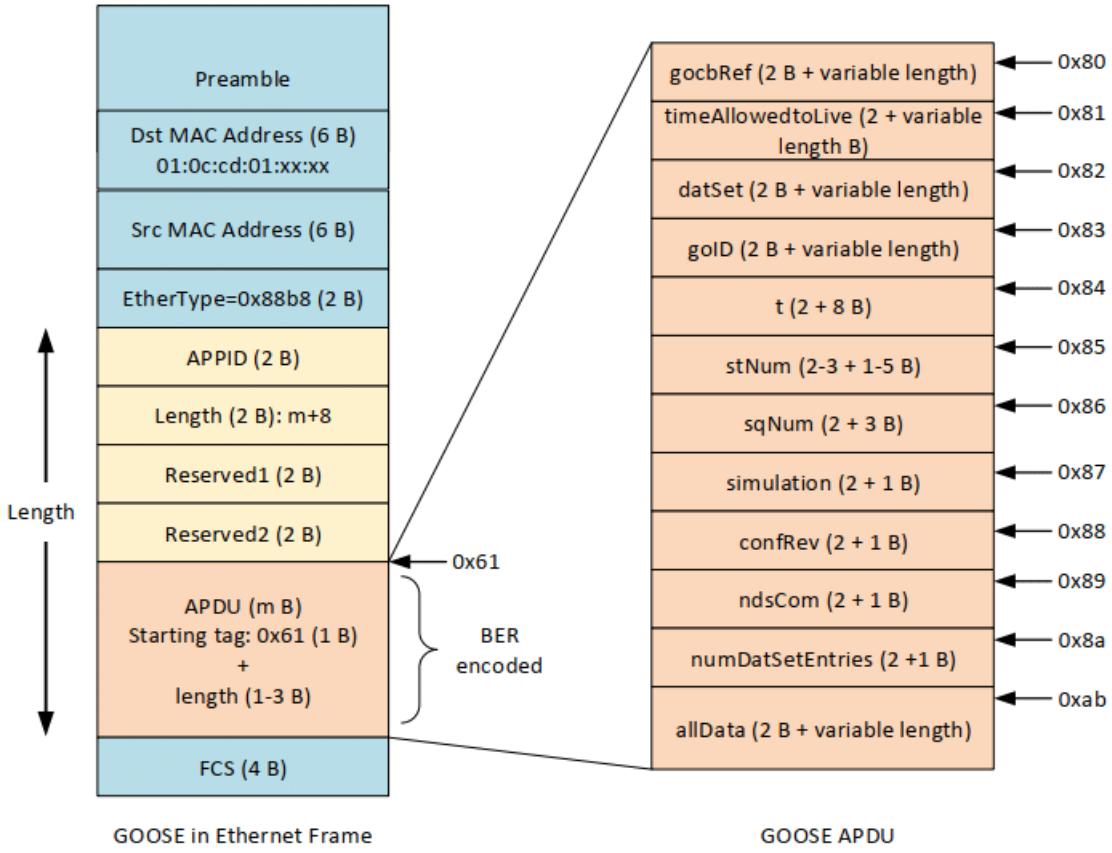



Figura 2 – Estrutura do Protocolo GOOSE

Fonte: (MATOUŠEK, 2018)

Observa-se na Figura 2 que a composição da APDU inclui referências ao bloco de controle (gocbRef), tempo de espera para o receptor (timeAllowedToLive), nome do conjunto de dados (datSet), número de elementos no conjunto de dados (numDatSetEntries) e identificador do remetente (goID). Mas, principalmente o *timestamp*(t) em que o stNum foi incrementado, contadores de envio do Goose (stNum e sqNum), indicador de teste (test), revisão da configuração contendo a contagem do número de vezes que a configuração do conjunto de dados foi mudado(confRev). Os próprios dados trocados estão no allData e as necessidade de reconfiguração em ndsCom.

De acordo com o Baigent et al. (2004) que descreve o funcionamento para os usuários, as mensagens GOOSE são mensagens *multicast* e são endereçadas a um endereço de grupo *multicast*. Todo IED interessado em receber a mensagem GOOSE assina o endereço do grupo *multicast*. Quando ocorre uma mudança de status em um IED, ele envia uma mensagem GOOSE para todos os outros IEDs que assinaram o endereço do grupo *multicast*. Todos os IEDs inscritos recebem a mensagem GOOSE e atuam na mudança de status.

As mensagens GOOSE podem ser enviadas de um IED para vários IEDs ou de vários IEDs para um IED. A mensagem GOOSE inclui a identidade do remetente e o status ou informações de controle que estão sendo transmitidas ou solicitadas. O IED destinatário então usa essas informações para atualizar seu status local ou tomar as medidas de controle apropriadas.

### 2.2.3 Protocolo SV

O protocolo de comunicação SV é baseado no padrão IEC-61850 definido por (IEC, 2003). Ele foi projetado para transmitir de maneira eficiente as medições de grandezas analógicas a partir dos sensores nos equipamentos de subestações para os sistemas de processamento e monitoramento na subestação. As medições de grandeza analógica são originadas das medidas em transformadores, disjuntores e seccionadores, entre outros dispositivos.

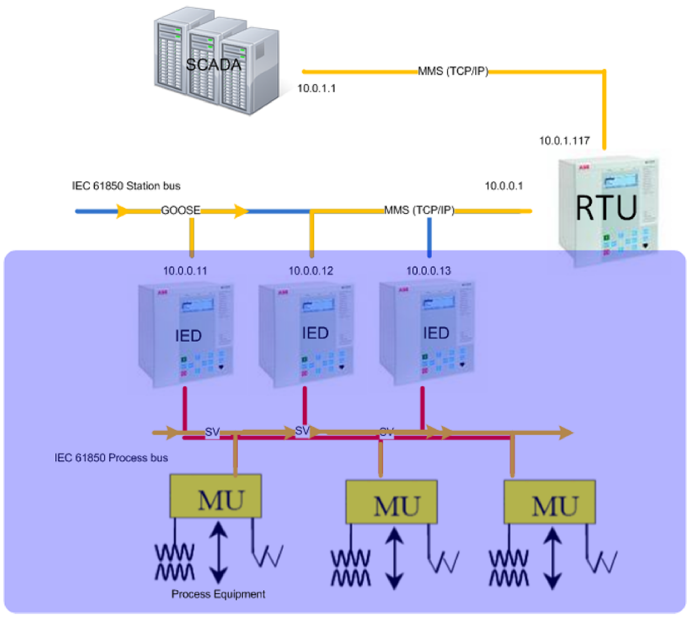



Figura 3 – Estrutura de comunicação Subestação Elétrica

Fonte: (ZHAO, 2012)

Na Figura 3, é esclarecido que o SV é predominantemente utilizado para troca de informações entre *Merging Unit*s (MUs) e IEDs. O atributo-chave do protocolo SV é sua capacidade de transmitir em tempo real as amostras de grandeza com altíssima precisão e um baixo atraso. Que (QUINCOZES et al., 2023) explicou sendo medições coletadas por meio de sinais analógicos de equipamentos elétricos e convertidas em sinais digitais por MUs, tal como está ilustrado na Figura 3, sendo enviadas periodicamente em alta taxa de transmissão para fins de proteção e medição.

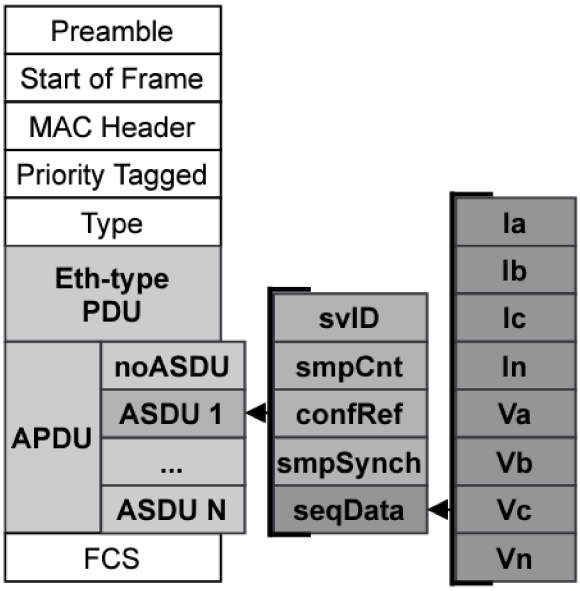



Figura 4 – Estrutura do protocolo SV

Fonte: (QUINCOZES et al., 2023)

A Figura 4 revela a estrutura do protocolo SV, onde de acordo com trabalho (SOLOMIN; TOPOLSKY, 2015) a APDU na estrutura é usada para transmitir amostras de grandezas analógicas, como tensão e corrente do protocolo. Além dos campos padrões para os protocolos, no SV cada campo *Application Service Data Unit* (ASDU) contido na APDU inclui svID (identificador de valores amostrados), smpCnt (índice da mensagem), confRev (revisão da configuração), smpSynch (modelo da sincronização do relógio) e seqData (sequência de valores de tensão e corrente). Em suma, o ASDU contém as amostras de grandezas analógicas, enquanto a APDU é a estrutura de dados que transporta elas.

## 2.3 Explainable Artificial Intelligence (XAI)

Trata-se de uma área da Inteligência Artificial, que vem ganhando forças no tempo presente e que está atrelada ao escopo de *Ethical AI*, que de acordo com Vainio-Pekka et al. (2023), é frequentemente dividida em princípios como transparência, responsabilidade, confiança, privacidade, sustentabilidade, autonomia e dignidade. Por isso, é possível dizer que grande parte desses objetivos hoje podem ser cumpridos com técnicas XAI. Entretanto, vários pesquisadores buscam já há algum tempo criar maneiras de tornar isso possível e divagaram entre técnicas para atender essa necessidade da compreensão humana na decisão das IAs.

Essas buscas que resultam no campo *Explainable AI* no presente, de certa forma, existe desde 1970, de forma indireta, quando surgiu a necessidade dos sistemas inteligentes tal como o MYCIN em se tornarem transparentes e confiáveis para humanos (CONFALONIERI et al., 2021), mas só ultimamente ela vem ganhando força com o avanço das IAs. Pode-se observar que existem avanços principalmente de ferramentas tal como SHAP, LIME dentre outros técnicas já utilizadas para esse conceito e aplicadas a muitos outros IDS.

Por essa razão, na época presente já existem diversas técnicas desenvolvidas para tornar modelos de AM em modelos explicativos e não mais apenas interpretáveis. Os quais Molnar (2022), gosta de distinguir entre dois conceitos, sendo eles Interpretabilidade e Explicabilidade. Ou seja, um modelo de IA pode ter a capacidade de ser entendido ou interpretado por seres humanos, mas nem sempre tem a capacidade de explicar suas decisões de uma forma que seja compreensível para os seres humanos.

### 2.3.1 SHAP

Em 2017, os pesquisadores em Lundberg e Lee (2017) propuseram um *framework* unificado chamado SHAP(*SHapley Additive exPlanations*), como sendo uma adaptação e extensão do conceito de "Valores de Shapley" (*Shapley values*), desenvolvido por Shapley et al. (1953), um economista e matemático americano. O SHAP *framework* utiliza o método do matemático que se baseia da teoria dos jogos de coalizão onde uma previsão pode ser entendida como um jogo, onde cada valor de recurso é um jogador e a previsão é o prêmio, nos ajudando a distribuir de maneira justa o prêmio entre os recursos.

Para entender a fundo, é viável imaginar que através de um conjunto de recursos para um modelo e pode-se calcular o valor de *Shapley* para um determinado recurso. O valor de *Shapley* é a medida de contribuição média(ponderada) do valor desse recurso para a previsão final do modelo em diferentes coalizões, portanto, ele não pode ser confundido como sendo a diferença na previsão quando se remove o recurso do modelo. A fórmula abaixo

simplificada, mostra que para calcular o valor de *Shapley* de um recurso é dada em termos da sua contribuição marginal para todas as possíveis combinações de recursos.

$$\phi_i = \sum_{S \subseteq F \backslash \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[ f_{S \cup \{i\}} \left( x_{S \cup \{i\}} \right) - f_S \left( x_S \right) \right] \tag{2.1}$$

Onde:

- $\phi_i$: representa o valor *Shapley* do recurso $i$.
- $|S|$: Um subconjunto de recursos excluindo $i$.
- $|F|$: O conjunto completo de recursos.
- $S \subseteq F \backslash \{i\}$ : significa que o conjunto $S$ é um subconjunto dos recursos restantes após remover o recurso $i$, ou seja, $F \backslash \{i\}$ isso significa que foi removido o recurso i do conjunto total de recursos.
- $\frac{|S|!(|F|-|S|-1)!}{|F|!}$ : é o termo de normalização que ajusta o peso das contribuições marginais.
- $x_S$: os valores dos recursos de entrada no conjunto S.
- $f_{S \cup \{i\}} \left( x_{S \cup \{i\}} \right)$: O valor predito pelo modelo quando todos os recursos em $S$ e $i$ estão presentes.
- $f_S \left( x_S \right)$: O valor predito pelo modelo quando apenas os recursos em $S$ estão presentes.
- $\left[ f_{S \cup \{i\}} \left( x_{S \cup \{i\}} \right) - f_S \left( x_S \right) \right]$: é a diferença entre a previsão quando o recurso $i$ está presente (em conjunto com os recursos em $S$) e a previsão quando apenas os recursos em $S$ estão presentes.

No trabalho (MOLNAR, 2022), o autor afirma que os valores *Shapley* são a única solução que satisfaz as propriedades de Eficiência, Simetria, Dummy e Aditividade, e que o *Framework de Lundberg* também calcula valores de *Shapley*, mas ele define as propriedades principais como sendo Precisão Local, Falta e Consistência, seguindo depois com as mesmas do *Shapley* como a Linearidade, Dummy e Simetria. Resumindo sua abordagem, é viável interpretar cada propriedade da seguinte forma:

- **Precisão Local**: garante que as explicações geradas reproduzam com precisão as previsões do modelo original para instâncias individuais.

- **Falta**: A falta assegura que características irrelevantes recebam uma contribuição de Shapley próxima de zero. Em (LUNDBERG; LEE, 2017) o autor diz que diferente do cálculo de Shapley a presença de 0 pode ser um valor arbitrário para representar recursos ausentes sem prejudicar a precisão local e ajudar a aplicar o cálculo em modelos de AM. Portanto, a falta pode ser expressa como $\phi_i \approx 0$.

- **Consistência**: A consistência implica que se a contribuição marginal de uma característica aumentar ou diminuir para uma instância de entrada alterada em comparação com a instância original, então a ordem relativa das contribuições de Shapley para todas as características não deve mudar drasticamente , podendo ser definida como:

$$f'_x(z') - f'_x(z' \backslash i) \geq f_x(z') - f_x(z' \backslash i) \tag{2.2}$$

  Para todos os *inputs* $z' \in \{0,1\}^M$, então $\phi_i(f', x) \geq \phi_i(f, x)$. Onde *Lundberg* afirma que a mudança na previsão do modelo ao remover característica $i$ deve ser pelo menos tão grande para a instância $x$ quanto para a instância $x'$. Isso é importante para garantir que a consistência seja mantida entre diferentes instâncias de entrada.

- **Linearidade**: tanto para *Shapley* quanto para *Lundberg*, essa propriedade exige que a soma das contribuições de *Shapley* para um conjunto de características seja igual à contribuição para o conjunto inteiro. Podendo ser expressa como: $\sum_{i=1}^{m} \phi_i = \phi(N)$

- **Dummy**: Ambos afirmam que se uma característica $i$ não tem influência em nenhuma coalizão, sua contribuição de *Shapley* é zero: $\phi_i = 0$

- **Simetria**: Para duas características $i$ e $j$ que contribuem igualmente para todas as coalizões, suas contribuições de *Shapley* são iguais, ou seja, $\phi_i = \phi_j$

## 2.4 Aprendizado de Máquina (AM)

Nascido das teorias criadas no campo da Inteligência Artificial, o termo foi desenvolvido e aplicado pela primeira vez como sendo "o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados"(SAMUEL, 1959). Arthur Samuel demonstrou a teoria utilizando um jogo de damas chamado de *Game of Checkers*, sendo o primeiro programa que aprendia sozinho a realizar uma tarefa.

Desde então, o conceito tem sido amplamente discutido e chegado a uma definição mais formal. Que de acordo com (MITCHELL, 1997) é "um programa de computador que aprende pela experiência $E$ a respeito de alguma classe de tarefas $T$ e performance mensurada $P$, se a performance em tarefas $T$, conforme medido por $P$, melhora com a experiência $E$".

Essa definição então, enfatiza a capacidade de um algoritmo ou programa ser autodidata e melhorar sua performance em alguma tarefa com base na experiência adquirida ao longo do tempo.

Dito isto, na época atual utiliza-se o AM em diferentes áreas, inclusive a segurança da informação. Os algoritmos de AM mais utilizados para detecção de intrusões são Árvores de decisões, *Random Forest*, *Support Vector Machines*(SVM), *K-nearest Neighbor*(KNN), *Redes Bayesianas*, Modelo Oculto de Markov(HMM) e *Artificial Neural Network*(ANN).

Existem 3 categorias, até agora, de AM. A primeira delas é o **Aprendizado Supervisionado**, onde um conjunto de entradas e saídas são fornecidas por um humano, para ajudar o algoritmo a aprender quais são as entradas/saídas desejáveis para o objetivo em questão. Em seguida, vem o **Aprendizado Não Supervisionado**, onde nenhum tipo de categorização dos dados é feita, o próprio algoritmo, nesse caso, vai buscar formular ou encontrar padrões nos conjuntos de dados e fornecer algum resultado para atingir algum fim.

Por fim o **Aprendizado por reforço**, onde o aprendizado é baseado em recompensas, o algoritmo vai interagir com o ambiente dinâmico e maximizar suas recompensas. A diferença com o Aprendizado Supervisionado é que, neste caso a única interferência seria de uma sinalização de recompensa por ter atingido um resultado satisfatório, mas para atingir tais objetivos ele pode fazer suas próprias escolhas arbitrárias.

## 2.5 Feature Engineering

A *Feature Engineering* ou Engenharia de Atributos, de acordo com (ZHENG; CASARI, 2018) é o processo de usar o conhecimento do domínio para selecionar, modificar ou criar novas características a partir de dados brutos. Logo, é um conjunto de técnicas amplamente utilizados pelos cientistas de dados para garantir que os dados estejam o mais adequado possível, na busca por uma melhoria na qualidade dos resultados. (TURNER et al., 1999) é um trabalho onde argumenta-se que, para que o ciclo de vida de execução de um *software* seja garantido com êxito se faz necessário a adição de etapas para preparar os recursos.

Inicialmente, os dados podem servir como uma porta de entrada para a compreensão do sistema e sobre o problema em questão. Com isso, na primeira etapa, a análise da natureza dos dados é imprescindível para saber quais técnicas aplicar e alcançar uma boa qualidade nos dados. Para isso, existem inúmeras técnicas e cada uma colabora pra algum objetivo para a preparação e manipulação dos dados, algumas das mais utilizadas são Codificação, *Principal component analysis* (PCA), *Clustering*, Transformação numérica, Normalização,

Padronização, Agrupamentos e Cruzamento de recursos para geração de novos parâmetros.

Por fim, algumas das técnicas geralmente são aplicadas de forma manual através de análise exaustiva dos recursos e outras já se tornaram automatizadas. Portanto, para preparar os dados para o modelo de AM, foi abordado um enriquecimento temporal, juntamente com técnicas mais automáticas de pré-processamento. Sendo elas, normalização de caracteres categóricos com *Label Encoder*, *Min Max* ou *One Hot Encoder* para dados numéricos.

## 2.6 Trabalhos Correlatos

O objetivo desta seção é analisar e revisar criticamente os trabalhos relacionados com o propósito central deste trabalho de detecção de intrusões em subestações elétricas através de IDS e XAI, com conjuntos de dados próprio para isso. Assim sendo, buscando identificar os desafios atravessados na área e avanços explorados que vão auxiliar na abordagem de um ponto inicial, foi priorizado os trabalhos que apresentavam abordagens relevantes e resultados significativos. Ao final desta seção, foi introduzido a Tabela 2.6 para auxiliar na identificação principal das diferenças entre os trabalhos.

Um trabalho muito útil para ajudar na perspectiva inicial foi realizado por (NEUPANE et al., 2022) com uma revisão abrangente sobre os métodos, desafios e oportunidades atuais para desenvolver IDSs que utilizam técnicas de IA/AM e que também sejam explicáveis. O autor não chegou a discutir todas as técnicas de XAI, mas de forma ampla, ele explicou pela abordagem de "caixa branca" e a abordagem "caixa preta". Com base nisso, ele propôs uma nova taxonomia para ajudar na definição formal sendo "X-IDS" para explicabilidade na detecção de intrusão e fornecer uma direção para pesquisas futuras.

No trabalho de (WANG et al., 2020) foi proposto um *Framework*, ou seja, uma estrutura ou base para iniciar a resolução de um problema ou desafio. No caso, o objetivo era tornar os modelos de IDSs, que utilizam de aprendizados de máquina, mais explicáveis, apresentar as explicações de forma acessível aos usuários não especialistas e por fim implementá-lo buscando balancear entre a precisão e a relevância das explicações. Para isso, ele utilizou o método SHAP e coordenou entre explicações globais e locais, com o objetivo de comparar os resultados de dois classificadores principais e avaliaram a eficácia do modelo a um conjunto de dados apenas, que foi o NSL-KDD.

O trabalho foca no uso de duas abordagens diferentes para lidar com problemas de classificação multi classe, sendo eles o *One-vs-All Classifier*(OvA) e o *Multiclass Classifier*. O primeiro busca dividir o problema de classificação em várias classificações mais simples, que decidem entre uma classe (de cada vez) a todas as outras. O segundo faz isso diretamente

buscando a classificação de dados em mais de duas categorias diferentes em um único passo. Eles realizaram um pré-processamento de dados baseado no *Label Encoder* e *min-max values* para normalização dos dados contidos entre intervalos. Por fim, a avaliação de performance utilizada teve como métricas a acurácia, precisão, *recall* e *F1-score* e eles obtiveram bons resultados desses dois classificadores em comparação com os de trabalhos anteriores.

Em Sivamohan, Sridhar e Krishnaveni (2023) foi criado um novo esquema de IDS denominado por ele por TEA-EKHO-IDS que usa seleção bioinspirada de variáveis e técnicas Bayesianas para encontrar melhores conjuntos de hiperparâmetros para otimizar a classificação. O campo foco de atuação do autor foi voltado para Sistemas Ciberfísicos, na sigla inglês(CPS), predominante de indústrias. O principal motivo foi a quantidade heterogênea de dados para esses sistemas e o quão difícil é para otimizar uma classificação nesses tipos de conjuntos de dados. Para alcançar aprimoramento nesses quesitos, ele adotou algoritmos otimização do rebanho de krill e implementou também um modelo mais fácil pra trabalhar com dados temporais usando rede neural *Bidirectional Long Short-Term Memory* (Bi-LSTM).

Os *datasets* selecionados por ele foram NSL-KDD 2015 e CIC-IDS2018. Apesar de ele contemplar várias métricas tal como, precisão, acurácia, *recall*, *F1-score* e taxa de alarme falso, no classificador proposto usando seleção bioinspirada para um total de 9 classes de ataques, demonstraram no gráfico de valores SHAP universais e de força para 4 classes de ataques observados. Além disso, foi explicado brevemente quais era as relações das estimativas do gráfico, porém sem relacionar com o comportamento dos ataques e se estava condizente os *Shapley values* universal pra cada classe. Contudo, o autor reforçou bem o quão bom pode ser para um modelo de classificação a utilização de técnicas bioinspiradas de seleção e adições de meta-heurística.

No trabalho de (KUZLU et al., 2020) o autor implementou e discutiu o uso de algumas técnicas de XAI para aprimorar as suas previsões de geração de potência fotovoltaicas. Foi utilizado para isso, dados históricos de carga de energia elétrica obtido a partir de um *dataset* comumente utilizado em uma competição internacional que desafia pesquisadores e profissionais a desenvolverem modelos de previsão de carga de energia elétrica, chamado GEFCOM (*Global Energy Forecasting Competition*). Com isso, usando XAI, ele abordou o problema central que era a dificuldade de modelos de previsão tradicionais terem em explicar as previsões em certas condições climáticas, como clima nublado ou na variabilidade da irradiação solar (intensidade da luz) devido às condições meteorológicas.

O trabalho colabora bastante no quesito XAI, justamente por trazer uma variedade de aplicações XAP, ele utilizou 3 no total foi possível extrair bastante *insights* sobre as previsões. Ao utilizar LIME ele conseguiu destacando que SSRD (dados de radiação solar

de superfície), HORA e TSR (radiação solar terrestre) são os recursos mais importantes, enquanto TCWL (topo do dados de céu claro de radiação atmosférica), U (velocidade do vento) e TP (precipitação total) são os menos importantes. Já a sua abordagem com o SHAP revela contribuições positivas e negativas de cada recurso, destacando SSRD como o mais influente. Já o ELI5 fornece pesos e destaca novamente que SSRD tem o maior impacto na saída do modelo. Essas análises visaram melhorar os modelos de previsão solar.

Em (MUNIR; SHETTY; RAWAT, 2023) foi proposto um *framework* de Inteligência Artificial confiável que detecta e explica a causa raiz de ataques cibernéticos em recursos energéticos distribuídos. Nomeados como *Distributed Energy Resources*(DERs), eles podem ser integrados com as *smarts grids*, tal como painéis solares, geração eólica de pequena escala ou integração de veículos elétricos à rede, todos esses podem ser considerados uma forma de DER. O principal problema que ele tenta resolver é justamente a dificuldade e nível da confiabilidade na identificação de forma proativa e interpretável de riscos cibernéticos causados pela mensagem de controle/status dos DERs integrados na *smart grid*.

Os autores testaram sua estrutura de IA confiável proposta usando um conjunto de dados de sistema ciber-físico chamado SCADA WUSTL-IIOT-2018. O conjunto contém 5 tipos de ataques relacionados a Sistema SCADA. Os autores usaram vários modelos de AM, como *Extra Tree*, *Random Forest*, *Gradient Boosting*, *AdaBoost* e regressão linear como classificadores. Além disso, aplicaram métodos SHAP e incorporaram um esquema de agrupamento hierárquico baseado em variância mínima de *Ward* para garantir a reprodutibilidade e explicabilidade da estrutura de IA confiável proposta.

Com isso, esses trabalhos estão relacionados com o trabalho atual, por utilizarem técnicas XAI na busca de tornar previsões mais interpretáveis. Além disso, mesmo que o foco experimental de alguns dos trabalhos esteja relacionado com rede elétrica, nenhum deles foi usado especificamente para as subestações elétricas, que se trata de uma das partes da rede elétrica inteligente (*Smart Grids*) como um todo. Em adição, esse trabalho foca em ataques específicos desse ambiente e utiliza o enriquecimento temporal como uma contribuição a mais no quesito melhora de previsões e qualidade de dados, que são providos por protocolos baseados na norma IEC-61850.

Portanto, todos os trabalhos analisados foram destacados na Tabela 2.6 abaixo, juntamente com os discutidos nesta seção. Eles demonstram a possibilidade de aplicar técnicas XAI a diversos ambientes, principalmente relacionado ao setor elétrico. Além disso, as observações desses estudos anteriores fundamentaram a escolha do SHAP como ferramenta de XAI desse trabalho, por revelarem a eficácia da ferramenta para mostrar as contribuições positivas e negativas de cada recurso nas previsões, tanto em análise global, local e relacio-

nada entre recursos de forma parcial, além de ter uma documentação robusta. Sendo assim, a Tabela 2.6 destaca bem as diferenças e focos de cada trabalho.

Tabela 1 – Trabalhos Correlatos

| Referência | Domínio | Técnicas XAI | Datasets | Classificadores | Ataques |
|---|---|---|---|---|---|
| Este trabalho | Subestações Elétricas | SHAP | ERENO | XGBoost<br>Random Forest<br>Decision Tree<br>CatBoost<br>KNN<br>SVM | High StNum Injection<br>Inverse Replay<br>Masquerade<br>Poisoned high rate<br>Random replay |
| (WANG et al., 2020) | Rede Corporativa | SHAP | NSL-KDD | One-vs-All Multiclass | DoS<br>U2R<br>R2L<br>Probe |
| (SIVAMOHAN; SRIDHAR; KRISHNA-VENI, 2023) | Rede Corporativa | SHAP | NSL-KDD<br>CIC-IDS | LSTM<br>GRU<br>BiLSTM<br>TEA-EKHO-IDS | DoS<br>Probe<br>R2L<br>U2R<br>Bot Web<br>Bruteforce<br>Infiltration<br>Injection |
| (KUZLU et al., 2020) | PV Power | SHAP<br>LIME<br>ELi5 | GEFCOM | RFR | Não se aplica |
| (MUNIR; SHETTY; RAWAT, 2023) | DERs SmartGrids | SHAP | SCADA<br>WUSTL-IIOT-2018 | Random Forest<br>Extra Tree<br>Gradient Boosting<br>AdaBoost<br>Linear Regression | PortScanner<br>AdressScan<br>Device Id.Atack<br>Agressive mode<br>Exploit |
| (ZOLANVARI et al., 2021) | IIoT | TRUST<br>LIME | SCADA<br>WUSTL-IIOT-2018<br>NSL-KDD<br>UNSW | ANN | Backdoor<br>Injection<br>DoS<br>Reconnaissance |

# 3 Desenvolvimento

Este capítulo é dedicado a explicar os métodos e passos utilizados no trabalho para o aprimoramento de IDSs em Subestações Elétricas, garantindo previsões imparciais e transparentes das ameaças cibernéticas desse ambiente através de técnicas de XAI e dados do *Framework* ERENO. A metodologia será experimental, usando o SHAP atrelado as previsões do IDS, *Feature Engineering* e interpretação de dados. Porém, visando muito mais em experimentar a técnica XAI, para demonstrar que adicionar o fator explicabilidade em qualquer etapa de desenvolvimento do IDS é muito mais vantajosos, tornando-o mais interpretável, confiável e simplificado para o uso em Subestações elétricas.

As etapas iniciais do desenvolvimento incluem a geração de dados via *Framework* ERENO, preparação básica e normalização de dados, além da aplicação do *Decision Tree* como um classificador inicial para que a primeira análise seja realizada. Com isso, a qualidade do IDS é avaliada inicialmente com métricas resultantes de matrizes de confusão e gráficos SHAP das previsões globais e parciais. Após a análise inicial, correções paliativas necessárias foram aplicadas, seguidas de etapas para o enriquecimento de dados e uma nova avaliação dos resultados do IDS, juntamente com os gráficos SHAP.

## 3.1 Cenário contextual

Para que faça sentido o contexto de inserção das técnicas XAI, é necessário compreender melhor os dados trafegados no ambiente de subestações elétricas e como isso foi replicado para tornar esse trabalho possível. A fonte dos dados é originada do *Ereno Framework* (QUINCOZES et al., 2023) que usou da ferramenta PSCAD/EMTDC para modelar a rede elétrica, simulando a operação dos circuitos em casos normais e cenários de falhas ou ataques e extraindo as medições elétricas tanto GOOSE e SV, de cada caso.

Com isso, os dados realistas são imputados pela ferramenta *Ereno Framework*, que inclusive é feito em JAVA e possui uma interface UI, que possibilita inputs manuais de algumas informações, tornando possível algumas alterações até mesmo da quantidade de ataques ou casos de ataque. Os recursos gerados na criação do conjunto de dados envolvem recursos básicos extraídos de pacotes de rede e recursos enriquecidos, sendo eles para o protocolo SV, com um total de 37 recursos (13 básicos e 24 enriquecidos), enquanto que para o protocolo GOOSE há um total de 32 recursos (22 básicos e 10 enriquecidos).

O *Ereno Framework* foi uma maneira de sanar a falta de dados realistas para tornar o

objetivo de melhorar o desempenho de IDS nesse setor possível. Além disso foi realizado uma etapa de seleção de *Features* em trabalhos correlacionados a esse trabalho, para que possam escolher as *features* que realmente são mais significativas para o processo de detecção.

Por fim, a Figura 5 abaixo mostra o ambiente que o IDS vai interagir acoplado especificamente ao monitoramento. Com informações desta rede, o *ERENO Framework* vai simular a parte necessária desse ambiente com os dados realistas da rede até os componentes foco.

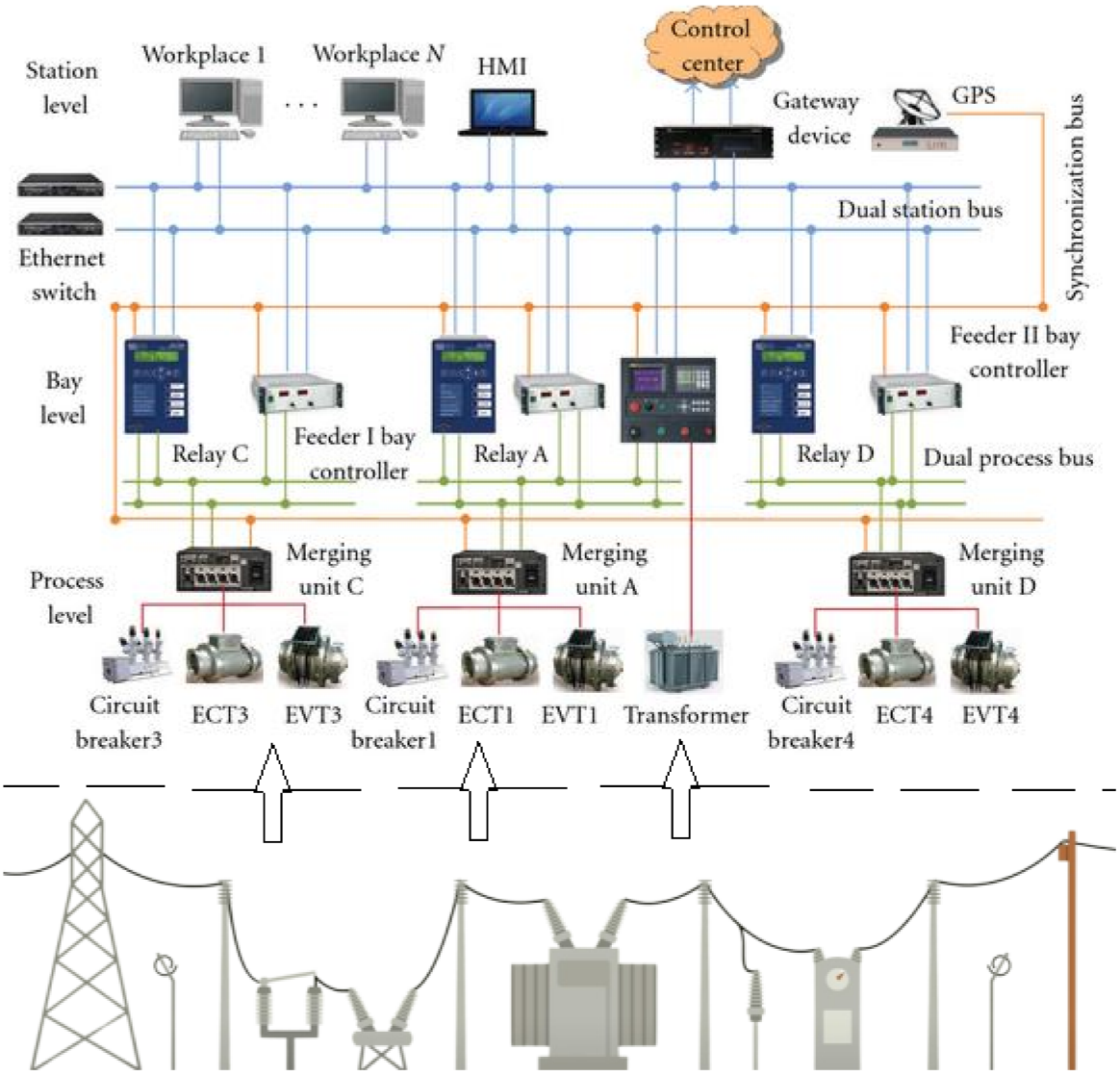



Figura 5 – Etapas desde a extração de energia até a parte de monitoramento da rede

Fonte: Adaptado de (LU; WANG; MA, 2012)

## 3.2 Arquitetura do aprimoramento do *Substation IDS* com XAI e FS

Para representar de forma geral todo o processo de construção e aprimoramento do modelo de detecção de intrusões em subestações elétricas, baseado na norma IEC-61850, pode ser observado na Figura 6 a baixo. As etapas são apresentadas, desde a exportação, processamento, enriquecimento e extração de *features* dos dados até a geração de gráficos interpretáveis e a avaliação da performance do modelo.

O fluxo começa com a carga, pré-processamento dos dados, seguido pelo treinamento do modelo e a extração de explicações XAI, usando a biblioteca SHAP. Após isso, é explicado parte por parte e o objetivo de cada etapa aplicada, seguindo uma abordagem mais técnica de cada uma, destacando as escolhas feitas e as metodologias aplicadas.

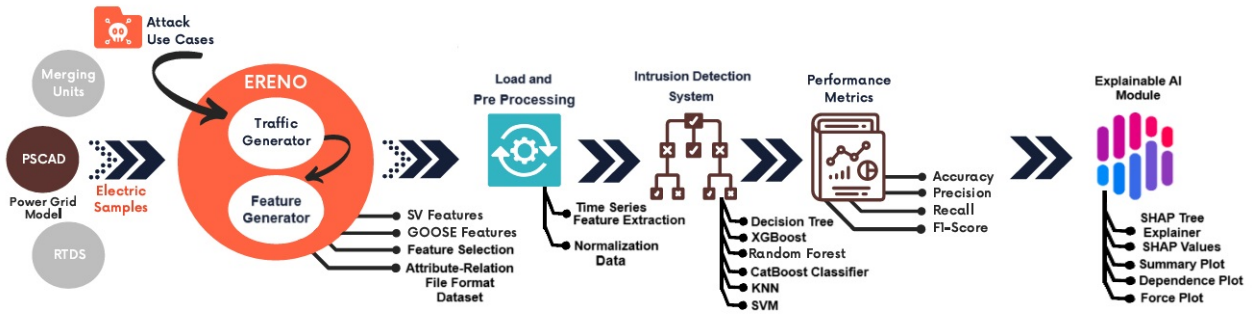



Figura 6 – Imagem mostrando o processo de acoplamento da Feature Selection e XAI ao IDS

Fonte: Adaptado de (QUINCOZES et al., 2023), pag.58

## 3.3 Coleta e Pré-processamento dos Dados

A primeira etapa do processo consiste na geração e carregamento dos dados. Foi criada uma função responsável por carregar os *dataset* de treino e teste, resultados da configuração da simulação de tráfego de rede com eventos legítimos e anormais pelo *ERENO UI Framework*. Esses *datasets* de tráfego customizável, são exportados da ferramenta no formato *Attribute-Relation File Format* (ARFF) em uma pasta compartilhada com o módulo *Python*, que tem um *trigger* que é ativado ao receber o arquivo, que irá passar ele para o formato tabular mais apropriado para o trabalho. Após isso, um formato tabular em memória é criado, com respectivos arquivos de teste e treino do modelo classificador do IDS, ou seja, dois arquivos tabulares são entradas do módulo e transformados para *Dataframes* do "*Pandas*" no *Python*.

Em seguida, ao analisar os gráficos de histogramas e explorar a natureza dos dados,

notou-se que eles estavam em magnitudes e distribuições discrepantes entre os seus valores, principalmente dados de *Timestamp*. Com isso, foi criado uma função de pré-processamento dos dados, que irá normalizar, transformar, codificar e enriquecer os dados e de forma automatizada, recebendo os dados e *labels* de *Dataframe Pandas* como entrada, executando, depois, vários métodos de pré processamento com base na natureza das colunas.

### 3.3.1 *Feature Extraction* com técnica de segmentação de séries temporais

A aplicação de técnicas de extração de novos atributos pela ideia de extrair estatísticas de segmentos de janelas móveis de tempo não sobrepostas, foi levada em consideração a partir do momento em que se finalizou a investigação da primeira análise de métricas e matrizes de confusão, das classificações dos ataques. Usou-se inicialmente o *Decision Tree* no IDS base, com apenas normalização básica. Pois, no primeiro resultado ele obteve um desempenho perfeito demais para maioria dos ataques, como mostrado na Figura 7, altos valores na diagonal e uma alta taxa de acertos. O *dataset* de treino continha 1000 eventos normais e 1000 de cada ataque, contendo no total 7000 linhas, já o de teste continha 200 para ataques.

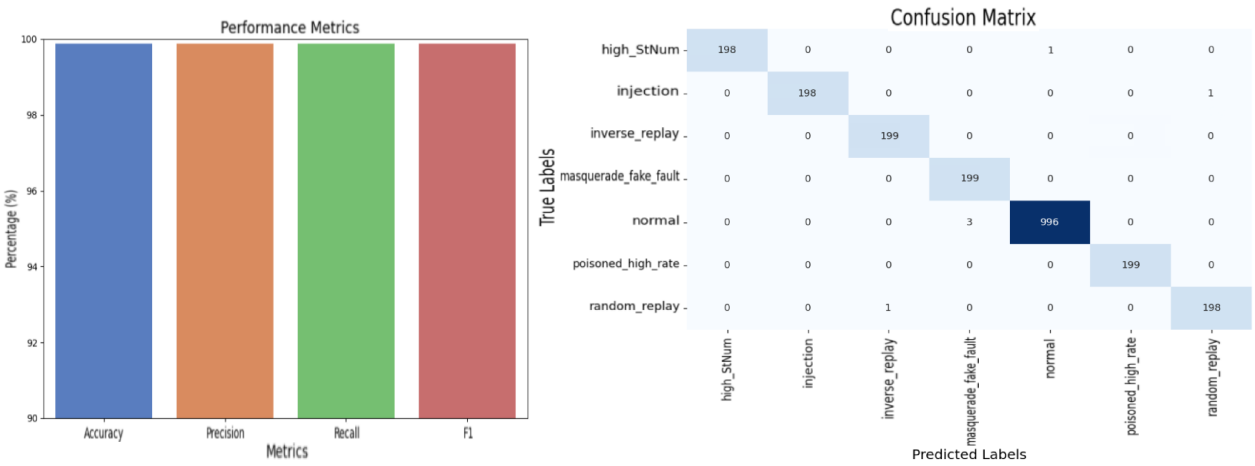



Figura 7 – Resultados da primeira análise com *Decision Tree* para o dataset de teste.

Apesar dos resultados parecerem ótimos, uma investigação cautelosa foi iniciada em relação aos ataques do tipo *DoS* por inundação ou injeção. No ataque *highStnum*, de acordo com (SILVEIRA; FRANCO, 2019), o objetivo é enviar mensagens GOOSE falsificadas com um valor alto de *StNum* (Número do Estado), que sinaliza aos dispositivos receptores a ocorrência de um novo evento (FERNANDES; BORKAR; GOHIL, 2014), superior ao das mensagens legítimas. O intuito ao disparar o ataque seria forçar o sistema a incrementar seu próprio *StNum* ao das mensagens falsificadas.

Portanto, tendo em vista o comportamento do ataque e a *feature* que se espera ser mais impactante para ele, foram exportados dois *datasets* binários com a mesma configuração e quantidade dos *datasets* multi-classe. Em seguida, após a previsão do *Decision Tree*, foi gerado o gráfico de importância global com SHAP, focando na previsão para a classe normal e atribuindo os pesos distribuídos (*Shapley values*) para quando os eventos eram de ataque. Isso revelou o que interferia na predição de um comportamento como normal, ou seja, o que reduzia a probabilidade da classe legítima. A visualização dessa forma tornou mais evidente a anomalia do ataque em relação aos eventos legítimos e destacou um aspecto preocupante.

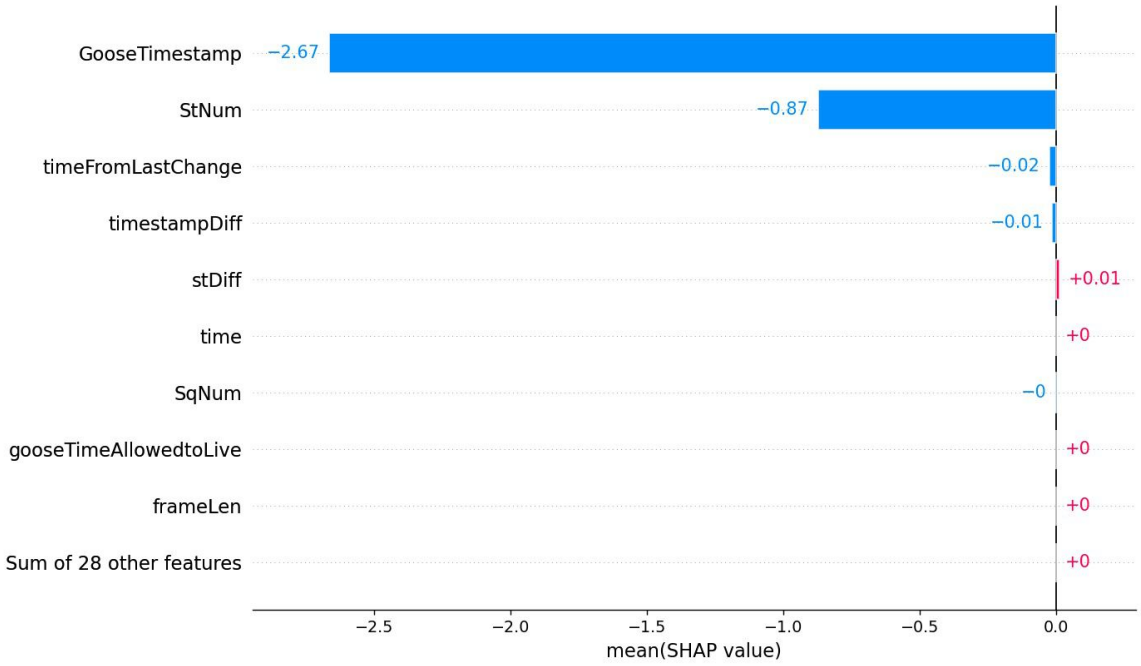



Figura 8 – Importância global das *features* no ataque *High Status Number Attack*

Dado os *Shap values* do gráfico, não é surpreendente que o *GooseTimestamp* esteja entre as variáveis mais impactantes, pois ele realmente pode indicar algum problema potencial para esse ataque. No entanto, o grau em que se destaca, sendo muito maior do que o *StNum*, sugere que a previsão foi dominada por essa *feature*. Em uma previsão desse tipo de ataque, ter um modelo que depende predominantemente de uma variável de tempo, como mostrado pelo SHAP, representa um problema potencial a ser abordado. Isso pode causar um desempenho discrepante do IDS quando for inserido em um ambiente real de subestações. Pois, aproveitou-se de idiossincrasias muito específicas dos dados de treino e teste do simulador de tráfego, resultando em previsões não tão precisas sobre o funcionamento fidedigno do ataque.

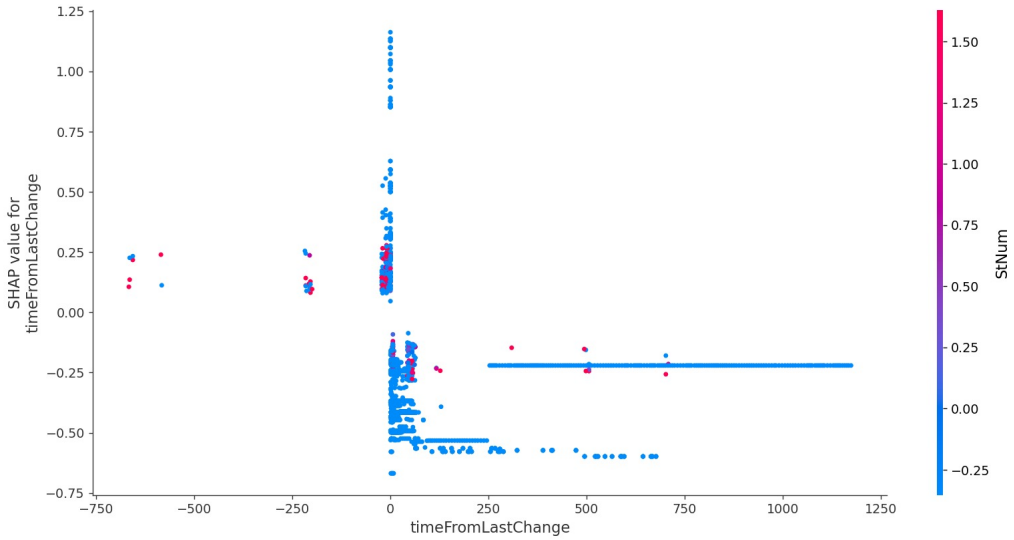



Figura 9 – Gráfico de dependência SHAP da variável TimeFromLastChange com o StNum.

Além disso, ao retomar a análise da previsão multiclasse, a Figura 9 exibe o Gráfico SHAP de dependência parcial das variáveis para as predições. Este gráfico demonstrou que valores negativos na diferença de tempo eram significativos na previsão, apontando para uma peculiaridade que pode ser mais evidente em outros ataques. Por sua vez, a Figura 10 realça a inferioridade das métricas ao predizer valores de outra simulação/geração da rede pelo ERENO UI, com a mesma quantidade de eventos, mas ignorando o *GooseTimestamp* e considerando apenas a diferença de *timestamp*. Ficou evidente que a dependência dos dados em relação ao *GooseTimestamp* não era apenas dominante para o ataque HighStnum, mas também para outros ataques.

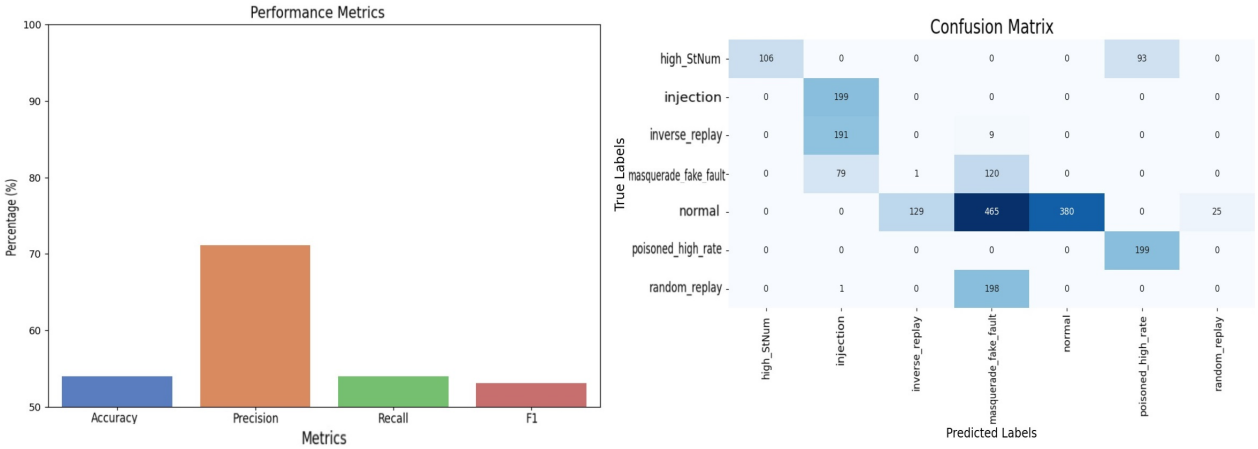



Figura 10 – Métricas e matriz de confusão para os dados de teste

Essas observações foram enviadas para a equipe do ERENO UI *Framework*, o que colaborou para que a equipe pudesse começar a correção do *bug* no próximo *path*, sendo elas principalmente a correção de diferenças de tempos negativos por falta de ordenação dos ataques em memória, durante a geração do *dataset* e possivelmente a adição do *timestamp unix epoch* aos dados temporais do *dataset*. Porém, para provar que adicionar as novas correções era algo primordial para as previsões de alguns ataques, comprovando que são realmente detalhes críticos, foi corrigido de forma paliativa diretamente no código do *path* atual do ERENO UI *Framework*, tornando possível o seu aprimoramento.

Ademais, após as correções paliativas das diferenças de tempo negativas utilizando *Python* e o incremento de *timestamp unix* na geração dos dados pelo código open-source do *Framework* utilizado, começou-se a possibilitar a extração de *features* por enriquecimento de dados de tempo. Essa técnica consiste em integrar o *dataset* a uma linha do tempo, facilitando a identificação de certos tipos de ataques, especialmente os de estilo *DoS*, tais como inundações de mensagens, injeções ou envenenamentos. Esses ataques visam comprometer o sistema, induzindo-o a falhas ou desligamento, cada um à sua maneira.

Assim sendo, adotou-se uma abordagem baseada na técnica de *Non-Overlapping Moving Window*, que divide os dados em intervalos de tempo fixos e não sobrepostos, realizando análises específicas dentro de cada intervalo. Desta forma, o conjunto de dados é segmentado em uma série temporal, na qual, neste trabalho, optou-se por intervalos heurísticos de 2 segundos. Isso permite realizar uma ou várias operações específicas dentro de cada intervalo sem sobreposição entre eles, destacando as tendências temporais relevantes.

Para melhor compreensão, pode-se elaborar matematicamente a técnica, definindo o procedimento que segmenta uma série temporal $y_t$ em intervalos fixos de tempo de 2 segundos, sem sobreposição entre os intervalos. Cada intervalo é denotado como $I_k$, onde $k$ é o índice desse intervalo. A ideia é agrupar os valores da série temporal em blocos de tempo fixos e aplicar operações ou análises dentro de cada intervalo separado. Eis uma descrição matemática da abordagem, sendo $t_1, t_2, t_3, \ldots, t_n$ uma sequência de tempos em que os dados são registrados. Há a possibilidade de definir os intervalos de tempo fixos de 2 segundos como:

$$I_k = [t_{2k}, t_{2k+1}] \text{ para } k = 0, 1, 2, \ldots$$

Onde:

- Cada $I_k$ é um intervalo de tempo de 2 segundos
- Para cada intervalo $I_k$, agrupa-se os valores da série temporal $y_t$ que estão dentro desse intervalo.

- Pode-se denotar os valores da série temporal dentro do intervalo $I_k$ como $y_{t_j}$ para $t_j \in I_k$.

Uma vez que os valores da série temporal dentro de $I_k$ estão identificados (ou seja, $y_{t_j}$ para $t_j \in I_k$ ), pode-se aplicar qualquer análise ou operação desejada dentro desse intervalo, sendo $|I_k|$ o número de pontos de dados dentro de $I_k$. Por exemplo, para calcular a média dos valores dentro de $I_k$, é possível aplicar:

$$\mathrm{f}\left(I_k\right) = \frac{1}{|I_k|} \sum_{t_j \in I_k} y_{t_j}$$

Desta forma, cinco novos atributos foram extraídos usando enriquecimento por janela de tempo: média, variância, desvio padrão, *kurtosis* e *skewness*. Cada evento carrega consigo as estatísticas de sua janela temporal. A *kurtosis* mede o achatamento da distribuição dos tempos em relação à média, indicando o quanto os dados estão concentrados; uma *kurtosis* alta sugere uma distribuição com caudas pesadas e um pico mais proeminente, enquanto uma *kurtosis* baixa indica um pico mais achatado, sugerindo uma variação menos extrema dos dados ao redor da média.

Por outro lado, a *skewness* determina a assimetria da distribuição em relação à média: uma *skewness* positiva implica que a cauda da distribuição se estende para a direita, indicando uma distribuição com inclinação positiva, enquanto uma *skewness* negativa mostra que a cauda se estende para a esquerda, caracterizando uma distribuição com inclinação negativa. Além dos atributos extraídos, um atributo chamada *dt_clock* também foi criada baseada no *GooseTimestamp*, mas em um formato granular, no intuito de substituir esse atributo, dominante nas previsões, de forma que não fosse perdido as informações úteis que ele trazia.

### 3.3.2 Normalização e Codificação dos Dados

O estágio de normalização e codificação dos dados é crucial nas etapas de processamento. Neste trabalho, são selecionadas técnicas apropriadas para colunas numéricas e categóricas, de acordo com a natureza dos dados. Para colunas numéricas, o *Scaler* é cuidadosamente definido para considerar a presença de muitos valores que, embora sejam considerados *outliers*, são essenciais para preservar as relações entre os dados originais. Isso inclui os novos recursos obtidos pela extração de atributos estatísticos das janelas de tempo.

Para colunas categóricas, foi analisado o número de valores únicos em cada uma. Por isso, o *LabelEncoder* foi aplicado exclusivamente para *feature* de classes, pois era a única com muitos valores únicos e para as demais colunas categóricas, utilizou-se o *OneHotEncoder*. Após a codificação, os dados processados são concatenados com as colunas especiais, e as

colunas originais são removidas deste *DataFrame*. Este procedimento normaliza a natureza dos dados, especialmente os categóricos, melhorando significativamente a eficácia da previsão em classificadores que se alinham bem com técnicas XAI.

**Algoritmo 1** Pré-processamento de Dados

**Função** PREPROCESS_DATA($X\_train, y\_train, X\_test, y\_test$)
**Entrada:**
$X\_train$: Conjunto de treinamento de features.
$y\_train$: Rótulos do conjunto de treinamento.
$X\_test$: Conjunto de teste de features.
$y\_test$: Rótulos do conjunto de teste.
**Saída:**
$y\_train$: Rótulos de treinamento após a codificação.
$y\_test$: Rótulos de teste após a codificação.
$X\_train$: Features de treinamento após o pré-processamento.
$X\_test$: Features de teste após o pré-processamento.
$le$: Objeto LabelEncoder utilizado para a codificação dos rótulos.
**Aplicação da Função de enriquecimento de dados Temporais:**
$time\_columns \leftarrow$ Colunas relacionadas a tempo do *Dataframe*
$X\_train\_time \leftarrow$ Aplicação do 2-TSNOMI enrichment Function em $X\_train[time\_columns]$
$X\_test_time \leftarrow$ Aplicação do 2-TSNOMI enrichment Function em $X\_train[time\_columns]$
$X\_train \leftarrow$ Concatenação $X\_train$ com features extraídas em $X\_train_time$
$X\_test \leftarrow$ Concatenação $X\_test$ com features extraídas em $X\_test_time$
**Identificação de Colunas Numéricas e Categóricas:**
$num\_cols \leftarrow$ Colunas numéricas de $X\_train$
$cat\_cols \leftarrow$ Colunas categóricas de $X\_train$
**Normalização dos Dados Numéricos:**
Inicializar MinMaxScaler
Aplicar MinMaxScaler em $X\_train[num\_cols]$
Aplicar MinMaxScaler em $X\_test[num\_cols]$
**Codificação One-Hot para Colunas Categóricas:**
**if** $cat\_cols$ não estiver vazio **then**
    Inicializar OneHotEncoder com *sparse=False* e *handle_unknown='ignore'*
    Aplicar OneHotEncoder em $X\_train[cat\_cols]$ e armazenar em $X\_train\_cat$
    Aplicar OneHotEncoder em $X\_test[cat\_cols]$ e armazenar em $X\_test\_cat$
    **Recuperação dos Nomes das Colunas Categóricas:**
    **for all** $col, categories$ em $cat\_cols, encoder.categories$ **do**
        Adicionar nomes das colunas ao $cat\_column\_names$
    **end for**
    **Criação de *DataFrames* para Dados Categóricos:**
    Criar *DataFrame* $X\_train\_cat\_df$ com colunas $cat\_column\_names$ a partir de $X\_train\_cat$
    Criar *DataFrame* $X\_test\_cat\_df$ com colunas $cat\_column\_names$ a partir de $X\_test\_cat$
    **Concatenação dos Dados Numéricos e Categóricos:**
    Concatenar $X\_train\_num\_df$ e $X\_train\_cat\_df$ em $X\_train$
    Concatenar $X\_test\_num\_df$ e $X\_test\_cat\_df$ em $X\_test$
**end if**
**Codificação de Rótulos:**
Inicializar LabelEncoder $le$
Codificar $y\_train$ utilizando $le$ se $y\_train$ for do tipo objeto
Codificar $y\_test$ utilizando $le$ se $y\_test$ for do tipo objeto
**Retorno dos Dados Pré-Processados e LabelEncoder:**
**Retornar** $y\_train, y\_test, X\_train, X\_test, le$

Portanto, com o pré-processamento definido no Algoritmo 1, novas características foram criadas tanto para o enriquecimento de dados temporais quanto para as variáveis derivadas da aplicação do *OneHotEncoder*. Especificamente, para a coluna *Protocol*, que continha os valores *GOOSE* e *SV*, a transformação resultou na criação de duas novas colunas nomeadas após esses protocolos, utilizando valores 0 ou 1. Essa abordagem melhora a capacidade do modelo de mensurar e interpretar as influências desses protocolos ao longo do tempo.

## 3.4 Treinamento do Modelo

A etapa de treinamento dos classificadores e predição dos modelos iniciou com a importação de bibliotecas *python* necessária para definição do ambiente de trabalho, tal como está descrito em (OLIVEIRA, 2024a). Após a etapa de pré-processamento, foi realizado uma classificação sob os dados do *dataset* normalizado, para 6 classificadores, sendo eles Random Forest, XGBoost, KNN, *Decision Tree*, *CatBoost Classifier* e SVM.

O objetivo inicial foi explorar as limitações de tempo de execuções, notou-se inicialmente que a etapa de pré-processamento para um *dataset* com 38 mil linhas, levava um tempo médio de 56 segundos para ser realizado, após isso foi realizado o treinamento dos classificadores observando inicialmente os tempos de execuções com o *dataset* de treino e de teste com uma quantidade de linhas de 2 mil linhas, como mostra a Figura 11 e Figura 12.

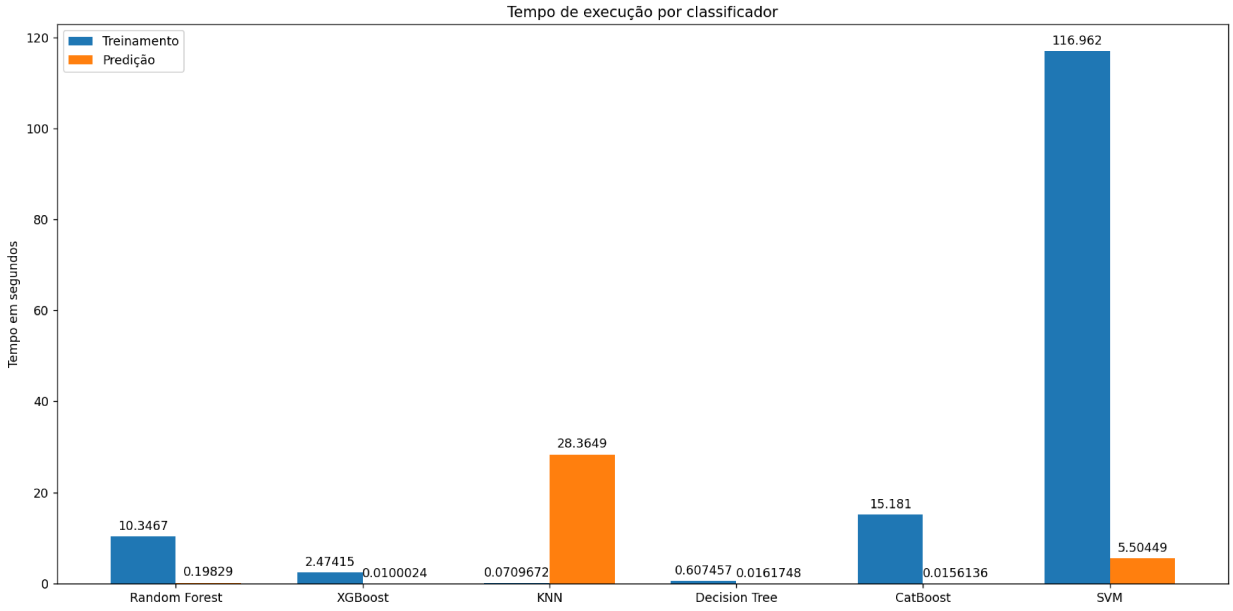



Figura 11 – Figura dos tempos de treinamento e previsão para 38 mil linhas

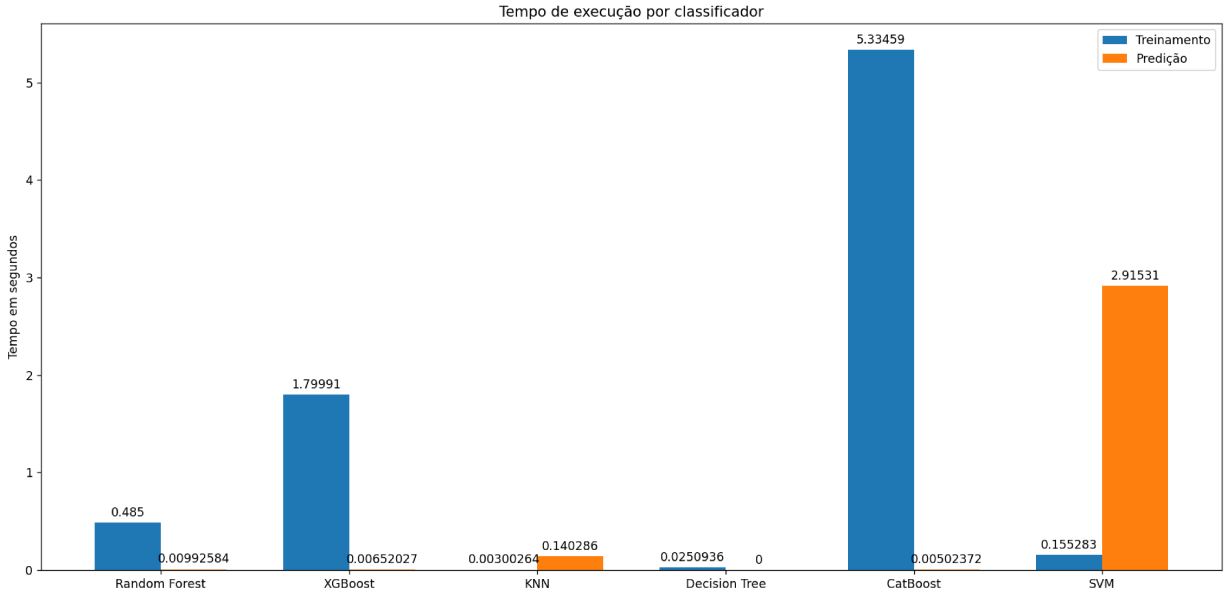



Figura 12 – Figura dos tempos de treinamento e previsão para 2 mil linhas

O próximo passo foi o ajuste de hiper-parâmetros do modelo, realizado de forma iterativa, visando otimizar sua performance a cada execução. Para isso, foi utilizado o *RandomForest* para ajuste de parâmetros ideais, por ser o algoritmo de árvore que relativamente foi mais rápido do que os demais nesse caso, possibilitando muitas alterações e testes recorrentes.

Com isso, o *XGBoost* e os demais classificadores de árvore foram configurado com os hiper-parâmetros ótimos encontrados, mas com outras etapas de ajustes próprios. Durante o ajuste de hiper-parâmetros, foram testadas diversas combinações de valores para os hiper-parâmetros do modelo, como número de árvores, profundidade máxima das árvores, taxa de aprendizado, entre outros. Após essa etapa, alguns classificadores foram descartados como o *CatBoost Regressor* e o *Naive Bayes*, por não terem tido bons resultados com as configurações encontradas, tanto em tempo quanto em resultado de métricas.

Em seguida, foi adotado as mesmas práticas de ajuste de parâmetros para os outros tipos de classificadores baseados em regressão por exemplo, em seguida os modelos foram avaliados novamente para garantir que as mudanças nos hiper-parâmetros resultaram em uma melhoria significativa na performance ao todo. Por fim, o modelo treinado com os hiper-parâmetros configurados de forma ideal foi utilizado para fazer as validações finais, se as métricas estão apresentando algum *overfitting* ou estão alinhadas com um bom ajuste de classificação.

## 3.5 Implementação do Componente XAI ao IDS

No estágio mais importante do trabalho, a extração de explicações XAI, foi realizado usando a biblioteca SHAP (*SHapley Additive exPlanations*), que fornece uma estrutura bem madura de exemplos e avanços para entender as previsões dos modelos de AM. A forma como foi feita despendeu bastante tempo de estudo a cerca de como aperfeiçoar o bastante para o tempo ideal de execução não fugir o bastante da ideia de ser uma solução rápida, apesar de não ser em tempo real, já que isso em trabalho nenhum foi alcançado, pela dificuldade de tornar isso possível já que depende muito da quantidade de dados processados e do tipo de classificação.

Após o treinamento do modelo, foi gerado um objeto *explainer* utilizando *TreeExplainer* do SHAP, passando o modelo para esse módulo. Isso permitiu calcular os valores para cada variável de entrada. Os valores SHAP representam a contribuição de cada variável para a diferença entre a saída do modelo e o valor médio de saída do conjunto de dados de treinamento.

Para aprimorar o modelo, foi usado a função *calculate shap parallel*, que paralelizou o cálculo dos valores SHAP para acelerar o processo. Isso foi necessário devido ao grande número de instâncias no conjunto de teste. Uma vez que foi obtido os valores SHAP, foi gerado vários gráficos para visualizar e interpretar as explicações do modelo. Isso incluiu gráficos de dependência para entender como uma variável afeta as previsões do modelo em relação a outra, bem como gráficos de resumo global para identificar quais variáveis são mais importantes para o modelo.

Durante o processo de geração de gráficos, foram identificados alguns problemas e soluções:

- Um dos problemas foi a correta manipulação dos valores SHAP para geração dos gráficos de dependência, importância global e força. Foi necessário garantir que os valores SHAP fossem passados corretamente para as funções de plotagem, garantindo que as dimensões dos dados estivessem corretas.
- Outro problema foi a visualização dos gráficos com grande aglomeração de pontos em um local. Uma solução seria aumentar a quantidade de pontos amostrados ou ajustar os parâmetros de plotagem para uma melhor visualização.
- Um erro comum foi a tentativa de passar o *array* de valores SHAP diretamente para as funções de plotagem, resultando em erros de tipo. A solução foi converter os valores SHAP para uma estrutura de dados adequada antes de passá-los para as funções de plotagem.

- Os tempos de execução prolongados devido a grande quantidade de dados no conjunto de dados e ao cálculo intensivo de SHAP também foi um problema, que para resolver, otimizou-se o código com ajustes de parâmetros do próprio SHAP e foi ajustado os hiper-parâmetros do modelo para reduzir a complexidade e o tempo de execução na etapa de extração de explicações.

Em resumo, o processo de extração de explicações XAI envolveu o treinamento de um modelo de detecção de intrusões, a geração de valores SHAP para explicar as previsões do modelo e a visualização desses valores em gráficos interpretáveis. O aprimoramento do modelo foi realizado buscando melhorar sua performance durante o processo. No entanto, enfrentados os diversos problemas relacionados à manipulação e visualização dos valores SHAP, foram solucionados iterativamente para obter uma interpretação mais precisa e compreensível das previsões do modelo.

## 3.6 Configuração da máquina utilizada nos testes

O computador empregado nos testes de classificação dos modelos de AM foi o Dell G15. Este computador conta com um processador Intel Core i7-13650HX, o qual possui 20 núcleos e chega a velocidades de até 5.0 GHz, disponibilizando potencial para processar grandes conjuntos de dados de modo veloz e eficiente. A memória RAM é de 16 GB DDR5, apta para manipular múltiplos processos e quantidades de dados enormes em um intervalo de tempo pequeno. além disso, o computador contou com um SSD de 512 gigabytes (GB), mas o armazenamento, contando com todos gráficos extraídos e o código, foi de aproximadamente 1.5GB.

Por fim, o laptop inclui uma placa gráfica NVIDIA GeForce RTX 3050 de 4 GB de GDDR6. Ele ajudou em testes de aceleração de operações usando o paralelismo, mas as diferenças não foram muito críticas, então se manteve o código sem paralelismo depois dessa obervação. Todo esse *setup* é operado no Windows 11, que possui suporte para todas as ferramentas de desenvolvimento utilizadas para a implementação e testes das classificações e extração de explicações SHAP.

# 4 Resultados

Este capítulo tem como objetivo mostrar os avanços obtidos com as técnicas aplicadas ao IDS com proposta de aprimora-lo. Primeiramente demonstra-se como era a natureza dos dados de tempo que estavam impactando as previsões antes e depois da aplicação da correção e depois os avanços obtidos acrescentando o enriquecimento temporal e aprimoramento da normalização dos dados. Por fim, são apresentados alguns gráficos com os resultados de métricas das classificações juntamente com os gráficos SHAP explicando as causas das previsões e em seguida a avaliação dessas explicações e colaborações adquiridas.

## 4.1 Estrutura e natureza dos dados no processo

A análise da natureza dos dados trouxe uma melhoria importante para a qualidade e integridade dos dados utilizados para treinar e avaliar o modelo. Foi revelado na observação que o problema residia na presença da diferença de tempos negativas, o que poderia estar induzindo as interpretações errôneas de alguns ataques e impactando negativamente o desempenho do modelo AM como um todo. Os Histogramas na Figura 13 e Figura 14 mostra a distribuição das métricas, ou seja a frequência em que os valores ocorrem em relação ao intervalo de valores do eixo x.

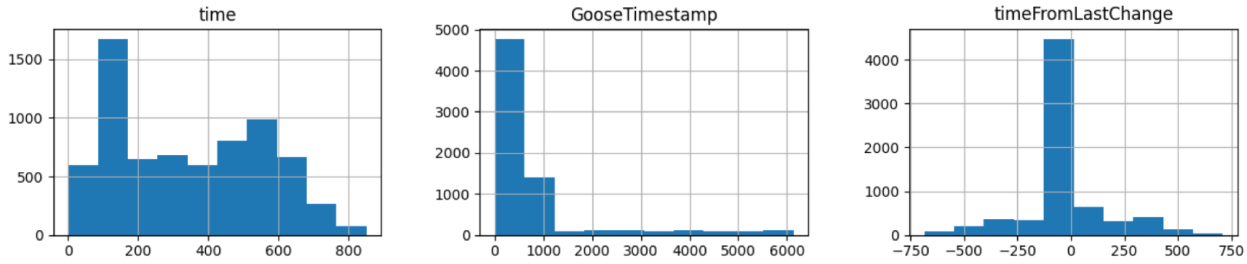



Figura 13 – Histograma dos dados de tempo irregulares do *Dataset* antes da correção

Tomando como exemplo o *"GooseTimestamp"*, pode-se concluir pelas frequências que a maioria das respostas ocorre rapidamente de 0 a 1000, com alguns eventos sendo mais demorados. Isso realmente era o esperado, tendo em vista que são a níveis de milissegundos, porém a diferença do *"time"* com *"GooseTimestamp"* não pode resultar em tempos negativos, isso poderia ser uma distribuição confusa para alguns modelos. Ao remover as diferenças de tempo negativas, conseguiu-se obter uma representação mais precisa dos intervalos de tempo

entre eventos, resultando em maior confiabilidade e consistência dos resultados das análises de AM.

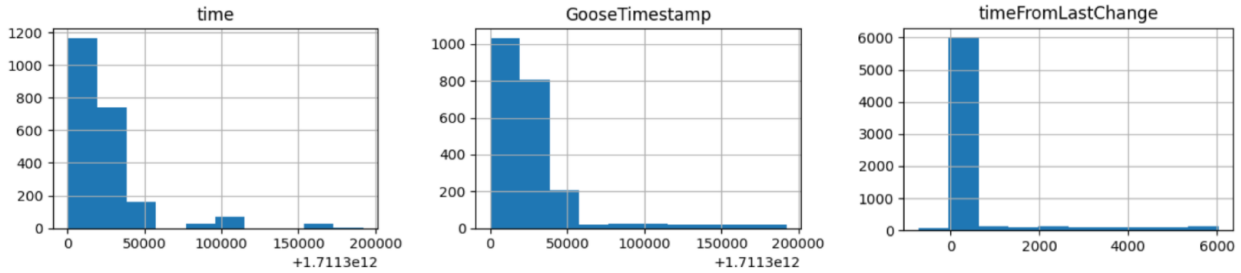



Figura 14 – Histograma dos dados de tempo do *Dataset* depois da correção

Mais importante do que isso, entretanto, essa correção melhorou a interpretabilidade das relações entre as variáveis de entrada e a variável alvo. Pois assim, criou-se a possibilidade de adicionar o enriquecimento por janelas de tempos, como está apresentado na Figura 15. Além disso, as previsões produzidas pelos modelos agora capturam melhor os padrões e o comportamento subjacente nos dados que eles visam em relação ao tempo, fornecendo assim uma fundação ainda mais sólida para análises futuras e tomada de decisões. Portanto, essa abordagem de corrigir problemas de pré processamento de dados destacou a necessidade de análise exploratória de dados e otimização de pipeline de dados em projetos de IDS.

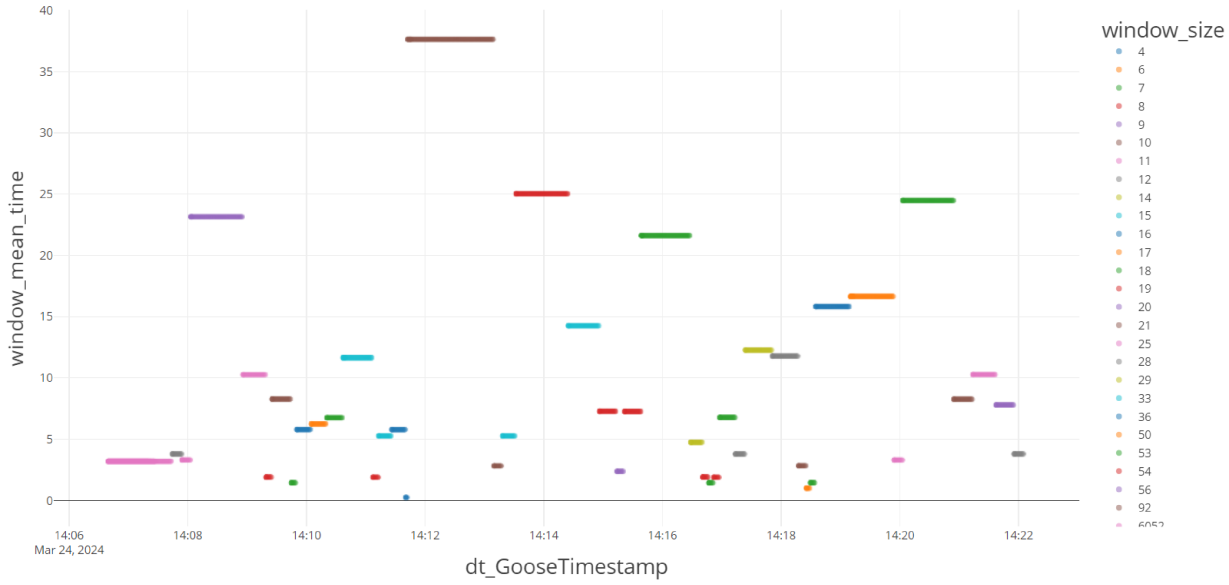



Figura 15 – Gráfico representando as médias internas de tempo em relação a cada janela

## 4.2 Avaliação das métricas e Resultados das classificações

A seguir, são apresentados os resultados obtidos a partir das classificações realizadas pelo IDS após a aplicação das técnicas de aprimoramento. Foram gerados diversos gráficos, primeiramente da classificação do IDS resultados dos cálculos da matriz de confusão, sendo as métricas de desempenho, acurácia, precisão, *recall* e *F1-Score*. A Figura 16 apresenta essas métricas e depois é evidenciada na Figura 17 com a Matriz de Confusão.

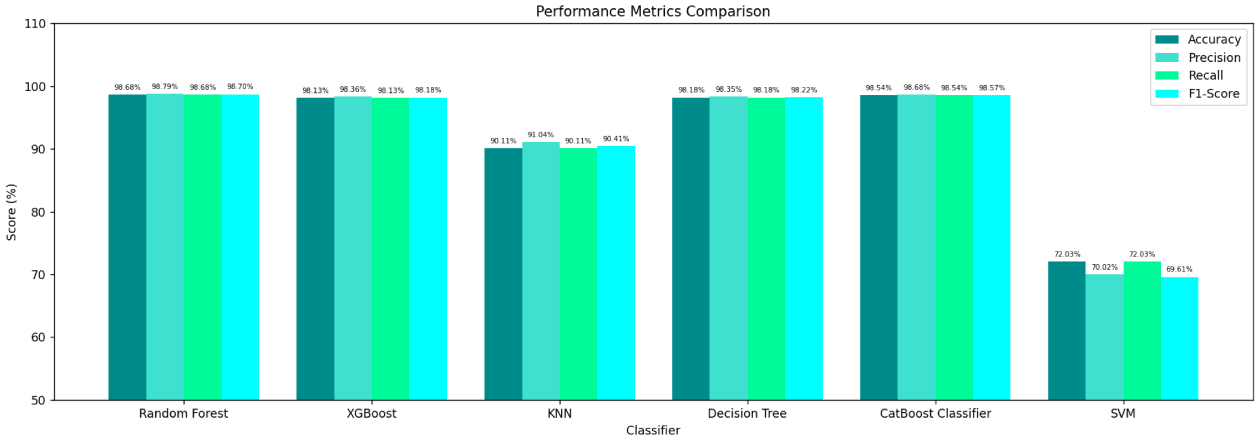



Figura 16 – Gráficos resultados métricas para os classificadores

O modelo *Random Forest*, obteve cerca de 98.68% de acurácia com precisão e *recall* próximo a esse valor. O valor do *F1-score* , que é a média ponderada da precisão e *recall* , foi de cerca de 98.70%. Já o modelo *XGBoost* obteve uma acurácia, precisão, *recall* e *F1-score* um pouco menor, na faixa média de 98.2%, 0.5% a menos que o modelo *Random Forest*.

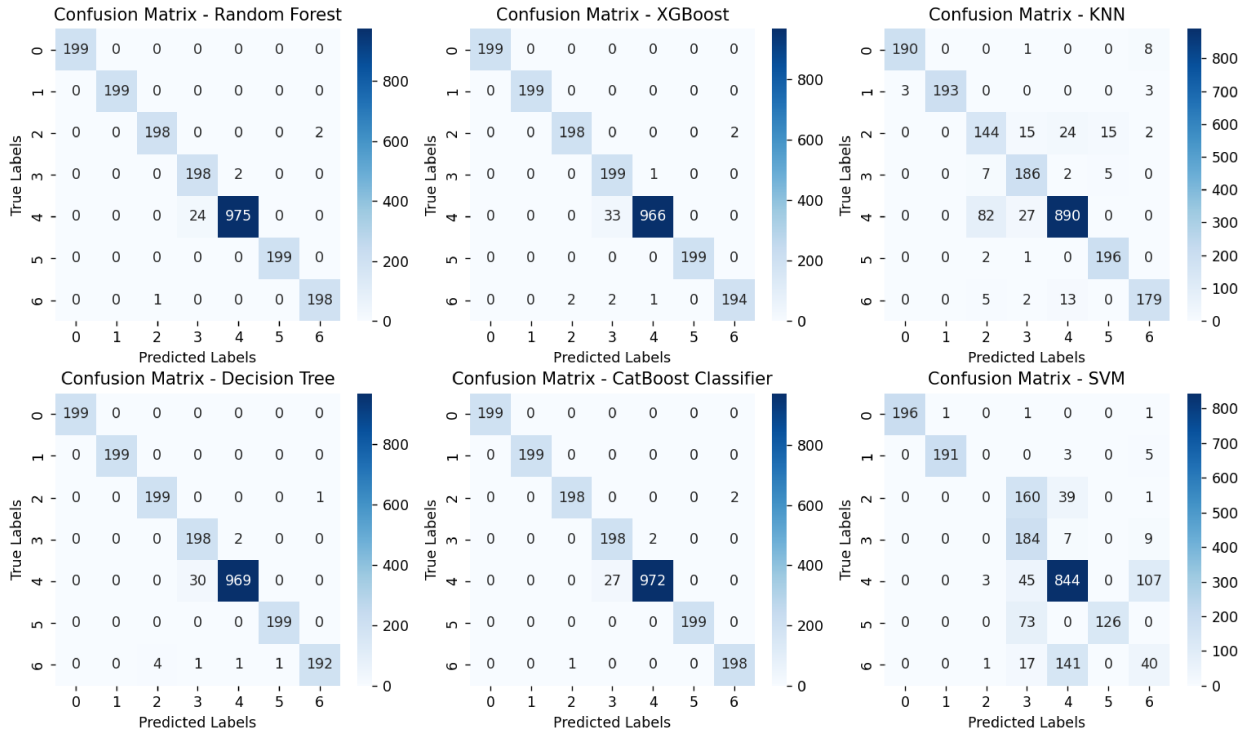
Ao considerar o classificador KNN (*K-Nearest Neighbors)*, observa-se uma média das métricas de 90,41%. Uma diferença considerável em relação aos modelos anteriores baseados em árvores, o que pode ser comparável devido à premissa do algoritmo KNN, que é baseado mais na proximidade entre os vizinhos mais próximos. Para o *Decision Tree*, os resultados são muito semelhantes ao *XGBoost*, porém um pouco melhor com valor da acurácia e *F1-score* de 98,18% e 98,22%.

Para o *CatBoost Classifier*, foi obtido uma acurácia de aproximadamente 98.54%, com precisão, *recall* e *F1-score* também próximos a esse valor, perdendo apenas para o *Random Forest*. Sendo o segundo melhor classificador dentre os utilizados, mostrando que utilizar técnicas de árvores simétricas, usando nós de decisão com a mesma condição de divisão na mesma profundidade, pode ser bastante útil.

Por fim, o modelo SVM apresentou um desempenho inferior em relação aos demais,

com uma acurácia de aproximadamente 72.03% e valores de precisão, *recall* e *F1-score* em torno de 70.02% e 69.61%, respectivamente. Isso pode ser atribuído à natureza linear do SVM, dificuldade de lidar com dados de alta dimensionalidade e à complexidade do problema em questão.

Em resumo, os modelos baseados em árvores (*Random Forest*, *XGBoost*, *Decision Tree* e *CatBoost*) apresentaram desempenho superior em comparação ao KNN e SVM, conforme evidenciado pelas métricas de avaliação (acurácia, precisão, *recall* e *F1-score*) e na Matriz de Confusão na Figura 17. Pode-se ressaltar nela as classificações exatas dos algoritmo de acordo com a natureza dos dados.




0: *high StNum*, 1: *injection*, 2: *inverse replay*, 3: *masquerade fake fault*,
4: *normal*, 5: *poisoned High Rate*, 6: *random replay*

Figura 17 – Matriz Confusão da Avaliação de Resultados

Em resumo, os resultados apresentados neste capítulo demonstram os benefícios significativos obtidos ao aplicar as técnicas de enriquecimento temporal, normalização dos dados ao IDS. As melhorias implementadas contribuíram para tornar o IDS menos tendencioso, não ocorrendo mais previsões muito perfeitas de dados mostrando claramente que o problema de viés de dados foi resolvido. A eficiência e a confiabilidade do sistema de detecção de intrusões foi fortalecido, obtendo mais capacidade de identificar e responder a ameaças cibernéticas de forma realista e precisa.

## 4.3 Gráficos SHAP para Explicação das Predições dos Ataques

Nesta seção mostra-se os benefícios de aprimoramentos adquiridos com a extração de gráficos explicativos utilizando técnicas *Explainable* AI com o SHAP, para extrair formas de compreensão intuitiva e valiosos sobre as previsões. Os tipo de gráficos adiante destaca as contribuições das variáveis de entrada de diferentes formas, seja para a previsão global ou entre a forte relação de uma variável com outra no modelo. É apresentado os principais tipos de gráficos escolhidos para análise no trabalho para cada ataque, excluindo aqueles para KNN e SVM, por não terem suporte na versão 0.44 do SHAP, utilizada no trabalho.

Iniciando com gráficos SHAP de impacto global das variáveis*(summary_plot)* para os 6 principais ataques do ERENO e para cada classificador suportados pelo SHAP, sendo eles gerados após o treinamento e previsões do modelo, já discutidas anteriormente. Com isso, foi compreendido como diferentes características influenciam de forma global a classificação de cada ataque para cada evento na rede de subestações elétricas que operam sob o protocolo IEC 61850 simulada pelo *Framework Ereno*.Neste tipo de gráfico, as barras representam a média do valor absoluto dos valores SHAP para cada característica, indicando o impacto global da variável para cada ataque.

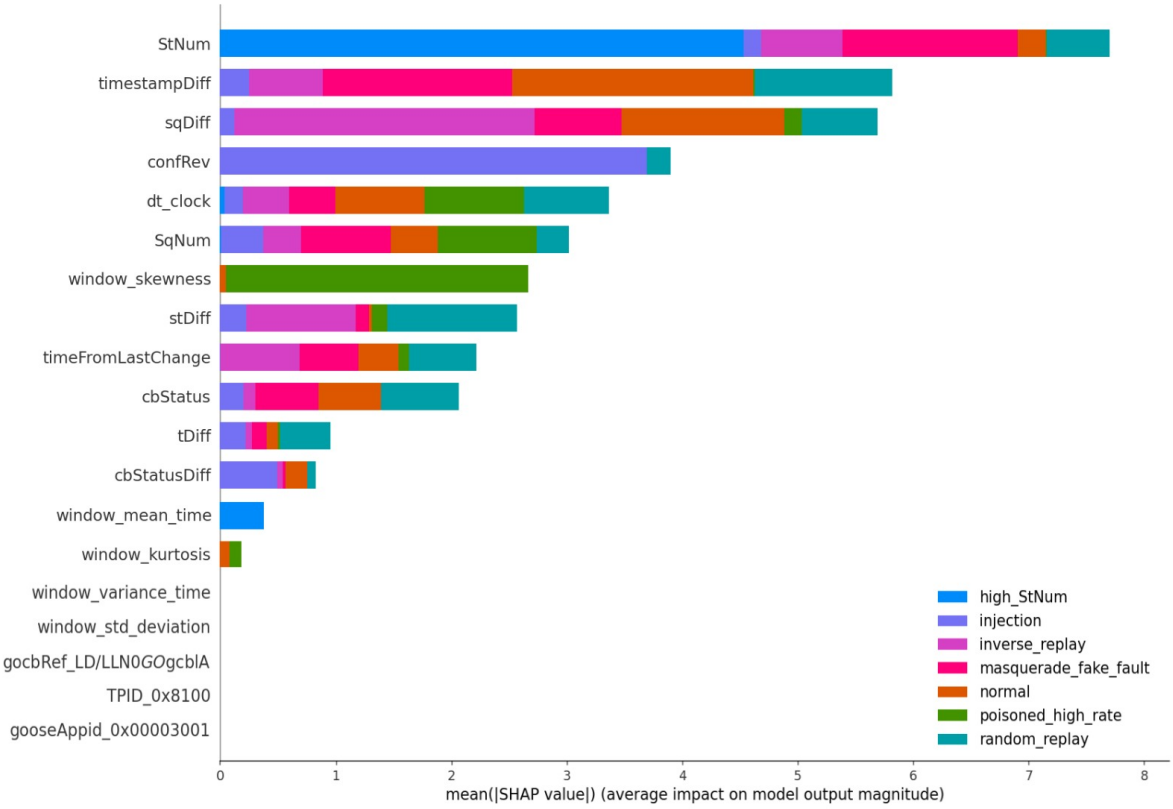



Figura 18 – Importância global das variáveis na Avaliação de Resultados no *XGBoost*

Na Figura 18, é possível notar notar que, para o classificador *XGBoost*, as novas *features* extraídas do enriquecimento de dados realmente contribuíram na classificação dos ataques. Os principais ataques que tiveram uma melhora na classificação nesse modelo foram justamente os que operam como classes *DoS*, que buscam diminuição do desempenho, sobrecarga e indisponibilidade de serviços da subestação. Agora, são sutilmente melhor percebidos pelas observações a mais sobre a assimetria e média de tempo.

Lembrando que os resultados são após a identificação e correção paliativa do *bug* de formato do *timestamp* e diferença negativa. Pode-se notar que nas Figuras 18, Figura 19, Figura 20 e Figura 21 as *features* originadas do enriquecimento de série temporal realizado, principalmente *window_kurtosis*, *window_mean_time* e *window_skewness*, foram de grande valor para a previsão dos modelos utilizados e suportados na versão do SHAP.

Inicialmente foi retirado o time e *GooseTimestamp features*, para que prevalecesse os valores das *features* de diferença de tempo e do *dt_clock* que contém a data em formato granular em escala menor. Tanto as barras do gráfico na Figura 18 quanto nas barras da Figura 19 demonstra o que era esperado, um benefício da média aplicada às janelas de tempo(window_mean_time) em relação ao ataque *high_Stnum*. Além disso, para os classificadores nas Figuras 19, Figura 20 e Figura 21 o ataque *Masquerade* passou a ser melhor identificado pela nova *feature window_kurtosis* que ajuda a revelar a assimetria ou forma de dispersão do tempo da janela em relação a média dela.

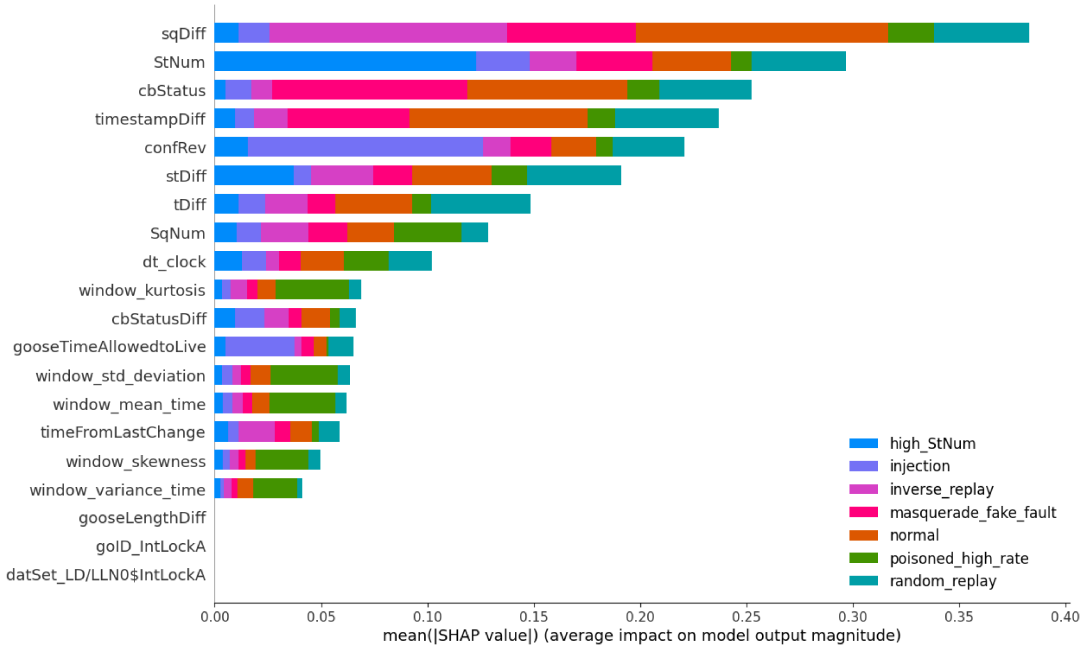



Figura 19 – Importância global das variáveis na Avaliação de Resultados no *Random Forest*

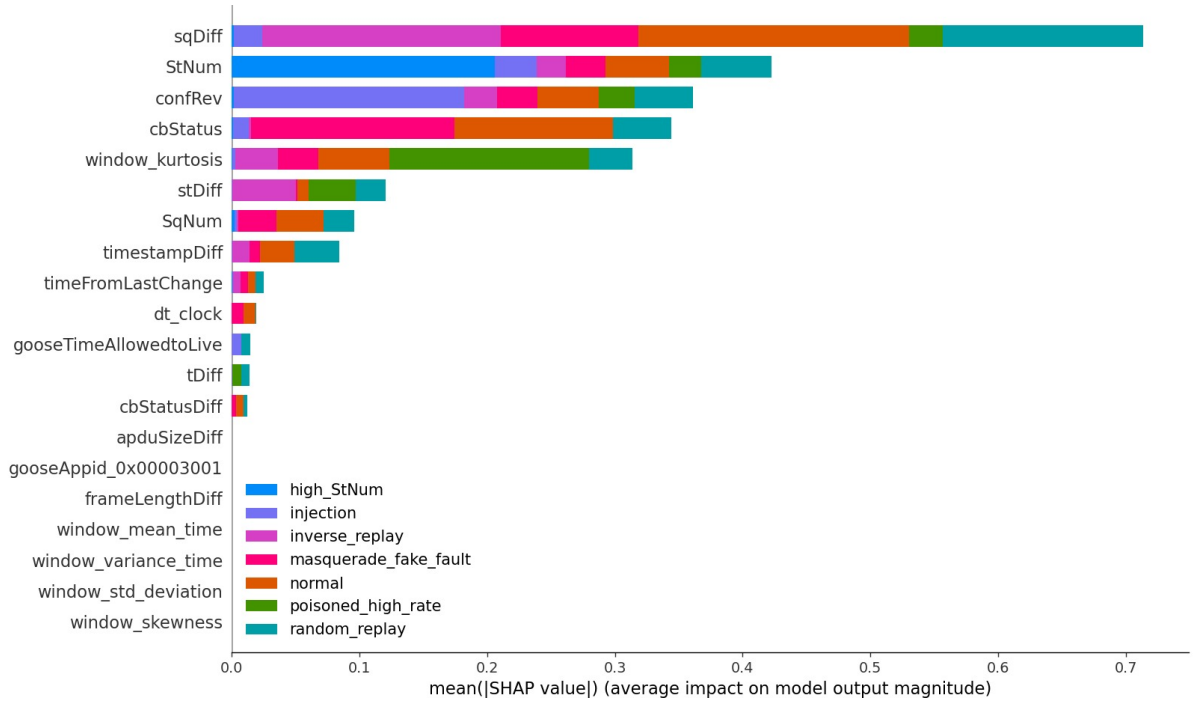



Figura 20 – Importância global das variáveis na Avaliação de Resultados no *Decision Tree*

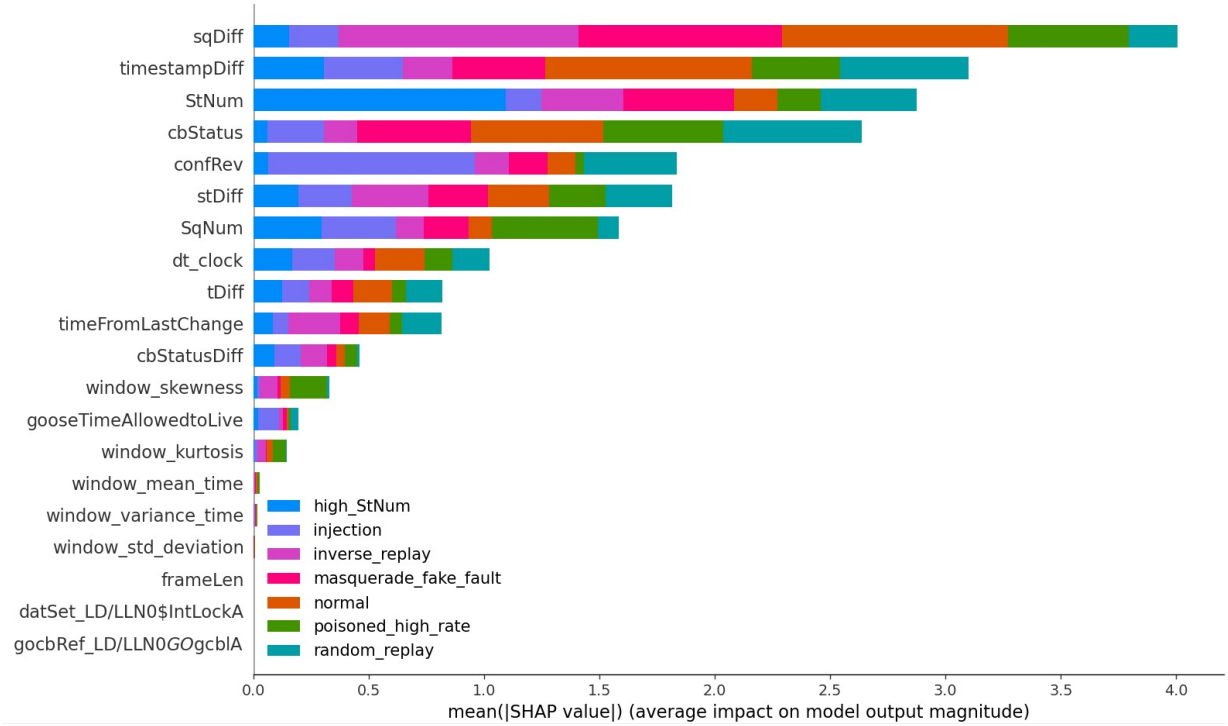



Figura 21 – Importância global das variáveis na Avaliação de Resultados no *CatBoost Classifier*

Para que se possa visualizar a relação entre um recurso a outro, é possível traçar os gráficos de dependência parcial. Um dos tipos de gráficos que se pode usar para extrair explicações da previsão, podendo trazer uma interpretabilidade a mais ao modelo, sendo bastante útil para entender como o modelo se comporta em relação a determinadas *features*. Portanto, para compreender o comportamento do modelo em relação a determinados recursos, os gráficos SHAP *dependence_plot* foram gerados a partir da combinação das variáveis com maior impacto identificadas em comum em todos os gráficos de importância Global, mas apenas para os ataques *Masquerade*, pois ele é o mais difícil de identificar comportamentos anormais.

Os gráficos contém várias informações sobre como essas variáveis interagem e influenciam as previsões do modelo de detecção desse ataque. Mas antes, para esclarecimento, os pontos estão coloridos de acordo com o valor da variável y, que está normalizado entre 0 e 1. A cor dos pontos varia do azul (valores baixos de y) ao vermelho (valores altos de y) e o eixo Y em si representa o impacto do valor SHAP para Ataque ou não. Os valores positivos de SHAP no eixo Y indicam uma maior probabilidade de ser um ataque, enquanto valores negativos indicam uma menor probabilidade.

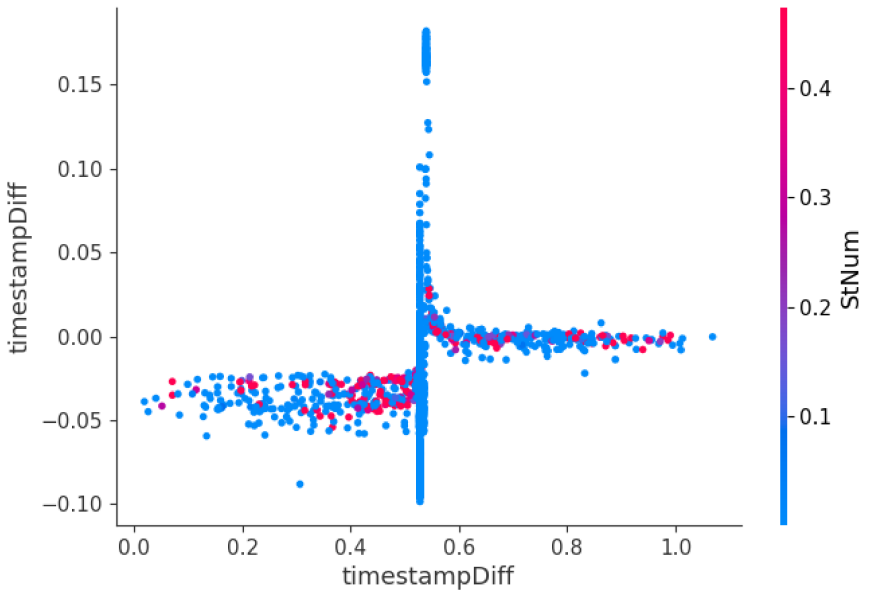



Figura 22 – Relação das variáveis *timestampDiff* com *StNum*

Interpretando a Figura 22, gerada pelo *Random Forest* para a classe de ataque *Masquerade*, percebe-se claramente a intenção do ataque sendo realizado. Há uma linha vertical

de pontos com valores SHAP positivos significativos quando o *timestampDiff* é próximo de zero em relação aos valores SHAP. Isso sugere que, quando há uma mudança de estado (*StNum* aumenta) sem um aumento correspondente no intervalo de tempo entre as mensagens (*timestampDiff* baixo), o modelo interpreta isso como um possível ataque *Masquerade*, mostrando que o invasor desse ataque tenta imitar o tráfego legítimo com mudanças rápidas de estado.

Na Figura 23, na previsão do Modelo no ataque *Masquerade*, foi melhor evidenciado a predição de ataque, mostrando que, quando a *kurtosis* da janela é alta em relação a uma média baixa do tempo da janela, o modelo conseguiu prever mais ataques *Masquerade*. Isso demonstra que ele se comporta realmente em janelas de média de tempo comum, mas que, com a adição da *kurtosis*, foi possível verificar que existia distribuição anormal, sugerindo que existia valores extremos agrupados em torno da média interna da janela, levando a uma melhora na interpretação desse ataque.

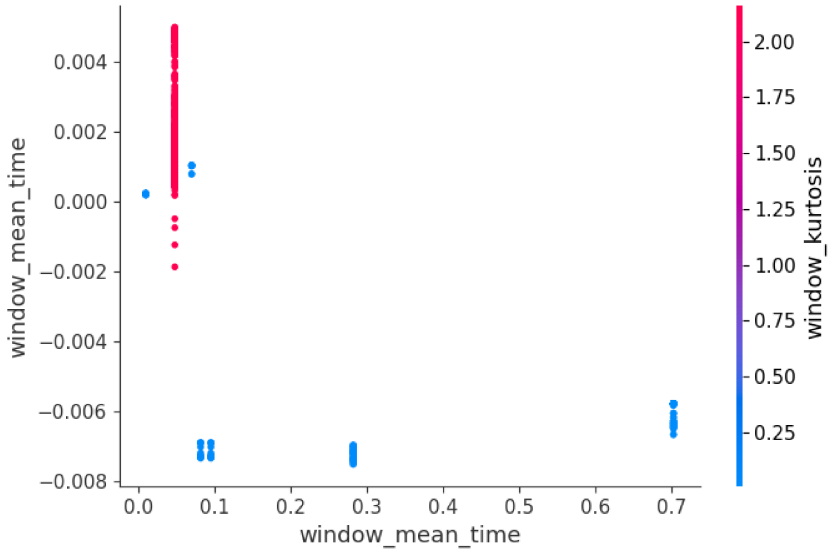



Figura 23 – Relação de dependência parcial das variáveis do enriquecimento temporal

Em síntese, existem 7 classes, 6 ameaças e uma normal, além de 4 modelos com suporte no SHAP e por fim 5 recursos de tempo mais impactantes e comuns entre a previsão de cada modelo, sendo analisada a cada 12 recursos mais comuns na previsão desde *StNum* até *StDiff*, entre outros. Tudo isso totalizou um total de $5 \times 12 \times 7 \times 4 = 1680$ gráficos de dependência e um total de 4 gráficos de importância global e de força das variáveis, que expressam a mesma ideia em formatos diferentes, todos estão disponibilizados em (OLIVEIRA, 2024b), além do código em (OLIVEIRA, 2024a), da implementação do aprimoramento do IDS.

# 5 Conclusão

Em conclusão, este estudo demonstrou o avanço substancial em usar as técnicas de extração de Explicações com técnicas XAI, justamente para dar confiabilidade e tornar o IDS uma caixa transparente, dando assim direção para saber o que está causando problemas. Como foi o caso, possibilitando correções ao *Framework* que gera o *dataset*, fornecendo meios para o entendimento das relações dos dados, gerando então tomadas de decisões mais assertivas, como a extração de dados enriquecidos através de análise de séries temporais e extração de novas características por meio de classes relacionadas.

A comparação dos dados de natureza anterior e posterior aos processos temporais de enriquecimento e normalização, destacou após a aplicação do aperfeiçoamento das técnicas, nos resultados de classificação do IDS, que houve uma melhoria considerável na precisão e confiabilidade das previsões. Indicadores de desempenho, como taxa de precisão, acurácia, *recall* e *F1-score* mostraram um aumento significativo e coerente após as modificações.

Os gráficos SHAP utilizados para explicar as previsões do modelo forneceram *insights* valiosos sobre as contribuições individuais e globais das variáveis de entrada para as decisões do IDS. A possibilidade agora de uma análise interpretativa no processo, contribuiu para uma compreensão mais detalhada dos padrões e comportamentos de ataques, tornando eles relevantes para a detecção de ameaças cibernéticas.

Além disso, os resultados obtidos em relação a eficácia das técnicas de *feature extraction* de série temporal, na otimização do IDS para lidar com ambientes dinâmicos e complexos, foi algo crucial nesse aprimoramento. Recomenda-se que futuros trabalhos explorem ainda mais técnicas de enriquecimento de dados e principalmente paralelismo, para executar técnicas de XAI em tempo real, além de investigar sua aplicação em diferentes cenários e contextos de segurança cibernética.

Em suma, os avanços alcançados neste estudo contribuíram para o fortalecimento e aprimoramento contínuo dos sistemas de detecção de intrusões em Subestações Elétricas. Capacitando o setor elétrico a agir de forma eficiente com a ajuda da IA ao processo de investigações dos motivos das previsões e os capacitando para melhoria contínua.

# Referências

BACE, R. G.; MELL, P. et al. Intrusion detection systems. US Department of Commerce, Technology Administration, National Institute of . . . , 2001. Citado na página 13.

BAIGENT, D.; ADAMIAK, M.; MACKIEWICZ, R.; SISCO, G. Iec 61850 communication networks and systems in substations: An overview for users. **SISCO Systems**, Citeseer, 2004. Citado 2 vezes nas páginas 15 e 17.

BIERMANN, E.; CLOETE, E.; VENTER, L. M. A comparison of intrusion detection systems. **Computers & Security**, Elsevier, v. 20, n. 8, p. 676–683, 2001. Citado na página 13.

CONFALONIERI, R.; COBA, L.; WAGNER, B.; BESOLD, T. R. A historical perspective of explainable artificial intelligence. **Wiley interdisciplinary reviews. Data mining and knowledge discovery**, Wiley Periodicals, Inc, Hoboken, USA, v. 11, n. 1, p. e1391–n/a, 2021. ISSN 1942-4787. Citado na página 19.

DAVENPORT, T. H. **The AI advantage: How to put the artificial intelligence revolution to work**. [S.l.]: mit Press, 2018. Citado na página 9.

DOSHI-VELEZ, F.; KIM, B. **Towards A Rigorous Science of Interpretable Machine Learning**. 2017. Citado na página 11.

FERNANDES, C.; BORKAR, S.; GOHIL, J. Testing of goose protocol of iec61850 standard in protection ied. **International journal of computer applications**, Citeseer, v. 93, n. 16, 2014. Citado na página 30.

GAUSHELL, D.; DARLINGTON, H. Supervisory control and data acquisition. **Proceedings of the IEEE**, v. 75, n. 12, p. 1645–1658, 1987. Citado na página 14.

HOYOS, J.; DEHUS, M.; BROWN, T. X. Exploiting the goose protocol: A practical attack on cyber-infrastructure. In: **2012 IEEE Globecom Workshops**. [S.l.: s.n.], 2012. p. 1508–1513. Citado na página 16.

IEC, T. Communication networks and systems in substations. **IEC61850**, 2003. Citado 2 vezes nas páginas 15 e 17.

ILASCU, I. **Eletrobras, Copel Energy Companies Hit by Ransomware Attacks**. 2021. BleepingComputer. Accessed: date-of-access. Disponível em: <https://www.bleepingcomputer.com/news/security/eletrobras-copel-energy-companies-hit-by-ransomware-attacks/>. Citado na página 11.

KSHETRI, N.; VOAS, J. Hacking power grids: A current problem. **Computer**, v. 50, n. 12, p. 91–95, 2017. Citado na página 11.

KUZLU, M.; CALI, U.; SHARMA, V.; GüLER, Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. **IEEE Access**, v. 8, p. 187814–187823, 2020. Citado 2 vezes nas páginas 24 e 26.

LEE, W.; STOLFO, S. J.; MOK, K. W. Adaptive intrusion detection: A data mining approach. **Artificial Intelligence Review**, Springer, v. 14, p. 533–567, 2000. Citado na página 13.

LU, X.; WANG, W.; MA, J. Authentication and integrity in the smart grid: An empirical study in substation automation systems. **International Journal of Distributed Sensor Networks**, v. 2012, 06 2012. Citado na página 28.

LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. **Advances in neural information processing systems**, v. 30, 2017. Citado 2 vezes nas páginas 19 e 21.

MATOUŠEK, P. Description of iec 61850 communication. In: **Technical Report**. [S.l.]: Brno University of Technology, 2018. Citado na página 16.

MCDONALD, J. D. **Electric power substations engineering**. [S.l.]: CRC press, 2003. Citado na página 14.

MENDES, C.; BORTOLI, F.; COSTA, C. Industria 4.0 a digitalizaÇÃo da manufatura: Um caso de estudo industry 4.0 the digitalization of manufacturing: A case study. 04 2021. Citado na página 14.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw-Hill Education, 1997. Citado na página 21.

MOLNAR, C. **Interpretable Machine Learning**: A guide for making black box models explainable. 2. ed. [s.n.], 2022. Disponível em: <https://christophm.github.io/interpretable-ml-book>. Citado 4 vezes nas páginas 9, 11, 19 e 20.

MUNIR, M. S.; SHETTY, S.; RAWAT, D. B. **Trustworthy Artificial Intelligence Framework for Proactive Detection and Risk Explanation of Cyber Attacks in Smart Grid**. 2023. Citado 2 vezes nas páginas 25 e 26.

NEUPANE, S.; ABLES, J.; ANDERSON, W.; MITTAL, S.; RAHIMI, S.; BANICESCU, I.; SEALE, M. Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities. **IEEE Access**, v. 10, p. 112392–112415, 2022. Citado na página 23.

OLIVEIRA, H. **Projeto de TCC: Implementação de um Sistema de Detecção de Intrusão Utilizando Explicabilidade de IA**. [S.l.]: GitHub, 2024. <https://github.com/Henriqw3/tcc-ereno-xai-ids>. Citado 2 vezes nas páginas 36 e 48.

_____. **Resultado dos Classificadores**. 2024. Dados dos resultados disponível em Google Drive. Disponível em: <https://drive.google.com/drive/folders/1LmckCV-fmZ-wRatbuaC1ZD-L57C4Ct39?usp=sharing>. Citado na página 48.

QUINCOZES, S. E.; ALBUQUERQUE, C.; PASSOS, D.; MOSSÉ, D. A survey on intrusion detection and prevention systems in digital substations. **Computer Networks**, Elsevier, v. 184, p. 107679, 2021. Citado na página 9.

QUINCOZES, S. E.; ALBUQUERQUE, C.; PASSOS, D.; MOSSé, D. Ereno: A framework for generating realistic iec–61850 intrusion detection datasets for smart grids. **IEEE Transactions on Dependable and Secure Computing**, p. 1–15, 2023. Citado 4 vezes nas páginas 10, 18, 27 e 29.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, 1959. Citado na página 21.

SHAPLEY, L. S. et al. A value for n-person games. Princeton University Press Princeton, 1953. Citado na página 19.

SILVEIRA, M. G. da; FRANCO, P. H. Iec 61850 network cybersecurity: Mitigating goose message vulnerabilities. In: **Proc. 6th Annual PAC World Americas Conf**. [S.l.: s.n.], 2019. p. 1–9. Citado na página 30.

SIVAMOHAN, S.; SRIDHAR, S.; KRISHNAVENI, S. Tea-ekho-ids: An intrusion detection system for industrial cps with trustworthy explainable ai and enhanced krill herd optimization. **Peer-to-Peer Networking and Applications**, Springer, v. 16, n. 4, p. 1993–2021, 2023. Citado 2 vezes nas páginas 24 e 26.

SOLOMIN, E.; TOPOLSKY, D. Arrangement of data exchange between adaptive digital current and voltage transformer and scada-system under iec 61850 standard. **Procedia Engineering**, v. 129, p. 207–212, 2015. ISSN 1877-7058. International Conference on Industrial Engineering (ICIE-2015). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877705815039181>. Citado na página 18.

TANENBAUM, A. S. **Computer networks**. [S.l.]: Pearson Education India, 2003. Citado na página 15.

TIMES, H. **WannaCry Ransomware: Bengal Power Distribution Company Hit by Cyberattack**. 2017. Online. Available at: <https://www.hindustantimes.com/india-news/wannacry-ransomware-bengal-power-distribution-company-hit-by-cyberattack-say-officials/story-biqMQN5cPKng36cIyho2oJ.html>. Citado na página 11.

TURNER, C. R.; FUGGETTA, A.; LAVAZZA, L.; WOLF, A. L. A conceptual basis for feature engineering. **Journal of Systems and Software**, Elsevier, v. 49, n. 1, p. 3–15, 1999. Citado na página 22.

VAINIO-PEKKA, H.; AGBESE, M. O.-O.; JANTUNEN, M.; VAKKURI, V.; MIKKONEN, T.; ROUSI, R.; ABRAHAMSSON, P. The role of explainable ai in the research field of ai ethics. **ACM Transactions on Interactive Intelligent Systems**, ACM New York, NY, v. 13, n. 4, p. 1–39, 2023. Citado na página 19.

WANG, M.; ZHENG, K.; YANG, Y.; WANG, X. An explainable machine learning framework for intrusion detection systems. **IEEE Access**, v. 8, p. 73127–73141, 2020. Citado 2 vezes nas páginas 23 e 26.

YOUSSEF, T. A.; HARIRI, M. E.; BUGAY, N.; MOHAMMED, O. Iec 61850: Technology standards and cyber-threats. In: IEEE. **2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)**. [S.l.], 2016. p. 1–6. Citado na página 9.

ZETTER, K. The ukrainian power grid was hacked again. **Motherboard**, 2017. Citado na página 11.

ZHAO, P. **IEC 61850-9-2 Process Bus Communication Interface for Light Weight Merging Unit Testing Environment**. 2012. Citado na página 17.

ZHENG, A.; CASARI, A. **Feature engineering for machine learning: principles and techniques for data scientists**. [S.l.]: "O'Reilly Media, Inc.", 2018. Citado na página 22.

ZOLANVARI, M.; YANG, Z.; KHAN, K.; JAIN, R.; MESKIN, N. Trust xai: Model-agnostic explanations for ai with a case study on iiot security. **IEEE internet of things journal**, IEEE, v. 10, n. 4, p. 2967–2978, 2021. Citado na página 26.