Categorização de Incidentes de Segurança utilizando Engenharia de *Prompts* em LLMs

Alex Sandre Pinheiro Severo[1], Douglas Paim Lautert[1], Diego Kreutz[1], Leandro Márcio Bertholdo[2], Marcio Pohlmann[1], Silvio Ereno Quincozes[1]

[1] AI Horizon Labs - Universidade Federal do Pampa (Unipampa)
[2]Universidade Federal do Rio Grande do Sul (UFRGS)
[3]Universidade Federal de Uberlândia (UFU)

{alexsevero.aluno,douglaslautert.aluno,marciopohlmann.aluno}@unipampa.edu.br
{diegokreutz,silvioquincozes}@unipampa.edu.br,leandro.bertholdo@ufrgs.br

Resumo. *A crescente complexidade e volume de incidentes de cibersegurança têm gerado grandes quantidades de dados não estruturados, dificultando sua triagem por equipes humanas. Este trabalho propõe o uso de Engenharia de Prompts aplicada a LLMs para a categorização automatizada desses incidentes. A metodologia foi testada em um conjunto de dados reais e anonimizados, avaliando a consistência das classificações em diferentes cenários: categorização livre, categorização guiada por taxonomia (NIST), com e sem refinamento progressivo dos prompts. Os resultados indicam que a combinação entre Progressive-hint Prompting (PHP) e o uso de taxonomia estruturada favorece a normalização semântica, reduz a ambiguidade e melhora a confiabilidade das classificações, com alto grau de assertividade na categorização de incidentes.*

1. Introdução

O aumento no volume e complexidade dos incidentes de cibersegurança tem gerado um número expressivo de notificações, pressionando as equipes de resposta a atuarem com maior eficiência. Apenas em 2024, o Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br) registrou 516.556 notificações de incidentes [CERT.br 2025], o que corresponde a uma média de aproximadamente 1.411 incidentes por dia. Este cenário evidencia a necessidade de soluções escaláveis que auxiliem no processo de triagem, categorização e priorização destes eventos de segurança.

A categorização de incidentes de segurança permite uma identificação estruturada dos tipos de ataques, a detecção de padrões e ameaças recorrentes, bem como uma compreensão mais aprofundada da diversidade dos eventos, favorecendo análises estratégicas e a melhoria contínua dos processos de defesa. No entanto, é comum haver ambiguidade de determinados relatos de incidentes, falta de padronização nas categorias adotadas pelas instituições e limitação de recursos humanos especializados. Como consequência desses problemas, os times de segurança tendem a priorizar ações corretivas imediatas, como a erradicação e a recuperação, em detrimento do registro estruturado e da classificação detalhada dos incidentes [Grispos et al. 2019].

Nesse contexto, o uso de soluções automatizadas para filtrar e analisar dados de eventos adversos, configura-se como uma estratégia promissora para acelerar a categorização de incidentes e ampliar a eficiência operacional dos processos de resposta

[Ogundairo and Broklyn 2024]. Contudo, os métodos automatizados enfrentam desafios relevantes: a escassez de dados rotulados, a ambiguidade semântica dos textos de incidentes, a definição de categorias e a variabilidade nos formatos e padrões de ataques [Ibrishimova 2019].

Estudos recentes indicam que abordagens híbridas, que combinam o conhecimento especializado humano com técnicas computacionais avançadas, têm potencial para superar estas limitações, tornando a classificação de incidentes mais confiável, contextualizada e útil para a tomada de decisão [Grispos 2016]. Entre essas abordagens, destacam-se os *Large Language Models* (LLMs), capazes de processar texto não estruturado, correlacionar informações e gerar respostas coerentes com alta velocidade, facilitando decisões rápidas e estratégicas [F5 Networks 2024]. Técnicas como a Engenharia de *Prompts* (*Prompt Engineering*) [IBM 2024] viabilizam a customização do comportamento desses modelos por meio da formulação estratégica de *prompts*, instruções textuais que orientam a geração de respostas com maior consistência semântica.

Frameworks como o NIST SP 800-61 oferecem diretrizes para categorizar incidentes de segurança, mas ainda há desafios na integração com modelos de linguagem, devido à ambiguidade dos termos e variações entre organizações. Isso dificulta a padronização e o uso de técnicas de processamento de linguagem natural (NLP), exigindo abordagens que conciliem dados textuais ricos com categorias precisas. [Lukwaro et al. 2024] A heurística *Progressive-Hint Prompting* (PHP) [Zheng et al. 2023] busca melhorar a precisão por meio de refinamento iterativo dos *prompts*.

Este trabalho busca investigar essa integração, combinando heurísticas progressivas com taxonomias estruturadas. Aplicando a heurística PHP na categorização automatizada de incidentes de um CSIRT (*Computer Security Incident Response Team*), o refinamento de *prompts* foi avaliado em modelos comerciais de LLM, testados em sua capacidade de categorizar livremente estes incidentes, ou orientado por taxonomias padronizadas, como as sugeridas pelo NIST. A análise compara diferentes estratégias de *prompting* e destaca o impacto da utilização dessas normativas como referências.

O restante deste artigo está organizado em 5 seções: Fundamentação Teórica (Seção 2), Trabalhos Relacionados (Seção 3), Metodologia (Seção 4), Resultados e Discussão (Seção 5) e Considerações Finais e Trabalhos Futuros (Seção 6).

2. Fundamentação Teórica

Esta seção aborda a Categorização de Incidentes de Segurança (Subseção 2.1), examinando as principais taxonomias e modelos de classificação utilizados na área, e a Engenharia de *Prompts* para Categorização de Incidentes (Subseção 2.2), explorando as técnicas de *design* e otimização de *prompts*, conceitos essenciais para a compreensão e validação dos resultados experimentais apresentados neste artigo.

2.1. Categorização de Incidentes de Segurança

Existem diversas normas e *frameworks* amplamente reconhecidos que propõem diretrizes para a classificação e categorização de incidentes de segurança da informação. Entre os mais utilizados, destaca-se a NIST SP 800-61 [Cichonski et al. 2012], e a taxonomia da *European Union Agency for Network and Information Security* (ENISA), que

fornece uma base conceitual útil para a harmonização da comunicação sobre ameaças no contexto europeu. [ENISA 2018]

O trabalho [Kim and Kwon 2022] concentra-se na otimização de modelos de classificação de ameaças para SIEM, superando as limitações das abordagens manuais e baseadas em assinaturas. A pesquisa avaliou a acurácia e, crucialmente, a eficiência de modelos de *deep learning* (CNN, LSTM, GRU) usando 2,6 milhões de eventos de ameaças reais. Embora todos os modelos tenham mostrado alta acurácia (com *recall* superior a 94%, essencial para SOC), o CNN-static(1D), um modelo proposto, se destacou significativamente em tempo de aprendizado e classificação, revelando-se o mais econômico para resposta a ataques de dia zero e operações de classificação em um ambiente SOC.

Defini-se no artigo [Rastenis et al. 2021] a falta de pesquisas sobre a distinção automatizada entre emails de *spam e phishing*, essencial para a segurança digital. Os autores propõem uma abordagem baseada na análise textual de emails em inglês, russo e lituano, avaliando o uso de tradução automática para expandir dados sem comprometer a precisão. Reforçam também a importância de adaptar os conjuntos de dados ao contexto organizacional. O método *Fast Large Margin* obteve o melhor desempenho, com 90,07% de precisão.

Enquanto que o trabalho [Patel et al. 2024] propõe um *pipeline* de aprendizado de máquina para a detecção automatizada de exploração de vulnerabilidades em tempo real a partir de *feeds* de Inteligência de Ameaças (TI). A pesquisa utiliza técnicas de processamento de linguagem natural (NLP), incluindo *embeddings* especializados como TIBERT, semelhante ao BERT, para analisar o volume e a heterogeneidade dos dados de TI e identificar eventos de exploração. Esta metodologia demonstrou precisão alcançando 78% de acurácia média na identificação de explorações.

Outras abordagens relevantes incluem o framework ATT&CK da MITRE [MITRE 2025], que organiza táticas e técnicas adversárias observadas em incidentes reais e é amplamente empregado em programas de *threat intelligence*; a norma ISO/IEC 27035, que trata da gestão de incidentes de segurança da informação e sugere práticas para detecção, classificação e resposta; o VERIS (*Vocabulary for Event Recording and Incident Sharing*) [VERIS 2025], criado para padronizar o registro e compartilhamento de eventos; o *framework* de serviços de CSIRTs do FIRST [FIRST 2025], que propõe categorias operacionais para atuação de equipes de resposta; e o modelo ITIL [AXELOS 2025], usado em ambientes corporativos para categorizar incidentes com base em impacto e urgência.

Essas taxonomias, embora variem em escopo e grau de formalização, evidenciam a importância de abordagens sistemáticas para a categorização de incidentes de segurança da informação. A padronização terminológica possibilita uma resposta mais coordenada a incidentes, facilita a análise forense e promove o compartilhamento estruturado de informações entre organizações. Além disso, contribui para a consolidação de uma linguagem comum no domínio da segurança cibernética, o que fortalece a interoperabilidade e a eficácia das ações preventivas e corretivas. [ENISA 2018]

Neste trabalho, utilizamos como base a publicação NIST SP 800-61 Rev.3 [Nelson et al. 2025], cuja taxonomia funcional de incidentes é amplamente adotada. Embora não imponha um padrão rígido, o NIST sugere categorias funcionais adaptáveis a cada instituição. Um modelo comum é o do US-CERT, que inclui acesso não autorizado,

código malicioso, negação de serviço, uso indevido, varreduras e investigações. O modelo pode ser ajustado via engenharia de *prompts*, conforme as necessidades da instituição. As categorias adotadas estão no Apêndice A.

2.2. Engenharia de *Prompts* para Categorização de Incidentes

A engenharia de *prompt* compreende um conjunto de técnicas voltadas à formulação estratégica de instruções textuais, com o objetivo de otimizar a entrada fornecida aos modelos de linguagem (LLMs). Essa prática busca melhorar a qualidade e a relevância das respostas geradas, especialmente em tarefas complexas e sensíveis ao contexto, como classificação, sumarização e geração de texto.

Uma técnica promissora nesse campo é o PHP, um modelo de engenharia de *prompts* que utiliza "dicas"progressivas para refinar as respostas geradas por modelos de linguagem. A abordagem consiste em reenviar a mesma pergunta acompanhada de uma dica adicional, permitindo que o modelo de linguagem ajuste sua resposta com base nesse novo contexto. Essa técnica tem se mostrado eficaz, especialmente em domínios específicos como a resolução de problemas matemáticos [Zheng et al. 2023].

Na implementação desenvolvida, a sequência de refinamentos é interrompida quando a resposta atinge um nível satisfatório de similaridade em relação à resposta anterior. Esse método, aliado a *prompts* otimizados pelo *Progressive-Hint*, melhora a precisão das respostas geradas. Na Figura 1, ilustramos o esquema do funcionamento da heurística PHP aplicada à categorização de incidentes de segurança. Inicialmente, um *prompt* básico (PROMPT Question 0 - Q0) é enviado ao modelo de linguagem, que retorna uma resposta (Hint 1 - H1).

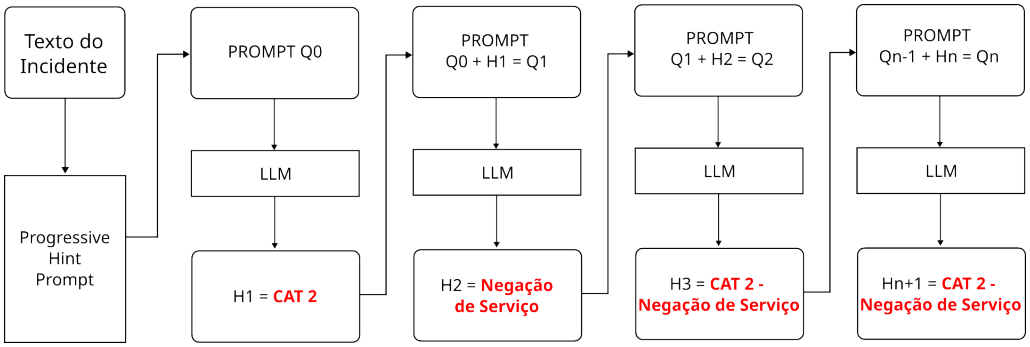

Figura 1. Adaptação do Método PHP aplicado em texto.

A resposta gerada é avaliada quanto à similaridade com o padrão definido. Caso a similaridade não atinja o limiar pré-definido, uma nova dica baseada na resposta anterior é incorporada ao próximo *prompt*. Este processo iterativo continua até que o nível de similaridade atenda ao critério estabelecido ou seja alcançado o número máximo de refinamentos (H3 seja "Altamente Similar"à Hn+1).

3. Trabalhos Relacionados

Pesquisas e publicações recentes tem contribuído para a necessidade de classificação automatizada de incidentes de segurança. A Tabela 1 apresenta uma síntese dos

principais trabalhos analisados, destacando o domínio de aplicação, os métodos de engenharia de *prompt* utilizados, os modelos de linguagem empregados e o escopo/limitações de cada proposta.

O primeiro agrupamento da Tabela 1 corresponde aos trabalhos aplicados ao domínio "Incidentes de Segurança". Mesmo antes da popularização dos modelos de linguagem, [Grispos 2016], forneceu uma base teórica sobre categorização de incidentes, destacando a importância de taxonomias e normas como o NIST e ISO/IEC. Contudo, atualmente, o emprego de LLMs vem se popularizando. Um exemplo é o *framework* SEVENLLM [Ji et al. 2024], que combina *fine-tuning* multitarefa em modelos como Llama e Qwen, juntamente com um sistema de moderação fundamentado em bases estruturadas como MITRE ATT&CK [MITRE 2025] e OASIS CTI [OASIS 2025], obtendo resultados em tarefas de inteligência contra ameaças.

Tabela 1. Classificação dos trabalhos relacionados quanto ao domínio, método de engenharia de *prompt* e modelos de linguagens utilizados.

Artigo	Domínio	Método	Modelos	Escopo e Limitações
[Grispos 2016]	Inc. Seg.	Não se aplica	Não se aplica	Referência teórica sobre taxonomias; não utiliza LLMs nem prompting.
[Molleti et al. 2024]	Inc. Seg.	HPTSA	GPT (em geral), BERT, T5, PaLM, LaMDA	Uso voltado à mitigação e resposta; não aplica prompting heurístico estruturado.
[Zhao et al. 2024]	Inc. Seg.	DA	ChatGPT (base GPT-3.5)	Enfoque em análise automática, sem integração com taxonomias ou PHP.
[Ji et al. 2024]	Inc. Seg.	SevenLLM	GPT-4, Llama 2, Qwen 1.5	Foco em threat intelligence; não combina prompting heurístico com estrutura NIST.
[Chen et al. 2023]	Matemática	SKiC	ChatGPT, Llama2-70B, Minerva-540B	Raciocínio incremental em matemática; sem relação com incidentes de segurança.
[Wu et al. 2024]	Matemática	PRP	GPT-3.5, GPT-4	Explora raciocínio progressivo em matemática; não aborda categorização ou segurança.
[Chen et al. 2024]	Matemática	SHP	text-davinci-003, GPT-3.5-Turbo	Hint automática aplicada a tarefas matemáticas; sem foco em segurança.
[Li et al. 2024]	Matemática	HTP	GPT-3.5-Turbo	Geração de hipóteses em matemática; sem taxonomia ou PHP.
[Zheng et al. 2023]	Matemática	PHP	text-davinci-003, GPT-3.5-Turbo, GPT-4	Aplica PHP em tarefas gerais; não integra taxonomia nem trata segurança.
Solução Proposta	**Inc. Seg.**	**PHP**	**Gemini, Grok, GPT-4, Llama3**	**Aplicar PHP com estrutura NIST para referência e consistência na classificação.**

No trabalho [Molleti et al. 2024], é proposta a integração de agentes baseados em LLMs a sistemas de resposta automatizada, por meio da arquitetura *Hierarchical Planning and Task-Specific Agents* (HPTSA), com suporte a *Security Information and Event Management* (SIEM) e *Artificial Intelligence for IT Operations* (AI-Ops). Os três estudos adotam metodologias que reforçam tanto a necessidade de automação quanto a viabilidade de aplicação de LLMs no contexto das classificações.

Outros estudos ampliam essa base ao explorar técnicas específicas com

LLMs aplicadas à detecção e categorização de incidentes. Por exemplo, o estudo [Zhao et al. 2024] avaliou o uso do ChatGPT (GPT-3.5) para melhorar a performance em cenários de dados desbalanceados, aplicando técnicas de *data augmentation* (DA) via engenharia de prompt e obtendo ganhos relevantes de *F1-score* na classificação de amostras minoritárias.

Ademais, uma revisão sistemática sobre o uso de LLMs na segurança cibernética é apresentada em [Silva and Westphall 2024], destacando aplicações como detecção de ameaças, análise de *logs*, geração de código e automação de respostas. Os *prompts* utilizados nessas aplicações baseiam-se, em grande parte, nas técnicas fundamentais de *zero-shot* e *few-shot*. Além de mapear essas aplicações, o estudo identifica desafios, incluindo problemas como escassez de dados específicos e confiabilidade das respostas. Esses achados reforçam a importância de soluções em NLP para a classificação automatizada de incidentes de segurança.

Por sua vez, os autores de [Zheng et al. 2023] propuseram o método *Progressive-Hint Prompt*, uma estratégia de *prompting* iterativo que utiliza respostas anteriores para refinar raciocínios complexos. Embora o foco da desse *framework* não seja a segurança cibernética, sua demonstração de maior assertividade em tarefas complexas torna-se uma inspiração direta para este estudo.

O segundo agrupamento da Tabela 1 reúne os trabalhos do domínio "Matemática", onde diferentes abordagens de Engenharia de *prompt* são investigadas. Os estudos abordam metodologias como *Skills-in-Context* (SKiC), *Prompt Engineering Through Optimal Control* (PETOC) e *Progressive Rectification Prompting* (PRP), que ampliam os objetivos do PHP, propondo soluções baseadas em ciclos de verificação [Chen et al. 2023] e raciocínio por habilidades compostas [Wu et al. 2024].

Além disso, abordagens como o *Self-Hint Prompting* (SHP) [Chen et al. 2024] e o *Hypothesis Testing Prompting* (HTP) [Li et al. 2024] aprofundam a lógica iterativa e reflexiva no processo de geração de respostas por LLMs. Embora esses trabalhos não atuem no domínio "Incidentes de Segurança", sua relevância metodológica está em oferecer comparações consistentes com o PHP e ampliar a gama das heurísticas que podem ser adaptadas à classificação automatizada de incidentes, especialmente quando alinhadas a referenciais como o NIST.

Dessa forma, o presente trabalho, fundamentado nas abordagens teóricas analisadas, apresenta a aplicação do PHP como estratégia central para o refinamento progressivo e contextualizado da classificação automática de incidentes de segurança utilizando LLMs. Esta abordagem é integrada a referenciais formais consolidados tecnicamente, como o NIST, com o objetivo de assegurar robustez e conformidade ao processo de categorização.

4. Metodologia

Este estudo propõe a categorização de incidentes de segurança da informação utilizando técnicas de engenharia de *prompts*, adotando uma abordagem experimental centrada na aplicação prática dessas técnicas, com o objetivo de otimizar a precisão e a eficiência na classificação automatizada de incidentes em ambientes reais, melhorando a resposta a ameaças e a tomada de decisões em tempo hábil.

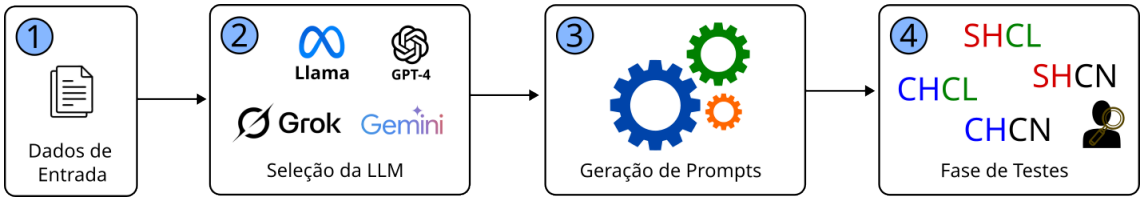

Figura 2. Etapas do Experimento

A hipótese central é que a combinação de *prompts* progressivos e uma taxonomia padronizada, como a tabela do NIST, melhora a precisão das respostas geradas por LLMs. O projeto propõe um sistema capaz de processar bases de dados realistas, derivadas de relatos reais de incidentes, com o objetivo de avaliar o desempenho de diferentes LLMs. Para testar essa hipótese, foi estruturado o experimento nas seguintes etapas.

4.1. Preparação e organização dos dados de entrada

A etapa de preparação e organização dos dados de entrada utiliza um conjunto de 194 incidentes reais anonimizados, no formato xlsx, fornecidos pelo POP-RS. A anonimização seguiu protocolos de remoção de identificadores sensíveis, garantindo que os dados preservassem a estrutura semântica dos relatos sem comprometer a privacidade dos envolvidos. Além disso, os incidentes foram rotulados previamente por dois analistas do POP-RS/RNP, utilizando a taxonomia do NIST SP 800-61r3 como referência. Essa rotulação serviu como base para o cenário de análise humana e para o cálculo das métricas de similaridade entre os rótulos humanos e os gerados pelas LLMs

4.2. Seleção dos Modelos de Processamento de Linguagem Natural

O sistema desenvolvido foi projetado para integrar diferentes LLMs selecionados para uso neste trabalho (Tabela 2). Estão incluídos quatro LLMs de diferentes provedores, Gemini (Google), Grok (X), GPT-4 (OpenAI) e Llama (Meta), com as respectivas versões de software (presentes na coluna "Modelo").

Tabela 2. Lista de LLMs e versões de software utilizadas.

Referência	LLM	Provedor	Modelo
[Google 2025]	Gemini	Google	Gemini-2.0-flash
[X 2024]	Grok	X	Grok-beta
[OpenIA 2024]	GPT-4	OpenAI	gpt-4.0
[Meta 2025]	Llama	Meta	Llama3.1-70b

4.3. Geração de *prompts* customizados

Para conexão entre os dados de entrada com as LLMs, foi criado um *script* na linguagem *Python* 3.10. O arquivo config.json possui a parametrização de cada LLM, enquanto o arquivo *main.py* possui funções para carga dos arquivos de entrada, envio do *prompt* para API de cada modelo de linguagem, bem como seleção do tipo de categorização e adição (ou não) de dicas. A Figura 3 ilustra passos executados pelo programa.

Importante destacar que o mesmo texto original do incidente foi utilizado em todos os cenários, de forma padronizada. Essa consistência assegura que as variações nos

resultados se devem exclusivamente às diferentes estratégias de *prompting* e não a variações na entrada textual.

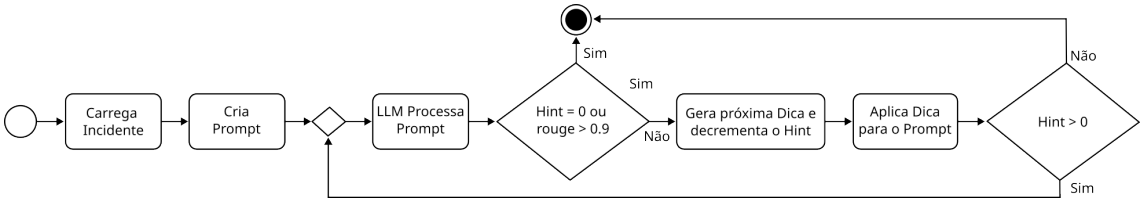

Figura 3. Fluxograma da implementação do PHP

Cada incidente carregado cria um *prompt* e este é enviado para a LLM, resultando em uma categorização. O refinamento ocorre iterativamente, onde o ciclo se encerra quando: 1. o teste é realizado sem dica; 2. quando não há mais dicas (Hint = 0); 3. quando a similaridade entre respostas (ROUGE) é alta ($\geq 0{,}9$). ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) é uma métrica comum para avaliar a qualidade de textos gerados por LLMs, medindo a similaridade entre respostas produzidas e referências pré-registradas através da sobreposição de n-gramas e outras correspondências semânticas. Valores de ROUGE próximos de 1,0 indicam alta similaridade, com valores acima de 0,9 sendo considerados satisfatórios e usados como critério de parada no ajuste de *prompts*. [Lin 2004]

O resultado é armazenado em uma lista e pode ser exportado nos formatos CSV, XLSX ou JSON. Caso existam dicas disponíveis, elas são incorporadas ao *prompt*, o que acarreta na redução do contador de dicas (Hint). O processo é então repetido iterativamente até que se atinja o limite mínimo de dicas ou se obtenha uma resposta considerada satisfatória. Para fins de reprodutibilidade e transparência, disponibilizamos no GitHub[1]. um conjunto com cinco exemplos anonimizados de incidentes, acompanhados dos respectivos *prompts* gerados e respostas obtidas. Esses exemplos foram selecionados por sua representatividade e ilustram os diferentes comportamentos dos modelos frente às técnicas de *prompting* adotadas.

4.4. Fase de testes

Os experimentos foram estruturados em cinco cenários distintos, cada um concebido para explorar aspectos específicos da dinâmica dos incidentes analisados. Essa organização visa garantir uma abordagem sistemática e abrangente da investigação, permitindo uma análise comparativa entre os diferentes contextos. As categorias dos incidentes observados, bem como sua classificação detalhada, estão descritas no Apêndice A, onde são apresentados os critérios utilizados para a sua definição e agrupamento.

No primeiro cenário, foi realizada a *Análise Sem PHP e categorização livre (SHCL).* Nesse cenário, os incidentes foram inseridos em *prompts* simples e enviados diretamente para os modelos via API, sem categorização pré-definida. Um exemplo prático deste cenário se dá como segue: Um incidente descrito como "Usuário detectou atividade incomum em sua conta"pode ser classificado erroneamente como "Incidente geral de segurança", sem especificação. Sem um direcionamento adequado, o modelo pode não identificar que se trata de um *Comprometimento de Conta (CAT1).*

[1] `https://github.com/AI-Horizon-Labs/SecLINC`

No segundo cenário, foi explorada a *Análise Sem PHP e com categorização NIST (SHCN).* Nesse cenário, os LLMs receberam categorias pré-definidas da taxonomia NIST, melhorando a precisão, mas limitando a flexibilidade interpretativa. Um exemplo prático deste cenário é um relatório contendo *"Ataque volumétrico via UDP flood"* pode ser diretamente categorizado como *Ataque de Negação de Serviço (CAT3)*, garantindo maior alinhamento com a classificação humana.

No terceiro cenário, é realizada a *Análise Com PHP e categorização livre (CHCL)*, onde aplicou-se o método PHP, sem a estrutura fixa da taxonomia NIST. O PHP oferece dicas progressivas para guiar os modelos na categorização de incidentes, por exemplo: para um evento descrito como "Exposição de banco de dados público na nuvem", o PHP pode orientar a LLM com sugestões como "Uma dica é a resposta anterior que envolve acesso não autorizado a informações?", ajudando o modelo a classificar **corretamente** como *Exfiltração ou Vazamento de Dados (CAT4)*.

No quarto cenário, é realizada a *Análise Com PHP e com categorização NIST (CHCN)*, no qual combinou-se o PHP com a taxonomia NIST para maximizar a precisão da categorização. Um exemplo desse cenário se dá por um ataque identificado como "Acesso externo suspeito ao firewall"pode ser refinado para *Exploração de Vulnerabilidade com tentativa de intrusão (CAT5 + CAT12)*, após o PHP orientar o modelo a considerar múltiplos fatores.

No quinto cenário, o conceito de *Human-In-The-Loop* é explorado através de uma etapa de *Análise Humana*. Esse cenário é executado como referência comparativa, no qual os incidentes foram classificados manualmente por dois analistas da equipe técnica do POP-RS, seguindo a tabela de incidentes (Apêndice A). Por exemplo, um incidente descrito como "E-mail enganoso simulando comunicação interna solicitando credenciais"foi identificado como *Engenharia Social (CAT7)* e tratativas foram recomendadas, como reforço na conscientização dos usuários.

Para medir a similaridade semântica entre respostas de LLMs e análises humanas, utilizamos o modelo *paraphrase-MiniLM-L6-v2*, reconhecido por seu bom desempenho na detecção de paráfrases e eficiência computacional. Como descrito em [Zhou et al. 2022], comparamos os *embeddings* gerados por meio de faixas de similaridade como os baseadas no cosseno entre vetores.

Outros trabalhos exploram faixas de similaridade distintas das nossas. Por exemplo, [Cer et al. 2018] propõe um modelo de codificação universal de sentenças com limiares diferentes, enquanto [Ming et al. 2021] discute o impacto de correlações espúrias em distribuições fora do domínio, usando métricas alternativas. Como não há um consenso na literatura sobre os limiares ideais, definimos empiricamente as seguintes faixas: Altamente semelhante (> 0.8), Moderadamente semelhante (0.6–0.8), Levemente semelhante (0.6-0.4) e Divergente (< 0.4).

5. Resultados e Discussão

Esta seção apresenta os resultados dos testes dos cenários da Subseção 4.4, destacando padrões de desempenho entre modelos e o impacto das estratégias de *prompting* e taxonomias na qualidade das classificações. Também são discutidas implicações práticas, com foco na consistência entre modelos e sua proximidade com classificações humanas.

5.1. Discussão Comparativa

Com base na metodologia (Seção 4), os dados foram analisados para identificar padrões, variações e relações relevantes. A seguir, os principais achados são organizados para facilitar a comparação entre os cenários avaliados.

O cenário SHCL (Figura 4) evidenciou a limitação dos LLMs em manter consistência sem apoio taxonômico ou de heurística, resultando em rotulagens variadas para incidentes semelhantes. Um exemplo foi a classificação de ataques DDoS, que apareceu com pelo menos quatro denominações distintas (ver Apêndice B). Esse cenário evidenciou o pior desempenho geral, o seu melhor resultado foi 92 classificações altamente semelhantes na LLM Gemini e o pior resultado aconteceu com a LLM Grok com 61 divergentes sobre os 194 incidentes analisados, o que indica grande dispersão semântica.

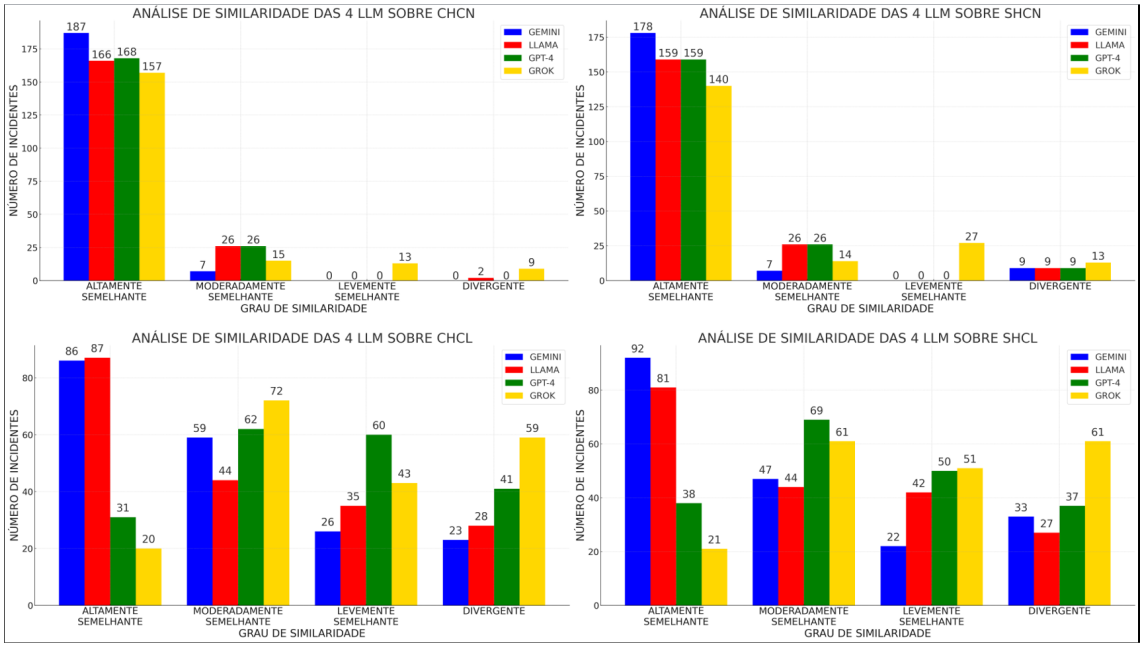

Figura 4. Comparação dos resultados entre os cenários CHCN, SHCN, SHCL e CHCL

O uso exclusivo da taxonomia NIST (SHCN) promoveu ganhos significativos na uniformidade, tendo a LLM Gemini apresentado um resultado de 178 de "Altamente Semelhante", uma tendência em todas outras LLM. No entanto, o modelo Grok também se destacou negativamente por apresentar 27 ocorrências levemente semelhantes e 13 divergentes, revelando que o Grok teve uma análise textual ruim, contrastando com os demais, que tiveram apenas 9 casos divergentes cada. A presença de respostas intermediárias em Llama (26) e GPT-4 (26), demonstram que a ausência de heurísticas como o PHP ainda permite variações interpretativas relevantes.

Já o uso isolado da heurística PHP (CHCL) demonstrou capacidade de organizar o raciocínio das LLMs, mas sem suporte taxonômico, a consistência não foi satisfatória, pois os números de divergência foram quase iguais nos cenários com categorização livre. A ausência da taxonomia fez com que o PHP mantivesse a dispersão de resultados entre as LLM que foram empregadas no estudo.

A melhor performance foi observada no cenário CHCN, que combina a orientação PHP com a taxonomia NIST, os modelos apresentaram desempenho mais consistente,

com alta concentração na faixa "Altamente semelhante". O modelo Gemini novamente se destacou com 187 ocorrências nessa faixa, seguido por Llama (166), GPT-4 (168) e Grok (157). A presença de respostas "Moderadamente semelhantes"caiu significativamente em Gemini (7) e GPT-4 (15), mantendo-se mais elevada em Llama (26) e Grok (26). Casos de divergência foram mínimos em GPT-4 (2) e Grok (9), sendo inexistentes em Gemini e Llama. Esses dados confirmam que a combinação de taxonomia NIST com a orientação heurística do PHP, contribuiu para maior uniformidade semântica entre os modelos, reduzindo variações e classificações ambíguas.

5.2. Análise comparativa com classificação humana

A análise humana visou compreender o grau de aproximação e aderência conceitual dos modelos em relação ao conhecimento dos analistas especializados em categorização de incidentes, permitindo identificar padrões de erro, vieses e possíveis limitações na capacidade interpretativa das LLMs. Para essa etapa, realizamos um comparativo utilizando o modelo paraphrase-MiniLM-L6-v2 para avaliar a similaridade semântica das respostas geradas em contrapartida aos quatro cenários de categorização (SHCL, CHCL, SHCN e CHCN). Esse processo contribuiu para um ciclo iterativo de refinamento de *prompts* e ajustes metodológicos, garantindo maior precisão nas classificações.

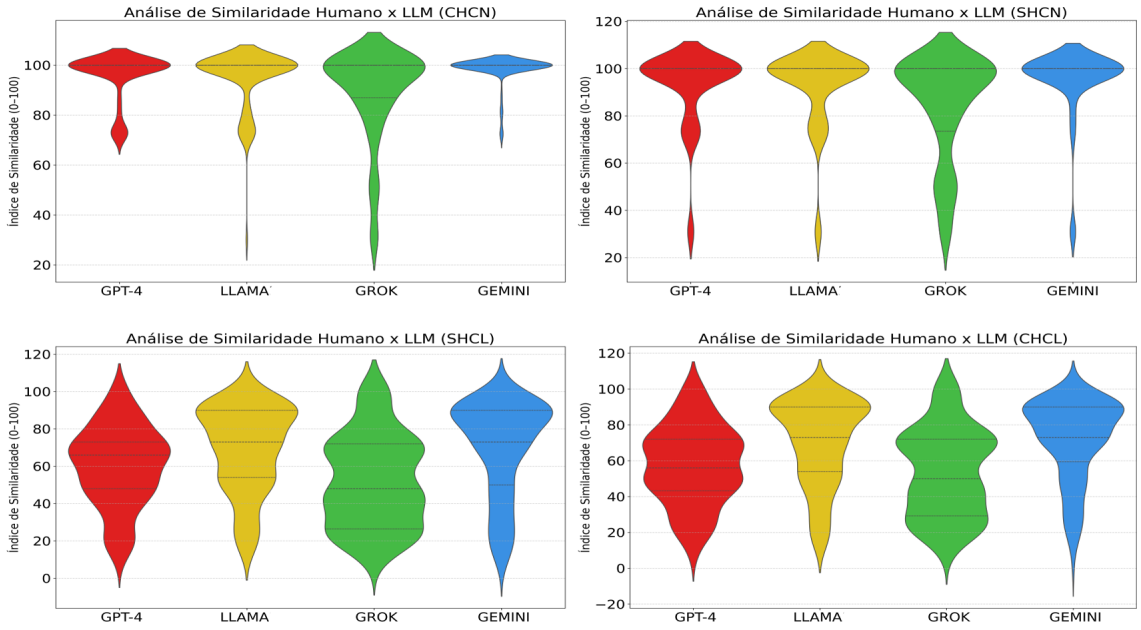

Figura 5. Gráficos de Similaridade CHCN, SHCN, SHCL, CHCL das quatro LLMs

Como ilustrado na Figura 5, a análise de similaridade entre as classificações humanas e as respostas das LLMs no cenário CHCN evidencia alta convergência semântica. Em particular, observa-se a maior concentração de respostas com alto índice de similaridade semântica em relação à análise humana. Os modelos Gemini e GPT-4 se destacaram, com distribuições altamente concentradas acima de 90 pontos, refletindo elevado alinhamento conceitual. A pior performance relativa ficou com o modelo Grok, que embora tenha atingido bons níveis em média, apresentou maior dispersão e uma cauda inferior indicando menor consistência em parte das respostas. Este cenário demonstra a eficácia da integração entre heurísticas e categorias de referência para orientar os modelos,

resultando em classificações mais precisas e homogêneas.

Adicionalmente, a análise permitiu verificar a proximidade das LLMs em relação à avaliação humana, servindo como indicador preliminar de alinhamento conceitual. O modelo Gemini apresentou o maior grau de aderência semântica aos rótulos humanos, sugerindo maior estabilidade interpretativa nesse cenário.

No cenário SHCN, observa-se um leve decréscimo nos volumes de classificação. As LLMs ainda mantêm boa concentração de similaridade, com destaque para Gemini, que continua entre os mais estáveis. Por outro lado, Grok novamente apresenta a maior variabilidade, com uma distribuição mais larga e pontos de baixa similaridade. Embora a taxonomia traga ganhos de padronização, a ausência do direcionamento heurístico resulta em maior dispersão nos julgamentos semânticos, evidenciando a importância de uma condução mais refinada do modelo por meio de *prompt* progressivos. Assim, o cenário SHCN reforça a importância da combinação entre categorização orientada e guias heurísticos para obtenção de respostas mais aderentes ao julgamento humano.

Os modelos Grok e GPT-4 demonstraram um alinhamento geral inferior com a análise humana em comparação com Gemini e Llama, especialmente nas categorias de maior similaridade. A maior incidência de classificações nos índices mais baixos pode indicar maior incerteza ou uma interpretação semântica diferente em relação ao conhecimento dos especialistas humanos. Como as análises são baseadas no conhecimento prévio assimilado pelas LLMs sobre o tema em questão, observa-se que os modelos Llama e Gemini apresentaram maior alinhamento com o entendimento e os critérios utilizados pelos analistas humanos. Embora não seja possível afirmar se as divergências entre os resultados indicam erros ou acertos, elas evidenciam a necessidade de especialização dos modelos e do aprimoramento contínuo de seus mecanismos de aprendizagem, com o objetivo de tornar as classificações mais precisas e coerentes com a avaliação humana.

No entanto, um ponto crucial que emerge da análise individualizada é a consistência do padrão observado entre os cenários: independentemente das diferenças de desempenho absoluto entre os LLMs, o cenário CHCN (PHP + NIST) consistentemente apresentou os melhores resultados, ou uma das melhores combinações, em termos de maior similaridade e menor divergência com a referência humana para cada um dos quatro modelos avaliados.

Isso reforça significativamente a tese principal deste trabalho: embora a escolha do LLM base influencie o teto de desempenho alcançável, a abordagem metodológica de combinar a heurística *Progressive-Hint Prompting* com a taxonomia estruturada do NIST (CHCN) demonstra ser adequada para a tarefa de categorização automatizada de incidentes. Seus benefícios podem ser observados através de diferentes arquiteturas de modelos, validando-a como uma estratégia promissora para melhorar a consistência e o alinhamento com o entendimento humano especializado. A necessidade de especialização e aperfeiçoamento dos LLMs permanece, mas a eficácia da abordagem CHCN se destaca como um resultado desta investigação.

5.3. Análise Estatística com Kruskal-Wallis

Para verificar se as diferenças entre os modelos LLMs nos cenários testados são estatisticamente significativas, utilizou-se o teste de Kruskal-Wallis, adequado para comparar três ou mais grupos independentes sem pressupor normalidade nos dados, especial-

mente para escores de similaridade semântica assimétricos [Siegel and Jr. 2006].

Tabela 3. Resultado do teste de Kruskal-Wallis por cenário

Cenário	Estatística H	p-valor	Significativo (p < 0,05)
CHCN	55.43	$< 0{,}001$	Sim
SHCN	50.93	$< 0{,}001$	Sim
CHSN	94.59	$< 0{,}001$	Sim
SHSN	79.96	$< 0{,}001$	Sim

O teste foi aplicado separadamente nos quatro cenários (CHCN, SHCN, CHSN, SHSN), com os resultados consolidados na Tabela 3. Os achados confirmam diferenças significativas entre os modelos, validando as análises comparativas. A Figura 6 mostra a distribuição dos escores por modelo e cenário, evidenciando padrões como concentração e mediana.

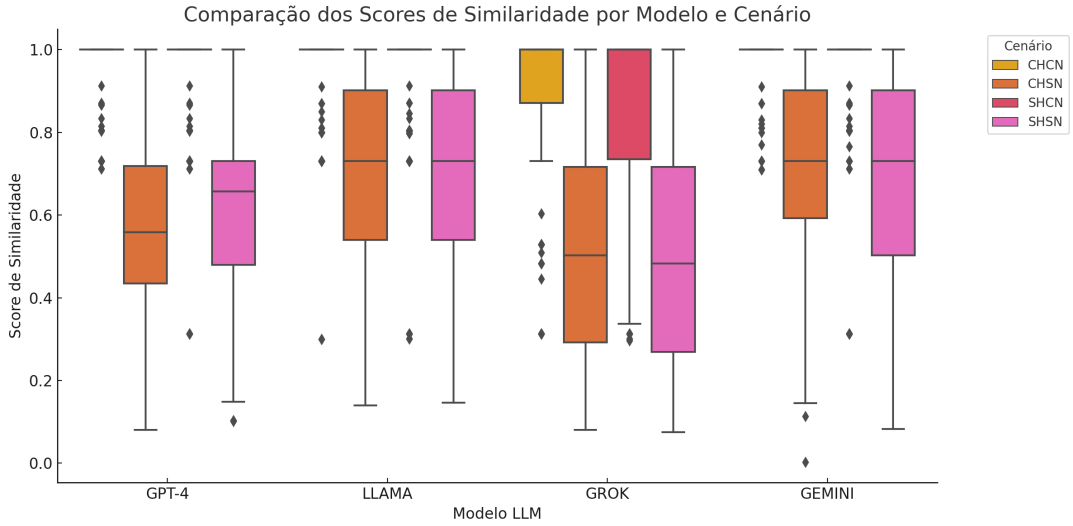

Figura 6. Distribuição dos escores de similaridade por modelo e cenário

É possível observar que o cenário CHCN (com PHP e NIST) apresentou os melhores resultados em termos de concentração de similaridade e baixa dispersão. Já o modelo Grok apresentou desempenho consistentemente inferior, com ampla variabilidade e menor aderência semântica. Gemini foi o modelo com melhor mediana em todos os cenários, especialmente em CHCN. Por fim, GPT-4 e Llama mostraram desempenho semelhante e mais estável nos cenários orientados por taxonomia (SHCN, CHCN). Esses achados reforçam a importância de considerar não apenas a escolha do modelo de linguagem, mas também o uso de heurísticas (PHP) e taxonomias estruturadas (NIST) para garantir resultados mais consistentes e alinhados com a análise humana.

5.4. Limitações Práticas e Análise Qualitativa dos Erros

A proposta enfrenta limitações práticas importantes, como a dependência de APIs comerciais de LLMs sem *fine-tuning*, o que implica em custos por *token*, instabilidade e

riscos à privacidade, mesmo com dados anonimizados. Como alternativa, está em desenvolvimento um *framework* para avaliação de *Small Language Models* (SLMs) treinados localmente, visando reduzir custos e garantir maior controle sobre dados sensíveis.

Adicionalmente, a análise qualitativa dos erros evidenciou dificuldades dos modelos em lidar com descrições ambíguas ou genéricas, como "atividade incomum" ou "tentativa de login não reconhecida", resultando em classificações divergentes ou imprecisas. Também foram observadas falhas em incidentes com possibilidade de classificação de mais de uma categoria (ex.: CAT5 + CAT3), agravadas por sobreposições conceituais, ausência de contexto técnico e ambiguidade linguística.

Essas limitações reforçam a necessidade de especialização dos modelos, da melhoria contínua nas dicas do PHP e da construção de *datasets* mais ricos semanticamente. Como medida complementar, recomenda-se incorporar mecanismos de análise humana assistida para casos ambíguos, especialmente em categorias de interpretação subjetiva, como Engenharia Social (CAT7) e Tentativa de Intrusão (CAT12).

6. Considerações Finais e Trabalhos Futuros

A análise de incidentes de segurança tem se tornado um desafio crescente devido ao aumento do volume e da complexidade dos eventos, impulsionados por inovações tecnológicas e o crescimento exponencial de atacantes. Isso dificulta a triagem eficiente, a identificação de padrões de ataque e a tomada de decisões rápidas, uma vez que esses processos exigem tempo, um recurso muitas vezes escasso durante eventos críticos.

A construção da solução envolveu uma análise cuidadosa dos dados coletados previamente para prever os resultados possíveis. Os quatro modelos de implementação (SHCL, SHCN, CHCL e CHCN) auxiliaram na compreensão de como a classificação prévia de incidentes pode ser realizada de forma rápida e eficiente. A combinação da estratégia PHP com a tabela de categorização do NIST mostrou-se eficaz na automatização da categorização de incidentes, contribuindo para a normalização dos tipos de categorização gerados pelas LLMs, reduzindo ambiguidades e promovendo maior consistência nos resultados. Os experimentos com engenharia de *prompt* demonstraram uma convergência para categorias padronizadas, embora haja espaço para melhorias. A primeira dica, baseada no NIST, pode ser refinada para melhor contextualizar os termos, aprimorando sua eficácia como ponto de partida do PHP. Futuros aprimoramentos devem focar na qualidade semântica e contextual dessa dica, buscando maior precisão nas categorizações dos modelos.

Trabalhos Futuros. Uma análise qualitativa dos erros sistemáticos pode melhorar a categorização de incidentes, otimizando a triagem, resposta e comunicação entre equipes em SOCs e CSIRTs, especialmente com o apoio de *playbooks* automatizados. A solução demonstrou escalabilidade nos PoPs da RNP, centralizando incidentes de diversas instituições, e será validada em parceria com uma organização global para confirmar sua aplicabilidade em larga escala. Futuramente, a abordagem será integrada a soluções como plataformas SOAR, sistemas SIEM e painéis de resposta, com a geração automática de rótulos por LLMs para aprimorar fluxos de trabalho e priorizar respostas. A heurística PHP será expandida para explorar variações diretas, descritivas e indutivas, associando cada categoria da taxonomia NIST SP 800-61r3 a um incidente representativo, permitindo refinar estratégias de *prompt engineering* e melhorar a classificação semântica.

Agradecimentos. Esta pesquisa contou com apoio parcial da CAPES, código de financiamento 001; da RNP, por meio do Programa Hackers do Bem e do GT LFI – *Learn From Incidents*; e da FAPERGS, por meio dos termos de outorga 24/2551-0001368-7 e 24/2551-0000726-1.

Referências

AXELOS (2025). What is it service management. https://www.axelos.com/certifications/itil-service-management/what-is-it-service-management.

Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder.

CERT.br (2025). Incidentes notificados ao cert.br. https://stats.cert.br/incidentes/.

Chen, J., Pan, X., Yu, D., Song, K., Wang, X., Yu, D., and Chen, J. (2023). Skills-in-context: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*.

Chen, J., Tian, J., and Jin, Y. (2024). Self-hint prompting improves zero-shot reasoning in large language models via reflective cycle. In *Proceedings of the 46th Annual Conference of the Cognitive Science Society*.

Cichonski, P., Millar, T., Grance, T., and Scarfone, K. (2012). Computer security incident handling guide. Technical Report NIST Special Publication 800-61 Revision 2.

ENISA (2018). Reference incident classification taxonomy. https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy.

F5 Networks (2024). Generative ai for threat modeling and incident response. https://www.f5.com/company/blog/generative-ai-for-threat-modeling-and-incident-response.

FIRST (2025). First csirt services framework. https://www.first.org/standards/frameworks/csirts/.

Google (2025). Gemini models. https://ai.google.dev/gemini-api/docs/models.

Grispos, G. (2016). *Cybercrime and Organizational Response: Exploring the Roles of Digital Forensics Investigations and Information Security Policy*. PhD thesis.

Grispos, G., Glisson, W. B., and Storer, T. (2019). How good is your data? investigating the quality of data generated during security incident response investigations.

IBM (2024). O que é engenharia de prompt? https://www.ibm.com/br-pt/think/topics/prompt-engineering.

Ibrishimova, M. D. (2019). Cyber incident classification: Issues and challenges. In Xhafa, F., Leu, F.-Y., Ficco, M., and Yang, C.-T., editors, *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*.

Ji, H., Yang, J., Chai, L., Wei, C., Yang, L., Duan, Y., Wang, Y., Sun, T., Guo, H., Li, T., Ren, C., and Li, Z. (2024). Sevenllm: Benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence. *arXiv preprint arXiv:2405.03446*.

Kim, J.-y. and Kwon, H.-Y. (2022). Threat classification model for security information event management focusing on model efficiency. *Computers & Security*, 120:102789.

Li, Y., Tian, J., He, H., and Jin, Y. (2024). Hypothesis testing prompting improves deductive reasoning in large language models. *arXiv preprint arXiv:2405.06707*.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Lukwaro, E., Kalegele, K., and Nyambo, D. (2024). A review on nlp techniques and associated challenges in extracting features from education data. *International Journal of Computing and Digital Systems*, 16:2210–142.

Meta (2025). LLAMA Models. https://ai.meta.com/blog/meta-llama-3.

Ming, Y., Yin, H., and Li, Y. (2021). On the impact of spurious correlation for out-of-distribution detection.

MITRE (2025). Att&ck matrix for enterprise. https://attack.mitre.org/.

Molleti, R., Goje, V., Luthra, P., and Raghavan, P. (2024). Automated threat detection and response using llm agents. *Journal of Advanced Research and Reviews*, 24(2).

Nelson, A., Rekhi, S., Souppaya, M., and Scarfone, K. (2025). Incident response recommendations and considerations for cybersecurity risk management: A csf 2.0 community profile. Technical Report NIST SP 800-61r3.

OASIS (2025). Sharing threat intelligence just got a lot easier! https://oasis-open.github.io/cti-documentation/.

Ogundairo, O. and Broklyn, P. (2024). Natural language processing for cybersecurity incident analysis. *Journal of Cyber Security*.

OpenIA (2024). GPT-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

Patel, K., Shafiq, Z., Nogueira, M., Menasché, D., Lovat, E., Kashif, T., Woiwood, A., and Martins, M. (2024). Harnessing ti feeds for exploitation detection. In *IEEE CSR*.

Rastenis, J., Ramanauskaitė, S., Suzdalev, I., Tunaitytė, K., Janulevičius, J., and Čenys, A. (2021). Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation. *Electronics*, 10(668).

Siegel, S. and Jr., N. J. C. (2006). *Estatística não-paramétrica para ciências do comportamento*. Artmed, Porto Alegre, 2 edition.

Silva, G. C. and Westphall, C. B. (2024). A survey of large language models in cybersecurity. *arXiv preprint arXiv:2402.16968*.

VERIS (2025). Veris: The vocabulary for event recording and incident sharing. https://verisframework.org/index.html.

Wu, Z., Jiang, M., and Shen, C. (2024). Get an a in math: Progressive rectification prompting. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*.

X (2024). Grok-2 Beta Release. https://x.ai/news/grok-2.

Zhao, H., Chen, H., Ruggles, T. A., Feng, Y., Singh, D., and Yoon, H.-J. (2024). Improving text classification with large language model-based data augmentation. *Electronics*, (2535).

Zheng, Liu, X. et al. (2023). Progressive-hint prompting improves reasoning in large language models.

Zhou, K., Ethayarajh, K., Card, D., and Jurafsky, D. (2022). Problems with cosine as a measure of embedding similarity for high frequency words.

A. Apêndice: Categorias de Incidentes adotada segundo orientações do NIST SP 800-61r3

A tabela de classificação de incidentes neste apêndice organiza e avalia a gravidade de eventos adversos de forma estruturada. Com base em impacto, urgência e risco, facilita a priorização e decisões rápidas, auxiliando na resposta adequada e garantindo a continuidade das operações.

Tabela 4. Categorização de Incidentes de Segurança (com prioridade)

Código	Categoria	Descrição	Prioridade
CAT1	Comprometimento de Conta	Acesso não autorizado a contas de usuários ou administradores.	5
CAT2	Malware	Infecção por código malicioso que compromete dispositivos ou dados.	5
CAT3	Ataque de Negação de Serviço (DoS/DDoS)	Tornar sistemas ou redes indisponíveis.	4
CAT4	Exfiltração ou Vazamento de Dados	Acesso, cópia ou divulgação não autorizada de dados sensíveis.	5
CAT5	Exploração de Vulnerabilidade	Uso de falhas conhecidas ou desconhecidas para comprometer ativos.	5
CAT6	Abuso Interno	Ações intencionais ou negligentes de usuários internos.	5
CAT7	Engenharia Social	Engano de pessoas para obter acesso ou informações.	3
CAT8	Incidente Físico ou de Infraestrutura	Violação física que impacta ativos computacionais.	4
CAT9	Alteração Não Autorizada	Modificação não autorizada em sistemas, dados ou configurações.	3
CAT10	Uso Indevido de Recursos	Uso não autorizado de sistemas para outros fins.	2
CAT11	Problema de Fornecedor/Terceiro	Incidente originado por falha de segurança de terceiros.	4
CAT12	Tentativa de Intrusão	Tentativas hostis de invasão ainda não confirmadas como bem-sucedidas.	3

B. Apêndice: Categorias únicas extraídas do conjunto de dados analisados

A tabela de categorias únicas extraídas dos dados analisados apresenta uma visão detalhada das classificações identificadas, mostrando agrupamentos distintos que evidenciam tendências, padrões e variações nos dados.

Tabela 5. Categorias únicas extraídas do conjunto de dados analisados

Categoria	Categoria (continuação)	Categoria (continuação)
Brute Force Attack	Information Disclosure	Port Scanning/Network Attack
Category	Information Sharing/Awareness	Privilege Escalation
Compromised System/DDoS Attack	Informational	Remote Code Execution
Compromised System/Malicious Activity	Informational Announcement	Spam/Phishing
Compromised System/Malware Infection	Malware Infection	SSH Brute-Force Attack
DDoS Attack	Malware Infection/Compromise	Suspicious Network Activity
Distributed Denial of Service (DDoS)	Malware/Compromised Host	Unauthorized Access Attempt
Distributed Denial-of-Service (DDoS)	Network Intrusion	Vulnerability
Distributed Denial-of-Service (DDoS) Attack	Network Misconfiguration/Vulnerability	Vulnerability Alert
Email Abuse/Spam	Network Scanning/Reconnaissance	Vulnerability Assessment
Email Account Compromise	Network Security	Vulnerability Exploitation
Email Security	Network Security Vulnerability	Vulnerability Management
Fraud/Phishing	Phishing	Website Defacement
Information	Port Scanning/Brute Force Attack	