UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Patrick Andrei Caron Guerra

AGRUPAMENTO DE RECOMENDAÇÕES DE RACIOCÍNIO BASEADO EM CASOS NA RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

Santa Maria, RS
2024

Patrick Andrei Caron Guerra

AGRUPAMENTO DE RECOMENDAÇÕES DE RACIOCÍNIO BASEADO EM CASOS NA RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

Orientador: Prof. Dr. Luís Alvaro de Lima Silva

Coorientador: Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS
2024

Guerra, Patrick Andrei Caron
Agrupamento de Recomendações de Raciocínio Baseado em Casos na Resposta a Incidentes de Segurança Cibernética / Patrick Andrei Caron Guerra.- 2024.
130 p.; 30 cm

Orientador: Luís Alvaro de Lima Silva
Coorientador: Raul Ceretta Nunes
Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Ciência da Computação , RS, 2024

1. Raciocínio Baseado em Casos 2. Agrupamento de Dados 3. Ontologias de Aplicação 4. Resposta a Incidentes 5. Segurança Cibernética I. Silva, Luís Alvaro de Lima II. Nunes, Raul Ceretta III. Título.

Sistema de geração automática de ficha catalográfica da UFSM. Dados fornecidos pelo autor(a). Sob supervisão da Direção da Divisão de Processos Técnicos da Biblioteca Central. Bibliotecária responsável Paula Schoenfeldt Patta CRB 10/1728.

Declaro, PATRICK ANDREI CARON GUERRA, para os devidos fins e sob as penas da lei, que a pesquisa constante neste trabalho de conclusão de curso (Dissertação) foi por mim elaborada e que as informações necessárias objeto de consulta em literatura e outras fontes estão devidamente referenciadas. Declaro, ainda, que este trabalho ou parte dele não foi apresentado anteriormente para obtenção de qualquer outro grau acadêmico, estando ciente de que a inveracidade da presente declaração poderá resultar na anulação da titulação pela Universidade, entre outras consequências legais.

Patrick Andrei Caron Guerra

AGRUPAMENTO DE RECOMENDAÇÕES DE RACIOCÍNIO BASEADO EM CASOS NA RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação**.

Aprovado em 26 de abril de 2024:

_____

Luís Alvaro de Lima Silva, Dr. (UFSM)
(Presidente/Orientador)

_____

Ana Trindade Winck, Dra. (UFCSPA) (videoconferência)

_____

Carlos Raniery Paula dos Santos, Dr. (UFSM)

Santa Maria, RS
2024

DEDICATÓRIA

À minha família, cujo amor e apoio incondicionais foram a minha força e inspiração em cada passo deste caminho.

Àqueles que ousam sonhar e, através dos sonhos, revelam o universo a si mesmos.

Àqueles que plantam sementes de esperança em solos de desafios, cultivando futuros onde sonhos podem florescer.

À luz que brilha na escuridão, guiando-nos pelo desconhecido.

Aos que acreditam no poder das pequenas ações para mudar o mundo.

AGRADECIMENTOS

Ao chegar ao final desta jornada, muitas são as pessoas que tenho a agradecer por terem contribuído, de diversas formas, para a realização deste trabalho e para o meu crescimento pessoal e profissional ao longo do mestrado.

Em primeiro lugar, gostaria de expressar minha mais profunda gratidão aos meus pais, Cleonice e Sergio, pelo amor incondicional, apoio e compreensão que me proporcionaram desde sempre. Sem a base sólida que me deram, nada disto teria sido possível.

Não poderia deixar de expressar minha mais profunda gratidão à minha família, pelo amor incondicional, compreensão e apoio inabalável em todos os momentos desta jornada. Vocês foram a minha fortaleza nos momentos difíceis e a fonte de minha inspiração e motivação para seguir em frente. Agradeço por sempre acreditarem em mim e por estarem ao meu lado, celebrando cada conquista e oferecendo conforto em cada desafio. O apoio de vocês foi essencial não apenas para a realização deste trabalho, mas em cada passo que dei na vida. A minha gratidão a vocês é eterna.

Agradeço também aos meus amigos, Beatris, Eduardo, Inácio, Adriano, Ana, Lucas, Fernando, Leandro, Kelvis, Daniel e Andressa, cuja presença e apoio foram essenciais nos momentos de alegria e, especialmente, nos desafios. A amizade de vocês é um dos maiores tesouros que levo desta fase.

Aos colegas e ex-colegas do mestrado, Diego e Crhistopher, agradeço pelas discussões enriquecedoras, pela camaradagem e pelo ambiente colaborativo que construímos juntos. Vocês tornaram este percurso acadêmico muito mais valioso e significativo.

Um agradecimento especial aos servidores da UFSM, em especial, Marcelo, Fábio e Guilherme, pela ajuda e informações sempre fornecidas com prontidão e gentileza. A disposição em auxiliar foi fundamental para a realização deste trabalho.

Agradeço ao meu orientador, Prof. Dr. Luís Alvaro de Lima Silva, e ao meu coorientador, Prof. Dr. Raul Ceretta Nunes, pela paciência, dedicação e sabedoria com que me guiaram nesta pesquisa. Suas orientações, críticas construtivas e incentivo foram cruciais para o desenvolvimento deste estudo e para o meu amadurecimento como pesquisador.

Além disso, não posso deixar de expressar minha imensa gratidão ao meu amigo, Prof. Dr. Sidnei Renato Silveira, pelo incentivo e força que me ofereceu nos momentos mais desafiadores desta jornada. Suas palavras de encorajamento foram pilares fundamentais para a minha perseverança e sucesso.

Por fim, expresso minha gratidão aos integrantes da banca, Prof.ª Dra. Ana Trindade Winck e Prof. Dr. Carlos Raniery Paula dos Santos, por aceitarem avaliar este trabalho e por contribuírem com sugestões valiosas que enriqueceram esta dissertação.

A todos vocês, meu sincero obrigado!

“Palavras são [...] nossa inesgotável fonte de magia. Capazes de causar grandes sofrimentos e também de remediá-los.”

(Alvo Dumbledore)

RESUMO

AGRUPAMENTO DE RECOMENDAÇÕES DE RACIOCÍNIO BASEADO EM CASOS NA RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

AUTOR: Patrick Andrei Caron Guerra
Orientador: Luís Alvaro de Lima Silva
Coorientador: Raul Ceretta Nunes

Raciocínio Baseado em Casos (Case-Based Reasoning - CBR) e Agrupamento de Dados (clustering) têm reconhecida relevância na resolução de diferentes problemas de aplicação. Entretanto, a integração destas técnicas de Inteligência Artificial (IA) ainda apresenta desafios de pesquisa importantes relacionados ao desenvolvimento de sistemas na área de segurança cibernética. O problema de pesquisa abordado nesta dissertação envolve como recomendar e reusar planos de resposta para incidentes de segurança cibernética. Neste contexto, a dinâmica e sofisticação crescente dos ataques cibernéticos e a exploração de vulnerabilidades instituem a necessidade de novas abordagens de IA que auxiliem na manutenção da resiliência cibernética em organizações. Este trabalho investiga a construção de uma memória reusável de experiências de resposta a incidentes de segurança cibernética, capturando experiências em estruturas de casos armazenadas numa base de casos. Casos possuem detalhes do contexto do incidente (problema) e planos com ações de resposta (solução). Métodos de similaridade são empregados para consultar essa memória, partindo de um contexto de incidente especificado (consulta), para recomendar casos relevantes para reúso. As principais contribuições deste trabalho incluem: o desenvolvimento de um método que integra CBR e clustering na organização das soluções recuperadas em clusters; e a modelagem de uma nova ontologia de aplicação para facilitar a aquisição e representação de planos de resposta a incidentes. Experimentos de validação cruzada e com novos incidentes foram desenvolvidos para avaliação da abordagem proposta. Os resultados indicam que a integração de CBR e clustering pode aumentar a precisão na seleção de planos de resposta para reúso, especialmente quando os analistas de segurança conseguem identificar e escolher o grupo mais adequado resultante do agrupamento das recomendações apresentadas para consultas CBR. A seleção aleatória de um grupo de recomendações pode apresentar resultados equivalentes de precisão ao uso exclusivo de consultas CBR. Por outro lado, a seleção do pior grupo obtido implica numa queda de precisão em relação ao uso exclusivo de recomendações apresentadas em consultas CBR. Isso demonstra que o refinamento de recomendações obtidas para consultas CBR, com base em clustering, pode otimizar a análise e o reúso de recomendações na resposta a incidentes, embora a seleção de grupos de casos obtidos e realizada pelo analista possa impactar significativamente nos resultados de precisão alcançados.

Palavras-chave: Raciocínio Baseado em Casos. Agrupamento de Dados. Ontologias de Aplicação. Resposta a Incidentes. Segurança Cibernética.

ABSTRACT

CLUSTERING OF CASE-BASED REASONING RECOMMENDATIONS IN CYBERSECURITY INCIDENT RESPONSE

AUTHOR: Patrick Andrei Caron Guerra
ADVISOR: Luís Alvaro de Lima Silva
CO-ADVISOR: Raul Ceretta Nunes

Case-Based Reasoning (CBR) and Clustering are recognized for their relevance in solving various application problems. However, integrating these Artificial Intelligence (AI) techniques still presents significant research challenges related to the development of systems in the field of cybersecurity. The research problem addressed in this dissertation involves recommending and reusing response plans for cybersecurity incidents. In this context, the increasing dynamics and sophistication of cyber attacks and the exploitation of vulnerabilities establish the need for new AI approaches that assist in maintaining cyber resilience in organizations. This work investigates the construction of a reusable memory of cybersecurity incident response experiences, capturing experiences in case structures stored in a case base. Cases contain details of the incident context (problem) and plans with response actions (solution). Similarity methods are employed to query this memory, starting from a specified incident context (query), to recommend relevant cases for reuse. The main contributions of this work include: the development of a method that integrates CBR and clustering in organizing the retrieved solutions into clusters; and the modeling of a new application ontology to facilitate the acquisition and representation of incident response plans. Cross-validation experiments and with new incidents were developed to evaluate the proposed approach. The results indicate that the integration of CBR and clustering can increase the precision in selecting response plans for reuse, especially when security analysts can identify and choose the most appropriate group resulting from the clustering of recommendations presented for CBR queries. Random selection of a group of recommendations can yield precision results equivalent to the exclusive use of CBR queries. On the other hand, selecting the worst group obtained implies a decrease in precision compared to the exclusive use of recommendations presented in CBR queries. This demonstrates that refining recommendations obtained for CBR queries, based on clustering, can optimize the analysis and reuse of recommendations in incident response, although the selection of case groups obtained and carried out by the analyst can significantly impact the precision results achieved.

Keywords: Case-Based Reasoning. Clustering. Application Ontologies. Incident Response. Cybersecurity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AL	Average Linkage
C&C	Command and Control
CBR	Case-Based Reasoning
CL	Complete Linkage
CSIRT	Computer Security Incident Response Team
CSOC	CyberSecurity Operations Center
CVE	Common Vulnerabilities and Exposures
CVO	Cybersecurity Vulnerability Ontology
DDoS	Distributed Denial of Service
DNS	Domain Name System
FP	Falso Positivo
FN	Falso Negativo
GTSeg	Gestão e Tecnologia em Segurança da Informação
GMT	Greenwich Mean Time
HC	Hierarchical Clustering
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IODEF	Incident Object Description Exchange Format
IoT	Internet of Things
IP	Internet Protocol
K-Med	K-Medoids
K-NN	K-Nearest Neighbors

LOO	Leave-One-Out
LR	Limiar de Recuperação
NAT	Network Address Translation
OWL	Web Ontology Language
PDF	Portable Document Format
SL	Single Linkage
SIEM	Security Information and Event Management
SOC	Security Operations Center
SO	Sistema Operacional
SSL	Secure Socket Layers
STIX	Structured Threat Information eXpression
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1 INTRODUÇÃO

Investimentos extensivos têm sido realizados para melhorar a segurança cibernética de ativos de informação em organizações. Entretanto, a dinamicidade de estratégias de ataque nem sempre podem ser detectadas por equivalentes estratégias de proteção. Além disso, vulnerabilidades de diferentes naturezas dão margem à continuada ocorrência de incidentes de segurança. Como descrito em Carias et al. (2020), esses problemas apontam para a necessidade de novas abordagens, a fim de manter a resiliência cibernética em organizações.

Segurança cibernética é cada vez mais importante devido à informatização de processos e atividades desenvolvidas nas organizações. O tamanho da organização (ANSON, 2020) não representa mais um grande fator para os atacantes na hora de selecionar as empresas que serão vítimas. Com a alta vulnerabilidade dos sistemas computacionais, a questão não é mais se o incidente irá ocorrer, mas quando deverá acontecer. Assim como investigado nesse trabalho, esse problema impõe a necessidade de implementação de estratégias reativas e proativas para a defesa dos sistemas.

A manutenção de um ambiente organizacional seguro envolve esforços voltados para a resposta a incidentes de segurança cibernética (KAUR; GABRIJELČIČ; KLOBUČAR, 2023). Embora muito do esforço ainda seja direcionado para a prevenção e detecção de incidentes, essas tarefas não são suficientes para manutenção de um ambiente seguro (LAHCEN et al., 2020). É importante notar que a resposta para a ocorrência de incidentes deve tratar esses problemas, assim como auxiliar na prevenção de novos incidentes. Para atacar esse problema, a estratégia investigada neste trabalho é o desenvolvimento de sistemas inteligentes para apoiar as organizações na coleção e reúso de planos de resposta a ocorrências de incidentes. Esses planos devem manter informações organizadas e detalhadas sobre quais ações de resposta que analistas de segurança devem executar. Portanto, este trabalho identifica que a reutilização de experiências passadas, empregadas na resposta de incidentes, tem um papel significante na mitigação de novos problemas de segurança.

Experiências concretas de resposta a incidentes podem ser mantidas em repositórios de conhecimento no formato de casos para sistemas CBR (AAMODT; PLAZA, 1994). Isso possibilita o reúso dessas experiências na resposta a novos incidentes. Diferentes estratégias podem ser agregadas para a construção e manutenção desses planos de resposta, como, por exemplo, documentar o ambiente técnico da organização. É possível nomear dispositivos, identificar serviços hospedados, enumerar portas abertas, identificar protocolos utilizados e agentes de usuários presentes na rede. É recomendado, também, armazenar relatórios de tráfego de rede que possam fornecer parâmetros para a análise e identificação de ataques com base na análise do tráfego. No entanto, a captura dos inci-

dentes e dos procedimentos empregados na resposta destes problemas não é uma tarefa trivial. Entre outras dificuldades, descrições informais (geralmente usando texto livre, por exemplo) de procedimentos de resolução ocasionam uma duplicidade de ações de resposta para incidentes com características similares. Por exemplo, uma ação de resposta associada à adição de um dispositivo *Internet of Things* (IoT) em uma rede dedicada para hosts IoT pode ser descrita por um analista como *"Set up a Host in a Segregated Subnet for IoT Devices"*, e por outro analista como *"Add a Host to the IoT Subnet for Network Isolation"*. Logo, a aquisição e representação de conhecimento de resposta a incidentes de segurança é um problema importante neste domínio. Neste trabalho, este problema é abordado com a modelagem de uma ontologia de aplicação voltada para representação de ações de resposta a incidentes.

Uma vez que memórias contendo experiências de resposta a incidentes de segurança sejam construídas, mecanismos de recuperação de informações, geralmente usados em sistemas de recomendação, podem utilizar ranqueamento baseados em medidas de similaridade (RICCI; ROKACH; SHAPIRA, 2022). Como resultado, sistemas de recomendação construídos com essas técnicas são capazes de apresentar aos analistas de segurança problemas e soluções similares aos problemas capturados nas consultas formuladas e executadas. Em muitos sentidos, as respostas para essas consultas podem apoiar a obtenção de informações que podem significativamente guiar a resposta a incidentes de segurança. O problema é que consultas com informações vagas ou ambíguas sobre incidentes preliminarmente analisados podem resultar na recuperação de uma variedade/diversidade grande de experiências de resposta a incidentes armazenados nessas memórias. Isso dificulta o processo de reúso de procedimentos de resposta, uma vez que nem sempre os casos mais similares recomendados como resultado de consultas executadas contêm as ações de resposta a incidentes mais relevantes/indicadas para serem aplicadas na solução do problema corrente.

Em geral, a apresentação de uma lista de experiências de solução de problemas contendo recomendações de procedimentos de resposta a incidentes, que pode ser grande e/ou com uma grande variedade de situações e problemas, pode dificultar o processo de análise e reúso de informações pelo analista de segurança. Quando isso ocorre, o reúso de planos de resposta recuperados para consultas é dificultado.

Variadas técnicas de IA têm sido exploradas na resposta a incidentes de segurança. Essas técnicas contribuem para a predição de ataques cibernéticos e a mitigação dos seus impactos (SUN et al., 2018; ZHANG et al., 2022). No entanto, nem sempre as técnicas de IA implementam Inteligência Artificial Explicável (XAI), o que pode tornar compreensível os processos de decisão executados, possibilitando que os usuários entendam, confiem e gerenciem as soluções apresentadas para problemas de segurança cibernética. Entretanto, técnicas como Raciocínio Baseado em Casos (*Case-Based Reasoning* - CBR) e clustering possuem relevantes capacidades explanatórias, promovendo maior transparên-

cia e compreensibilidade nos processos de resposta a incidentes. Neste caso, o uso de técnica de CBR permite recuperar (recomendar) soluções registradas em casos passados para resolver problemas novos, permitindo aos usuários reusar procedimentos de resposta a incidentes experimentados no passado. Por sua vez, o clustering organiza casos de resposta a incidentes em grupos significativos, com base em suas características, facilitando a análise e a compreensão de padrões de resposta por analistas de segurança. A combinação dessas técnicas não só pode organizar a resposta prática a incidentes de segurança cibernética, como também apoiar uma tomada de decisões mais informada, aumentando, consequentemente, a confiança nos sistemas de segurança baseados por IA.

Assim como investigado neste trabalho, planos de resposta a incidentes de segurança podem ser detalhados por ações de resposta de diferentes naturezas (por exemplo, analíticas, corretivas, etc). A execução dessas ações aborda o problema de aquisição e representação de dados e conhecimento de resposta a incidentes. O problema é que diferentes analistas de segurança podem descrever uma mesma ação de resposta a incidentes de diferentes formas. Além disso, repositórios que armazenam essas ações para futuro reúso apresentam extensa falta de padronização, o que prejudica a avaliação dos planos de resposta a incidentes possivelmente reusáveis na resposta de novos incidentes. Para abordar esse problema de aquisição e representação de conhecimento de segurança cibernética, este trabalho explora o emprego de ontologias de aplicação (JAKUS et al., 2013; SÁNCHEZ-ZAS et al., 2023; SYED, 2020).

Além do apoio à aquisição e representação de planos de resposta a incidentes de segurança cibernética, o emprego de sistemas de segurança cibernética está fortemente relacionado ao armazenamento, recuperação e reúso de dados, além do conhecimento de experiências concretas de resposta a incidentes de segurança, tal como proposto em trabalhos passados, desenvolvidos no grupo de pesquisa[1] onde esta dissertação está inserida (NUNES et al., 2019; BARCELOS, 2020; GUERRA et al., 2023). Nesse contexto, o framework de CBR (AAMODT; PLAZA, 1994; RICHTER; WEBER, 2013) permite que analistas de segurança mantenham repositórios sobre incidentes e seus planos de resposta no formato de casos. Com isso, essas experiências concretas de solução de problemas podem ser recuperadas de uma memória e reusadas na melhor resposta a novos incidentes de segurança. Para responder a esses incidentes, analistas de segurança descrevem esses problemas como consultas a serem executadas nesses sistemas, sendo que essas consultas capturam as principais informações disponíveis sobre os incidentes detectados. Entretanto, a forma como tais consultas são inicialmente descritas é suscetível a vários problemas, visto que informações sobre incidentes correntes podem estar incompletas ou incorretas por diferentes motivos.

Para atacar esse problema, este trabalho explora diferentes técnicas de clustering (EVERITT et al., 2011) na análise e reúso de experiências de resposta a incidentes, re-

[1] Gestão e Tecnologia em Segurança da Informação (GTSeg)

cuperadas de bases de conhecimento. Motivado pelo que é desenvolvido em sistemas de Recuperação de Informações (SADAF; ALAM, 2012) na web, este trabalho inova pela análise e organização de resultados de consultas em diferentes grupos, permitindo, assim, filtrar as informações mais relevantes para o reúso de procedimentos de resposta a incidentes correntes. A investigação apresentada neste trabalho emprega técnicas de XAI, especialmente a combinação de CBR e clustering.

No contexto de pesquisa apresentado, as principais contribuições deste trabalho são:

a) um método de emprego de uma nova ontologia de aplicação para apoiar o processo de aquisição e representação de planos de respostas a incidentes de segurança (soluções para esses problemas) em casos para CBR;

b) um método de emprego de algoritmos de clustering na organização e priorização de planos de resposta a incidentes de segurança recuperados por consultas CBR, as quais são construídas para expressar as principais características de problemas de segurança correntes que precisam ser resolvidos;

c) uma solução que integra CBR e clustering para aprimorar a resposta a incidentes de segurança cibernética em organizações.

1.1 ESTRUTURA DO TEXTO

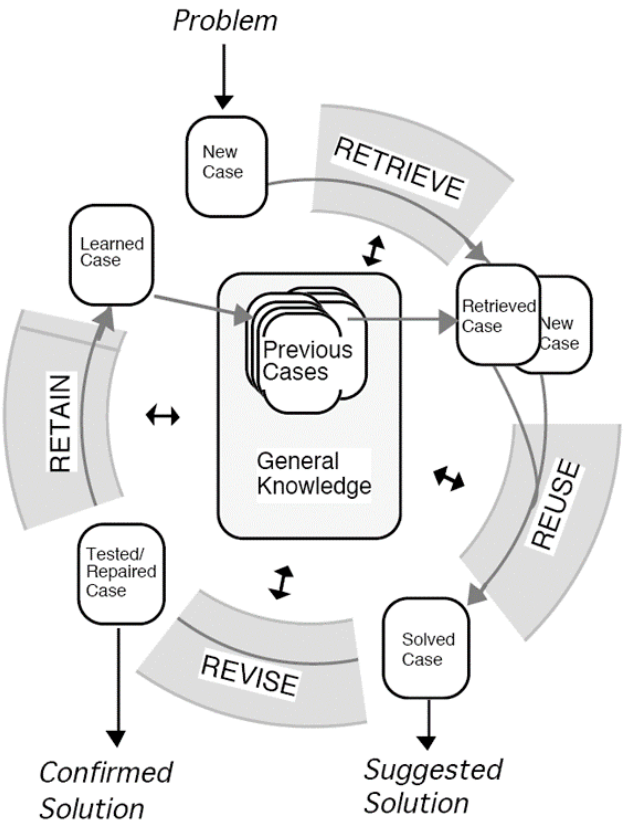
O texto está estruturado da seguinte forma: o Capítulo 2 aborda o referencial teórico desta pesquisa. O Capítulo 3 descreve os trabalhos relacionados. O Capítulo 4 discute o emprego de ontologias na aquisição e representação de planos de resposta a incidentes. A representação e o fluxo de resposta a incidentes de segurança cibernética, com base em técnicas de CBR e clustering, são explorados no Capítulo 5. O Capítulo 6 detalha o sistema de resposta a incidentes de segurança cibernética implementado. Um exemplo de caso de resposta a incidente de segurança é explorado no Capítulo 7. Já no Capítulo 8 são apresentados os experimentos desenvolvidos, os resultados e as discussões acerca destes. Por fim, o Capítulo 9 explora conclusões e possibilidades de trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 RACIOCÍNIO BASEADO EM CASOS - CBR

CBR permite que soluções utilizadas em experiências de solução de problemas passados sejam reaplicadas na resolução de novos problemas. Em sistemas CBR, as coleções onde tais experiências ficam armazenadas, em um formato de “casos”, são conhecidas como base de casos. De acordo com (RICHTER; WEBER, 2013), cada caso é composto por uma lista de atributos capazes de representar tais experiências. Para permitir que experiências de solução de problemas sejam utilizadas para propor soluções em novos problemas, diferentes etapas de raciocínio podem ser realizadas. Em sistemas CBR, estas etapas compõem o ciclo CBR (MANTARAS et al., 2005), o qual pode ser dividido em: recuperação, reúso, revisão e retenção. A Figura 1 apresenta o Ciclo de CBR.

Figura 1 – Ciclo CBR.



Fonte: (AAMODT; PLAZA, 1994).

A fase de recuperação visa calcular a similaridade entre os casos, identificando aqueles que são mais próximos ao problema especificado na consulta. O método *K-Nearest Neighbors* (K-NN) (RICHTER; WEBER, 2013) é amplamente empregado para

identificar os casos mais similares presentes na base de casos, com esse processo sendo efetuado por meio do uso de métricas de similaridade. Existem dois principais tipos de cálculo de similaridade: locais e globais. Os métodos de similaridades locais avaliam a similaridade em cada atributo individual que compõe a representação dos casos, variando as métricas de acordo com a natureza do atributo e sua relevância para a aplicação. Isso resulta em uma diversidade de métodos de similaridade locais, cada um projetado com diferentes finalidades. O método de similaridade global, por sua vez, agrega os valores das similaridades locais para avaliar a semelhança geral entre a consulta e os casos na base de casos. A similaridade global, usada na recomendação de casos na etapa de recuperação no sistema CBR, desenvolvido nesta dissertação, é expressa pela Equação (2.1):

$$Similarity_{global}(q, c) = \frac{1}{|A|} \sum_{i=1}^{A} Similarity_{local}(q_i, c_i) \tag{2.1}$$

Na Equação (2.1), $Similarity_local$ denota o cálculo de similaridade local, q refere-se à consulta (atributos do problema do caso consultado), A representa os atributos considerados para o cálculo da similaridade global e $|A|$ o número de atributos avaliados na consulta. O valor de similaridade global é obtido pela soma dos valores de similaridades locais entre o caso consultado e os casos armazenados ($CB = \{c_1, c_2, ..., c_n\}$), dividida pelo número de atributos analisados (presentes em ambos, na consulta e no caso armazenado sendo comparado).

Com base na Equação (2.1), o algoritmo K-NN seleciona um conjunto de casos que são suficientemente semelhantes a uma consulta específica q, com base em um Limiar de Recuperação (*Retrieval Threshold* - LR) definido. A similaridade global entre a consulta e cada caso (q) presente na base de casos (CB) é calculada, e apenas aqueles casos cuja similaridade é maior ou igual ao LR definido são recuperados. O Algoritmo (1) apresenta o uso do algoritmo K-NN, explorado nesta dissertação, para realizar a recomendação de casos para uma determinada consulta CBR.

No Algoritmo (1), a entrada é composta por três parâmetros: a base de casos (CB), a consulta (q) e o LR ($RetrievalThreshold$). A saída do algoritmo é uma lista ordenada de casos recuperados ($retrievedCases$), onde cada caso é acompanhado de sua similaridade (global), calculada em relação à consulta, e todos são ordenados de forma decrescente pela similaridade. Assim, o algoritmo permite identificar os casos mais similares ao problema representado na consulta.

Após recuperar uma lista dos casos mais similares presentes na base de casos, esses casos são examinados para que haja uma seleção da solução mais adequada ao problema em questão. Essa tarefa seria mais simples se fosse possível manter uma base de casos abrangendo todas as variantes de problemas dentro do domínio em estudo. No entanto, a seleção da melhor solução muitas vezes requer a análise de vários casos similares, dado que é raro encontrar um caso passado que se alinhe perfeitamente com o

Algoritmo 1 Método de Recuperação em sistemas CBR com base em K-NN para recomendação de casos

procedure RETRIEVAL($CB, q, RetrievalThreshold$)
    $retrievedCases \leftarrow []$
    for each c in CB **do**
        $similarity \leftarrow Similarity_{global}(q, c)$
        if $similarity \geq RetrievalThreshold$ **then**
            append $(similarity, c)$ to $retrievedCases$
        end if
    end for
    $retrievedCases \leftarrow sort(retrievedCases, descending = True)$
    return $retrievedCases$
end procedure

problema atual.

Além de escolher a solução a ser aplicada, a fase de reutilização também adapta as soluções dos casos recuperados para atender às especificidades do novo problema. Assim, pode-se afirmar que esta fase engloba duas subfases: a seleção da solução (definida pelas políticas de reutilização) e a adaptação da solução ao contexto atual. Frequentemente, o processo de adaptação começa durante a implementação da política de reutilização e só se conclui ao fornecer uma solução para o problema.

Concluída a fase de reutilização, a solução adaptada passa pela fase de revisão, onde se avalia sua eficácia na resolução do problema. Esta etapa, conforme descrito por Richter e Weber (2013), visa aprimorar a qualidade das soluções utilizando conhecimentos que não estavam disponíveis nos casos analisados.

Uma vez que a solução proposta tenha sido revisada, essa nova experiência pode ser integrada à base de casos. A construção de sistemas inteligentes demanda que este processo de aprendizado ocorra continuamente, especialmente se a base inicial de casos não cobre adequadamente os problemas que o sistema visa resolver. No contexto do CBR, a fase de retenção é crucial para este aprendizado, estabelecendo critérios para o enriquecimento da base de casos com uma vasta gama de problemas e soluções. A aprendizagem em sistemas CBR se dá pela incorporação de novas experiências à base de casos, permitindo que as políticas de reutilização proponham soluções para desafios inéditos. Assim, quanto mais abrangente for a base de casos, mais eficazes serão as soluções sugeridas.

2.2 ONTOLOGIAS

A construção e manutenção de ontologias (GUARINO, 1995) representam um dos mais importantes avanços da IA para apoiar tarefas de aquisição e representação de conhecimento. Ontologias descrevem formas de especificar explicitamente um conjunto de conceitos compartilhados. Os conceitos (JAKUS et al., 2013) são utilizados para expressar e organizar termos comuns da terminologia do domínio explorado, cobrindo parcialmente ou completamente um domínio-alvo.

Ontologias são artefatos de engenharia de conhecimento (SCHREIBER et al., 2000), que podem representar uma hierarquia de conceitos relacionados entre si. Com isso, as ontologias permitem a definição de conceitos de forma padronizada, aprimorando o reúso dos conceitos definidos e amenizando problemas como ambiguidades na definição de descrições, repetição e/ou duplicação de conceitos e termos. Também oportunizam uma forma estruturada de retenção de conhecimentos e a definição de relações entre conceitos, possibilitando a representação de conceitos de um ou mais domínios (JAKUS et al., 2013).

A ontologia define uma terminologia que é compreensível para os usuários deste modelo reusável. Elas são construídas principalmente para: a) oportunizar um entendimento comum de estruturas de informação, tanto para pessoas quanto para agentes inteligentes (EHRLINGER; WÖSS, 2016); b) proporcionar a reutilização de conhecimento de um determinado domínio, permitindo que, por meio de uma modelagem adequada de um determinado conhecimento, a ontologia possa ser compartilhada e reutilizada na construção de outras ontologias relacionadas ao domínio; c) definir, de forma explícita, suposições de um determinado domínio, com base no uso de um vocabulário de conceitos bem definidos, que evitem a ocorrência de conceitos ambíguos que possam levar a diferentes interpretações (JAKUS et al., 2013).

Na construção de ontologias, os principais componentes usados incluem as classes, relações, axiomas e instâncias. As classes representam os conceitos, de forma ampla. As relações representam associações entre os conceitos e outros componentes. Os axiomas formais definem frases que são verdadeiras, normalmente usadas para verificar a consistência da ontologia e também a realização de inferências de novos conhecimentos. Finalmente, as instâncias representam elementos ou *individuals* na ontologia (CALERO; RUIZ; PIATTINI, 2006).

A construção de uma ontologia pode ser conduzida/orientada conforme um conjunto de etapas, que auxiliará na obtenção de uma ontologia útil para o domínio-alvo (LIVITCKAIA et al., 2019). A seguir, conforme a Figura 2, estas etapas são as seguintes:

i) Especificação: Esta é a fase inicial onde o domínio, objetivo, escopo, linguagem de implementação, usuários-alvo e usos pretendidos da ontologia são especificados. Aqui, define-se o contexto e os limites dentro dos quais a ontologia operará, bem como os seus objetivos e a audiência que ela atenderá.

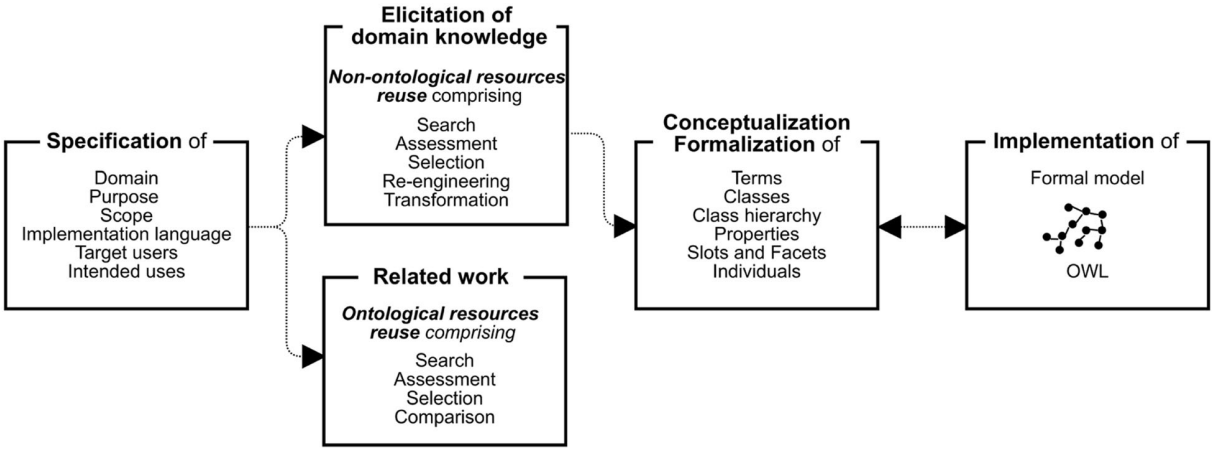
ii) Elicitação do conhecimento do domínio: Neste estágio, o conhecimento do domínio é reunido usando recursos não ontológicos. Isso envolve a procura e avaliação de informações relevantes, a seleção das partes mais pertinentes e, em seguida, a reengenharia e transformação dessas informações para atender às necessidades da ontologia. Esta fase é crítica para assegurar que a ontologia reflita com precisão o domínio a ser modelado.

iii) Trabalhos relacionados: Este estágio envolve a análise de recursos ontológicos de trabalhos relacionados, seguindo etapas semelhantes: pesquisa, avaliação, seleção e comparação. Aqui, busca-se evitar a reinvenção da roda, aproveitando os esforços prévios que podem ser adaptados ou estendidos para o novo contexto.

iv) Conceptualização/Formalização: Aqui, a estrutura real da ontologia é desenvolvida. Termos são definidos, classes e hierarquias de classes são estabelecidas, propriedades e instâncias de classes são identificadas. Este passo é fundamental para a criação de uma ontologia bem estruturada e capaz de representar o conhecimento do domínio de forma eficaz.

v) Implementação: Por fim, a ontologia é implementada como um modelo formal, utilizando frequentemente a *Web Ontology Language* (OWL), que permite que a ontologia seja utilizada por sistemas computacionais. A OWL inclusive foi adotada na construção da ontologia deste trabalho. Esta etapa materializa a ontologia de maneira que possa ser utilizada por sistemas de informação, permitindo que sistemas computacionais processem e realizem inferência de conhecimentos a partir dos dados estruturados pela ontologia.

Figura 2 – Processo de Desenvolvimento de Ontologia.



Fonte: (LIVITCKAIA et al., 2019).

A construção de ontologias (STAAB; STUDER, 2010) envolve um processo interativo e contínuo, que vai sendo posto em prática e aprimorado à medida que novos conceitos e inter-relações são definidos. Um conjunto de atividades pode ser identificado nesse contexto, buscando assegurar a aplicabilidade da ontologia na área-alvo. O processo de refinamento da ontologia inclui assegurar que ela reflita com qualidade o domínio de aplicação a que se propõe descrever/formalizar. Isso envolve avaliar necessidades de alteração na estrutura, nos conceitos e relacionamentos, a remoção de conceitos que não sejam relevantes para o contexto da ontologia, bem como adicionar novos conceitos, relacionamentos e outros materiais adicionais que possam auxiliar na representação do domínio.

Embora existam diferentes tipos de ontologias, ontologias de aplicação são relevantes para este trabalho, já que desempenham um papel crucial ao conectar conceitos específicos de um domínio a tarefas particulares (CALERO; RUIZ; PIATTINI, 2006). Isso permite uma representação mais detalhada e contextualizada do conhecimento de um domínio específico. Ao fazer isso, as ontologias de aplicação não apenas enriquecem a compreensão de um domínio com base na adição de dimensões relacionadas a aspectos operacionais, como também melhoram a interoperabilidade e a capacidade de reúso de ontologias ao integrar os conceitos de um domínio com as necessidades práticas em tarefas específicas. Neste trabalho, uma nova ontologia de aplicação, com foco em ações de resposta a incidentes de segurança cibernética, foi construída.

2.3 CLUSTERING

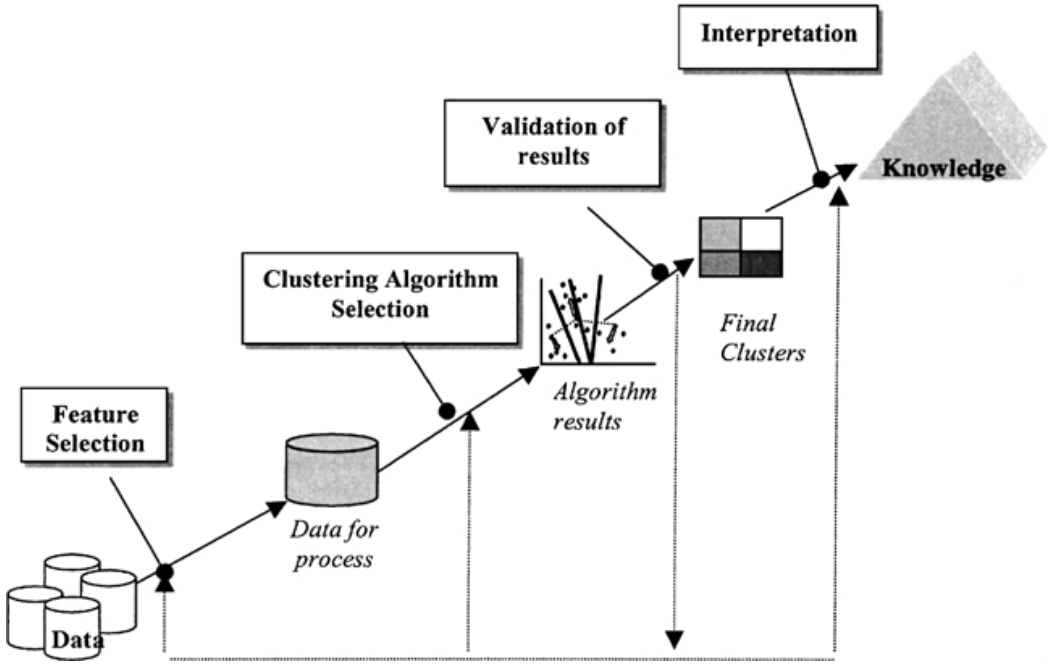
Clustering (EVERITT et al., 2011) envolve a descoberta de grupos, considerando a existência de um conjunto de n objetos a serem agrupados em q grupos. Este trabalho explora técnicas de Agrupamento Hierárquico (*Hierarchical Clustering* - HC) e K-Medoids (K-Med) (ROKACH; MAIMON, 2005).

O diagrama apresentado na Figura 3 descreve um processo estruturado para a realização do agrupamento de dados e análise de cluster. Este processo é descrito a seguir.

Inicialmente, com base nos atributos e/ou características escolhidos, os dados são processados e preparados, o que pode incluir passos como normalização ou transformação de dados para garantir que o algoritmo de clustering funcione de maneira adequada. Esta etapa assegura que as variações nos dados reflitam diferenças reais no comportamento ou nas propriedades que estamos tentando agrupar, e não distorções causadas por escalas ou unidades inconsistentes. Nesta pesquisa, os atributos do problema do caso utilizam os mesmos métodos de cálculo de similaridade local empregados na recuperação CBR.

Após a preparação dos dados, a seleção do algoritmo de clustering a ser utilizado

Figura 3 – Processo de Análise de Cluster.



Fonte: (HALKIDI; BATISTAKIS; VAZIRGIANNIS, 2001).

deve ocorrer. Há muitos algoritmos disponíveis, cada um com suas próprias forças e suposições, e a escolha pode ser influenciada por diversos fatores, incluindo o tipo de dado, a presença de ruído e o número de clusters desejado. Neste trabalho, os algoritmos de HC com os critérios de ligação *Single Linkage* (SL), *Average Linkage* (AL) e *Complete Linkage* (CL) e K-Med são empregados.

Quando um algoritmo é aplicado aos dados, ele identifica clusters com base nas características selecionadas. Estes são os grupos naturais em que os dados são divididos, de acordo com as similaridades que o algoritmo foi capaz de encontrar entre os grupos.

A etapa subsequente envolve a validação dos resultados. Técnicas de validação, como análise de silhueta ou índices de validação interna, podem ser utilizadas para avaliar a qualidade dos clusters obtidos.

Com os clusters validados, o próximo passo é a interpretação. Isso requer compreensão do contexto e conhecimento de domínio para entender o que cada cluster representa. A interpretação faz com que os dados se transformem em conhecimento útil, permitindo a tomada de decisões estratégicas com base neste conhecimento.

Por fim, o conhecimento adquirido com base no desenvolvimento deste processo pode ser aplicado em diversas áreas para resolver problemas, orientar decisões futuras, ou descobrir oportunidades. No contexto deste trabalho, o conhecimento deve orientar a seleção de um grupo de recomendações para reúso na resposta a novos incidentes. O diagrama enfatiza que o processo de agrupamento não é linear; as descobertas feitas durante a interpretação podem levar a revisões nas etapas anteriores, tornando o agrupamento uma atividade iterativa e reflexiva.

Neste estudo, uma matriz de distâncias é utilizada como entrada para os algoritmos de agrupamento, facilitando o processamento de similaridade de atributos de diferentes formatos e tipos, presentes em casos que foram recomendados em consultas CBR. O Algoritmo (2) apresenta como a matriz de distâncias pode ser construída, considerando um conjunto de casos recomendados (em uma ou mais consultas CBR).

Algoritmo 2 Método de Computação da Matriz de Distâncias entre Casos Recomendados em Consulta(s) CBR

```
1: function CalculateDistanceMatrix(RC)
2:     n ← |RC|
3:     for i ← 1 to n do
4:         for j ← 1 to n do
5:             distanceMatrix[i][j] ← 0
6:         end for
7:     end for
8:     for i ← 0 to n − 1 do
9:         for j ← i + 1 to n − 1 do
10:            distanceMatrix[i][j] ← 1 − Similarity(RC_i, RC_j)
11:            distanceMatrix[j][i] ← distanceMatrix[i][j]
12:        end for
13:    end for
14:    return distanceMatrix
15: end function
```

Conforme o Algoritmo (2) descrito, a matriz é calculada tomando como entrada um conjunto de casos recomendados em uma ou mais consultas RC, onde cada elemento representa um caso. Para cada par de casos, a distância é inversamente proporcional à sua similaridade. A diagonal da matriz é preenchida com zeros, significando que a distância de um caso para si mesmo é nula.

O algoritmo HC é baseado em matrizes, que representam os resultados obtidos de cálculos de similaridade (ou distância) entre objetos, tomados como entrada nos algoritmos. Isso permite construir um dendrograma, que basicamente representa uma árvore com os diferentes grupos encontrados. Usando um método aglomerativo, a construção dessa árvore ocorre de forma inversa, começando pelas folhas, que inicialmente contêm todos os n objetos que estão sozinhos, cada um em um respectivo grupo e/ou *cluster*. Em contraste, o método divisivo realiza a construção da árvore de forma oposta, iniciando com um *cluster* contendo todos os objetos, que são separados até que todos os n objetos estejam sozinhos. Neste trabalho, o método aglomerativo é utilizado e três critérios de ligação são empregados no algoritmo HC, descritos a seguir.

O primeiro critério de ligação abordado é o *Single Linkage* (SL) (*nearest-neighbor*). Este critério considera a menor distância entre pares de objetos (WUNSCH; XU, 2009),

sendo um objeto de um cluster e um de outro. Ele tende a criar clusters dispersos e não leva em consideração a estrutura do *cluster*, entretanto, funciona bem quando os clusters estão distantes um do outro (ROKACH; MAIMON, 2005) e quando se deseja encontrar *outliers*. O funcionamento do critério de ligação SL é descrito no Algoritmo (3).

Algoritmo 3 Critério de Ligação *Single* (SL)

function SINGLELINKAGE($clusterA, clusterB, distanceMatrix$)
  $minDistance \leftarrow \infty$
  for each point a in $clusterA$ **do**
    for each point b in $clusterB$ **do**
      if $distanceMatrix[a][b] < minDistance$ **then**
        $minDistance \leftarrow distanceMatrix[a][b]$
      end if
    end for
  end for
  return $minDistance$
end function

Outro critério de ligação explorado nesta pesquisa é o *complete* (CL), também conhecido como *furthest-neighbor*, que utiliza a distância máxima entre pares de objetos, sendo um objeto de um *cluster* e um objeto de outro, realizando o *merge* de clusters. Os clusters formados tendem a ser compactos e com o mesmo diâmetro. A implementação deste critério de ligação (CL) é ilustrada no Algoritmo (4).

Algoritmo 4 Método Critério de Ligação *Complete* (CL)

function COMPLETELINKAGE($clusterA, clusterB, distanceMatrix$)
  $maxDistance \leftarrow 0$
  for each point a in $clusterA$ **do**
    for each point b in $clusterB$ **do**
      if $distanceMatrix[a][b] > maxDistance$ **then**
        $maxDistance \leftarrow distanceMatrix[a][b]$
      end if
    end for
  end for
  return $maxDistance$
end function

Por fim, o outro critério de ligação do HC empregado neste trabalho é o *average* (AL), também conhecido como *Unweighted Pair Group Method with Arithmetic Mean*, que emprega a distância média entre clusters e realiza o *merge* da menor distância intercluster. Este critério leva em consideração todos os pontos do grupo, sendo menos sensível a

outliers que os critérios SL e CL. Este critério de ligação (AL) é explicado no Algoritmo (5).

Algoritmo 5 Método Critério de Ligação *Average* (AL)

function AVERAGELINKAGE($clusterA, clusterB, distanceMatrix$)
    $totalDistance \leftarrow 0$
    $totalPairs \leftarrow 0$
    for each point a in $clusterA$ **do**
        for each point b in $clusterB$ **do**
            $totalDistance \leftarrow totalDistance + distanceMatrix[a][b]$
            $totalPairs \leftarrow totalPairs + 1$
        end for
    end for
    $averageDistance \leftarrow totalDistance/totalPairs$
    return $averageDistance$
end function

Conforme o critério de ligação escolhido, agrupam-se os n nós, até que reste apenas o nodo raiz, onde todos os n objetos fazem parte de um único grupo. O Algoritmo (6) descreve o pseudocódigo para o HC, utilizando como entrada a matriz de distâncias e o critério de ligação.

Algoritmo 6 Método de Agrupamento Hierárquico (HC) com base na matriz de distância e critério de ligação

function HIERARCHICALCLUSTERING($distanceMatrix, linkageCriteria$)
    Initialize $clusters$ as a list where each data point is a separate cluster.
    Initialize $mergeHistory$ as an empty list to record merges.
    while the number of clusters in $clusters > 1$ **do**
        Find the pair of clusters $C1$ and $C2$ in $clusters$ that are closest based on $linkageCriteria$.
        Calculate the distance d between $C1$ and $C2$.
        Merge $C1$ and $C2$ into a new cluster C_{new}.
        Record the merge in $mergeHistory$, including $C1$, $C2$, and their distance d.
        Remove $C1$ and $C2$ from $clusters$ and add C_{new} to $clusters$.
        Update the distance matrix for C_{new}.
    end while
    return $mergeHistory$.
end function

O dendrograma resultante da execução do algoritmo HC pode ser cortado em determinado limiar de similaridade para obter a organização dos n objetos em q grupos. Esta é uma das vantagem do HC: criar múltiplas partições à medida que diferentes limiares

de similaridade são usados para fatiar o dendrograma. Além disso, ele possui mais versatilidade na criação de grupos devido às várias possibilidades de critérios de ligação, proporcionando a criação de grupos com diferentes estruturas e formas.

Outros algoritmos de clustering são baseados em centroide, como o algoritmo K-Means. Esse algoritmo busca, na maioria das vezes, uma partição ótima dos dados com base em um critério, minimizando a soma dos erros ao quadrado (EVERITT et al., 2011). Entre as suas vantagens estão a simplicidade na implementação, velocidade rápida de convergência e adaptabilidade a dados esparsos. Por outro lado, as desvantagens deste algoritmo são: sensibilidade à inicialização dos grupos, sendo que a máxima local alcançada pode ser diferente da máxima global, devido a essa inicialização; e a possibilidade de ser afetado por ruídos e *outliers* se estes forem selecionados como medoids iniciais.

Utilizado neste trabalho, o algoritmo de clustering K-Med corresponde a uma versão modificada do algoritmo K-Means. O *medoid* representa um objeto atual da amostra de dados. O método K-Med é mais robusto que o algoritmo K-Means na presença de ruídos e *outliers*, devido ao uso de um *medoid* como ponto central do grupo, sendo menos influenciado por valores extremos do que o uso da média, por exemplo. Assim como no algoritmo K-Means, o K-Med utiliza um valor de K (número de grupos) como parâmetro para a execução do clustering (ROKACH; MAIMON, 2005). O Algoritmo (7) apresenta a implementação do K-Med.

Algoritmo 7 Método de clustering K-Medoids (K-Med) com base na matriz de distância e número de grupos

function KMEDOIDSCLUSTERING($distanceMatrix, k$)
    Select an initial set M of k medoids randomly from the n data points.
    Assign each point to the closest medoid to form initial clusters, using $distanceMatrix$ to determine distances.
    repeat
        for each medoid $m_i \in M$ **do**
            for each non-medoid point p_j, $j \neq i$ **do**
                Compute the total cost of swapping m_i with p_j, defined as the sum of distances from all points to their nearest medoid in the potential new configuration.
            end for
        end for
        Identify the swap (if any) that minimizes the total cost of clustering.
        if a swap that reduces the total cost is found **then**
            Perform the swap, updating M.
            Reassign each point to its closest medoid according to the updated M.
        end if
    until no swap reduces the total cost
    return The final set of clusters and their corresponding medoids.
end function

3 TRABALHOS RELACIONADOS

Esta seção revisa trabalhos na área de resposta a incidentes de segurança cibernética e procura, especialmente, elencar aqueles que exploram técnicas de IA.

O trabalho de Husák e Čermák (2022) analisa o uso de sistemas de recomendação na resolução de incidentes de segurança cibernética, especialmente os que incluem filtragem colaborativa, baseada em conteúdo ou baseada em conhecimento. O trabalho apresenta uma taxonomia para sistemas de recomendação voltados à resolução de incidentes onde quatro níveis são identificados: 1) triagem; 2) análise e resposta; 3) inteligência e prevenção; e 4) gerenciamento. Além da taxonomia, os autores apontam desafios, tais como informações insuficientes e dados incompletos coletados durante a análise e resposta a incidentes. Os autores ainda salientam a importância da realização de análises após à resolução do incidente, tanto no arquivamento quanto na documentação do incidente, embora isso possa ser negligenciado por equipes de *Security Operations Center* (SOC). Neste trabalho, tal análise é abordada no processo de revisão e retenção de casos (incidentes). Para isso, durante ou após o uso de planos de resposta recomendados pelo sistema, o analista de segurança verifica a necessidade de atualização dos casos e/ou criação de um novo caso.

O estudo de Applebaum et al. (2018) apresenta uma proposta para a captura, categorização e automatização de procedimentos de resposta a incidentes com base em um *playbook* estruturado. Um formato para especificar planos de ações de resposta a incidentes com base no uso de *plays* e *playbooks* é abordado. As *plays* correspondem a cursos de ações que podem ser executadas automaticamente ou manualmente como resposta a incidentes. Os *playbooks* apresentam conjuntos de cursos de ações a serem aplicados conforme diferentes características do contexto e evento do incidente ocorrido. Os *playbooks* são indexados com base em eventos, riscos, contextos e ações. Diferentemente do objetivo abordado no referido trabalho, que inclui automatizar procedimentos de resposta a incidentes, neste trabalho a resposta a incidentes é priorizada via processo de engenharia de conhecimento. Neste caso, o trabalho de Applebaum et al. (2018) propõe uma ontologia para consolidação do conhecimento representado nos *playbooks*. Em contrapartida, o uso da ontologia nesta pesquisa visa reduzir a divergência e a redundância na definição de descrições de ações nos planos de resposta a incidentes.

Os autores Grigorescu et al. (2022) apresentam uma proposta de mapeamento de *Common Vulnerabilities and Exposures* (CVEs) (WALTERMIRE; SCARFONE, 2011) para as técnicas do Modelo de Matriz de Ataques do MITRE (MITRE ATT&CK) (STROM et al., 2020) usando uma abordagem baseada em *Bidirectional Encoder Representations from Transformers* (DEVLIN et al., 2019). Esse mapeamento permite que uma vulnerabilidade CVE seja associada a uma ou mais categorias e especializações (técnicas) do MITRE

ATT&CK, possibilitando uma caracterização do comprometimento que a vulnerabilidade pode ocasionar. As categorias e técnicas do modelo MITRE ATT&CK não fornecem mitigações específicas para cada técnica de ataque listada em sua matriz; entretanto, incluem um conjunto de boas práticas de segurança para cada categoria tática. Este trabalho é relevante para esta pesquisa pois apresenta uma alternativa para tratar do problema de falta de informações sobre o incidente. Entretanto, isso só é aplicável a incidentes da categoria Exploração de Vulnerabilidades, sendo que a vulnerabilidade explorada precisa estar presente na base CVE.

A pesquisa de Schoenborn e Althoff (2023) apresenta um sistema de detecção de intrusão baseado em CBR. O sistema proposto utiliza uma abordagem multiagente para melhorar a eficiência e a escalabilidade da detecção de intrusão com base na análise de tráfego de rede. Cada agente é especializado em detectar um tipo específico de ataque utilizando CBR. Um agente de coordenação realiza uma votação com base nos votos recebidos de cada agente e apresenta os resultados das votações para um usuário humano. O usuário, por sua vez, pode decidir qual agente estava correto. O uso de CBR é explorado, assim como nesta pesquisa, onde diferentes categorias de incidentes são consolidados em uma base de casos, embora cada categoria possa ter diferentes atributos.

Por fim, os autores Nunes et al. (2019) e Guerra et al. (2023) apresentam uma proposta de resposta a incidentes de segurança cibernética que utiliza diferentes técnicas de CBR. Os incidentes são categorizados com base em padrões internacionais e nos formatos *Incident Object Description Exchange Format* (IODEF) (TAKAHASHI; LANDFIELD; KADOBAYASHI, 2014) e *Structured Threat Information eXpression* (STIX) (BARNUM, 2012). Diferentes métodos de cálculo de similaridade são utilizados na recuperação e ajustados conforme o tipo de atributo. Entretanto, não são exploradas estratégias de engenharia do conhecimento na aquisição e representação nos planos de resposta a incidentes de segurança. Planos de resposta são definidos com base uma lista ordenada de ações (definidas em texto livre). Estratégias de refinamento e apresentação de grupos/clusters das recomendações obtidas na recuperação como resultados da execução de uma consulta também não são exploradas.

A integração entre técnicas de clustering e CBR é um assunto importante para o desenvolvimento desta pesquisa.

Diferentes trabalhos envolvem o emprego de algoritmos de clustering na indexação de casos para CBR. O trabalho de Chen et al. (2018) emprega a análise de clusters para criar casos abstratos. Esses casos são utilizados em multiníveis de indexação de bases de casos. Na recuperação do CBR, a similaridade de um novo problema é calculada inicialmente em relação aos casos abstratos em um ou mais níveis e, então, para os casos da sub-base de casos indexada pelo incidente abstrato. Os autores de Zhu et al. (2015) propõem a seleção de *features* e análise de clusters, criando clusters com uma estrutura hierárquica para representar subcasos. A seleção de *features* é utilizada para encontrar

atributos representativos e não redundantes nos casos. O algoritmo de análise de clusters é executado para criar sub-bases de casos, as quais são utilizadas para a etapa de recuperação do CBR. Os autores Müller e Bergmann (2014) propõem um algoritmo de recuperação com base em índices. Um algoritmo de clustering hierárquico é utilizado para criar uma estrutura de índices com base em clusters e, com isso, um algoritmo de pesquisa em árvore é utilizado.

O uso de clustering é explorado para reduzir o espaço de busca na etapa de recuperação do trabalho apresentado em Mansoul e Atmani (2016). Um processo de clustering baseado em restrições é utilizado para escolher a melhor solução de um conjunto de soluções. Ao invés de uma recuperação massiva de casos, a pesquisa é focada em casos que atendam a critérios específicos. O uso de clustering permite a identificação dos casos coletados em circunstâncias semelhantes e a limitação da recuperação apenas a eles. Clustering pode também ser explorado anteriormente ao processo de recuperação do CBR, tal como proposto em Zhang e Yang (2022). Por exemplo, o algoritmo K-Means pode ser utilizado para agrupar os casos semelhantes em clusters. Considerando a etapa de recuperação, o problema atual, utilizado como entrada, é comparado apenas com os casos dentro do cluster mais semelhante, em vez de compará-lo com todos os casos na base de casos. Diferente dessas abordagens, a pesquisa descrita em Cocea e Magoulas (2012) envolve a utilização da saída do processo de recuperação de sistemas CBR como entrada para algoritmos de clustering. Os casos do CBR representam as estratégias utilizadas para resolução de problemas matemáticos pelos alunos. Uma matriz de incidência do(s) plano(s) utilizado(s) por cada aluno é usada e aplica-se o agrupamento com base nessa matriz.

No contexto de recuperação de informações na web, o trabalho descrito em Sadaf e Alam (2012) apresenta uma revisão sobre a utilização de clustering, com o objetivo de organizar resultados de pesquisas de motores de busca com palavras/frases ambíguas. Isso permite que um usuário selecione um grupo de documentos recuperados conforme o rótulo de um grupo, o qual indica o tópico/assunto contido em cada cluster. O trabalho utiliza uma proposta de HC, comparando a lista de casos resultantes, ordenados por similaridade decrescente, com os resultados de consultas web executadas por um motor de busca. O principal objetivo do agrupamento dos resultados de consultas executadas é o de rotular rapidamente as páginas recuperadas, permitindo que o usuário possa navegar pelos resultados da consulta realizada em um ou mais grupos, onde esses grupos podem organizar as páginas recuperadas. Esse processo é semelhante ao refinamento da consulta original, mas sem precisar fazer novas consultas no motor de busca. Este trabalho é relevante para a presente pesquisa, embora o processo de busca e recuperação de páginas web por motores de busca geralmente apresente diferenças significativas em relação ao modo com que consultas CBR são resolvidas.

Trabalhos relacionados sobre ontologias no domínio de segurança cibernética são

revisados nesta seção. Estes trabalhos envolvem o gerenciamento de vulnerabilidades de segurança e informações operacionais.

O autor Syed (2020) propõe o gerenciamento de vulnerabilidades de segurança utilizando ontologias. O trabalho também apresenta um sistema de alerta inteligente que realiza os seguintes passos: a) a coleta de informações de vulnerabilidade de várias fontes; b) integração de informações heterogêneas de acordo com o domínio de vulnerabilidade, representado pela *Cybersecurity Vulnerability Ontology* (CVO); e c) o aprimoramento da CVO com conceitos adicionais necessários para emitir alertas cibernéticos e geração da Ontologia de Inteligência Cibernética, com conceitos adicionais e propriedades inferidas, que são utilizadas para emitir alertas.

O trabalho de Takahashi e Kadobayashi (2015) contempla a criação de uma ontologia que representa informações operacionais de segurança cibernética para permitir a colaboração entre organizações. A base de conhecimento de contramedidas armazena regras e critérios conhecidos para avaliar o nível de segurança dos ativos de Tecnologia da Informação e detectar e/ou proteger de ameaças à segurança. A ontologia apenas apresenta informações sobre detecção e proteção de incidentes (medidas preventivas antecedentes à ocorrência do incidente). Isso é diferente da presente pesquisa, pois ações para correção de incidentes de segurança cibernética são representadas na ontologia construída.

O estudo de Onwubiko (2018) analisa o processo desenvolvido em *CyberSecurity Operations Center* (CSOC), provendo um framework que pode ser reusado. Também cria um grafo de conhecimento de incidentes cibernéticos, com base no uso de uma ontologia. Esse grafo permite que analistas de CSOC formem uma consciência situacional dos ativos de informação monitorados, ameaças que a organização pode ser alvo, vulnerabilidades que podem ser exploradas, o caminho de compromisso e a superfície de ataque. O trabalho define um framework com base em ontologias para utilização em CSOC. O quesito de resposta a incidentes é contemplado por meio do conceito de *playbook*, que define instruções relacionadas à resposta a incidentes de segurança.

A pesquisa de Mundie et al. (2014) apresenta uma ontologia baseada em um metamodelo de gerenciamento de incidentes. O objetivo é o de comparar e analisar processos de diferentes equipes *Computer Security Incident Response Team* (CSIRT). A ontologia descreve procedimentos essenciais aplicados no gerenciamento de incidentes, possuindo foco nas relações entre as equipes.

Os autores Correa et al. (2022) apresentam um sistema inteligente de suporte à decisão para equipes de resposta a incidentes de segurança cibernética, tendo como objetivo mitigar suspeitas de ataques cibernéticos. Uma ontologia é utilizada para mapear o conhecimento sobre ataques cibernéticos para ações de mitigação.

O trabalho de Çakmakçı et al. (2021) apresenta um framework para detecção e resposta inteligente a ataques *Distributed Denial of Service* (DDoS), combinando ferramentas *Security Information and Event Management* (SIEM) com ontologias. A ontologia é utili-

zada para modelar explicitamente o conhecimento de uma organização, incluindo informações sobre a infraestrutura de TI, medidas de segurança e ataques DDoS. Um sistema de inferência é utilizado para propor medidas de resposta e recuperação para ataques DDoS.

Os autores Preuveneers e Joosen (2024) propõem um framework de segurança cibernética baseado em ontologia, focado em sistemas, aplicações e serviços que utilizam IA. A ontologia desenvolvida visa documentar ameaças, vulnerabilidades e ataques, direcionados a componentes de IA, além de estratégias preventivas, de defesa e contramedidas. Um enfoque particular é dado a ataques que comprometem a segurança e a privacidade de sistemas que implementam soluções de IA.

Em resumo, embora as ontologias propostas nos trabalhos de Syed (2020), Takahashi e Kadobayashi (2015), Onwubiko (2018), Mundie et al. (2014), Correa et al. (2022), Çakmakçı et al. (2021) e Preuveneers e Joosen (2024) sejam relevantes para a aquisição e representação de dados e conhecimento sobre segurança cibernética, elas possuem relevantes diferenças do que é proposto neste trabalho. O papel da ontologia sugerida neste estudo é apoiar a homogeneização e representação de categorias de ações de análise, resolução e mitigação de incidentes. O objetivo principal é empregar essa ontologia na representação de casos, em especial na representação de planos de resposta a incidentes.

4 EMPREGO DE ONTOLOGIAS NA AQUISIÇÃO E REPRESENTAÇÃO DE PLANOS DE RESPOSTA A INCIDENTES

4.1 AMBIGUIDADE E REDUNDÂNCIA

A análise dos incidentes registrados nas bases de casos dos trabalhos de Nunes et al. (2019) (257 casos) e Barcelos (2020) (269 casos) permitiu identificar questões de ambiguidade e redundância na representação de ações de resposta a incidentes.

A ambiguidade na descrição das ações é caracterizada por interpretações divergentes por parte dos analistas de segurança. A falta de clareza nas ações pode acarretar em atrasos na resposta ao incidente, uma vez que os analistas podem executar ações de maneira equivocada. Isso não apenas prolonga o tempo necessário para conter e tratar o incidente, mas também pode aumentar o risco de danos adicionais à infraestrutura e aos dados da organização.

A redundância de ações é caracterizada pela falta de consistência na documentação e registro de incidentes. Quando cada analista de segurança descreve as ações usadas na resposta a um incidente de maneira diferente, torna-se difícil rastrear e documentar a resposta aplicada ao incidente. Entretanto, essa documentação é essencial na consolidação das experiências de resposta a incidentes e reúso dessas experiências na resposta a novos incidentes de segurança.

Para exemplificar os problemas de ambiguidade e redundância considere um determinado incidente e dois planos de resolução com ações descritas por analistas de segurança diferentes, conforme ilustrado na Tabela 1.

Considerando a *ação 1* dos *planos A e B*, é possível observar descrições distintas, as quais podem levar à falta de clareza na interpretação da ação. A *ação 1* do *plano A* pode criar desafios para os analistas de segurança no processo de resolução do incidente, dada a ausência de instruções específicas sobre quais informações devem ser coletadas. Isso pode levar à coleta de dados dispersa e inconsistente, uma vez que diferentes analistas podem interpretar a ação de maneira distinta. Essa dificuldade na interpretação e execução da ação não apenas pode atrasar a resposta ao incidente, como também pode deixar lacunas importantes nas informações coletadas.

Por outro lado, no *plano B*, a descrição da *ação 1* define com melhor precisão o que deve ser coletado, ou seja, informações sobre o problema que gerou o incidente. Os analistas de segurança sabem que o problema é composto por um conjunto específico, detalhado e previamente definido de atributos. Essa clareza ajuda a garantir que as informações coletadas sejam relevantes, abordando diretamente o problema do incidente que está sendo tratado. Isso não apenas agiliza a resposta ao incidente, mas também garante

Tabela 1 – Exemplos de planos de resolução que apresentam ações ambíguas e redundantes.

Action	**Plan A**	**Plan B**
1	Gather comprehensive details about the incident at hand	Obtain information about the incident problem
	...	...
5	If there's no essential need for external access to the application, set up a device firewall policy to block external connections to the application port	If external access to the service isn't necessary, implement a rule within the device firewall to obstruct access to the service port
6	Upgrade the application to the most recent version	Install the up-to-date version of the service on the host
	...	...
8	Limit the frequency of connections to the application within a designated timeframe	Establish a firewall rule on the device to curtail the volume of connections to the service port within a specified time frame
	...	...

Fonte: Autor.

que a coleta de dados esteja alinhada com os objetivos de solução do problema.

Outro exemplo da falta de clareza na descrição das ações pode ser observado nos planos da Tabela 1. No *plano A*, a *ação 8* oferece uma descrição relativamente clara, mas que ainda permite diferentes interpretações. Um dado analista de segurança poderia implementar a limitação indicada ajustando as configurações da aplicação, podendo esta não ser a abordagem mais eficaz. Outro analista poderia interpretar diferente. A possibilidade de uma interpretação distinta decorre da especificação de uma ação ambígua. Diante desta descrição de ação, um processo mais demorado de execução do plano de resposta pode acabar ocorrendo. A *ação 8* do *plano B*, que visa a mesma solução, é mais pontual ao indicar o que deve ser feito: criar regras de tráfego firewall para limitar o número de conexões. Note que a ação poderia também ser cumprida pelo ajuste das configurações de parâmetros de aplicações específicas, mas a *ação 8* do *plano A* não deixa isso claro.

Em geral, o emprego de uma linguagem completamente informal para a especificação de ações de resposta a incidentes pode impactar na interpretação de como estas ações devem ser conduzidas, nas ações efetivamente executadas e no tempo de resposta ao incidente. No resolução do *plano A*, possivelmente a configuração de limites de conexões no contexto de uma aplicação específica demandará contato com outros profissionais da organização responsáveis pela aplicação. Essa interação vai permitir implementar uma limitação das conexões nas configurações da aplicação, reforçando que a falta de especificidade das ações pode levar a soluções incompletas. Por outro lado, a *ação 8* do *plano B* fornece uma descrição mais precisa da tarefa a ser realizada. Não apenas é claro que a ação envolve a configuração de uma regra de firewall no dispositivo, mas também que ela especifica exatamente o que deve ser feito: limitar o volume de conexões à porta do serviço dentro de um intervalo definido. Essa clareza elimina a ambiguidade e garante que

analistas de segurança compreendam o que precisa ser feito.

A redundância de ações também pode ser observada em ações dos *planos A e B*. Por exemplo, na descrição da *ação 5* nos *planos A e B*, ambas as ações têm o mesmo objetivo: restringir o acesso externo a uma determinada aplicação ou serviço, utilizando uma política de firewall no dispositivo. A redundância ocorre porque as ações são praticamente idênticas em termos de objetivo e método, embora tenham sido redigidas de maneira diferente. Outro exemplo de redundância na descrição de ações pode ser observado na *ação 6* dos *planos A e B*. Ambas as ações visam alcançar o mesmo resultado: atualizar a aplicação ou serviço para a versão mais recente disponível. Porém, a existência de redundância nas ações usadas para especificar os planos de resposta a incidentes pode criar confusão para os analistas de segurança.

4.2 ESPECIFICAÇÃO DE AÇÕES DE RESPOSTA A INCIDENTES

Este trabalho propõe a revisão contínua de ações de resposta a incidentes de segurança. O objetivo é melhorar a clareza e a compreensão destas ações e a especificação de ações que sejam objetivas e claras, de modo que não haja espaço para interpretações. Essa abordagem pode garantir que os envolvidos entendam o que precisa ser feito, minimizando a possibilidade de erros. Para que isso aconteça, é fundamental revisar o plano de resposta a incidentes, como exemplificado no *plano C*, ilustrado na Tabela 2. Nesse plano, as ações dos *planos A e B* (Tabela 1) foram reescritas de forma mais objetiva e concisa. Por exemplo, comparando a descrição da *ação 1* dos *planos A e B* com a *ação 1* do *plano C*, fica evidente que a *ação 1* do *plano C* é mais precisa, indicando ao analista de segurança a coleta de informações sobre o incidente. Outro exemplo é a *ação 8* do *plano C*, que ajusta a descrição da *ação 8* dos *planos A e B*, indicando a adição de uma política ao firewall para limitar o número de conexões na porta do serviço/aplicação em um determinado intervalo de tempo.

Através do *plano C* é também possível observar a resolução de redundância na descrição de ações especificadas em diferentes planos. As *ações 5 e 6* do *plano C* apresentam uma melhora em relação à redundância presente nas descrições das ações usadas na representação dos *planos A e B*. Na prática, elas apresentam uma redação mais genérica, mas mantendo a precisão. Isso permite o reúso destas especificações de ações nos *planos A e B*, dado que ambas tinham o mesmo objetivo: restringir o acesso externo a um serviço ou aplicação, apresentando especificações diferentes por estarem procurando identificar o usuário da porta (aplicação ou serviço).

Tabela 2 – Exemplo de plano de resolução ajustado para reduzir ações ambíguas e redundantes.

Action	**Plan C**
1	Collect information about the incident problem
	...
5	If external access to the service/application is not required, add a policy to the device firewall that prevents access to the service/application port
6	Update service/application with the latest version
	...
8	Add a firewall policy for the device to limit the number of connections to the service/application port in a given time interval

Fonte: Autor.

4.3 UMA ONTOLOGIA DE RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

Este trabalho propõe uma Ontologia de Resposta a Incidentes de Segurança Cibernética para tratar questões de ambiguidade e redundância em especificações de ações usadas na representação de planos de resposta a incidentes. A ontologia atua como um repositório de conhecimento que armazena descrições unificadas de ações de resposta a incidentes de segurança. O objetivo principal é empregar a ontologia no apoio às tarefas de especificação de planos de resposta para incidentes, de forma que os planos criados por diferentes analistas de segurança reusem e compartilhem ações de resolução definidas na ontologia. Dessa forma, diferentes descrições criadas para uma determinada ação, que na prática representam um mesmo procedimento de resposta para um incidente, podem ser consolidadas e indexadas como um único conceito da ontologia.

Os seguintes passos foram empregados na construção da ontologia para ações de resposta a incidentes de segurança:

1) Coleta de Ações de Resposta: Informações detalhadas sobre as ações de resposta a incidentes de segurança são coletadas. As ações existentes, com ambiguidade e duplicidade, foram coletadas a partir de planos de resolução existentes na base de casos (NUNES et al., 2019; GUERRA et al., 2023), livros (RAINS, 2023; MARTÍNEZ, 2022), ChatGPT (OpenAI, 2023), repositórios do GitHub (Meir Wahnon, 2023; Counteractive Security Inc., 2023; AWS Samples, 2023) e sites (Mozilla, 2023).

2) Especificação de Conceitos: Identificação dos principais conceitos relacionados às ações de resposta a incidentes de segurança. Em especial, a ontologia proposta foca na representação da natureza das ações.

3) Definição de Relacionamentos: Estabelece relacionamentos hierárquicos entre os conceitos definidos na ontologia, organizando esse conceitos como classes e subclasses. Uma ação é uma instância de um conceito especializado na onto-

logia. As ações sempre estão associadas a uma classe/conceito. Os relacionamentos entre conceitos podem ser usados para pesquisa de ações na ontologia, uma vez que um conceito faz parte da hierarquia.

4) Propriedades de Dados: Identifica as propriedades e atributos associados às ações de resposta a incidentes. As ações possuem propriedades de dados, como identificador obrigatório (número inteiro) e, opcionalmente, materiais complementares. A definição de identificadores para ações na ontologia permite referenciar essas ações durante a criação ou atualização de planos de resposta a incidentes.

5) Modelagem em OWL: Criação de uma representação de ações de resolução usando a linguagem de OWL (Web Ontology Language). O software Protégé foi utilizado para a construção da ontologia. Uma revisão das ações de resolução é então realizada, onde as ações são definidas como instâncias na ontologia.

A ontologia é representada por um conjunto estruturado de ações de resposta a incidentes, onde as ações são organizadas hierarquicamente. Isso inclui a definição de conceitos, relacionamentos e atributos que auxiliam na representação destas ações.

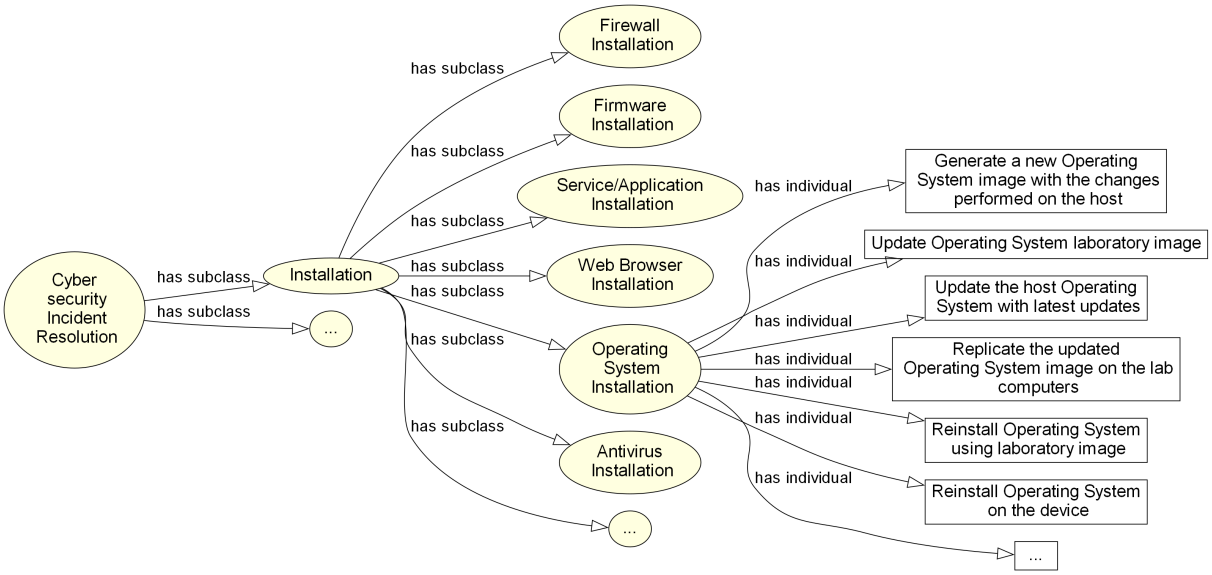
O núcleo da ontologia corresponde a um conjunto de classes que detalham conceitos e que definem a natureza das ações de resposta (por exemplo, sobre o que se tratam as ações de resposta, de modo que os analistas de segurança as compreendam). A ontologia especializa esses conceitos por meio de subclasses. Os conceitos folha da hierarquia são mais específicos e detalhados, onde as folhas da hierarquia capturam uma descrição concreta das ações de resposta (por exemplo, descrições simples e objetivas das ações de resposta).

As ações que compõem os planos de resposta a incidentes são especificadas como instâncias da ontologia. Isso significa que cada ação de resposta é considerada uma instância de uma classe da ontologia. Essa abordagem oferece flexibilidade na representação das ações, onde detalhes são capturados como propriedades de dados. Alguns exemplos de propriedades que podem fazer parte das ações/instâncias incluem identificadores e outras informações complementares, que ajudam na identificação e execução da ação.

A Figura 4 apresenta um exemplo de hierarquia de conceitos e ações de resposta a incidentes. Esse recorte da ontologia apresenta a hierarquia de conceitos e ações relacionadas à *"Operating System Installation"*. Na ontologia, a hierarquia de conceitos começa com o conceito raiz *"Cyber security Incident Resolution"*. A partir desse ponto, o conceito *"Installation"* especializa o conceito raiz. Em seguida, o conceito *"Operating System Installation"* especializa o conceito *"Installation"*. Portanto, o conceito *"Operating System Installation"* possui várias instâncias que representam ações concretas usadas na descrição de planos de resposta a incidentes. Algumas dessas instâncias incluem *"Generate a new Operating System image with the changes performed on the host"*, *"Reinstall Operating*

System on the device”, *“Reinstall Operating System using laboratory image”*, *“Replicate the updated Operating System image on the lab computers”*, *“Update Operating System laboratory image”* e *“Update the host Operating System with latest updates”*. Note que vários outros conceitos também especializam o conceito *“Installation”*. Alguns exemplos desses conceitos incluem *“Firewall Installation”*, *“Firmware Installation”*, *“Service/Application Installation”*, *“Web Browser Installation”* e *“Antivirus Installation”*.

Figura 4 – Recorte da ontologia apresentando a hierarquia de conceitos que define ações relacionadas à instalação de Sistema Operacional (SO).



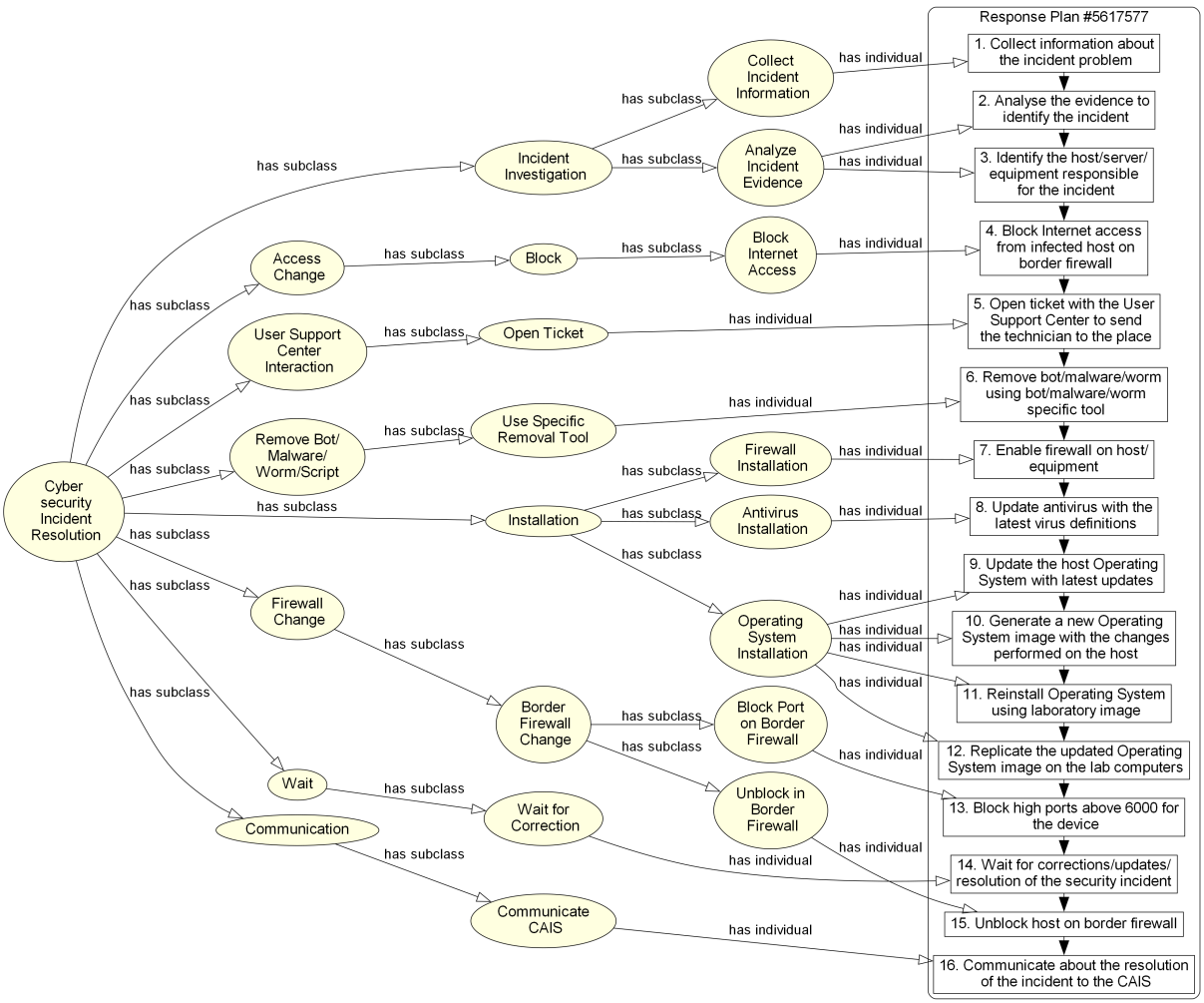
Fonte: Autor.

O exemplo na Figura 4 demonstra como a hierarquia de conceitos e ações de resposta a incidentes pode ser aplicada a um cenário específico, neste caso, à instalação de sistemas operacionais. Neste trabalho, esse exemplo ilustra como a ontologia fornece uma estrutura organizada para representar e reusar as ações de resposta a incidentes.

Um exemplo prático do relacionamento existente entre um plano e as ações de resposta capturadas pela ontologia é ilustrado na Figura 5. Neste exemplo, as naturezas destas ações de resposta são capturadas como conceitos mais gerais. Esses conceitos são especializados, até chegar ao nível de representação de ações concretas, ou seja, às instâncias. Por exemplo, o conceito *“User Support Center Interaction”* é especializado pelo conceito *“Open Ticket”*. Por sua vez, a ação *“Open ticket with the User Support Center to send the technician to the place”* foi adicionada como instância do conceito *“Open Ticket”*. Essa ação está presente no plano de resposta do incidente 5617577.

As instâncias da ontologia podem conter propriedades de dados usadas para definir informações complementares. Essas informações podem auxiliar na compreensão da ação de resposta e apresentam instruções para a sua execução. Por exemplo, informações complementares são capturadas por *Uniform Resource Identifiers* (URIs) com

Figura 5 – Conexão entre ações e planos de resposta usando conceitos da ontologia.

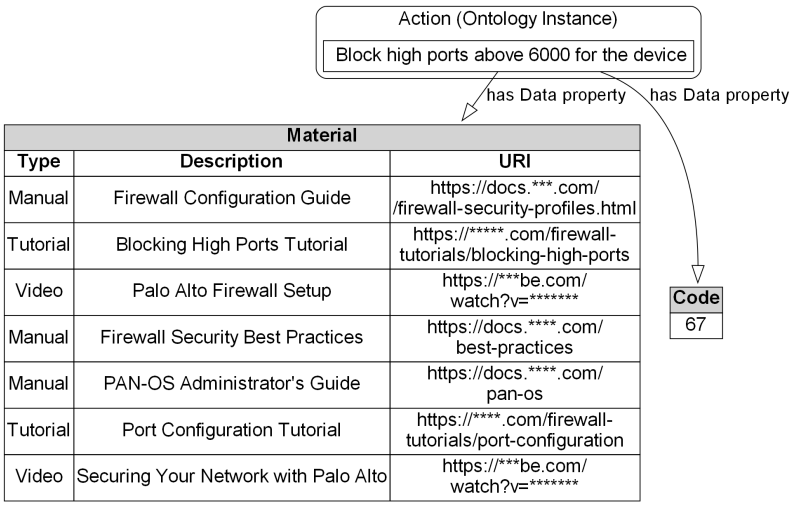


Fonte: Autor.

destino a sites internos ou externos, imagens, vídeos, tutoriais, manuais e documentos *Portable Document Format* (PDF). A Figura 6 apresenta exemplos de materiais complementares associados à ação *"Block high ports above 6000 for the device"*. Os materiais complementares desta ação incluem manuais, tutoriais e vídeos que fornecem instruções aos analistas de segurança, abordando temas como configuração de firewall para bloquear conexões em portas de redes altas (acima de 6000). Além desses materiais, há também referências a documentos PDF que detalham práticas recomendadas de segurança. Por exemplo, *"Firewall Security Best Practices"* oferece orientações abrangentes sobre como configurar e gerenciar um firewall de maneira segura. Em geral, o uso de links para materiais complementares torna os planos de resposta mais fáceis de seguir, facilitando o aprendizado. Novos analistas de segurança podem rapidamente se familiarizar com as melhores práticas e abordagens adotadas, acelerando a capacidade de contribuir para a resposta a incidentes.

A Tabela 3 apresenta, lado a lado, uma comparação das ações de resposta a inci-

Figura 6 – Materiais complementares associados à ação definida na ontologia.



Fonte: Autor.

dentes para três planos de resposta a incidentes distintos a um dispositivo servidor proxy com acesso aberto na Internet. Os *planos D e E* são delineados sem o emprego das ações definidas na ontologia criada, enquanto o *plano F* utiliza ações definidas na ontologia. Por exemplo, a *ação 8*, nos três planos, está relacionada à verificação de scripts maliciosos com execução automática no Sistema Operacional (SO) do dispositivo afetado. Isso ilustra a diferença que o uso da ontologia faz na clareza das ações de resposta a incidentes. A ação do *plano D*: *"Examine the device's operating system for potential malicious scripts"* é mais precisa do que a do *plano E*: *"Look for signs of malicious actions on the device"*, pois indica explicitamente que a análise se concentra em scripts maliciosos no SO. No entanto, ambas as ações de resposta podem deixar em aberto o método exato de análise, introduzindo alguma ambiguidade. Por outro lado, a *ação F*: *"Check for the presence of malicious scripts at device Operating System startup"* é a mais específica das três. Ela não apenas identifica o objetivo da verificação (scripts maliciosos), mas também diz qual local do sistema deve ser verificado (inicialização do SO). Isso torna a ação de resposta mais direta e menos sujeita a interpretações, garantindo uma abordagem mais eficaz para a segurança do dispositivo.

Outro exemplo de ação de resposta que pode ser observado é a *ação 10*, que aborda a segurança de serviços. Nos três planos, a ação apresenta nuances de clareza e especificidade. No *plano D*, a ação *"Enable encryption options for the service"* é relativamente direta, indicando a ativação de opções de criptografia para o serviço. No entanto, essa descrição não especifica quais opções de criptografia devem ser habilitadas, deixando espaço para interpretação. A ação do *plano E*: *"Activate encryption choices for the service"* é semelhante à primeira, mas não esclarece quais escolhas de criptografia devem ser feitas. Em contraste, a terceira ação: *"Configure use of Transport Layer Security (TLS)/Secure Socket Layers (SSL) on the application/service"* é a mais clara e específica.

Ela não apenas indica a necessidade de criptografia, como diz que o protocolo TLS/SSL deve ser configurado no aplicativo ou serviço, eliminando a ambiguidade e fornecendo orientações precisas para melhorar a segurança do serviço. Portanto, a ação de resposta do *plano F* é destacada por sua clareza, garantindo uma abordagem mais precisa para a segurança dos serviços.

Tabela 3 – Planos de resposta a incidentes com e sem o emprego do uso da ontologia para representação de ações.

Action	**Plan D**	**Plan E**	**Plan F**
	...	...	...
3	Limit Internet access from the affected host.	Restrict Internet access from the host.	Block Internet access from infected host on border firewall.
	...	...	...
8	Examine the device's operating system for potential autorun scripts.	Look for signs of malicious actions on the device.	Check for the presence of malicious scripts at device Operating System startup.
	...	...	...
10	Enable encryption options for the service.	Activate encryption choices for the service.	Configure use of TLS/SSL on the application/service.
	...	...	...

Fonte: Autor.

Um aspecto relevante para a proposição da Ontologia de Resposta a Incidentes de Segurança Cibernética é o processo de adaptação de planos de resposta a incidentes. Como parte do processo de reúso de planos de resposta a incidentes, a adaptação destes planos para novos problemas envolve a adição, alteração e remoção das ações de resposta que compõem os planos especificados. O cadastro de novas ações de resposta na representação ocorre por meio da seleção das ações na ontologia. Caso as ações a serem adicionadas não estejam presentes na ontologia, conceitos necessários para adicionar as ações à ontologia devem ser criados. A remoção de ações de um plano de resposta a incidentes não envolve o uso da ontologia; isso só acontece caso exista algum problema com a ação já definida na estrutura de conceitos da ontologia. Neste caso, uma nova ação de resposta pode ser cadastrada ou uma ação existente na ontologia pode ser alterada.

Por fim, cabe destacar que a ontologia desenvolvida não é estática, pois representa um conhecimento que é atualizado à medida que novos incidentes são tratados. Novas especializações dos conceitos podem ser criadas para melhor organizar o conjunto de ações de resposta a incidentes. Como consequência, esse processo permite a manutenção e melhoria contínua dos planos de resposta registrados nas estruturas dos casos da base de casos. À medida que a ontologia apresenta conceitos que atendem às necessidades de aquisição e representação de planos de resposta a incidentes ou, de forma mais pragmática, atende às necessidades de modelagem dos planos descritos nos casos armazenados na base de casos, a ontologia tende a sofrer menos atualizações.

5 RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

A resposta de incidentes inicia-se com o recebimento de um relatório de incidentes. Estes relatórios podem ser provenientes de sistemas de detecção, como *Intrusion Detection System* (IDS), dispositivos de análise de fluxo de rede, firewalls e outros. Eles também podem ser construídos a partir de detecções manuais recebidas de equipes de segurança internas ou externas à organização. Com base no incidente, o processo de resolução ocorre de acordo com as seguintes atividades:

Convert (Conversão): o incidente pode ser recebido de diferentes formas, formatos e estruturas. Por exemplo, e-mail contendo uma descrição em texto livre; logs, provenientes de ferramentas de detecção de incidentes; sistemas de notificação de incidentes; e formatos dedicados como IODEF e STIX. Uma vez recebidos, eles são convertidos para um formato padronizado de relatório de incidentes. O conhecimento sobre padronização de incidentes é usado na execução dessa atividade.

Analyze (Análise): o incidente nem sempre é recebido com informações precisas. Por exemplo, um incidente da categoria *Defacement* é recebido, mas durante a atividade de análise é identificado que uma estação cliente foi infectada por um *malware*, passando a fazer parte de uma *botnet*, e também passou a hospedar um site com conteúdo malicioso. Logo, a classificação correta deste incidente é *Botnet*, pois o dispositivo não possuía nenhum site hospedado anteriormente à infecção para a existência de uma desfiguração do conteúdo do site (*Defacement*).

Decompose (Decomposição): com base no problema do incidente definido, este pode ser decomposto em um ou mais problemas. Isso possibilita executar várias consultas no sistema, sendo que cada uma destas têm como entrada um dos subproblemas definidos. A decomposição do problema do incidente pode ser utilizada especialmente quando o incidente possui informações faltantes por diferentes motivos. Por exemplo, em um incidente da categoria *botnet*, o tipo do dispositivo (*Device Type*) afetado não foi identificado. Diante disso, diferentes problemas podem ser analisados para as várias categorias de tipos de dispositivos existentes.

Retrieve (Recuperação): a atividade de recuperação consiste em executar uma ou mais consultas, com base na aplicação de cálculos de similaridade entre o problema usado (como entrada) na consulta e os problemas presentes nos casos registrados na base de casos. Para cada consulta é retornada uma lista de casos, os quais são organizados em ordem decrescente de similaridade em relação ao problema usado como consulta. Para execução da consulta, diferentes cálculos de similaridade são usados para avaliar as similaridades entre os diferentes atributos e valores que representam a consulta e os casos armazenados na base de casos. Os casos retornados nas consultas, especialmente as soluções, são avaliados para verificar a possibilidade de reusar os planos de resposta

registrados nesses casos. Os resultados de cada consulta são armazenados no sistema e podem ser utilizados na atividade de clustering.

Clustering: o agrupamento demonstra os diferentes grupos (clusters) de casos presentes nos resultados de uma ou mais consultas ativas. A atividade de clustering pode ser realizada com base em consultas ativas, utilizando atributos do problema (por exemplo, incidente), solução (como o plano de resposta ao incidente) ou ambas as partes dos casos.

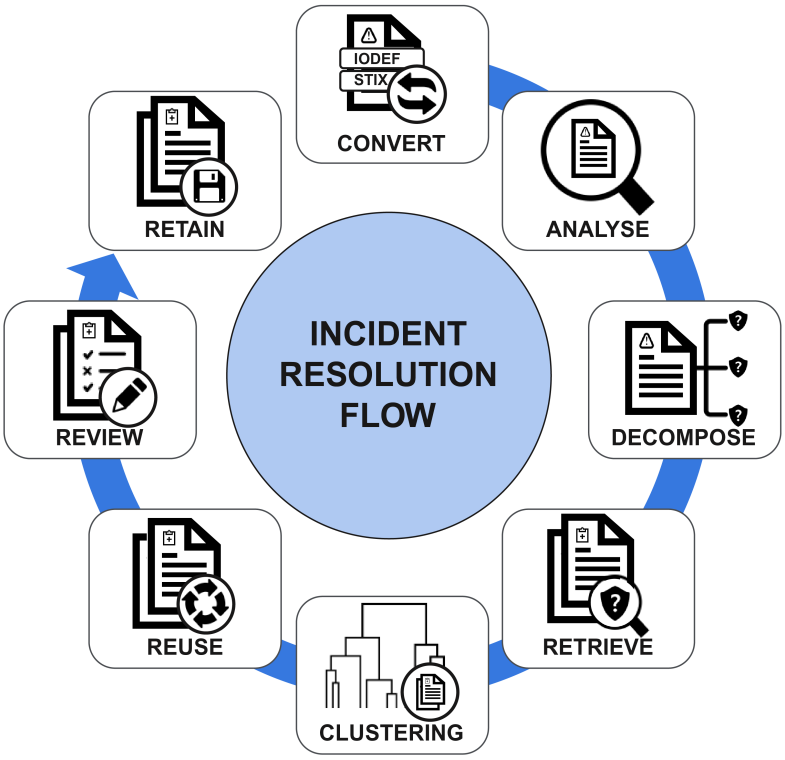
Reuse (Reúso): à medida que uma consulta é realizada no sistema para resolução do novo incidente, uma análise dos casos recuperados é realizada, especialmente sobre a aplicabilidade dos planos de resposta registrados. Com base nesta análise, planos de resposta a incidentes são reusados na solução do incidente problema.

Review (Revisão): durante o tratamento do novo incidente, com base nos planos recuperados, pode ser identificada a necessidade de alterar a ordem de execução da lista de ações de resposta. A adaptação das ações presentes nos planos para o contexto do novo incidente é realizada com base na adição de novas ações (selecionando ações para compor o plano usando a ontologia), alterações na ordem em que as ações aparecem no plano, ou ainda, caso não sejam aplicáveis ou não sejam necessárias para a resposta ao incidente, na remoção de ações. A combinação de ações de mais de um dos planos recuperados também pode ser explorada.

Retain (Retenção): uma vez que o novo incidente foi tratado, a experiência envolvida deve ser avaliada. Caso essa experiência de solução de problemas seja relevante, ela pode ser armazenada no sistema, seja com base no cadastro de um novo caso na base de casos, ou na atualização de um ou mais planos utilizados na resposta de incidentes. Embora a ontologia apresente um conjunto abrangente de ações de tratamento de incidentes, a adição de novas ações de resolução pode ser requerida à medida que novos incidentes e planos de resposta sejam cadastrados no sistema. Esse é um aspecto importante para a manutenção do conhecimento presente na base de casos e na ontologia, onde a solução de novos incidentes oportuniza uma crescente evolução do conhecimento armazenado no sistema.

O processo de resposta a incidentes é apresentado na Figura 7.

Figura 7 – Fluxo de resposta a incidentes.



Fonte: Autor.

5.1 REPRESENTAÇÃO DE INCIDENTES DE SEGURANÇA

A experiência de resposta a incidentes é normalmente mantida por analistas de segurança em diferentes organizações. Neste contexto, esse trabalho discute a aquisição e representação desse conhecimento no formato de casos para CBR.

Quando a detecção de um incidente realizada por um *Computer Emergency Response Team (CERT)* é recebida por organizações, busca-se representar o incidente como um novo caso a ser resolvido. A representação destes casos depende diretamente do modelo de casos. Neste trabalho, as principais partes do modelo de representação de incidentes proposto são as seguintes:

i) descrição do incidente: compreende os atributos de descrição contextuais do incidente, como identificador, título, descrição, data e hora da detecção do incidente;

ii) informações recebidas do CERT: define as informações que foram recebidas do CERT, geralmente relacionadas à detecção do incidente;

iii) investigação preliminar: aborda as informações levantadas pela organização sobre o incidente, por meio de uma investigação preliminar realizada. Em muitos sentidos, informações relevantes sobre a detecção do incidente são coletadas e inseridas no novo caso;

iv) risco do incidente: indica uma avaliação do risco/impacto do incidente no contexto da organização.

Conforme apresentado em Nunes et al. (2019) e Barcelos (2020), utilizar unicamente dados provenientes de relatórios de identificação de incidentes na construção de consultas CBR nem sempre é suficiente para caracterizar um incidente de segurança.

A descrição do novo incidente envolve a análise de relatórios recebidos com informações sobre novos incidentes identificados. Um relatório de incidente apresenta descrições sobre o incidente, como detalhes das infraestruturas organizacionais afetadas; padrões de ataque e informações operacionais relacionadas à classe do incidente, tais como informações sobre *Uniform Resource Locator* (URLs), malwares, portas e protocolos empregados, entre outros. Entretanto, nem sempre os relatórios de incidentes recebidos apresentam informações corretas e suficientes para caracterizar os problemas correntes, de forma que um analista de segurança possa recuperar planos de resolução relevantes para resolver o incidente em questão. Essa falta de informações em relatórios de incidentes pode ser atribuída à limitação do conhecimento inicial sobre o incidente ou até a erros humanos, como a interpretação inadequada de eventos e logs associados ao incidente. A detecção tardia de incidentes, a falta de monitoramento contínuo e a ausência de rastreamento adequada de ambientes cibernéticos também contribuem para a imprecisão dos relatórios. Além disso, a complexidade técnica dos incidentes, a evasão de detecção por parte dos atacantes e as limitações nos registros de log podem dificultar a obtenção de informações confiáveis. A falta de integração ou integração parcial entre ferramentas de segurança, a falta de treinamento adequado de analistas de segurança e colaboradores sobre como reportar incidentes também podem comprometer a qualidade dos relatórios de incidentes. Manipulações maliciosas em arquivos de logs e relatórios provenientes de sistemas e ferramentas de detecção por parte dos invasores também podem comprometer a correta detecção e, consequentemente, os relatórios de incidentes provenientes destas detecções. Portanto, é fundamental identificar esses desafios ao buscar garantir respostas mais eficazes às ameaças cibernéticas.

Para abordar esse problema, novos atributos foram incluídos ao modelo de casos proposto neste trabalho. Por exemplo, atributos propostos para a representação de casos do tipo de incidente *botnet* são:

i) *device_type*: identifica o tipo de dispositivo comprometido; por exemplo, um *Server*, *Client Station*;

ii) *type_specialization*: indica qual o contexto de funcionamento do dispositivo comprometido na *botnet*; por exemplo, *Command and Control* (C&C) (ALAVIZADEH et al., 2021) (dispositivo ao qual os *bots* se conectam para obter código malicioso e instruções), e *Bot Device* (dispositivo que faz parte de uma *botnet* e que se conecta a dispositivos C&C);

iii) *device_responsible*: indica qual o setor/departamento responsável pelo dispositivo comprometido; por exemplo, *Department A*, *Department B*;

iv) *incident_detected_on_device*: indica qual dispositivo o incidente foi detectado; por exemplo, *Device Itself*, *PROXY Server*, *Network Address Translation (NAT) Server*.

Em geral, o processo de recebimento de um novo incidente envolve o mapeamento das informações do incidente detectado para o conjunto de atributos de representação de incidentes proposto neste trabalho. Os formatos IODEF versão 2 e STIX versão 2.1 são exemplos de formatos para representação e compartilhamento de incidentes de segurança cibernética. Com o intuito de facilitar o mapeamento de incidentes, foram desenvolvidas tabelas de mapeamento para cada um dos atributos de cada categoria de incidente proposta neste trabalho. Essas equivalências também utilizam um mapeamento para o IODEF versão 2 e de objetos e propriedades para o STIX versão 2.1.

A Tabela 4 apresenta os mapeamentos dos formatos IODEF versão 2 e STIX versão 2.1 para os atributos que identificam um incidente. Todas as categorias de incidentes possuem o conjunto de atributos definido na Tabela 4. Por exemplo, a Tabela 5 apresenta o mapeamento de atributos da categoria *Botnet*.

O Apêndice B apresenta os respectivos mapeamentos para cada uma das demais categorias de incidentes.

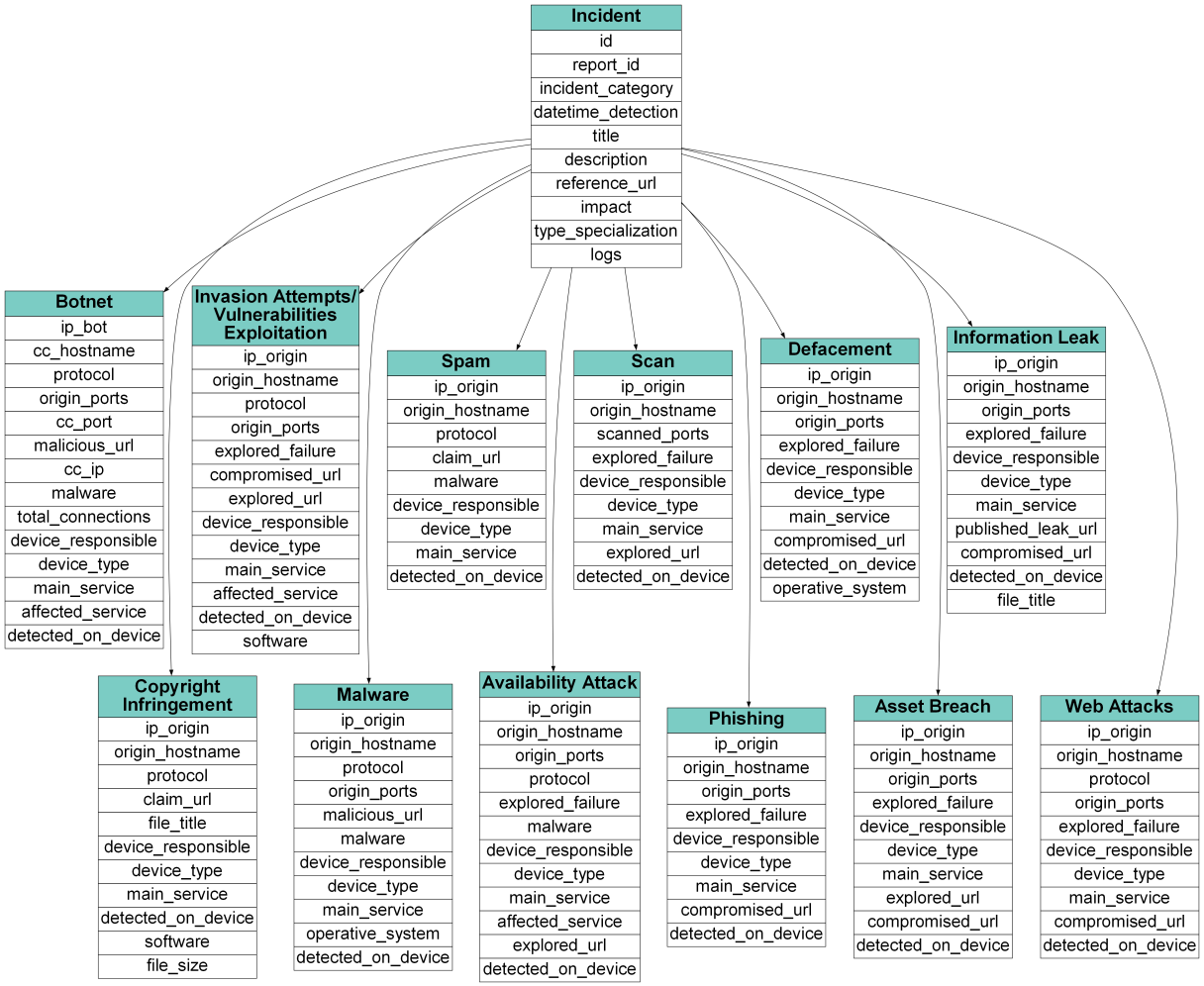
Tabela 4 – Mapeamento de atributos e propriedades de incidentes conforme os formatos IODEF e STIX.

Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Incident	IncidentID	Incident	id	report_id
Incident	DetectTime	Incident	created	datetime_detection
Assessment	IncidentCategory	Incident	incident_type	incident_category
Method Class	sci:AttackPattern	AttackPattern	name	type_specialization
Incident	Description	Incident	name	title
Discovery	Description	Incident	description	description
BusinessImpact	severity	Custom	impact	impact
Reference	URL	external-reference	source_name/url	reference_url
RecordData	RecordItem	Custom	logs	logs

Fonte: Autor.

A Figura 8 apresenta o conjunto de atributos utilizado por cada categoria de incidente. O conjunto de atributos da classe *incident* é herdado por todas as classes/categorias de incidentes.

Figura 8 – Atributos por categoria de incidente.



Fonte: Autor.

Tabela 5 – Mapeamento de atributos e propriedades de incidentes da categoria *Botnet* conforme os formatos IODEF e STIX.

Botnet				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_bot
DomainData	Name	domain-name	type/value	cc_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
Service	port	network-traffic	type/src_port	origin_ports
Service	port	network-traffic	type/src_port	cc_port
RelatedActivity	URL	external-reference	source_name/url	malicious_url
NodeRole	category/c2-server	IPv4 Address/ IPv6 Address	type/value	cc_ip
Reference	ReferenceName	Malware	name	malware
EventData	Counter	network-traffic	src_packets	total_connections
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure_ types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
Service	Application	Software	type/name/ version	affected_service
Discovery	description	Infrastructure	description	detected_on_device

Fonte: Autor.

5.2 RESPOSTA A INCIDENTES UTILIZANDO CBR

A resolução baseada em casos de novos incidentes de segurança é proposta em Nunes et al. (2019) e Guerra et al. (2023). Esse processo envolve o recebimento das informações de um incidente detectado, a formulação e execução de consultas CBR para selecionar planos de respostas para abordar esses problemas. O objetivo é recuperar experiências de resposta a incidentes de uma base de casos, que possam ser reusadas na solução do novo problema. Um exemplo de incidente e de plano de resposta recuperado de uma base de casos pode ser apresentado para explicar o funcionamento deste processo.

Dada a ocorrência de um novo incidente, um conjunto de informações sobre ele é recebida por um analista de segurança. Muitas vezes, essas informações não são precisas o suficiente para permitir a efetiva resposta dos incidentes detectados. Assim como proposto neste trabalho, tais informações devem ser complementadas com informações adicionais sobre o contexto do incidente.

Os vários detalhes que podem complementar as informações de detecção de incidentes são resultantes de análises preliminares realizadas internamente, na organização, por um analista de segurança. Essas análises podem identificar características do incidente e dos dispositivos de software e hardware afetados, por exemplo. Tais aspectos são

importantes para que o sistema CBR seja capaz de recuperar planos de resposta mais precisos para os incidentes descritos como consultas. Como parte desse processo de caracterização de incidentes, os atributos que detalham esses problemas são utilizados para construir e executar uma ou várias consultas (Figura 9 (a)).

Ao executar uma consulta, uma lista de casos ordenados, em ordem decrescente de similaridade, é retornada pelo sistema CBR (Figura 9 (b)). O usuário pode visualizar os casos recuperados, analisando o incidente problema e o plano de resposta registrado em cada caso. O analista verifica os casos recuperados para identificar se existem um ou mais planos de resposta que podem ser aplicados, ou adaptados, na resolução do problema. Em certas situações, por exemplo, um novo plano de resposta pode ser construído para o problema corrente. Assim como proposto neste trabalho, esse novo plano é construído a partir do reúso de casos de soluções de problemas recomendados pelo sistema CBR para a consulta dada.

Um exemplo de um novo incidente é apresentado na Figura 9 (a). Nesse incidente, um servidor de *Domain Name System* (DNS) está acessível publicamente na Internet, respondendo a consultas externas. Como resultado de uma consulta construída para esse incidente detectado, incidentes, juntamente com planos de resposta a incidentes passados, são ranqueados por similaridade, tal como apresentado na Figura 9 (b). Tais consultas podem ser registradas no sistema, onde um usuário pode informar uma descrição para o tipo de problema que a consulta expressa. Análises podem ser realizadas para considerar, por exemplo, que dentre os incidentes recuperados, existem apenas dispositivos servidores, mais especificamente servidores web, DNS e SSH.

Figura 9 – Exemplo de consulta e casos recuperados.

Query (Report ID #4474699) (a)	
Title	Open Resolver Vulnerability in a DNS Server
Description	A DNS server on Department B was responding to external clients, making it susceptible to open resolver vulnerabilities such as cache poisoning and distributed denial of service (DDoS) attacks.
Protocol	UDP
Type Specialization	Vulnerability Exploitation
Device Type	Internal Device - Server
Explored Failure	Open Resolver
Main Service	DNS
Origin Ports	53 (Domain Name Server)
Device Responsible	Department B
Detected On Device	Device Itself
Reference URL	https://***/docs/whitepapers/open-recursive-dns/
Impact	Medium

Retrieved Cases (b)				
Attributes	**Case #1818260 - 88.10%**	**Case #4504238 - 87.13%**	**Case #5078740 - 74.34%**	**...**
Title	Open Resolver Vulnerability in a Web Server with DNS Service	Open Resolver Vulnerability in a DNS Server	Open SSH Server Vulnerability on Department B	...
Description	A web server on Department A has a DNS service that was responding to external clients, making it susceptible to open resolver vulnerabilities such as cache poisoning and distributed denial of service (DDoS) attacks.	A DNS server on Department A was responding to external clients, making it susceptible to open resolver vulnerabilities such as cache poisoning and distributed denial of service (DDoS) attacks.	A SSH server on Department B was accessible on the Internet without any limitation and can introduce several security risks and vulnerabilities.	...
Protocol	UDP	TCP	SSH	...
Type Specialization	Vulnerability Exploitation	Vulnerability Exploitation	Vulnerability Exploitation	...
Device Type	Internal Device - Server	Internal Device - Server	Internal Device - Server	...
Explored Failure	Open Resolver	Open Resolver	Open Resolver	...
Main Service	WEB	DNS	SSH	...
Origin Ports	53 (Domain Name Server)	53 (Domain Name Server)	22 (SSH Remote Login Protocol)	...
Device Responsible	Department A	Department A	Department B	...
Detected On Device	Device Itself	Device Itself	Device Itself	...
Reference URL	https://***/docs/whitepapers/open-recursive-dns/	https://***/docs/whitepapers/open-recursive-dns/	https://****/*-actions-for-hardening***server-for-internet/	...
Impact	Low	Low	Medium	...
Actions				
...	...	...	...	...
8	Recommend responsible to use DNS Security Extensions.	Recommend responsible to use DNS Security Extensions.	Add a Firewall policy for the device to limits the number of connections to the service/ application port in a given time interval.	...
9	Recommend responsible to disable recursion for external clients on DNS server.	Recommend responsible to disable recursion for external clients on DNS server.	Recommend responsible to disable password-based authentication and consider the use of key-based authentication.	...
...	...	...	...	...

Fonte: Autor.

5.3 RESPOSTA A INCIDENTES UTILIZANDO CBR E CLUSTERING

O primeiro passo para a resposta a novos incidentes de segurança é recuperar uma lista de casos de solução de problemas passados que contenha planos de resposta para o incidente corrente. O problema é que listas de casos recuperados não indicam qualquer existência de grupos de problemas (por exemplo, incidentes) ou soluções (como planos de resposta) nesses resultados. Assim como proposto neste trabalho, algoritmos de clustering são usados na análise dos resultados dessas consultas CBR.

O uso de clustering sobre os resultados de consultas possibilita apresentar ao analista de segurança estruturas de grupos que estavam presentes nos resultados destas consultas, mas que não seriam facilmente identificadas pelos usuários através de uma análise manual da lista de casos recuperados. Em geral, a informação sobre a existência de grupos nestes resultados pode ser, por si só, relevante para apoiar a resolução desses problemas.

O agrupamento de resultados de consultas CBR é relevante quando a lista de casos recuperada apresenta situações-problema muito variadas, o que costuma ocorrer quando as informações utilizadas como entrada nas consultas são pouco específicas, por vários motivos. Esse é o caso de consultas formadas por informações vagas ou ambíguas. Isso ocorre quando um analista de segurança não consegue identificar informações importantes sobre o incidente, como os tipos de dispositivos afetados.

Em especial, muitos analistas podem não ter experiência na resolução do tipo de incidente detectado. Nessa situação, o uso de clustering permite identificar quais os tipos de dispositivos que normalmente são afetados pela categoria de incidente tratada. Também é relevante utilizar clustering para agrupar uma grande quantidade de resultados de consultas, de forma a obter uma lista de casos mais específica para abordar o problema corrente. Nesse contexto, o uso de clustering permite ao usuário focar a análise nos casos de um ou mais grupos selecionados. Entretanto, existem situações em que a recuperação de casos tão diversificados pode ser vista como uma vantagem. Na prática, esses casos podem indicar diferentes experiências de resposta a incidentes que possuem características similares. Nestas situações, a análise desses casos recuperados pode revelar que eles podem ser organizados em grupos distintos.

Por exemplo, os casos recuperados pelo sistema Figura 9 (b) para a resolução do incidente (Figura 9 (a)) retornam instâncias de incidentes com diferentes contextos. Neste caso, o incidente 4474699 ocorreu em um servidor DNS, que estava respondendo a clientes externos (provenientes da Internet) à rede da organização, o que poderia ocasionar ataques de DDoS. Considerando apenas os três incidentes recuperados/recomendados, ao analisar o tipo de dispositivos afetado, pode-se identificar que trata-se de diferentes contextos. O incidente 1818260 foi detectado em um servidor web, já o incidente 4504238 ocorreu em um servidor DNS e, por fim, o incidente 5078740 foi relacionado a um servidor

SSH. Estes são exemplos de como a execução de apenas uma consulta no sistema pode retornar casos com diferentes contextos.

Assim como proposto neste trabalho, grupos de casos organizados a partir de listas de casos recuperados para consultas CBR podem ser formados e investigados de acordo com diferentes aspectos. Em geral, esse grupos podem ser determinados a partir de computações de similaridade:

a) similaridades entre características dos incidentes capturados pelos casos recuperados, ou seja, atributos usados na representação de problemas no modelo de casos usado;

b) similaridades não somente entre incidentes mas também entre planos de resposta usados para abordar esses problemas, ou seja, atributos usados na representação de pares problema/solução no modelo de casos usado;

c) similaridades entre passos de resposta, capturados na representação de planos de resposta a incidentes, ou seja, atributos usados na representação de soluções no modelo de casos usado.

Usando essas entradas no algoritmo de clustering, os casos recuperados podem ser organizados de diferentes formas. O processo de resposta a incidentes, com base no uso de CBR e clustering, envolve atividades que são desenvolvidas após consultas CBR serem executadas. Essas atividades são as seguintes:

1) escolher algoritmo de clustering que deve ser utilizado;

2) selecionar uma ou mais consultas ativas a serem analisadas;

3) indicar atributos dos casos que devem ser utilizados como entrada nos algoritmos de clustering a serem executados;

4) definir o número de grupos almejado caso seja necessário;

5) identificar e selecionar os grupos relevantes para responder ao problema descrito na consulta.

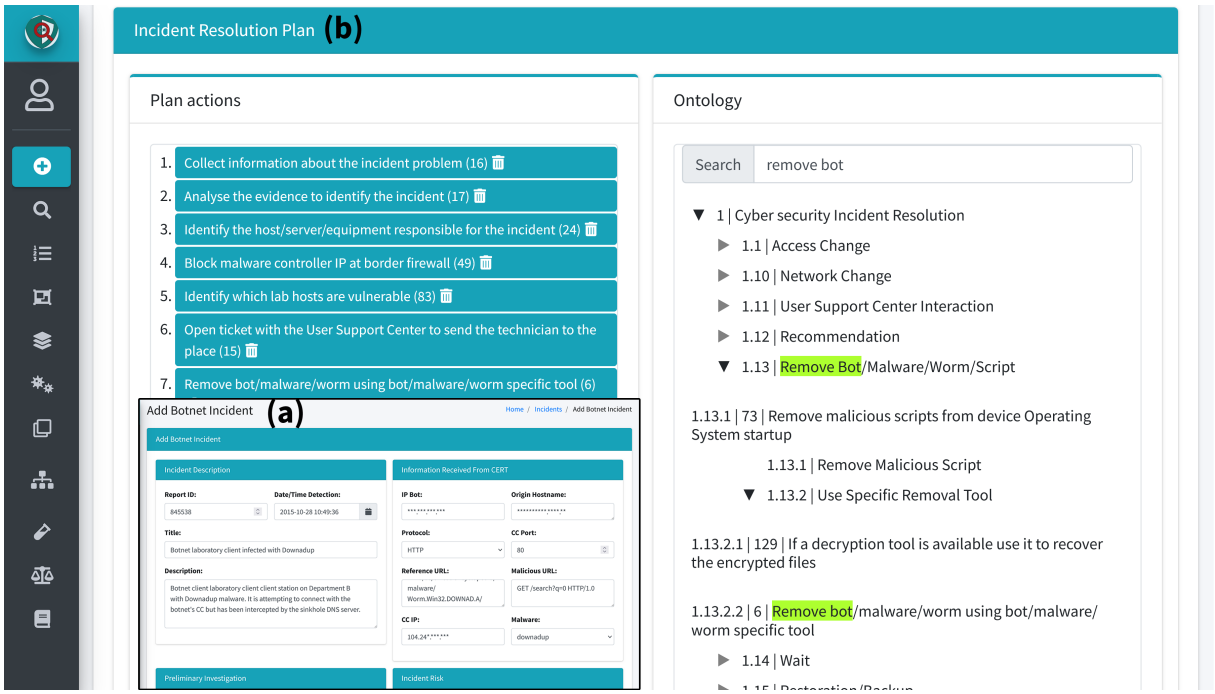
O exemplo na Figura 9 utiliza uma consulta no sistema *Cluster and Case-based Cybersecurity Incident Resolution* (CCCIR) para resolver um incidente. O fluxo de resposta a um novo incidente ao empregar o agrupamento é resumido a seguir: ao receber um novo incidente, um analista de segurança examina o seu contexto, levantando informações que são descritas por consultas CBR. Essas consultas são executadas e, opcionalmente, os resultados obtidos podem ser agrupados de acordo com diferentes critérios. O analista de segurança analisa as recomendações e avalia se os planos de resposta retornados podem ser adaptados para tratar o novo incidente.

6 SISTEMA DE RESPOSTA A INCIDENTES DE SEGURANÇA CIBERNÉTICA

Uma nova versão (2.0) do sistema CCCIR foi desenvolvida neste trabalho. Essa versão é implementada em Python versão 3.9 com o *framework* Django versão 3.2 e PostgreSQL versão 13. A Ontologia de Resposta a Incidentes de Segurança Cibernética proposta foi construída no software Protégé versão 5.5. Esse modelo é representado no formato OWL, o qual é importado no sistema CCCIR utilizando o pacote “owlready2”.

O sistema possui interfaces de cadastro de novos casos. O cadastro é baseado nos principais atributos de cada categoria de incidente, registrada na base de casos do sistema (GUERRA et al., 2023). A Figura 10 (a) apresenta um exemplo da interface de cadastro de um novo incidente do tipo *botnet*.

Figura 10 – Interface de cadastro de incidentes com a seleção de ações do plano de resposta usando a ontologia.



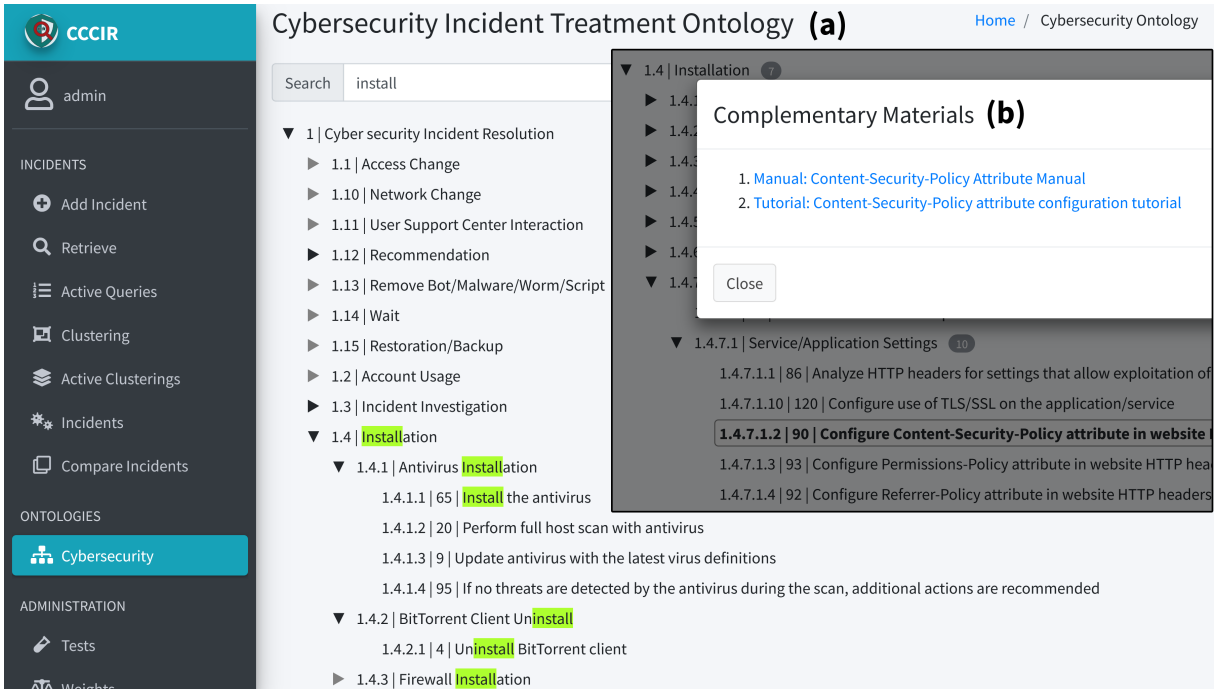
Fonte: Autor.

O cadastro de um novo incidente também envolve adicionar um plano de resposta para o incidente representado no modelo de caso. Neste trabalho, isso envolve selecionar e registrar a sequência de ações de resposta do incidente (Figura 10 (b)). Para apoiar esse processo, a Ontologia de Resposta a Incidentes de Segurança Cibernética é apresentada para o analista de segurança.

O sistema permite a visualização da hierarquia de conceitos e ações (instâncias) da ontologia construída, permitindo que o usuário navegue nos diferentes níveis da estrutura. Ele permite a realização de pesquisas por termos representados na ontologia, funcionando

como uma forma de filtro de conteúdo (Figura 11 (a)). Informações complementares associadas às ações de resposta a incidentes podem ser visualizadas com base na seleção de ações, onde uma lista com links para essas informações é apresentada pelo sistema (Figura 11 (b)).

Figura 11 – Interface de visualização da ontologia e materiais complementares às ações.



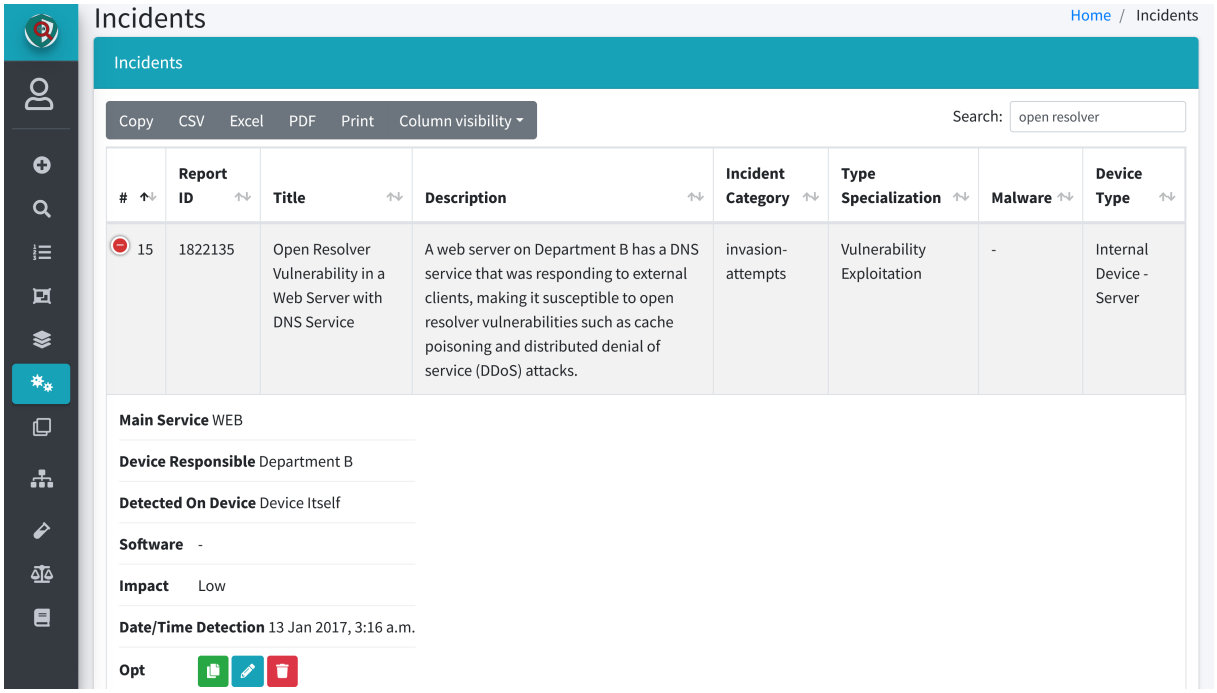
Fonte: Autor.

O usuário pode realizar as pesquisas ou explorar os conceitos da ontologia e, então, selecionar as ações que representam o plano de resposta desejado. À medida que o analista seleciona uma ação (instância da ontologia), ela é adicionada ao plano construído. O usuário pode alterar a ordem destas ações no plano de resposta e remover ações quando necessário.

O sistema apresenta para o usuário uma lista com todos os incidentes cadastrados, podendo este pesquisar e filtrar incidentes exibidos conforme um determinado valor ou termo de pesquisa (Figura 12), ou ainda ordenar os incidentes conforme valores de determinados atributos usados no modelo de casos. Na tabela apresentada, cada linha representa um incidente cadastrado, e para cada incidente, é exibido um conjunto de opções (*Opt*): remover o incidente, editar o incidente ou adicionar um novo incidente com base no incidente selecionado. Selecionando uma destas opções, a interface de cadastro de incidente é apresentada para o analista de segurança. Com isso, os valores dos atributos do incidente originalmente selecionado (reusado) são pré-preenchidos na interface de cadastro de novos incidentes. Essa funcionalidade permite um cadastro mais rápido de novos casos na base, onde o usuário precisa apenas atualizar os valores dos atributos que são distintos dos valores registrados no caso reusado. Tal funcionalidade também permite

o cadastro de novos planos de resposta a incidentes associados a novos casos registrados no sistema.

Figura 12 – Interface de exibição dos incidentes cadastrados no sistema.



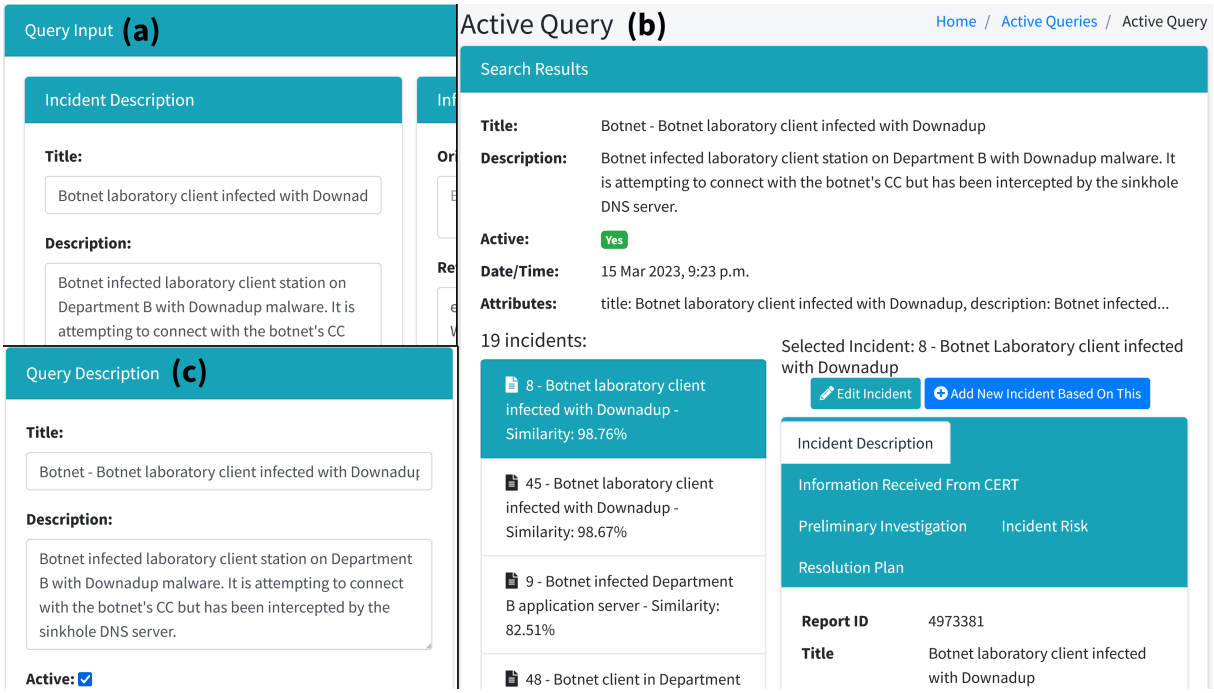
Fonte: Autor.

A interface de recuperação de casos (*active query*) do sistema permite que o analista preencha os valores dos atributos que descrevem o problema corrente. Cada categoria/tipo de incidente possui uma interface adaptada para os atributos daquela categoria. Por exemplo, a Figura 13 (a) apresenta um exemplo da interface de recuperação para incidentes da categoria *botnet*. Para a execução de consultas CBR no sistema, o usuário pode selecionar um conjunto de pesos usado nas computações de similaridade, conforme uma lista de pesos previamente cadastrada no sistema. O usuário também pode configurar seu próprio conjunto de pesos a ser usado nos cálculos de similaridade executados para obter respostas para consultas CBR. Após a execução de consultas, o sistema permite navegar pelos resultados das consultas executadas *active query*, tal como apresentado na (Figura 13 (b)).

O analista de segurança pode salvar informações de uma consulta *active query*. Assim, as informações relacionadas à consulta são registradas no sistema, tal como título, descrição, data/hora e atributos/valores utilizados, além dos resultados da consulta, ordenados de forma decrescente, conforme o valor de similaridade computado (Figura 13 (c)). O usuário pode percorrer cada um dos incidentes recuperados acessando as diferentes partes da descrição do incidente e do plano de resposta associado.

Na Figura 13, um novo incidente é usado na criação de uma consulta. O incidente 4973381 envolve o dispositivo de um laboratório detectado trafegando dados com ende-

Figura 13 – Interface de recuperação do sistema.



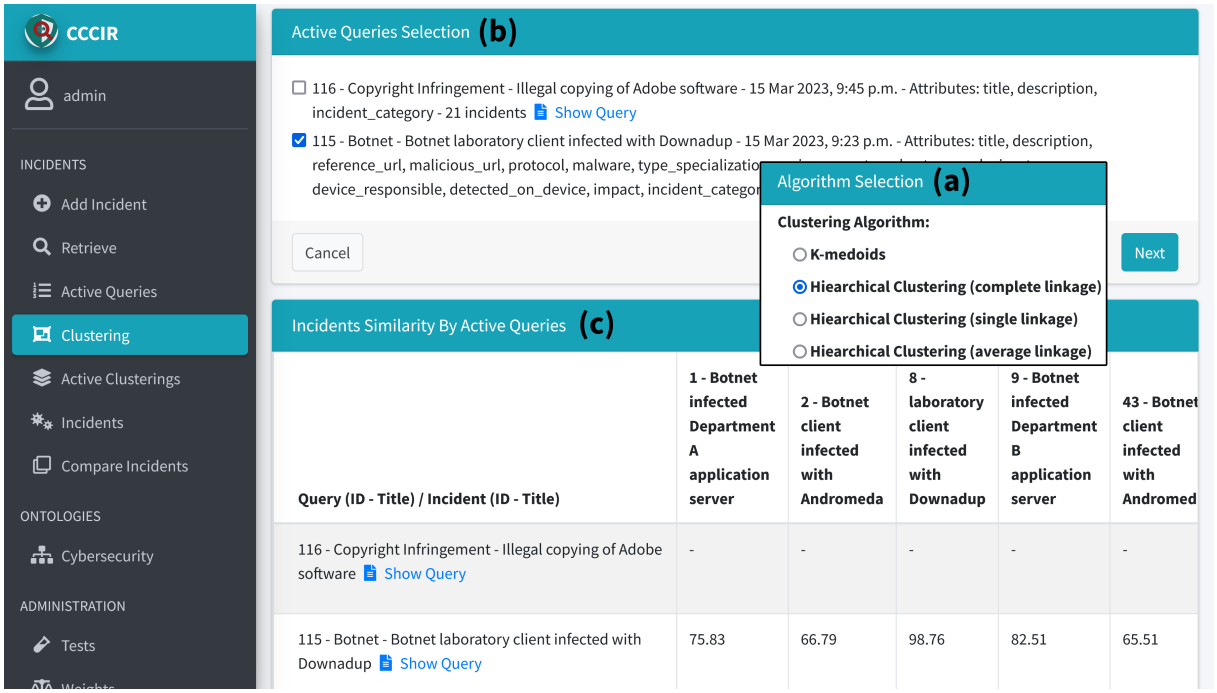
Fonte: Autor.

reços conhecidos por distribuir códigos maliciosos do *malware Downadup* e endereços de C&C de *botnets*. Neste exemplo, o clustering é empregado para formar grupos com base nos resultados retornados pela *active query*. O algoritmo de clustering é selecionado (Figura 14 (a)). Os algoritmos de clustering implementados no sistema são: Agrupamento Hierárquico (HC) com um dos critérios de *linkage*: *complete linkage* (CL), *single linkage* (SL), *average linkage* (AL); e *K-Med*. Em seguida, uma ou mais consultas ativas no sistema são selecionadas, onde os casos recuperados para essas consultas são usados como entrada no algoritmo de clustering. Para agrupar os casos, uma *active query* é exibida para o usuário, em conjunto com a data e a hora da execução e o número de casos recuperados (Figura 14 (b)). A similaridade dos incidentes retornados pela *active query* é apresentada (Figura 14 (c)), possibilitando a análise de casos retornados, por exemplo.

Os atributos usados no modelo de casos utilizado no processo de clustering devem ser selecionados pelo analista de segurança. O analista pode não somente selecionar os atributos usados na representação dos incidentes, como também pode selecionar os atributos utilizados na representação de planos de resposta destes incidentes (Figura 15). Nas computações dos algoritmos de clustering, os métodos de similaridade usados nas computações de recuperação de casos são também utilizados. Para o agrupamento de resultados de consultas, caso for selecionado o plano, como entrada para o algoritmo de clustering, a distância de Levenshtein (ONTAÑÓN, 2020) é usada na computação de similaridades entre planos de ações de resposta registradas nos casos recuperados.

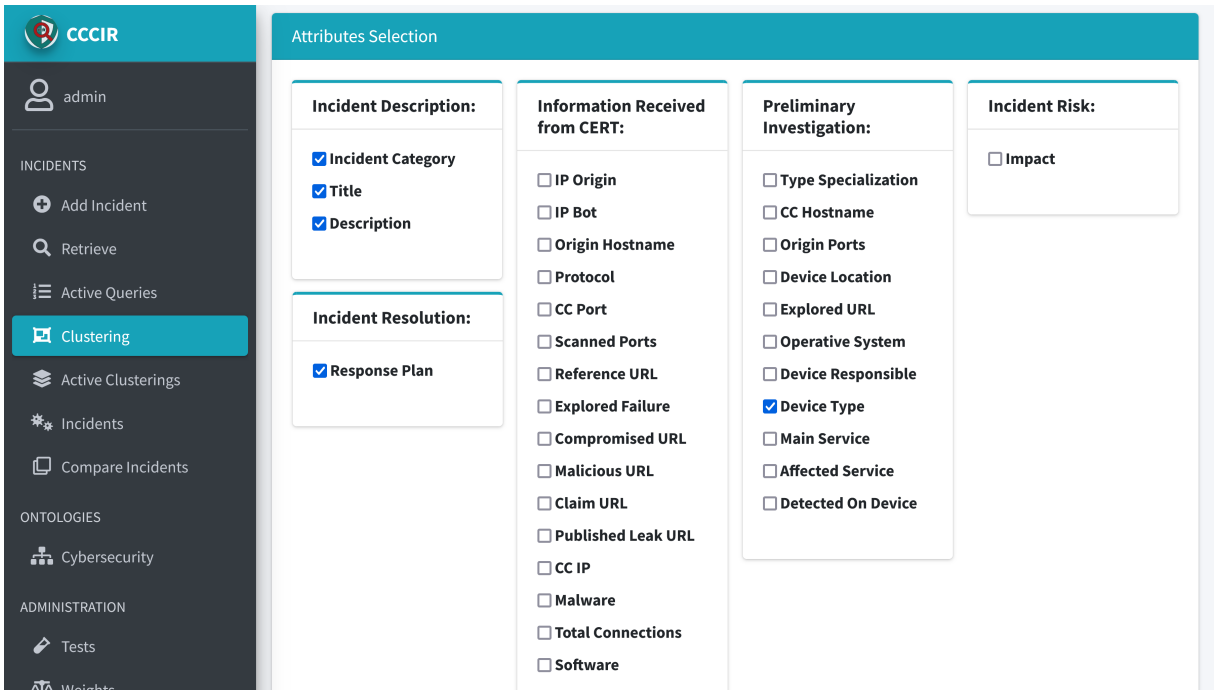
A definição do número de grupos ocorre conforme o algoritmo de clustering usado.

Figura 14 – Interface de seleção do algoritmo de clustering e *active queries*.



Fonte: Autor.

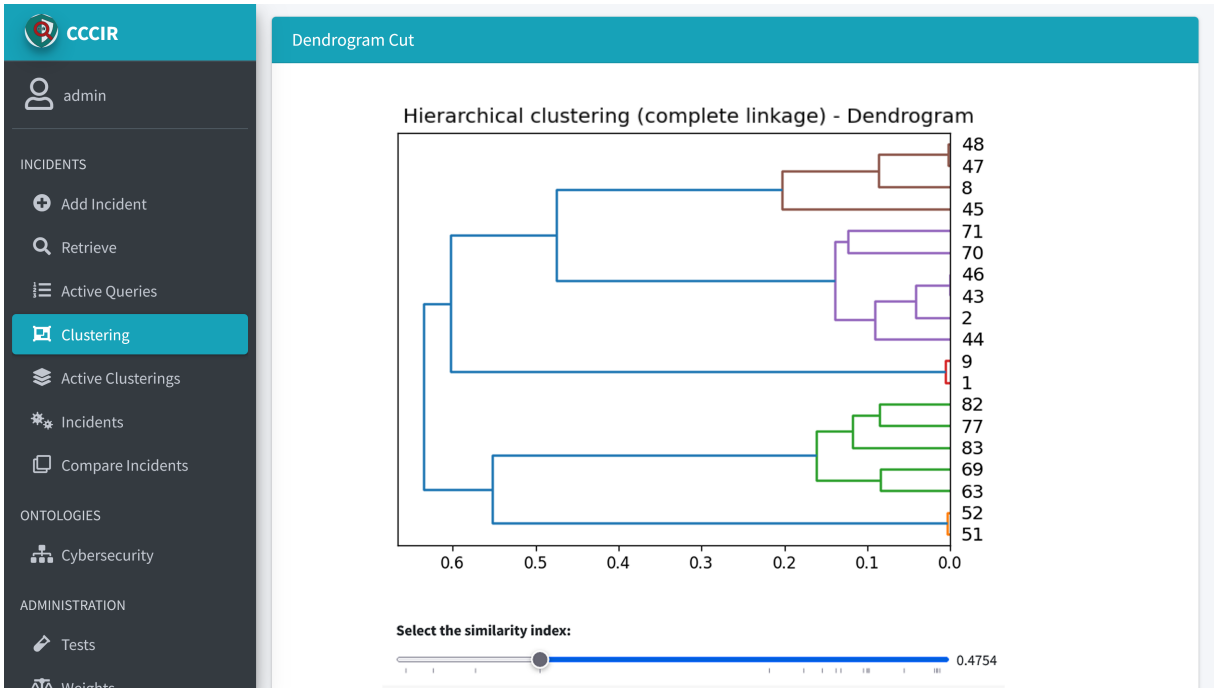
Figura 15 – Interface de seleção dos atributos usados no clustering.



Fonte: Autor.

No agrupamento hierárquico, um dendrograma contendo os casos analisados é apresentado. Para determinar os grupos, o analista de segurança seleciona o limiar de similaridade desejado (Figura 16) para cortar o dendrograma. No agrupamento do *K-Med*, a distorção dos grupos é apresentada de forma gráfica, o que permite ao usuário melhor determinar o número de grupos desejado. O título e descrição a cada um dos grupos formados podem ser definidos pelo usuário, como também o título e a descrição usados no armazenamento do *active clustering* (Figura 17).

Figura 16 – Interface de corte do dendrograma no clustering.



Fonte: Autor.

No exemplo apresentado na Figura 18, os atributos da consulta e do plano de resposta do incidente foram utilizados como entrada nos algoritmos de clustering. Cinco grupos foram formados:

i) grupo 1: incidentes relacionados ao tratamento de servidores de aplicação infectados com *downadup*;

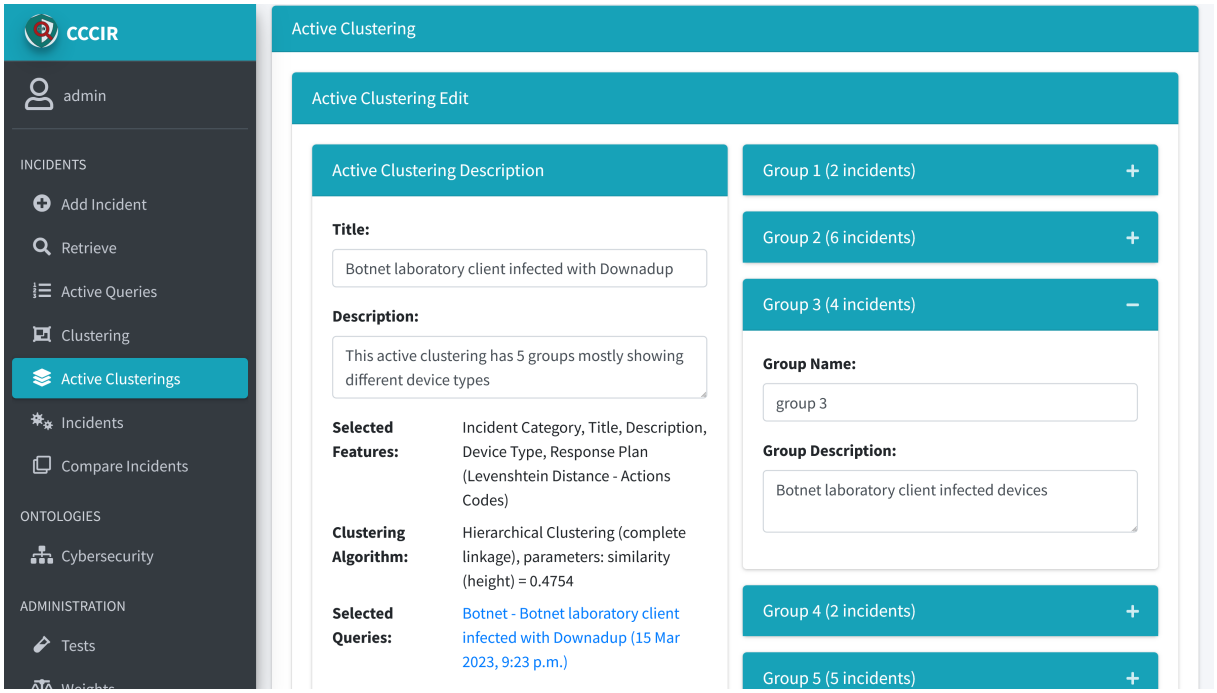
ii) grupo 2: dispositivo clientes infectados com *downadup*;

iii) grupo 3: dispositivo clientes de um laboratório infectados com *downadup* e *avalanche-andromeda*;

iv) grupo 4: dispositivos móveis infectados com *ghost-push*;

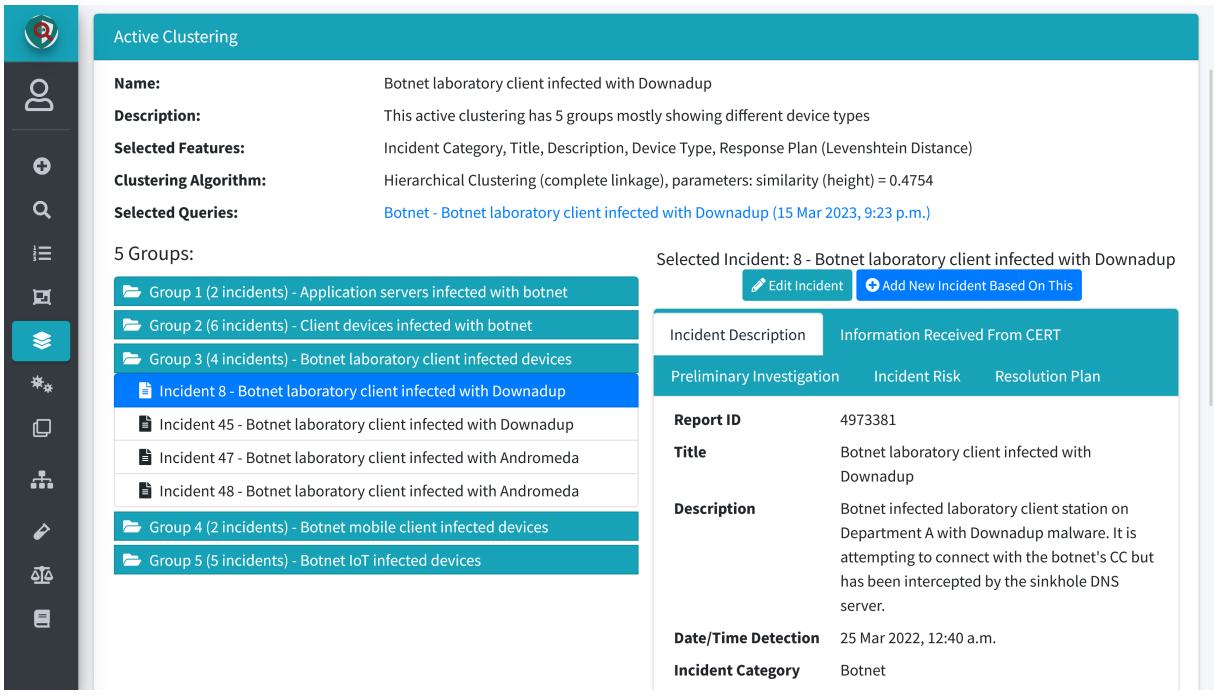
v) grupo 5: dispositivos IoT infectados com *mirai*.

Figura 17 – Interface de edição do *active clustering*.



Fonte: Autor.

Figura 18 – Interface de exibição do *active clustering*.

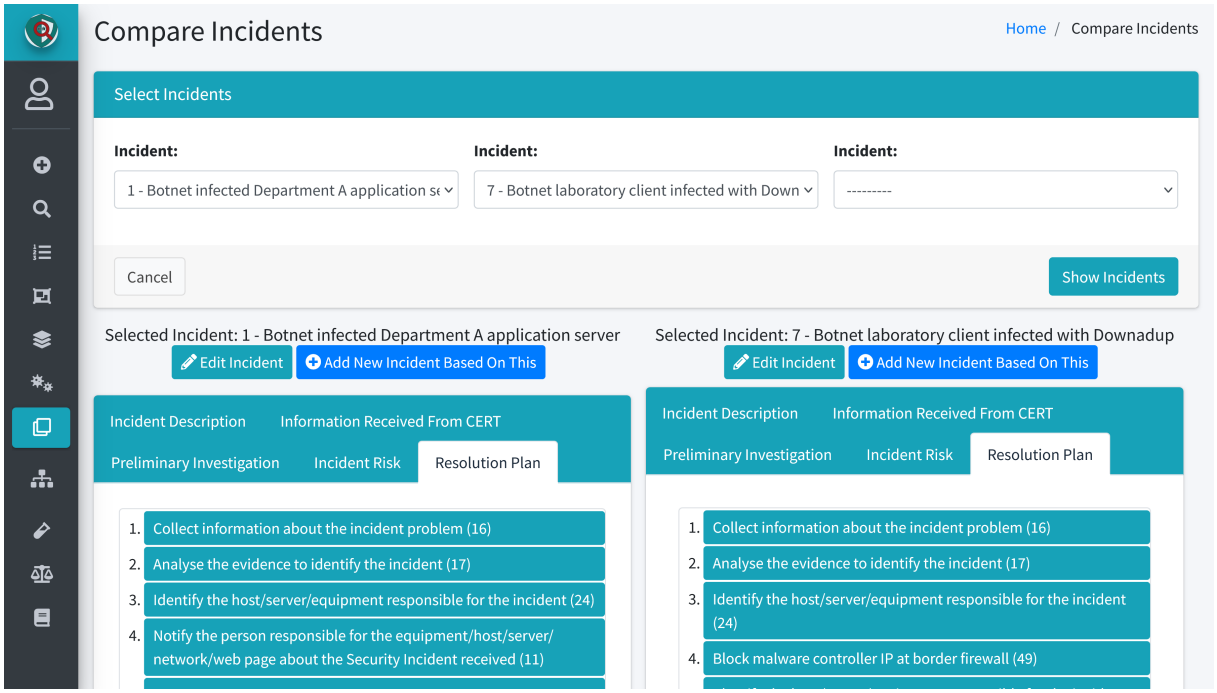


Fonte: Autor.

Complementando uma análise inicial de todos os 19 casos recomendados e ranqueados por similaridade (Figura 13), o uso de clustering permite uma análise dos grupos formados, demonstrando as diferentes variações de situações registradas nos casos selecionados.

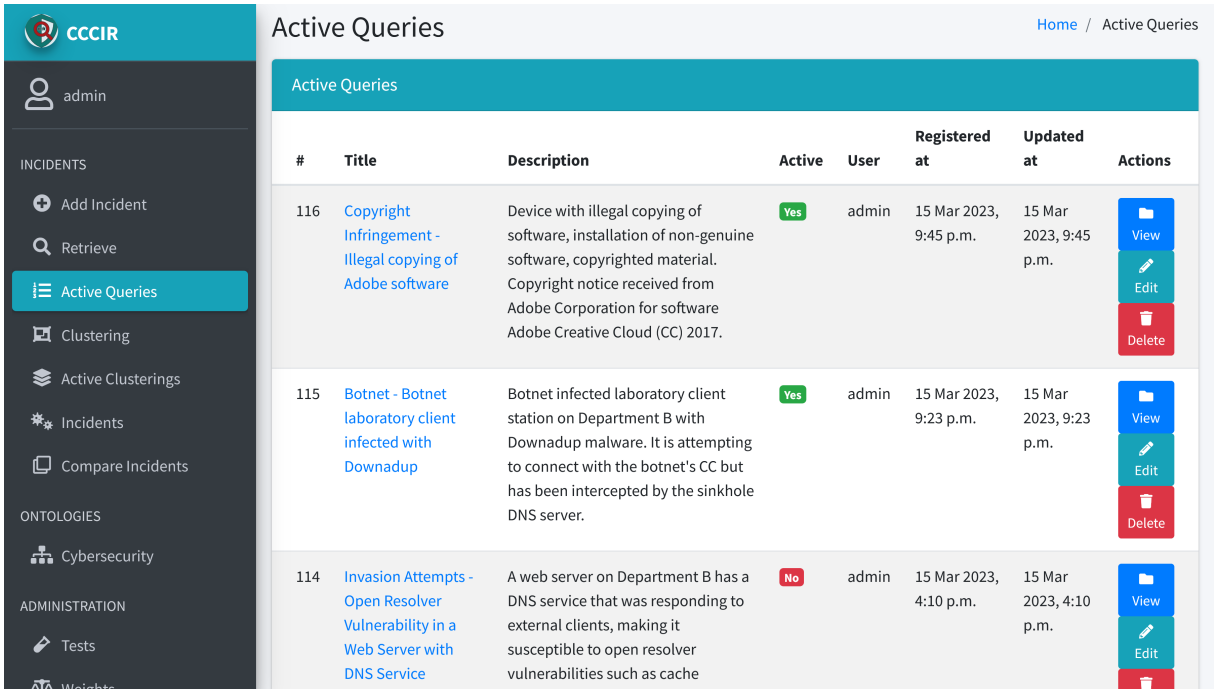
Por fim, o sistema permite que os usuários realizem comparações entre incidentes, seja na descrição dos problemas ou na solução destes incidentes (Figura 19). O usuário pode comparar essas descrições lado a lado, sendo que isso pode ocorrer durante a recuperação e/ou execução do algoritmo de clustering. Todas as *active queries* e os *active clusterings* são armazenados no sistema. Essas informações podem ser editadas e reutilizadas posteriormente, por exemplo, na resposta de incidentes ou na modelagem de novos casos. As interfaces de visualização das *active queries* e *active clusterings* são apresentadas, respectivamente, na Figura 20 e Figura 21.

Figura 19 – Interface de comparação de incidentes.



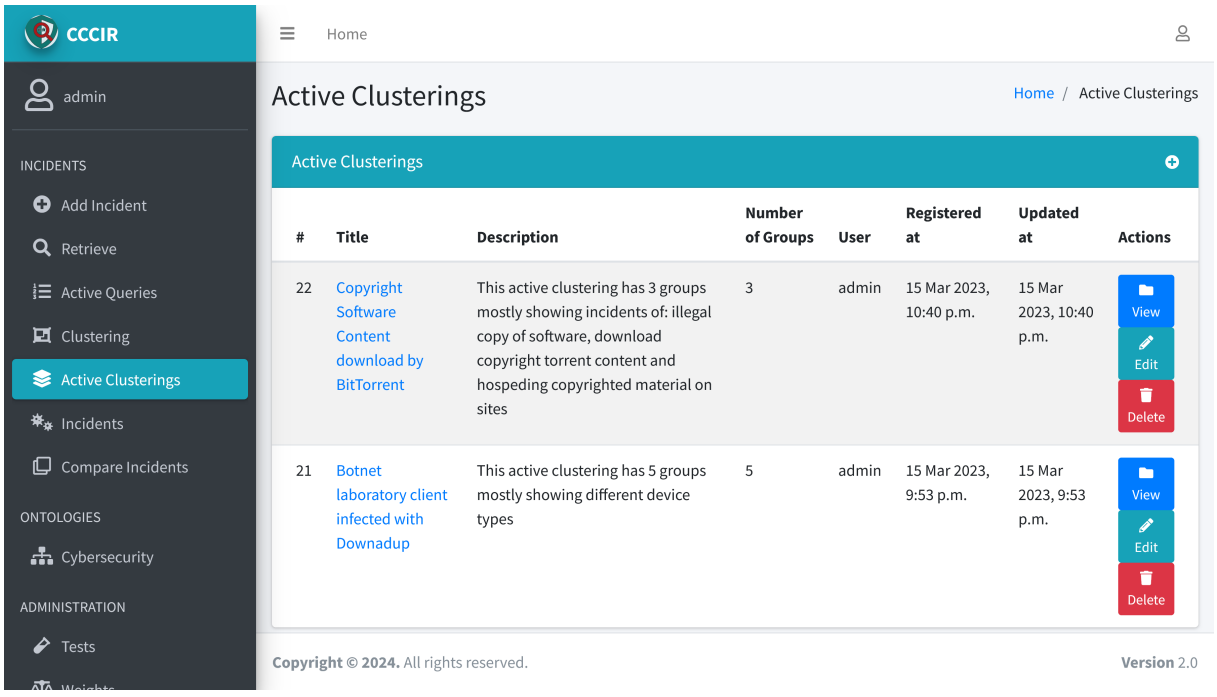
Fonte: Autor.

Figura 20 – Interface de exibição das *active queries*.



Fonte: Autor.

Figura 21 – Interface de exibição dos *active clusterings*.



Fonte: Autor.

7 UM CASO DE RESPOSTA A INCIDENTE DE SEGURANÇA

Um incidente detectado pode ser reportado de diferentes formas, seja através de logs; descrições em texto livre; utilizando um formato conhecido para relatório de incidentes, como IODEF e STIX; ou, ainda, outros formatos internos à organização. Por exemplo, um relatório de incidente recebido em IODEF é apresentado na Figura 22. Este relatório de incidente corresponde à detecção de uma consulta DNS partindo de um host da organização para uma piscina de mineração. Normalmente, esse comportamento é associado a atividades maliciosas, visto que a organização onde o incidente foi detectado não possui qualquer envolvimento com mineração de dados, nem permite a execução de mineração utilizando qualquer dispositivo dentro da rede da organização. Portanto, trata-se de uma ação maliciosa.

Figura 22 – Relatório de incidente em formato IODEF.

```
<ns0:IODEF-Document xmlns:ns0="urn:ietf:params:xml:ns:iodef-1.0" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" lang="en" version="1.00" xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
    <ns0:Incident lang="en" purpose="reporting">
        <ns0:IncidentID name="****-CAIS">5871154</ns0:IncidentID>
        <ns0:ReportTime>2022-12-30T13:11:27+00:00</ns0:ReportTime>
        <ns0:DetectTime>2022-12-30T13:13:46+00:00</ns0:DetectTime>
        <ns0:Description>Dear, CAIS, through its monitoring solutions, has detected that the host listed
below is being used to mine Cryptocurrencies.The ********* ******** network, provided and maintained by
***, aims to provide advanced network services for teaching and research applications. Here is a link to
the *** network usage policy: [...] We request that the system be checked to verify the source of such
activity and that measures are taken necessary to stop such action.[...]
---------------------------------------------------- IP Connected to Miner:Origin Port, IP
Miner:Destination Port, TimeStamp(GMT-3), Incident Description, Protocol ***.***.***.***:40497,
192.3**.***.***:53, 28/12/2022-10:02:24, ,UDP Payload (Base 64) Event name:ET POLICY DNS request for
Monero mining pool Payload Base64:[...]</ns0:Description>
        <ns0:History />
        <ns0:AdditionalData meaning="ip_cc">192.3**.***.***:53</ns0:AdditionalData>
        <ns0:AdditionalData meaning="other_logs">Nome Evento:ET POLICY DNS request for Monero mining
pool  Payload Base64:****</ns0:AdditionalData>
        <ns0:AdditionalData meaning="protocol">UDP</ns0:AdditionalData>
        <ns0:AdditionalData meaning="service_provider_email">cais@***.***.**</ns0:AdditionalData>
        <ns0:AdditionalData meaning="service_provider_entity">CAIS-***</ns0:AdditionalData>
        <ns0:AdditionalData meaning="subject">Host Performing Malicious Activity - CAIS Sensor</
ns0:AdditionalData>
        <ns0:EventData>
            <ns0:Flow>
                <ns0:System category="source">
                    <ns0:AdditionalData meaning="source_port">48597</ns0:AdditionalData>
                    <ns0:Node>
                        <ns0:Address category="ipv4-addr">***.***.***.***</ns0:Address>
                    </ns0:Node>
                </ns0:System>
                <ns0:System category="target">
                    <ns0:Node>
                        <ns0:Address category="ipv4-addr">192.3**.***.***</ns0:Address>
                    </ns0:Node>
                </ns0:System>
            </ns0:Flow>
        </ns0:EventData>
    </ns0:Incident>
</ns0:IODF-Document>
```

Fonte: Autor.

O primeiro passo do fluxo de resposta ao incidente corresponde à conversão do relatório recebido para um formato de problema usado pelo sistema. Para isso, informações adicionais sobre os incidentes são detalhadas na estrutura de casos representada no sistema. O processo de conversão utiliza o relatório de incidente como entrada. Os atributos deste relatório são mapeados para um conjunto de atributos utilizados pelo sistema. O resultado deste processo é um relatório de incidente convertido de acordo com uma tabela de mapeamentos para o conjunto de atributos utilizado no sistema. A tabela 6 (a) apresenta um exemplo do mapeamento do Incidente 5871154 apresentado na Figura 22.

Tabela 6 – Problema do Incidente 5871154 após conversão e análise

	Incident Problem	
Attribute	**After convert (a)**	**After analyse (b)**
Incident Category	-	Malware
Type Specialization	-	Criptominer
Title	Host Performing Malicious Activity - CAIS Sensor	Host infected with criptominer
Description	Dear, CAIS, through its monitoring solutions, has detected that the host listed below is being used to mine Cryptocurrencies. The ********* ******** network, provided and maintained by ***, aims to provide advanced network services for teaching and research applications. Here is a link to the Ipê network usage policy: [...] We request that the system be checked to verify the source of such activity and that measures be taken to stop such action.[...] IP Connected to Miner: Origin Port, IP Miner: Destination Port, TimeStamp(GMT[1]-3), Incident Description, Protocol ***.***.***.***:40497, 192.3**.***.***:53, 28/12/2022-10:02:24, , UDP[2] Payload (Base 64) Event name: ET POLICY DNS request for Monero mining pool Payload Base64:[...]	This incident involves a security event where a device from Department A has made a DNS request for a Monero mining pool. The event was triggered by an IDPS[3] rule that detected the DNS request for a known Monero mining pool domain name, indicating potential unauthorized or malicious activity. This behavior is related to a device infected with criptominer malware.
Malicious URL	-	pool.mine***.com
CC IP	192.3**.***.***	-
CC Port	53 (DNS)	80 (HTTP[4])
Origin Ports	48597	48597
IP Origin	***.***.***.***	***.***.***.***
Device Responsible	-	Department A
Detected On Device	-	Device Itself
Protocol	UDP	HTTP

Fonte: Autor.

[1] *Greenwich Mean Time* (GMT)
[2] *User Datagram Protocol* (UDP)
[3] *Intrusion Detection and Prevention System* (IDPS)
[4] *Hypertext Transfer Protocol* (HTTP)

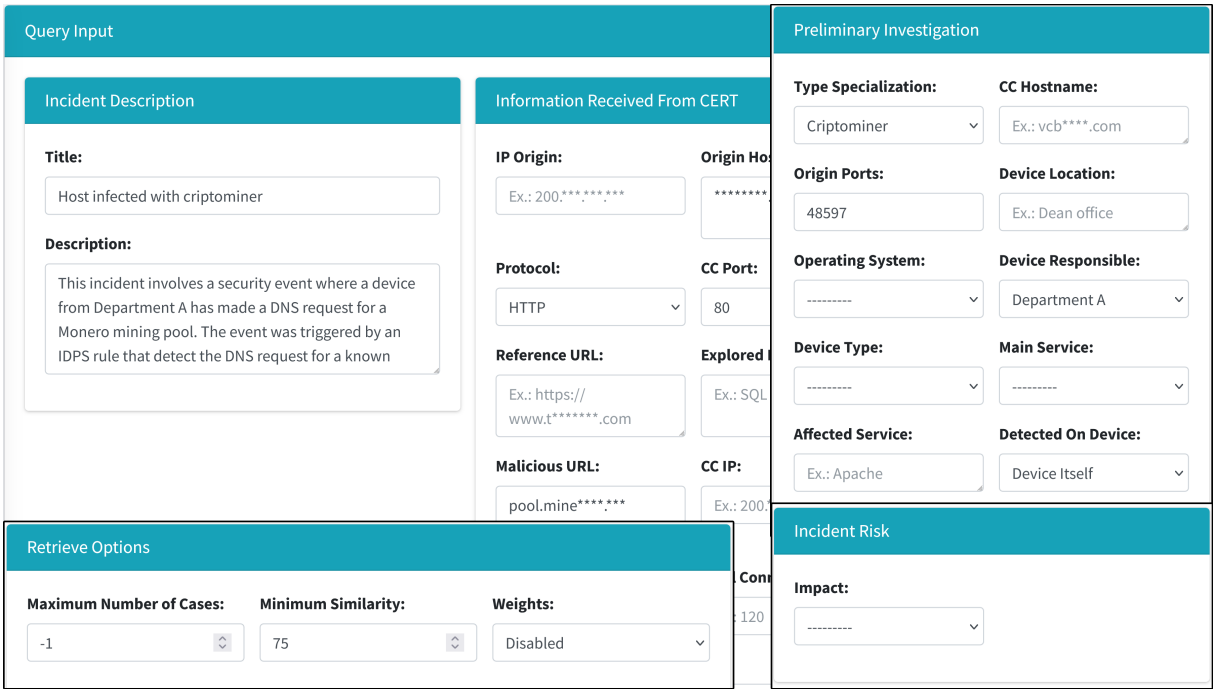
O passo de análise recebe como entrada o relatório. O conteúdo deste relatório é analisado e correções e/ou adições são realizadas. Por exemplo, no contexto apresentado na Tabela 6 (a), o incidente foi detectado com base na ocorrência de uma execução de consulta DNS para resolução de um domínio associado a uma piscina de mineração. No incidente 587154, o relatório inicialmente recebido corresponde à detecção de uma consulta DNS associada a uma piscina de mineração. Apesar disso, essa foi apenas a causa de detecção do incidente. Um ajuste do incidente de segurança, neste caso, é necessário.

Uma sugestão de infecção por *malware* pode ser levantada, em especial, um *malware* com características de *criptominer*. O endereço de destino mapeado para o atributo CC IP também precisa ser removido. Este endereço representa o servidor DNS que detectou a resolução de uma URL maliciosa e não possui relação com o endereço de *Internet Protocol* (IP) do C&C do *malware*. Por outro lado, o *payload* codificado em base 64, descrito no incidente, indica o domínio da piscina de mineração que o host infectado tentou resolver. A porta 53 está relacionada à comunicação com o servidor DNS do host que requisitou a resolução do domínio da piscina de mineração. Este não tem relação com a comunicação do host com o servidor C&C. A análise da faixa de IP do endereço de *IP Origin* também permite identificar informações internas à organização sobre o incidente. Por exemplo, o dispositivo estava conectado à infraestrutura de rede de uma unidade da organização, não correspondendo a um dispositivo conectado a quaisquer servidores PROXY ou NAT. Logo, o incidente foi detectado no próprio dispositivo que efetuou a requisição. A tabela 6 (b) apresenta os atributos do incidente após a execução do passo de análise.

O passo de decomposição envolve identificar os diferentes contextos do novo incidente, onde ele pode ser decomposto em subproblemas que podem ser utilizados na construção de consultas no sistema. Por exemplo, o relatório apresentado na Tabela 6 (b) não indica qual o tipo de dispositivo que originou o incidente. Logo, diferentes valores para o tipo de dispositivo podem ser explorados na definição de problemas a serem investigados por consultas, como *"Internal Device – Client Station"* ou *"Internal Device - Laboratory Client Station"*. Essa decomposição do problema é um processo opcional em que o analista de segurança pode optar ou não pela formação e execução de múltiplas consultas. A Figura 23 apresenta um exemplo da consulta construída com base nos atributos da Tabela 6 (b).

O passo de recuperação envolve a execução de uma ou mais consultas no sistema. Cada consulta resulta em uma lista de casos selecionados e ordenados em ordem decrescente de similaridade. A Figura 24 (a) apresenta a lista de casos recomendados para a consulta apresentada na Figura 24 (b), exibindo o identificador do caso no sistema, o título do incidente e o valor de similaridade. Uma análise dos incidentes recuperados pode ser realizada, avaliando os problemas armazenados nos casos recuperados. Isso permite selecionar um ou mais incidentes em que o plano de resposta pode ser aplicado na resolução do novo incidente.

Figura 23 – Exemplo de consulta para o incidente 5871154.



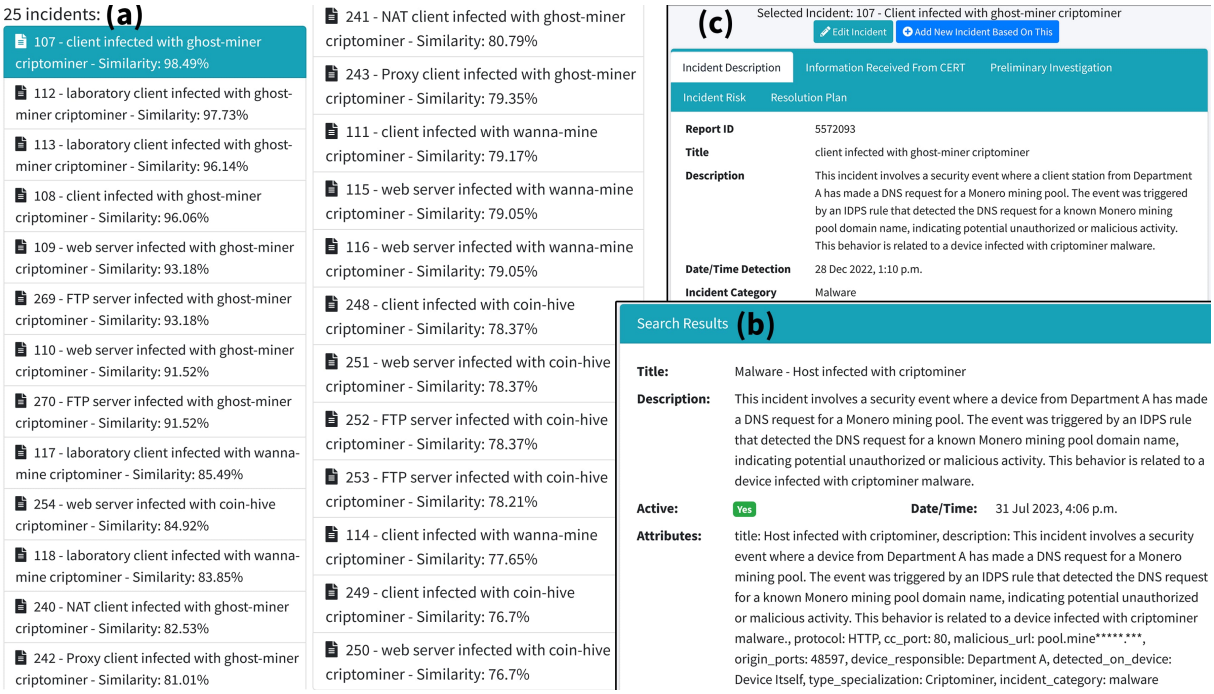
Fonte: Autor.

Considerando que o novo incidente possui um caso correspondente na base de casos com alta similaridade, os procedimentos de resposta registrados no primeiro caso recuperado podem ser reusados. Entretanto, o sistema disponibiliza recursos de clustering que também podem ser usados pelo analista de segurança. Um aspecto que pode dificultar a análise dos casos recuperados é a variedade de incidentes e soluções recuperadas, além da quantidade de casos recuperados, seja quando é realizada uma ou mais consultas no sistema. O agrupamento de recomendações apresentadas para consultas pode auxiliar o analista apresentando grupos de casos similares recomendados. O analista pode selecionar quais atributos devem ser utilizados no agrupamento, incluindo os atributos da descrição dos incidentes (problema) e/ou o plano de tratamento (solução). Nesse contexto, o agrupamento pode auxiliar a filtrar os casos recomendados pelo sistema com base em uma ou mais consultas, tornando mais fácil para o analista o processo de análise dos casos recomendados.

As recomendações de casos apresentadas na Figura 24 foram agrupadas com base nos planos de resposta presentes nos casos. Logo, um conjunto de grupos pode ser levado em conta para avaliação das recomendações apresentadas na consulta. A Figura 25 apresenta um conjunto de grupos obtidos. O analista pode analisar as recomendações partindo da organização em grupos apresentada, neste exemplo, com base nas variações de planos existentes nos casos recomendados.

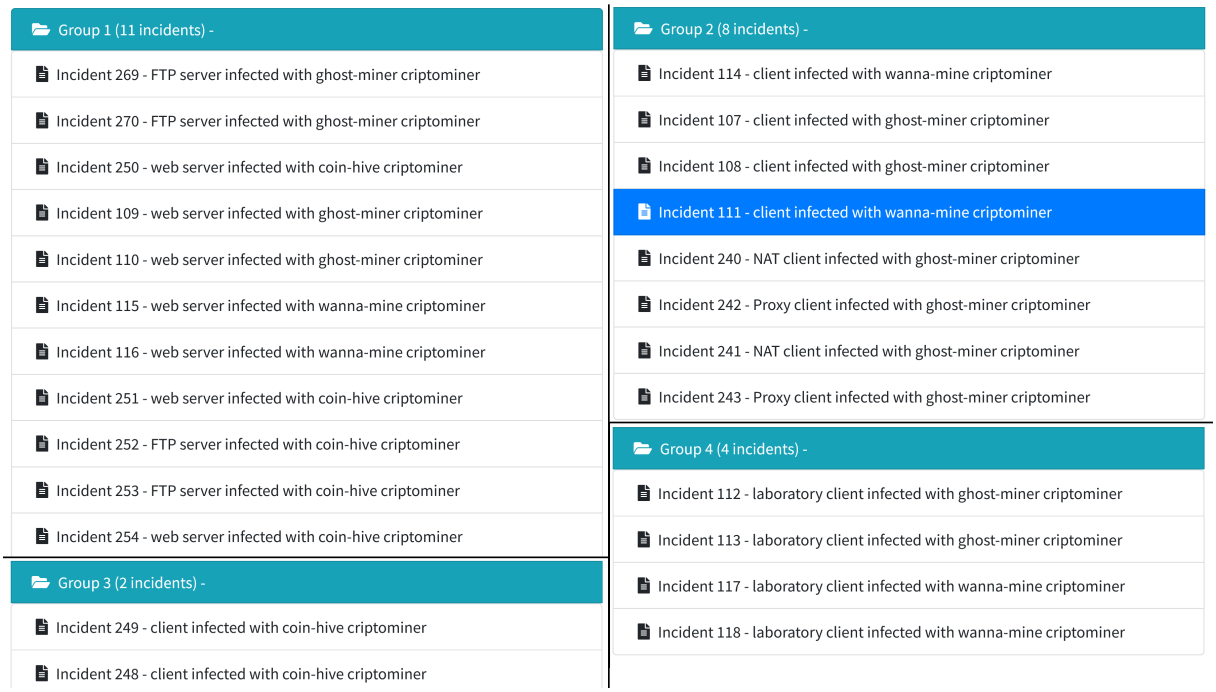
Em um primeiro momento, o analista pode concentrar a análise nos títulos dos casos e selecionar um dos grupos para uma análise mais detalhada. Por exemplo, o analista

Figura 24 – Exemplos de recomendações recuperadas para o incidente 5871154.



Fonte: Autor.

Figura 25 – Exemplos de grupos de recomendações obtidos para o incidente 5871154, com base no agrupamento das recomendações recuperadas para uma consulta.



Fonte: Autor.

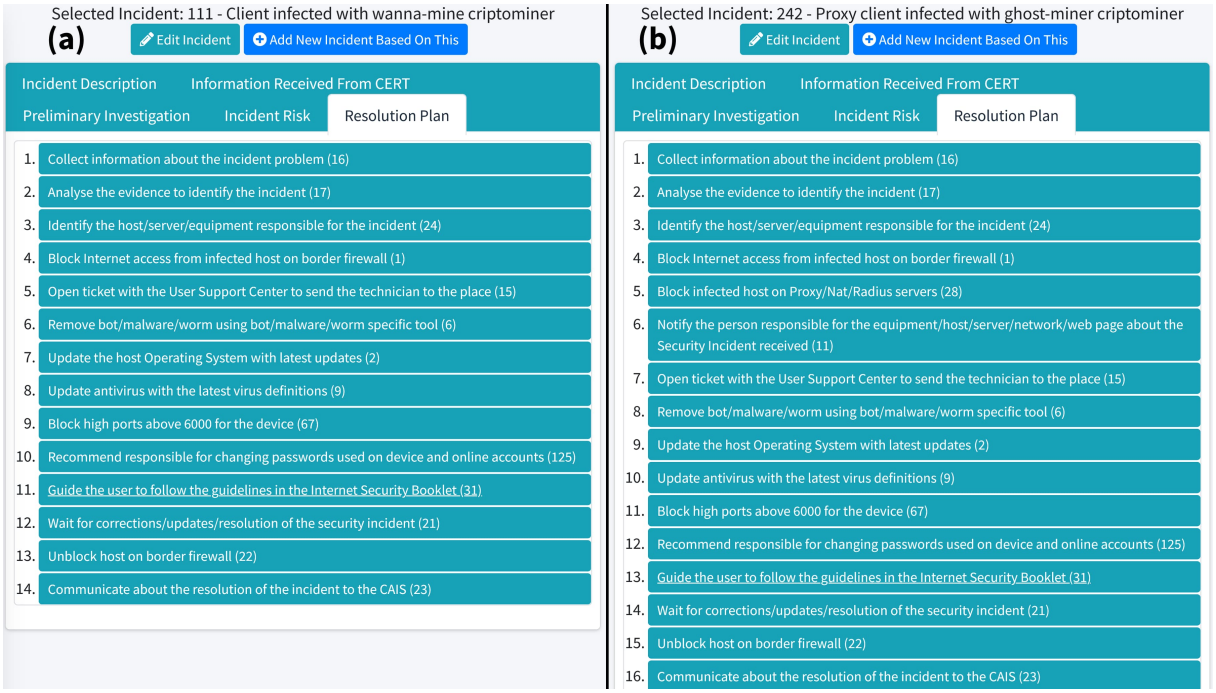
pode considerar a descrição do incidente e o plano de resposta. O primeiro grupo contém 11 incidentes de *criptominers* em servidores; o segundo grupo contempla 8 incidentes de clientes infectados com *criptominers*; o terceiro grupo possui 2 incidentes de clientes (estações de trabalho desktop) infectados com *coin-hive*; e o quarto grupo agrupa 4 incidentes envolvendo dispositivos clientes, provenientes de laboratórios infectados com *criptominers*. Com base nesta interpretação inicial dos grupos, o analista percebe a variedade de hosts previamente infectados por *malwares criptominers* na organização. O analista também percebe a necessidade de identificar qual é o tipo de host afetado, dentre os servidores e clientes (gerais internos e clientes de laboratórios que, neste exemplo, possuem imagens do sistema em espelho em cada laboratório e, ainda, clientes utilizando NAT ou PROXY).

No passo reúso, as soluções podem ser adaptadas e aplicadas levando em consideração as diferenças e particularidades do contexto do novo incidente. Com base na análise do endereço IP do dispositivo, foi descartada a hipótese deste ser um servidor, mas foi identificada uma porta aberta comumente usada para o compartilhamento de arquivos e impressoras em dispositivos *Windows*: *Transmission Control Protocol* (TCP)/445 *(Microsoft Directory Services)*. O host também está dentro da faixa de endereços IP do setor administrativo da organização, logo, o dispositivo corresponde a uma estação de cliente do setor administrativo. Com base nesta informação, o analista pode selecionar um dos grupos que tem maior similaridade com o contexto identificado - por exemplo, o grupo 2. Em seguida, o analista pode realizar uma análise das descrições dos incidentes. Nesse processo, foram selecionados dois incidentes que possuem planos de resposta que podem ser adaptados e aplicados na resposta do novo incidente. A Figura 26 apresenta estes dois planos selecionados: (a) cliente infectado com *wanna-mine criptominer*; e (b) cliente PROXY infectado com *ghost-miner criptominer*.

A Figura 26 (a) apresenta um plano de tratamento para uma estação cliente que não possui relação com o setor administrativo. Entretanto, a ação *“Block Internet access from infected host on border firewall (1)”* não deve ser executada no contexto do novo incidente, mas sim duas ações no lugar desta: *“Notify the person responsible for the equipment/host/server/network/web page about the Security Incident received (11)”*, com base no plano apresentado na Figura 26 (b), e a ação *“Request to the person responsible for the equipment/host/server to immediately stop communicating with the data network (35)”*, adicionada utilizando a ontologia. A ação *“Unblock host on border firewall (22)”* também não deve ser executada, visto que o host não teve o acesso à Internet, bloqueada anteriormente no firewall de borda. Isso permite que o usuário do dispositivo desative a conexão com a Internet assim que possível, mas não interrompendo qualquer ação crítica que possa estar sendo desempenhada no host. Uma vez criado o plano de resposta adaptado para o contexto do novo incidente, o analista executa as ações presentes no plano. O plano de resposta utilizado no incidente 5871154 é apresentado na Figura 27.

No passo revisão, ações relacionadas aos planos utilizados podem ser revisadas e

Figura 26 – Exemplos de planos selecionados para reúso no tratamento do incidente 5871154, com base no segundo grupo.

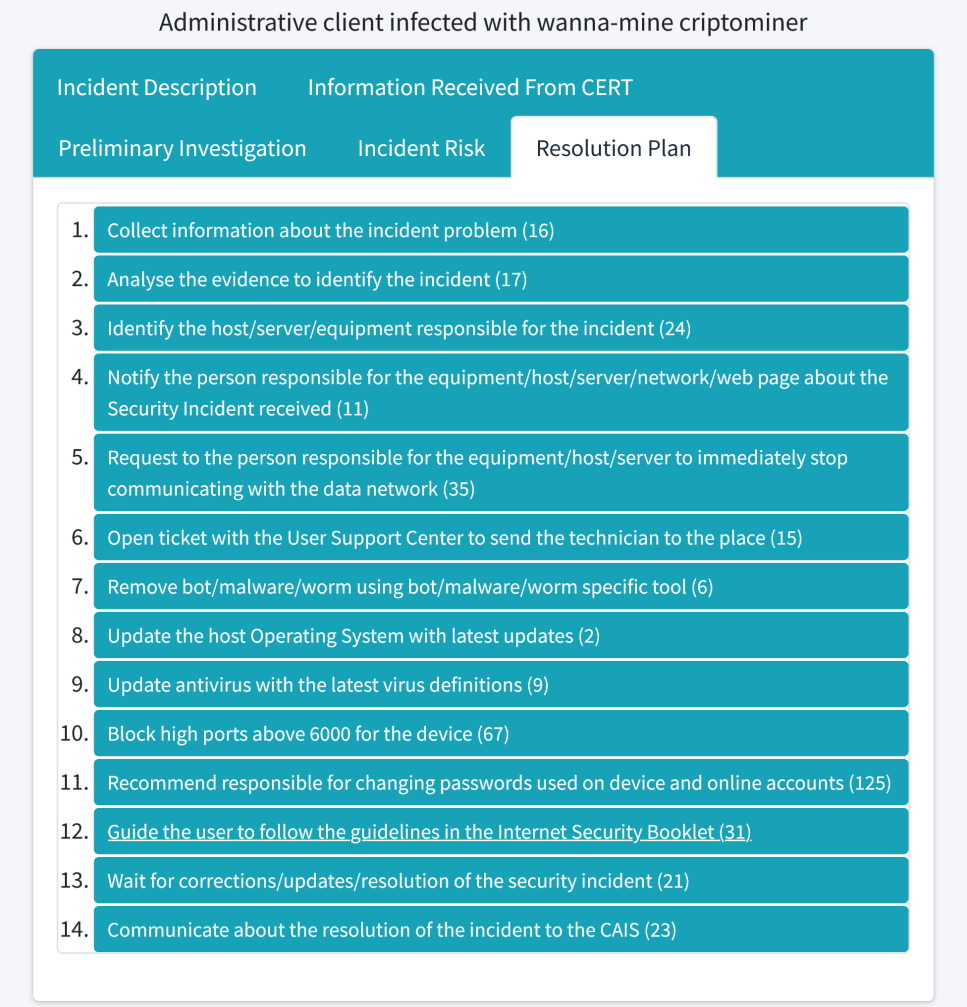


Fonte: Autor.

corrigidas, se necessário. Também devem ser identificadas quaisquer limitações ou ajustes necessários para o contexto do incidente, não apenas antes da execução do plano de resposta ao novo incidente, mas também durante ou após sua execução. Após finalizada a etapa de revisão das ações utilizadas para tratar o novo incidente, se necessário, podem ser feitas atualizações nos planos. Análises complementares podem ser desencadeadas neste passo, inclusive de atualização de outros casos com contextos relacionados, colaborando para a manutenção da base de casos.

Por fim, o passo de retenção contempla a atualização dos casos ou a criação de novos na base de casos, contemplando novas soluções ou contextos empregados na resposta a novos incidentes. Isso ocorre especialmente quando o incidente apresenta um novo contexto, como a resposta a um cliente administrativo infectado com *criptominer*, cujo plano aplicado para a resposta do incidente (5871154) é apresentado na Figura 27.

Figura 27 – Exemplo de plano para o tratamento do incidente 5871154, criado com base no reúso de planos e na ontologia.



Fonte: Autor.

8 EXPERIMENTOS E RESULTADOS

Este capítulo avalia o emprego de clusters e casos no apoio ao processo de resposta a incidentes de segurança cibernética (CCCIR). Diferentes tipos de experimentos foram conduzidos para avaliar os métodos desenvolvidos neste trabalho. Os resultados destes experimentos são apresentados e discutidos ao longo deste capítulo.

Uma versão inicial da base de casos, usada neste trabalho, foi originalmente organizada e investigada em Nunes et al. (2019). Neste trabalho, essa base de casos foi revisada e atualizada de várias formas. Essas tarefas envolveram a modificação e adição de novos atributos para caracterizar as informações de contexto e detecção dos incidentes de segurança cibernética.

Os experimentos utilizaram uma base contendo 354 casos, representando 5 categorias de incidentes. A primeira categoria, relacionada a *botnets*, inclui 102 casos. Esses casos detalham infecções de dispositivos por *bots* de *botnets* como *downadup*, *avalanche-andromeda*, *mirai*, entre outros. A segunda categoria, centrada na violação de direitos autorais, inclui 48 casos. Esses casos envolvem o uso de licenças de software inválidas, distribuição e hospedagem de conteúdo protegido por *copyright* sem haver as devidas permissões. A terceira categoria, abordando tentativas de invasão/exploração de vulnerabilidades, registra 127 casos. Esses casos detalham incidentes sobre ataques de força bruta, exploração de vulnerabilidades e configurações inadequadas de servidores web. A quarta categoria, *phishing*, inclui 21 casos. Esses casos estão relacionados, por exemplo, à perda de acesso a contas de usuários por meio de links maliciosos recebidos por e-mail. Finalmente, a quinta categoria, *malware*, engloba 56 casos, que incluem criptominers como *coin-hive*, *ghost-miner* e *wanna-mine*, além de outros tipos de *malware*, como *ransomwares*.

A estrutura de cada caso contempla duas partes: detalhes do contexto e da detecção do incidente (**problema**) e um plano de resposta com ações de resposta para o incidente (**solução**). Neste trabalho, foram realizadas melhorias na forma como os planos de resposta são representados como soluções para os incidentes registrados. Durante a formação da base de casos, os planos de resposta utilizavam conjuntos de ações descritas por analistas de segurança de maneira simplificada, com uma redação livre, ocasionando problemas como redundância de ações nos planos representados nestes casos. Diante deste cenário, este trabalho desenvolveu e empregou uma ontologia para organizar e padronizar a representação de procedimentos de resposta a incidentes de segurança. Resultados preliminares deste trabalho de aquisição e representação de conhecimento foram publicados em Guerra et al. (2023). Mais importante, os experimentos e resultados desenvolvidos neste trabalho amplamente utilizam essas representações de planos de resposta a incidentes, que são indexadas e organizadas de acordo com conceitos definidos na on-

tologia desenvolvida neste trabalho. Logo, o emprego da ontologia desenvolvida tem um impacto significativo nas computações que são realizadas nos experimentos apresentados neste capítulo. Sem tal padronização nas representações de planos de respostas de incidentes, análises de similaridades entre as sequências de ações de resposta, descritas nesses planos, não poderiam ser realizadas usando métricas de distância amplamente conhecidas na literatura.

Novos casos também foram coletados junto a um data center e representados de acordo com o modelo de casos proposto nesta pesquisa. Isso permitiu ampliar a base de casos utilizada em Guerra et al. (2023). As informações obtidas sobre os incidentes foram ajustadas e mapeadas para o conjunto de atributos utilizado na representação dos casos. Com base nesses incidentes coletados, um plano de resposta para cada incidente foi construído, utilizando ações de resposta padronizadas na ontologia construída.

Embora a pesquisa onde esta dissertação está inserida explore a ampliação das funcionalidades de um sistema de resposta a incidentes de segurança (NUNES et al., 2019) com 12 categorias de incidentes de segurança cibernética, a coleta de novos casos para os experimentos foi desafiadora, por vários motivos, assim como a obtenção de planos de ações de resposta para incidentes coletados. Adicionalmente, o ajuste das informações que caracterizam os incidentes foi crucial, uma vez que os registros iniciais de incidentes continham informações inconsistentes, registros de maneira incorreta e com valores de atributos faltantes. A maioria desses problemas foi resolvida nas etapas de aquisição e representação de conhecimento que resultou na construção da base de casos neste problema de aplicação.

Para o desenvolvimento de experimentos de validação envolvendo a resposta a novos incidentes de segurança (não vistos durante o desenvolvimento deste trabalho), 56 novos incidentes reais também foram coletados junto à equipe de resposta a incidentes de um data center. Em seguida, um plano com ações de resposta foi construído para cada um desses incidentes. Esses planos foram representados de acordo com os conceitos modelados na ontologia construída. Os novos incidentes utilizados nos experimentos possuem as mesmas categorias de casos presentes na base de casos. Esses novos problemas representam em torno de 20% dos incidentes de cada categoria presente na base de casos, conforme Tabela 7.

Os experimentos foram desenvolvidos em Python 3.9, servidor de desenvolvimento do Django versão 3.2 e browser Firefox.

Tabela 7 – Casos representando incidentes de segurança utilizados nos experimentos de validação desenvolvidos neste trabalho.

Categoria de Incidente	Novos Incidentes	Base de Casos
Botnet	20	102
Violação de Direitos Autorais	10	48
Tentativa de Invasão/Exploração de Vulnerabilidades	25	127
Malware	11	56
Phishing	4	21
Total	70	354

Fonte: Autor.

8.1 ANÁLISE DE PLANOS DE RESPOSTA PARA INCIDENTES DE SEGURANÇA

Nos experimentos desenvolvidos neste trabalho, casos usados como consulta no sistema possuem informações sobre o incidente. Essas informações são registradas nos atributos do **problema** representado no modelo de casos. Além disso, esses casos também registram um plano de resposta para o incidente (**solução**), onde tal plano foi construído com o apoio de analistas de segurança. Baseado nestes casos, o problema é determinar quando o sistema recuperou um caso (ou casos, dentre uma lista de casos mais similares recuperados) que é relevante para resolver o problema descrito no caso usado como consulta. Se a maioria dos casos mais similares recuperados para um incidente usado como consulta possui um plano de resposta similar (neste caso, reusável) ao plano de resposta registrado no caso usado como consulta, é possível definir que o sistema apresenta uma recomendação que adequadamente apoia a resposta do incidente a ser resolvido. Neste caso, o sistema apresenta uma resposta correta para o novo problema.

Utilizando a estrutura de casos descrita anteriormente, a avaliação de um plano de resposta a incidente recomendado (caso recuperado) como resposta a um novo incidente usado como consulta no sistema (caso consulta) pode ser associada à quantidade de modificações necessárias para transformar o plano recuperado pelo sistema ($ResponsePlan_{Case}$) no plano previamente registrado no caso usado como consulta ($ResponsePlan_{Query}$). Na prática, isso envolve transformar a sequência de ações do plano de resposta recomendado na sequência de ações registrada no plano registrado no caso consulta.

Em geral, os experimentos desenvolvidos neste trabalho consideram a similaridade entre os planos de resposta para incidentes. Utilizando a similaridade entre estes planos de resposta, é possível classificar o plano recomendado (isto é, registrado em um caso mais similar recuperado) como relevante ou não para apoiar a resposta a um novo incidente usado na consulta. Caso a similaridade entre um plano recomendado pelo sistema e um plano registrado em algum dos casos mais similares recuperados pelo sistema seja igual ou maior do que um determinado valor ou limiar de similaridade (75%), então o plano de

resposta registrado no caso recuperado é considerado como *relevante* para responder o incidente dado. Caso contrário, ele é considerado como *não relevante*.

Para analisar essa relevância, este trabalho utiliza a distância de Levenshtein para comparar planos de resposta para incidentes de segurança. Neste caso, a distância de Levenshtein mede a quantidade mínima de edições necessárias para transformar um plano de resposta em outro. Logo, essa métrica é aplicada para avaliar a similaridade entre ações de resposta registradas em diferentes planos de resposta. Essa métrica de distância também pode ser utilizada para interpretar o esforço de reúso envolvido na adaptação de um plano passado para a resposta a um problema corrente. A similaridade entre sequências de ações de resposta representadas nestes planos tem como base as adições, remoções e alterações nas ações do plano. Logo, quanto maior a similaridade entre os planos, menor é o esforço envolvido na adaptação do plano a ser reusado na resposta de um novo incidente.

Por exemplo, dados 2 planos, um plano recuperado e um plano-alvo, cada um composto por 10 ações de resposta. Dessas ações, 8 são idênticas (conforme análise apoiada pela ontologia proposta neste trabalho) e estão dispostas na mesma ordem em ambos os planos. Portanto, para tornar o plano recuperado igual ao plano base, apenas 2 ações precisam ser ajustadas. Logo, se a similaridade entre os planos é 80%, o esforço de reúso é inversamente proporcional, ou seja, 20%.

Em resumo, os experimentos desenvolvidos neste trabalho consideram o seguinte critério: $similarity_{Levenshtein}(ResponsePlan_{Query}, ResponsePlan_{Case}) \geq$ *PlanSimilarityThreshold*. Em particular, testes executados neste trabalho permitiram definir um *PlanSimilarityThreshold* de 75% como critério de análise da relevância/reusabilidade de planos de resposta. Na prática, um baixo esforço de reúso de planos de resposta a incidentes é obtido quando este limiar é usado. A utilização deste limiar permitiu uma avaliação da equivalência entre planos de resposta para incidentes, assegurando que apenas aqueles que compartilham uma porcentagem significativa de ações de resposta em comum (porcentagem que também considera o volume de casos representados na base de casos; portanto, a competência da base de casos em resolver esses problemas) sejam considerados como respostas relevantes para os problemas (incidentes) usados como consultas nos testes executados neste trabalho.

8.2 MÉTRICAS DE AVALIAÇÃO

Os experimentos desenvolvidos neste trabalho empregam métricas tradicionais de avaliação em Machine Learning, como: precisão, recall, F-score e acurácia. Estas métricas desempenham um papel crucial na avaliação de sistemas de recomendação em diversos domínios de aplicação (SEGOVIA-AGUAS; JIMÉNEZ; JONSSON, 2020) (CHEN

et al., 2023) (LIU et al., 2019). Em particular, os problemas de aplicação abordados nestes trabalhos envolvem problemas de recomendação em que soluções têm o formato de planos (sequências ordenadas de ações). Isso é similar ao enfoque adotado nesta pesquisa, onde casos de resposta a incidentes apresentam planos descritos como uma sequência de ações de resposta, as quais são indexadas e organizadas por meio de conceitos definidos em uma ontologia.

No problema investigado neste trabalho, o sistema analisa os casos existentes na base de casos para recomendar soluções usáveis (reusáveis) para um novo problema: nesta situação, um plano de resposta para um novo incidente. Em sistemas CBR, essa decisão é determinada pela análise de similaridade entre os atributos dos incidentes registrados nos casos. Ou seja, os detalhes dos incidentes registrados como problemas no caso de consulta e nos casos da base de casos. Ela também considera um LR de casos pré-definido no sistema CBR. Por exemplo, um caso é recomendado pelo sistema se $similarity_{CBR}(IncidentProblem_{Query}, IncidentProblem_{Case}) \geq$ *RetrievalThreshold*. Em particular, testes executados neste trabalho permitiram analisar diferentes valores de limiar de similaridade usados pelo algoritmo de recuperação implementado no sistema CBR. A partir desta análise (experimento 1), os experimentos subsequentes executados no trabalho (experimentos 2, 3 e 4) utilizaram o *RetrievalThreshold* (LR) de 75%.

Para avaliar a eficácia do sistema, é necessário analisar se os casos recomendados (neste caso, os casos recuperados pelo sistema para a consulta dada) são relevantes ou não para apoiar a resolução do novo problema. Isso é analisado com base na similaridade dos planos de resposta registrados nos casos recuperados pelo sistema. A decisão considera se os planos de resposta do caso consultado e de cada um dos casos recuperados são similares o suficiente para oferecer uma recomendação considerada reusável para resolver o novo incidente. Neste trabalho, o sistema analisa se os casos recuperados para a consulta apresentam um limiar de similaridade do plano adequado, ou seja, atendem ao seguinte critério: $similarity_{Levenshtein}(ResponsePlan_{Query}, ResponsePlan_{Case}) \geq$ 75%.

Mesmo que um plano de resposta não tenha sido recuperado pelo sistema por não atender ao critério de similaridade usado pelo algoritmos de recuperação, é essencial determinar se o caso contendo um plano de resposta reusável deveria ter sido recomendado pelo sistema. Esses casos que contêm planos reusáveis para o incidente dado como consulta, mas que não foram recuperados pelo sistema, são tomados neste trabalho como “falsos negativos”.

Em resumo, os critérios para popular uma matriz de confusão usada na análise de resultados experimentais computados neste trabalho são os seguintes:

i) Verdadeiros Positivos (VP):
$RetrievedCases_{aQuery}$ = $\{\{case_1, \ldots, case_n\} \in CB \mid$
$similarity_{CBR}(IncidentProblem_{aQuery}, IncidentProblem_{aCase}) \geq$
RetrievalThreshold AND

$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) \geq$ 75%$\}$

ii) Falsos Positivos (FP):
$RetrievedCases_{aQuery}$ = $\{\{case_1, \ldots, case_n\} \in CB \mid$
$similarity_{CBR}(IncidentProblem_{aQuery}, IncidentProblem_{aCase}) \geq$
RetrievalThreshold AND
$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) <$ 75%$\}$

iii) Verdadeiros Negativos (VN):
$RetrievedCases_{aQuery}$ = $\{\{case_1, \ldots, case_n\} \in CB \mid$
$similarity_{CBR}(IncidentProblem_{aQuery}, IncidentProblem_{aCase}) <$
RetrievalThreshold AND
$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) <$ 75%$\}$

iv) Falsos Negativos (FN):
$RetrievedCases_{aQuery}$ = $\{\{case_1, \ldots, case_n\} \in CB \mid$
$similarity_{CBR}(IncidentProblem_{aQuery}, IncidentProblem_{aCase}) <$
RetrievalThreshold AND
$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) \geq$ 75%$\}$

Mais ainda, as análises dos planos de resposta realizadas neste trabalho também consideram que esses planos foram construídos para incidentes que possuem as mesmas categorias, considerando as 5 categorias de incidentes investigadas: *botnets*, violação de direitos autorais (*copyright*), invasão/exploração de vulnerabilidades, *phishing* e *malware*. Baseado nestes critérios, as seguintes métricas são usadas na computação de resultados para os experimentos desenvolvidos neste trabalho:

a) Precisão *(Precision)*: A precisão mede a proporção de recomendações corretas entre todas as recomendações feitas pelo sistema. Em outras palavras, ela destaca a qualidade das recomendações positivas, indicando a capacidade do sistema de evitar FP. Equação 8.1 é utilizada para calcular a precisão:

$$pr = \frac{VP}{VP + FP} \tag{8.1}$$

b) Recall (Revocação ou Sensibilidade): O recall avalia a capacidade do sistema em recuperar todos os casos relevantes na base de casos. Ele é calculado como o número de recomendações corretas dividido pelo número total de casos relevantes na base de casos. O recall destaca a capacidade do sistema de evitar FN, ou seja, evitar deixar de apresentar todas as recomendações relevantes.

Equação 8.2:

$$re = \frac{VP}{VP + FN} \tag{8.2}$$

c) F-Score: O F-Score é uma métrica que combina precisão e recall em uma única medida, proporcionando uma visão equilibrada do desempenho do sistema. O F-Score é calculado como a média harmônica entre precisão e recall, fornecendo uma medida única que considera tanto FP quanto FN. A Equação 8.3 mostra como a métrica é obtida:

$$fs = 2 \times \frac{pr \times re}{pr + re} \tag{8.3}$$

d) Acurácia *(Accuracy)*: A acurácia é uma métrica que mede a proporção de previsões corretas (tanto positivas quanto negativas) em relação ao total de previsões feitas pelo sistema. Isso inclui tanto as recomendações corretamente identificadas (VP e VN) quanto as incorretas (FP e FN). O cálculo da acurácia é descrito pela Equação 8.4:

$$ac = \frac{VP + VN}{VP + VN + FP + FN} \tag{8.4}$$

Além de conduzir experimentos utilizando técnicas de CBR, este trabalho desenvolve diferentes tipos de testes envolvendo o uso de técnicas de clustering. A seleção de um grupo de casos formado a partir de casos mais similares recuperados para uma consulta executada é analisada neste trabalho de acordo com resultados de **precisão**.

Neste trabalho, os casos organizados em um determinado cluster podem corresponder a duas classes distintas:

i) Classe contendo casos com planos de resposta *reusáveis* para responder um incidente dado como consulta:
$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) \geq 75\%$

ii) Classe contendo casos com planos de resposta *não reusáveis* para responder um incidente dado como consulta:
$similarity_{Levenshtein}(ResponsePlan_{aQuery}, ResponsePlan_{aCase}) < 75\%$

8.3 EXPERIMENTOS DESENVOLVIDOS

Nesta pesquisa, foram desenvolvidos 4 diferentes experimentos relacionados às propostas de emprego de CBR e clustering na resposta a incidentes de segurança ciber-

nética. Estes experimentos são os seguintes:

Experimentos de validação cruzada: Os dois primeiros experimentos consistem de uma validação cruzada utilizando o método Leave-One-Out (LOO). Para desenvolver esse experimento, é removido um incidente da base de casos a cada iteração de teste. Isso indica que o mecanismo de recuperação do sistema tem disponível o restante da base de casos para fornecer recomendações para o incidente selecionado (caso) e usado como consulta no sistema. Dessa forma, é possível avaliar a capacidade do sistema em recomendar planos de resposta relevantes (reusáveis) para tratar o incidente corrente.

Experimentos com novos problemas: Os dois últimos experimentos consistem na avaliação da capacidade do sistema em fornecer recomendações para novos incidentes. Esse enfoque permite aferir a capacidade em recuperar planos relevantes para tratar incidentes não anteriormente vistos durante o desenvolvimento do sistema.

Nos experimentos realizados, diferentes limiares de recuperação foram adotados no mecanismo de recuperação de casos, o qual é usado para obter respostas para consultas CBR executadas no sistema. Esses limiares variaram de 60% a 100%, com variação de 5% entre cada execução, e representam o critério de similaridade que determina quais casos podem ser considerados e analisados como relevantes durante a fase de recuperação, onde esses casos são apresentados como recomendações para apoiar a solução de um incidente usado como consulta no sistema.

A variação nos valores do LR permite avaliar e ajustar a acurácia do sistema. Alguns cenários de solução de problemas com o apoio de sistemas CBR podem demandar uma abordagem mais flexível, admitindo casos recuperados com similaridades mais baixas. Em outros cenários, uma abordagem mais restritiva pode ser requerida, considerando apenas similaridades entre problemas que descrevem situações muito semelhantes. Dessa forma, a escolha de valores distintos para o LR pode influenciar diretamente o desempenho do sistema.

Especificamente, o valor de LR de 75% foi escolhido de forma experimental (conforme resultados do experimento 1) neste trabalho. Este LR representa um equilíbrio entre a exigência de similaridade entre o caso utilizado na consulta e os casos recuperados, em relação ao número de casos recuperados. O valor de LR definido considera o número de casos atualmente registrados na base de casos. A inclusão de novos casos na base de casos pode vir a permitir o aumento desse valor de similaridade mínimo, visto que a competência da base de casos em resolver novos problemas pode ser incrementada/melhorada com a inclusão destes casos. Essa escolha de LR de 75% também se justifica pela necessidade de haver certa flexibilidade diante de alterações, adições e remoções de ações de resposta necessárias para o reúso de um plano de resposta a incidentes. O limiar de similaridade mínima do plano definido garante adaptabilidade a mudanças nos planos recuperados pelo sistema e reusados na solução de problemas dados.

A seguir, o objetivo de cada experimento é descrito.

1) **Experimento 1: Resposta a incidentes de segurança utilizando o reúso de planos de resposta recuperados via técnicas de CBR - Validação Cruzada**

   Estes experimentos de validação cruzada avaliam a efetividade do sistema CBR em recuperar casos passados contendo planos de resposta reusáveis para a resposta a incidentes retirados da base de casos e usados como consultas. O experimento varia diferentes limiares de recuperação de casos pois essa configuração impacta na obtenção de resultados da recomendação obtidos pelo sistema CBR.

2) **Experimento 2: Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR e clustering - Validação Cruzada**

   Estes experimentos de validação cruzada avaliam a efetividade do sistema CBR em recuperar casos passados relevantes para a resposta de incidentes tomados como consultas. Utilizando técnicas de clustering, estes casos mais similares recuperados via técnicas de CBR são organizados em grupos de casos distintos. Conforme os analistas de segurança analisam e selecionam um desses grupos, apenas os planos de resposta contidos no grupo selecionado são reusados na resolução do incidente-alvo. Desta forma, este experimento avalia se o emprego de diferentes técnicas de clustering permite ampliar a efetividade do mecanismo de recuperação e seleção de casos implementado pelo sistema CBR.

3) **Experimento 3: Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR - Novos Incidentes**

   Estes experimentos avaliam a efetividade do sistema CBR em recuperar casos passados contendo planos de resposta reusáveis para resposta a novos incidentes usados como consultas.

4) **Experimento 4: Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR e clustering - Novos Incidentes**

   Este experimento replica o processo de testes desenvolvido no experimento 2, introduzindo a construção de respostas para novos incidentes. O objetivo do experimento envolve a avaliação das recomendações de planos de resposta na

resolução destes novos incidentes, avaliando se essas recomendações oferecidas pelo sistema são mais efetivas quando casos recuperados que pertencem a grupos de casos formados pela execução de algoritmos de clustering, e então selecionados por analistas de segurança, são considerados na construção de respostas para novos incidentes.

Os experimentos 1 e 3 visam avaliar a eficácia da recomendação de planos de resposta a incidentes com base em uma abordagem de recuperação de casos típica de CBR. Utilizando apenas os recursos de sistemas CBR, cada caso presente na base de casos é removido e usado como entrada para uma consulta no experimento 1. Esse processo permite analisar o desempenho de recomendação do sistema para casos presentes na base de casos. Similar processo de avaliação é realizado no experimento 3, embora esses testes focalizem na construção de recomendações de soluções para novos incidentes coletados.

Os experimentos 2 e 4 expandem a abordagem de CBR ao incorporar o uso de algoritmos de clustering para o refinamento de resultados de consultas. Com o objetivo de avaliar os resultados deste refinamento de recomendações geradas pela integração de técnicas de CBR e clustering propostas neste trabalho, estes testes utilizam diferentes algoritmos de clustering na seleção de casos passados reusáveis na solução de incidentes correntes. Em particular, 2 algoritmos de clustering são utilizados para o refinamento de consultas e avaliação da efetividade da recomendação de planos de resposta a incidentes. Os algoritmos empregados são K-Medoids (K-Med) e Agrupamento Hierárquico (Hierarchical Clustering - HC), com 3 critérios de ligação (linkage): *single* (SL), *average* (AL) e *complete* (CL).

Os algoritmos empregados neste estudo foram escolhidos devido às suas características distintas. O algoritmo K-Med é reconhecido por sua robustez na identificação de medoides, tornando-o eficaz na presença de *outliers*. No HC, cada critério de ligação apresenta vantagens específicas. O critério *single* favorece a formação de clusters, nos quais os casos agrupados estão mais próximos entre si. O critério *average* busca uma média de similaridade, promovendo a formação de clusters mais equilibrados e representativos da totalidade dos dados. O critério *complete* favorece a inclusão de elementos mais distantes, destacando-se na identificação de padrões globais nos dados.

Os algoritmos foram configurados para obter dois grupos/clusters nos experimentos desenvolvidos nesta pesquisa. A escolha de dois clusters foi feita com base na observação de que, embora um número maior de grupos possa levar a uma precisão mais elevada, isso tende a criar clusters com poucas recomendações/casos. Essa situação não representa uma melhoria nem facilita o contexto da análise de recomendações por analistas de segurança, uma vez que pequenas variações entre casos podem resultar na criação de outros grupos que possuem poucas variações entre si. Esse contexto dificulta a seleção de um grupo dentre muitos grupos para a análise das recomendações presentes em um

grupo pelo analista. Variações na similaridade, mesmo que sutis, entre os casos podem ocasionar a criação de grupos com um número muito pequeno de casos. Portanto, manter apenas dois clusters possibilita capturar as principais variações e padrões nas recomendações.

Para computar clusters com os casos recuperados por consultas CBR, estes experimentos testam o uso de atributos dos i) problemas, das ii) soluções, ou iii) problemas/soluções, representados nestes casos como entradas para as execuções dos algoritmos de clustering. Estes atributos são selecionados nos casos recuperados e usados nas computações de similaridade executadas pelos algoritmos de clustering. A combinação do uso de diferentes atributos no agrupamento tem por objetivo analisar variações nos resultados dos experimentos. Em resumo, as seguintes configurações foram testadas:

a) problema: atributos do problema (incidente) registrado no modelo de casos. Estes atributos descrevem detalhes do contexto e detecção do incidente;

b) solução: atributo que representa a solução (plano de resposta) registrado no modelo de casos. Estes atributos representam a sequência de ações de resposta que, se conduzidas, levam à resolução do incidente;

c) problema/solução: tanto os atributos do problema quanto da solução dos casos são utilizados como entrada para as execuções dos algoritmos de clustering.

O uso de atributos dos problemas/incidentes como entrada para os algoritmos de clustering resulta em grupos contendo casos com contextos e informações de detecção similares. O uso de atributos da solução ou planos de resposta como entrada para os agrupamentos produz grupos que incluem casos de resposta a incidentes de segurança com planos de resposta contendo ações semelhantes, tal como computado pelo emprego da distância de Levenshtein. Além disso, o uso de atributos de problemas/soluções como entrada nos algoritmos de clustering busca formar grupos em que as informações dos problemas, e das soluções representadas nos casos organizados nos grupos, tenham alta similaridade entre si.

Ao aplicar algoritmos de clustering nos resultados de consultas CBR (isto é, na lista de casos mais similares recuperados pela consulta executada), é importante notar que o recall obtido em cada grupo formado não pode superar os resultados de recall inicialmente alcançados pela recuperação de casos via CBR. Isso ocorre porque o processo de clustering filtra e reorganiza os casos originalmente recuperados, agrupando-os de acordo com suas semelhanças, sem introduzir novos casos ao conjunto de resultados obtido como resposta da consulta CBR. Dessa forma, os casos inicialmente recuperados permanecem os mesmos após os algoritmos de clustering serem executados. Na prática, espera-se que o recall diminua em comparação aos resultados obtidos exclusivamente com o uso de CBR. Essa redução dos resultados de recall também ocorre com os valores de F-Score

obtidos. Neste contexto, as análises dos resultados de emprego integrado de algoritmos de CBR e clustering, experimentados neste trabalho, são direcionadas para os resultados de métricas de precisão e acurácia, embora resultados de métricas recall e F-score sejam computadas e apresentadas.

Para a avaliação das recomendações resultantes do emprego de casos recuperados que são organizados em diferentes grupos, um dos grupos obtidos é escolhido, tendo como base o valor de uma das métrica de avaliação: neste trabalho, a métrica de precisão. Portanto, a avaliação dos diferentes grupos de casos resultantes da execução dos algoritmos de clustering é realizada com base na seleção e reúso de apenas um dos grupos de casos formados. Essa análise simula a tomada de decisão desenvolvida por um analista de segurança ao examinar os casos contidos nos grupos formados após aplicar técnicas de clustering. Na prática, essa seleção seria realizada manualmente pelo analista de segurança quando este utiliza o sistema, dadas as necessidades de resposta dos incidentes correntes.

Dada a seleção de um grupo de casos, este grupo pode ser analisado de acordo com a sua precisão como o a) melhor grupo ou b) pior grupo de acordo com diferentes critérios de avaliação. Mais, ainda, o analista de segurança pode apenas realizar a seleção aleatória de um grupo, tal como testado neste trabalho. Por exemplo, a seleção do melhor grupo de casos para reúso de planos de resposta para incidentes pode estar relacionada ao grupo que permite obter o melhor resultado de precisão associado à recomendação de planos reusáveis para a resposta do incidente corrente.

Nos experimentos 2 e 4, portanto, a precisão foi escolhida como a métrica principal para a análise e seleção de um grupo de casos formado pela execução de um algoritmo de clustering. Embora outras métricas, como recall, F-Score e acurácia, tenham sido avaliadas nos experimentos, este capítulo centraliza os resultados obtidos utilizando a métrica de precisão como critério de escolha. Os resultados obtidos nos experimentos 2 e 4, ao utilizar as métricas recall, F-Score e acurácia para escolha do grupo, são apresentados no Apêndice A.

8.4 RESULTADOS DOS EXPERIMENTOS DE VALIDAÇÃO CRUZADA

8.4.1 Resposta a incidentes de segurança utilizando o reúso de planos de resposta recuperados via técnicas de CBR - Validação Cruzada

Os resultados do experimento 1 são apresentados na Figura 28 (a). A precisão do sistema atinge 84.76% quando o LR é configurado em 80%. No entanto, o recall apresenta o valor de 93.29% com um LR de 60%. O F-Score, uma métrica de equilíbrio entre precisão

e recall, atinge 70.71% com um LR de 70%. A acurácia máxima obtida é de 98% quando o LR é configurado em 75%.

Os resultados mostram que a escolha do LR depende das necessidades de solução dos problemas do analista de segurança. Se a ênfase estiver na precisão das recomendações obtidas pelo sistema, um LR em torno de 80% pode ser escolhido. Por outro lado, se a prioridade for recuperar uma gama mais ampla de casos relevantes para o problema descrito na consulta, um valor mais baixo de precisão pode ser escolhido para configurar o mecanismo de recuperação de casos do sistema CBR.

Nos experimentos 1 (Validação Cruzada) e 3 (Novos Incidentes), o número médio de casos recuperados como resultado da execução de consultas com um LR de 75% foi de 12 e 14 casos, respectivamente. Esta recuperação evidencia um equilíbrio entre o volume de casos recuperados e a qualidade dos resultados destes testes. Neste trabalho, a escolha do LR de 75% é fundamentada nesse equilíbrio, assegurando não só a qualidade dos resultados de CBR, mas também a subsequente aplicação de técnicas de clustering sobre os casos recuperados, o que requer a recuperação de um número de casos que serão organizados e analisados como clusteres distintos.

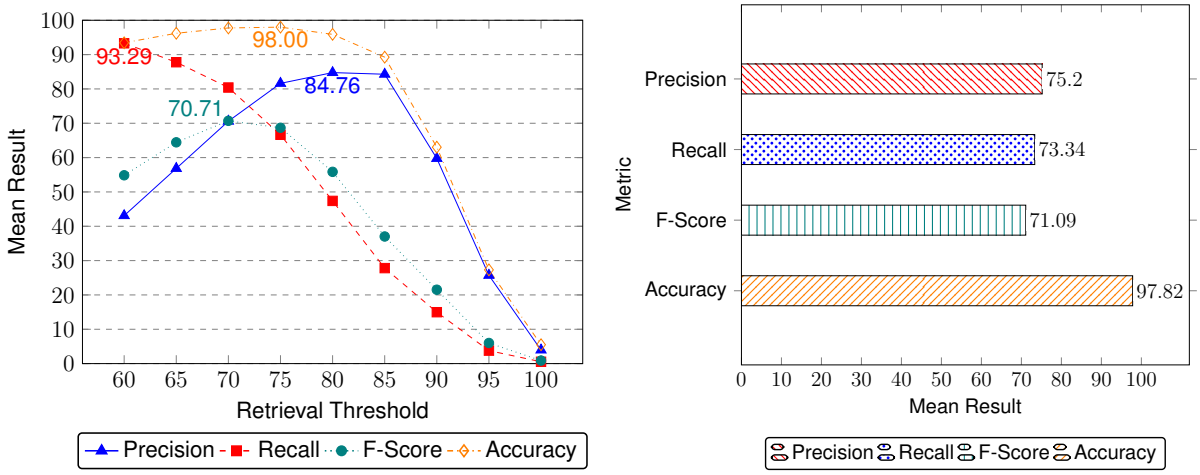
Embora o LR de 80% tenha apresentado uma precisão de 84.76% em comparação com 81.62% obtidos com o emprego de um LR de 75%, o número médio de casos recuperados para consultas executadas com o LR de 80% é inferior a 10 casos. Essa redução na quantidade de casos recuperados limita a análise desses casos via clustering, técnica geralmente usada para a análise de maiores volumes de informação. Consequentemente, com base nos resultados observados no experimento 1 (Figura 28 (a) e Tabela 8), o LR de 75% foi adotado nos experimentos 2, 3 e 4 deste trabalho, equilibrando a precisão retornada pelo sistema com a recuperação de um maior número de casos (e um maior valor de recall e F-Score), favorecendo o emprego de técnicas de clustering sobre conjuntos de casos recuperados para consultas dadas.

Tabela 8 – Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR - Validação Cruzada.

Metric	Retrieval Threshold								
	60	65	70	75	80	85	90	95	100
Precision	43.09	56.84	70.60	81.62	**84.76**	84.28	59.75	25.71	03.95
Recall	**93.29**	87.78	80.41	66.65	47.39	27.81	14.97	03.78	00.54
F-Score	54.85	64.45	**70.71**	68.69	55.85	37.02	21.52	06.00	00.88
Accuracy	93.44	96.21	97.76	**98.00**	95.90	89.25	63.03	27.30	05.50

Fonte: Autor.

Figura 28 – Experimentos 1 e 3: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR, LR de 75% (em (b)).



(a) Validação Cruzada

(b) New Incidents

Fonte: Autor.

8.4.2 Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR e clustering - Validação Cruzada - Precisão Como Métrica de Análise da Escolha de um Cluster de Casos

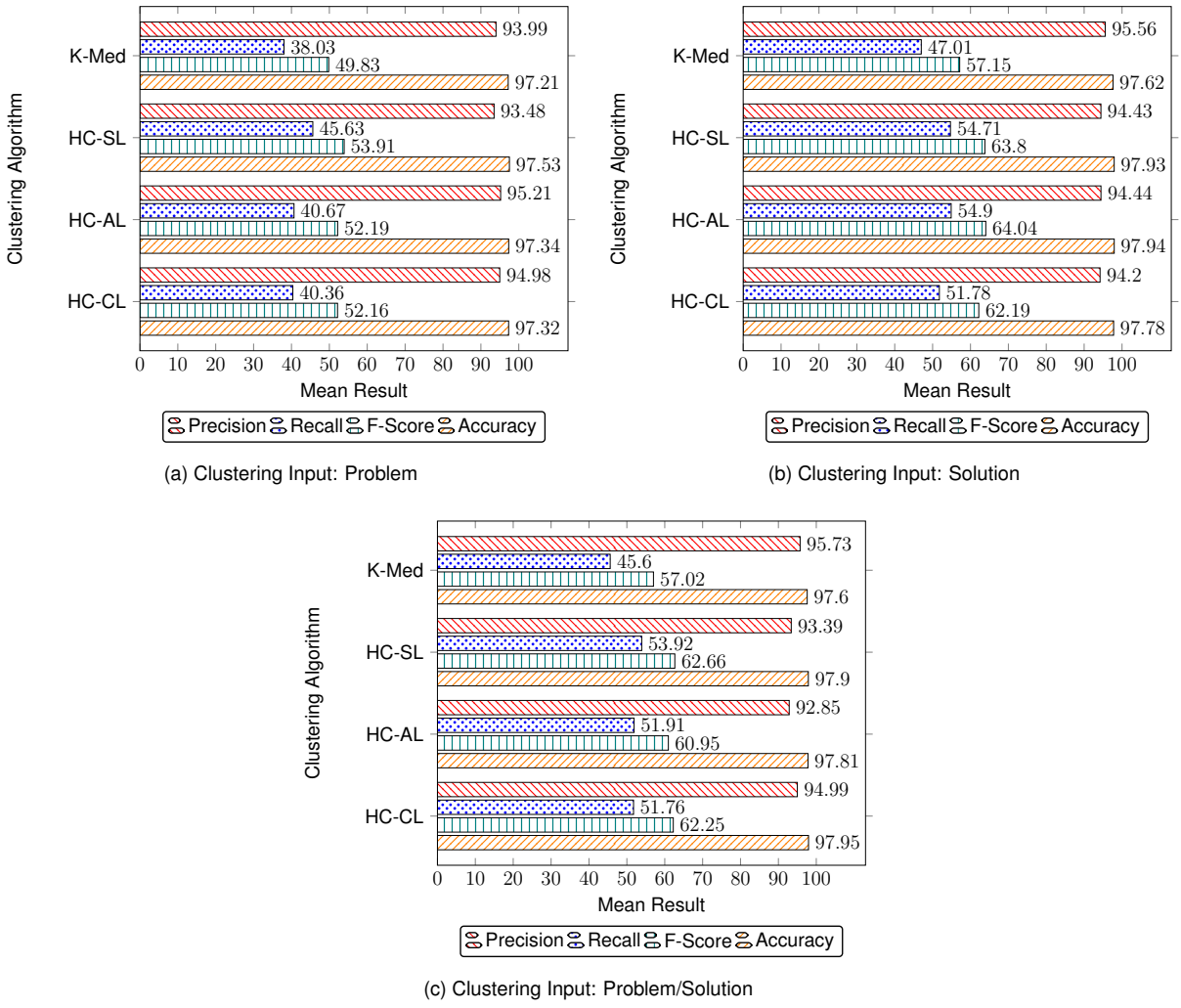
No experimento de **validação cruzada**, utilizando os algoritmos de CBR e de clustering, é possível analisar os resultados de precisão do sistema considerando que o analista de segurança efetuou a **escolha do melhor** grupo de casos formado. Esta escolha foi baseada na métrica de precisão computada para cada cluster formado, com o objetivo de determinar qual seria o grupo contendo casos com planos de resposta mais relevantes de serem reusados. Após esta seleção, portanto, ocorre o reúso de planos de resposta a incidentes registrados nos casos deste grupo escolhido para resposta ao incidente utilizado como consulta.

Neste experimento, os resultados de precisão do sistema foram muito similares entre si, conforme apresentado na Figura 29. Isso ocorreu para todos os algoritmos de clustering usados e para todos os diferentes tipos de entradas (problema, solução e problema/solução) usadas nas execuções destes algoritmos.

Nas configurações testadas nos experimentos de validação cruzada com os algoritmos de clustering (K-Med e HC com suas variações SL, AL e CL), usando entradas de atributos do problema, solução e problema/solução, os resultados de precisão do sistema superaram os alcançados apenas com o uso de técnicas de CBR. Especificamente, a precisão observada foi de 81.62% no uso de técnicas de CBR, com um LR de 75%.

Nenhum tipo de entrada (atributos do problema, solução e problema/solução) para os algoritmos de clustering consistentemente mostrou resultados de precisão superiores àqueles obtidos com as outras configurações de tipos de entradas testadas no experimento.

Figura 29 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *Precisão* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

O algoritmo K-Med, utilizando os atributos problema/solução como entrada, alcançou a mais alta precisão, 95.73%, neste experimento. Este resultado representa um aumento de 14.11% na precisão em comparação ao uso exclusivo de técnicas de CBR.

O algoritmo HC com SL, utilizando atributos da solução como entrada, alcançou uma precisão de 94.43%, mostrando ser superior em relação às outras configurações de entradas testadas para este algoritmo específico.

O algoritmo HC com AL, empregando atributos do problema como entrada, alcançou uma precisão de 95.21%, superando os resultados de outras configurações de entrada testadas nesta configuração de algoritmo.

Por fim, o algoritmo HC com CL, utilizando os atributos problema/solução como entrada, apresentou uma precisão de 94.99%, a mais elevada em comparação com os resultados de outras configurações de entrada testadas nesta configuração do algoritmo.

Em resumo, se o analista de segurança realizar a escolha do melhor grupo de casos

formado para obter planos de resposta a serem (re)usados na solução de novos incidentes, o emprego combinado de técnicas de CBR e clustering melhora significativamente a precisão do sistema. Isso é válido para qualquer configuração de algoritmos de clustering testada neste experimento de validação cruzada.

No experimento de **validação cruzada** com os algoritmos de CBR e clustering, considerando que o analista de segurança tenha realizado a **escolha do pior grupo** de casos formado, é possível analisar os resultados de precisão obtidos.

Diferente dos resultados obtidos quando um analista de segurança faz a melhor escolha de um grupo retornado pela execução dos algoritmos de clustering, usando algoritmos de clustering em todas as configurações testadas (K-Med ou HC com SL, AL, e CL), e considerando diferentes tipos de entrada (atributos do problema, solução, problema/solução), quando o analista de segurança faz a pior escolha de um grupo de casos a ser reusado na resposta do novo incidente, a precisão do sistema é inferior àquela alcançada somente com o uso de técnicas de CBR, que foi de 81.62% com um LR de 75%.

De acordo com os resultados obtidos neste outro experimento de validação cruzada, onde a pior escolha de um grupo de casos é realizada pelo analista de segurança, nenhuma técnica de clustering testada permite melhorar os resultados obtidos apenas com o uso de técnicas de CBR. Esse comportamento pode ser observado nos resultados apresentados na Figura 30.

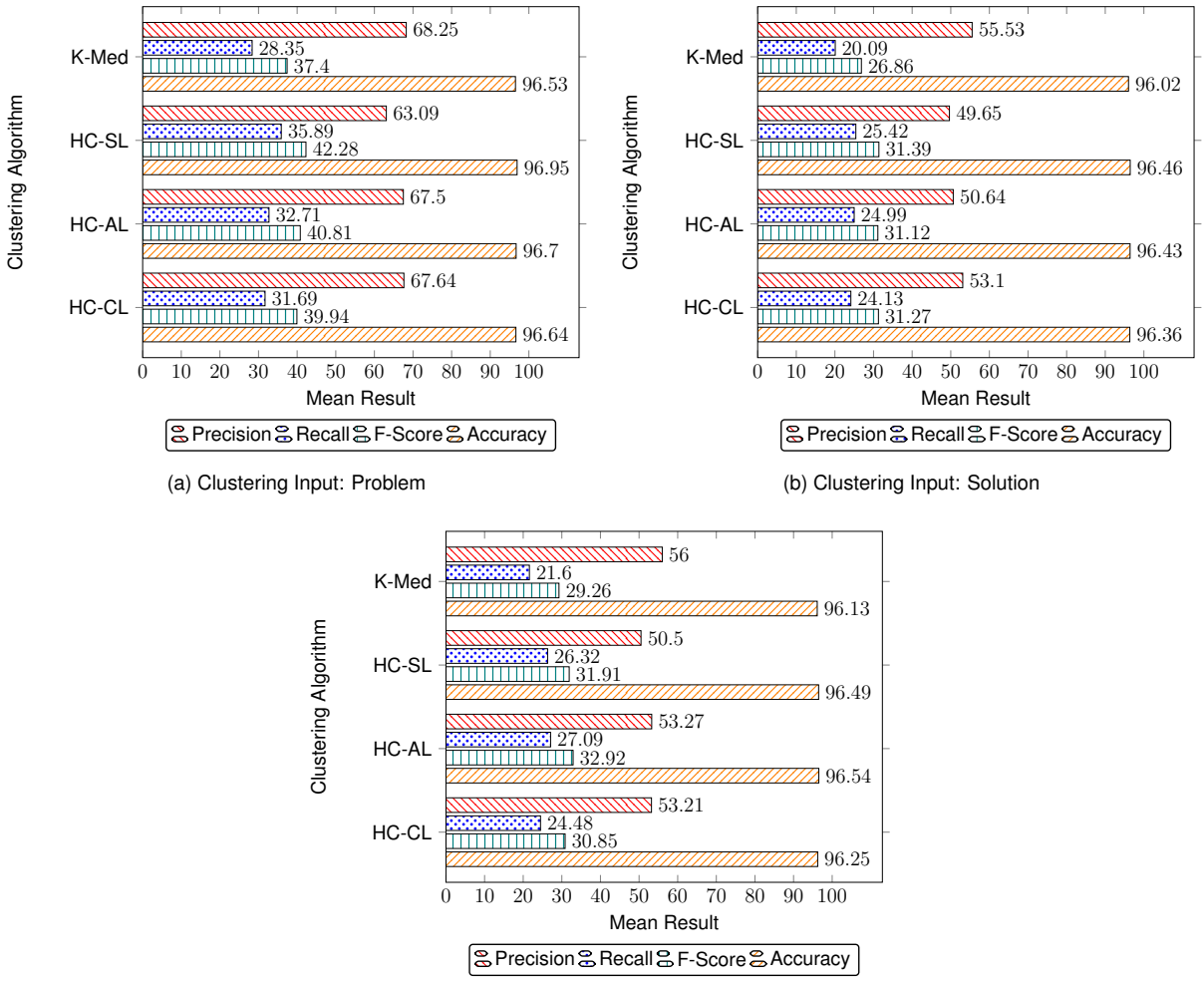
No entanto, os resultados menos piores de precisão associados à escolha do pior grupo de casos pelo analista de segurança foram obtidos quando os algoritmos de clustering usaram os atributos do problema como entrada. Em particular, o resultado menos pior obtido neste experimento de validação cruzada foi obtido pelo uso do algoritmo K-Med com atributos do problema usados como entrada nas execuções deste algoritmo. Esse resultado é 13.37% menor que o resultado de precisão obtido com o uso de técnicas de CBR somente.

Em resumo, se o analista de segurança realizar a pior escolha do grupo de casos formado para obter planos de resposta a serem (re)usados na solução de novos incidentes, o uso de técnicas de CBR e clustering permite que usuários focalizem as suas análises visando o reúso de planos de resposta nas características dos grupos de casos formados. Contudo, se estes usuários apenas reusarem os planos de resposta registrados em um grupo de casos que não seja uma boa escolha para abordar o incidente corrente, os resultados de precisão obtidos pelo sistema devem ser piores que aqueles resultados que podem ser obtidos pelo uso de técnicas de CBR somente.

No experimento de **validação cruzada** com os algoritmos de CBR e clustering, considerando que o analista de segurança tenha realizado uma **escolha aleatória** de um grupo de casos formado, os resultados de precisão obtidos são os seguintes.

Conforme os resultados apresentados na Figura 31, o uso de todos os algoritmos de clustering testados tiveram os piores resultados de precisão quando os atributos do pro-

Figura 30 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *Precisão* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

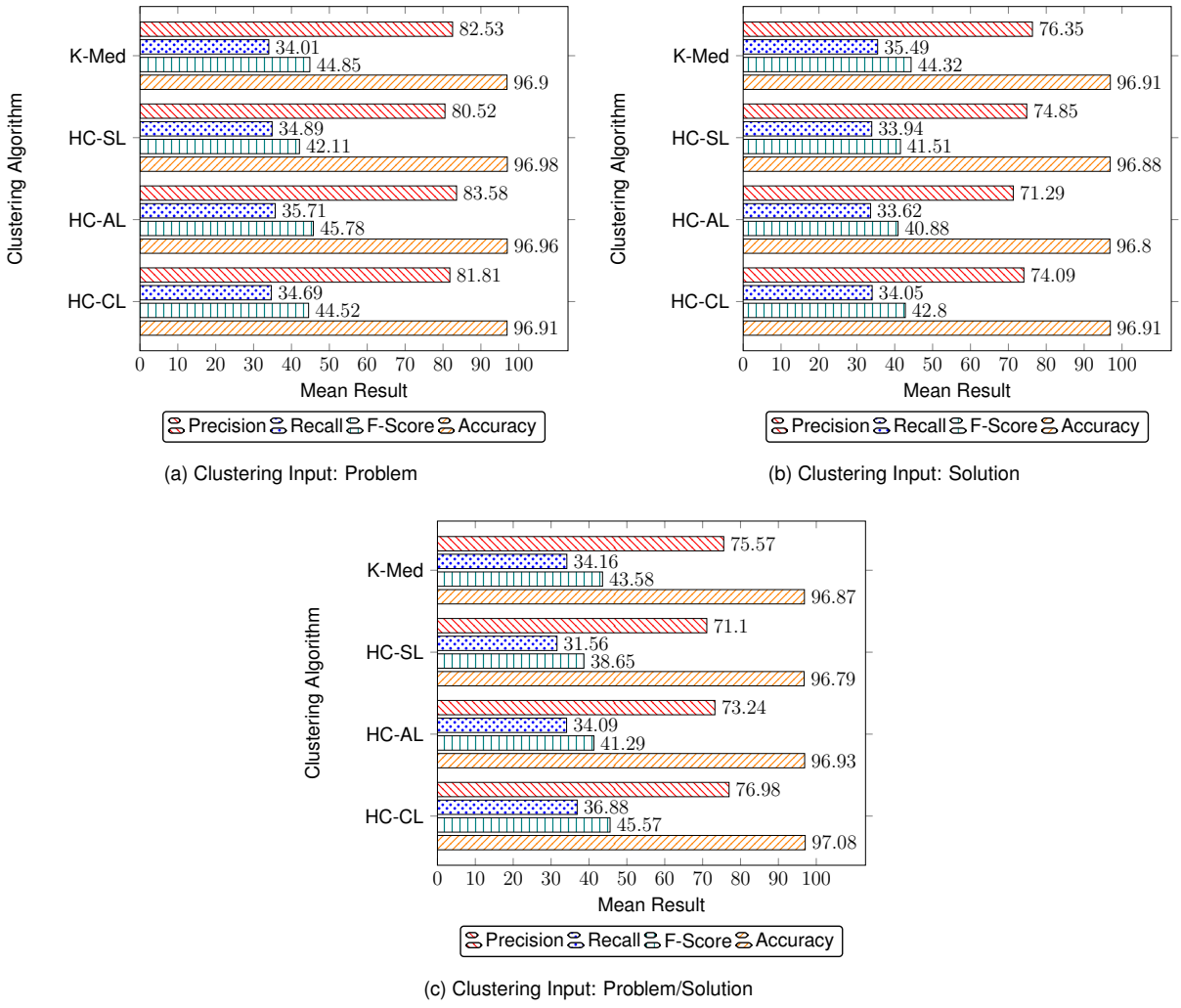
Fonte: Autor.

blema ou problema/solução foram utilizados, em comparação ao uso apenas de técnicas de CBR (81.62% com LR de 75%).

O emprego dos algoritmos de clustering testados (K-Med, HC com AL e CL), usando como entrada os atributos do problema, foram as únicas configurações de clustering testadas com a escolha aleatória de um grupo de casos que obteve resultados (levemente) melhores de precisão (respectivamente, 82.53%, 83.58%, 81.81%), maiores que aqueles obtidos pelo uso de técnicas de CBR apenas. Isso sugere que a aleatoriedade na seleção do grupo pode causar pequeno impacto na precisão, mantendo os resultados em um patamar comparável àqueles obtidos apenas com a utilização de CBR. Nesse contexto, a aplicação de técnicas de clustering não resultou em uma degradação notável dos resultados de precisão obtidos em relação ao uso de técnicas de CBR somente.

Os resultados de precisão obtidos quando os algoritmos de clustering testados usaram os atributos do problema como entrada foram maiores do que os resultados de

Figura 31 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *Precisão* da recomendação do sistema como métrica de análise da escolha **aleatória** de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

precisão obtidos por algoritmos correspondentes usando atributos de soluções ou problemas/soluções como entradas para as execuções destes algoritmos.

Se um analista de segurança fizer uma escolha aleatória de um grupo após executar um algoritmo de clustering que usa atributos do problema como entrada, o uso de técnicas de CBR e clustering, em combinação, permite que os usuários obtenham resultados de precisão levemente superiores àqueles obtidos pelo uso de técnicas de CBR somente. Os resultados obtidos não são tão inferiores quanto aqueles obtidos quando as melhores escolhas de grupos de casos formados por CBR e clustering são realizadas. Considerando os resultados obtidos neste experimento de validação cruzada, a escolha aleatória de um grupo de casos permite melhorar os resultados de CBR quando os algoritmos K-Med, HC com AL e CL, e atributos do problema são usados como entrada nas execuções desses algoritmos.

Em resumo, ao analisar os resultados deste experimento de validação cruzada,

comparando diferentes abordagens de seleção de grupos de casos retornados por consultas CBR executadas, observam-se diferentes impactos nas medidas de precisão do sistema. A escolha aleatória de um grupo resultou em valores próximos de precisão aos obtidos no experimento 1 (empregando apenas técnicas de CBR), indicando que a seleção aleatória impacta minimamente nos resultados desta métrica. Por outro lado, a seleção do pior grupo mostrou uma redução na precisão do sistema, devido à inclusão de casos não relevantes e *outliers* nos grupos a serem reusados na resposta do incidente corrente. Em contraste, a seleção do melhor grupo demonstrou uma melhora substancial no valor de precisão do sistema em relação ao uso de técnicas de CBR apenas.

Nos experimentos de validação cruzada, a variação da acurácia ao utilizar a combinação de técnicas de CBR e algoritmos de clustering foi muito sutil, não ultrapassando uma alteração de 3% em comparação aos resultados obtidos com o uso exclusivo de CBR.

8.5 RESULTADOS DOS EXPERIMENTOS COM NOVOS INCIDENTES

8.5.1 Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR - Novos Incidentes

O objetivo principal do experimento 3 é avaliar a eficácia do sistema no apoio à construção de respostas para novos incidentes de segurança (incidentes nunca vistos durante o desenvolvimento deste trabalho). O intuito é avaliar a capacidade do sistema de se adaptar a casos não vistos ou submetidos anteriormente no sistema, avaliando a recuperação de respostas reusáveis para novos problemas. Conforme apresentado na Figura 28 (b), para o LR de 75%, uma precisão de 75.20% e uma acurácia de 97.82% foram obtidos. Os resultados obtidos neste experimento (3) são inferiores aos valores de precisão e acurácia obtidos no experimento 1 de validação cruzada, sendo, respectivamente, 81.62% e 98%, conforme apresentado na Figura 28 (a). Estes resultados são, de certa forma, esperados, visto a queda nos resultados de precisão e acurácia, uma vez que este experimento testa a capacidade do sistema de generalizar o aprendizado de forma a permitir resolver problemas novos. Neste caso, os valores obtidos neste experimento onde não há o refinamento de resultados de consultas CBR com algoritmos de clustering continuam sendo relevantes e aceitáveis em comparação aos resultados obtidos no experimento 1. Em resumo, essa abordagem experimental realiza uma análise da capacidade do sistema em recomendar incidentes relevantes para reúso em novas situações ou cenários de incidentes de segurança.

8.5.2 Resposta a incidentes de segurança utilizando o reúso de planos de resposta passados recuperados via técnicas de CBR e clustering - Novos Incidentes - Precisão como Métrica de Análise da Escolha de um Cluster de Casos

O experimento 4 avalia diferentes algoritmos de agrupamento e seleções de grupos, envolvendo a seleção de casos passados para a construção de respostas para incidentes novos/inéditos. Isso difere do experimento 2, em que um caso é removido da base de casos e utilizado como consulta no sistema CBR.

No experimento envolvendo **novos incidentes**, usando ambos os algoritmos de CBR e clustering, considerando que o analista de segurança tenha realizado a **escolha do melhor grupo** de casos formado pela execução dos algoritmos de clustering testados, os resultados de precisão obtidos são os seguintes.

Os resultados de precisão do sistema foram muito próximos entre eles neste experimento. Isso ocorreu para todos os algoritmos de clustering usados e para todos os diferentes tipos de entradas (problema, solução, problema/solução) usadas nas execuções destes algoritmos. Estes resultados são apresentados na Figura 32.

Em todas as configurações testadas de algoritmos de clustering (K-Med, HC com SL, AL e CL), com entrada de atributos do problema, solução e problema/solução nas execuções destes algoritmos, os resultados de precisão do sistema foram significativamente maiores que aqueles obtidos pelo uso de técnicas de CBR somente, tal com obtido nos experimentos com novos incidentes desenvolvidos (75.20% com LR de 75%).

O algoritmo K-Med, usando atributos de problema/solução, permitiu obter os resultados mais altos de precisão (92.79%) em relação a outras configurações de entradas testadas para este algoritmo. Este foi o maior resultado de precisão obtido neste experimento com novos incidentes, o que melhora em 17,59% os resultados de precisão obtidos com o uso de técnicas de CBR somente.

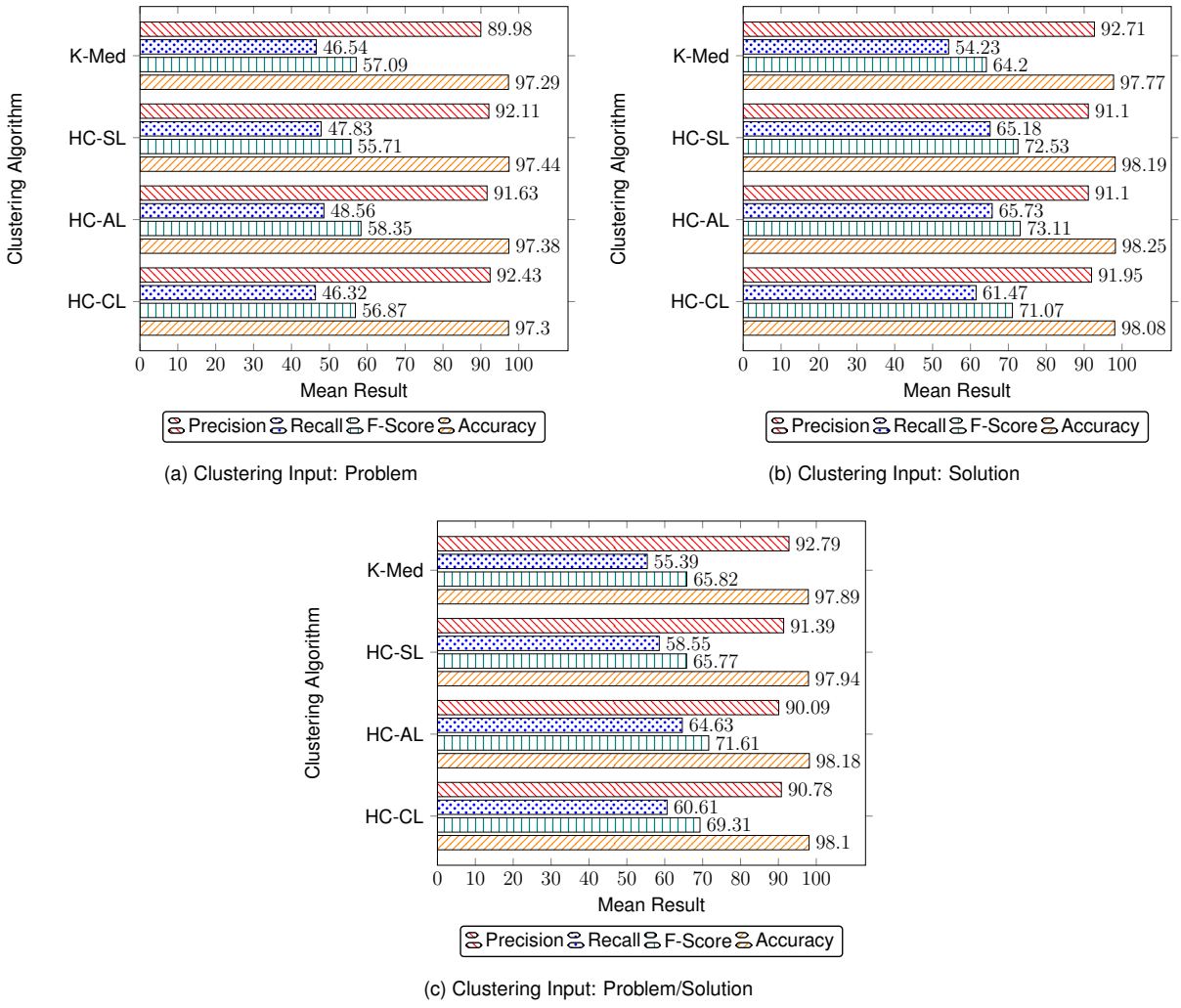
O algoritmo HC com SL, utilizando atributos do problema como entrada nas execuções deste algoritmo, alcançou uma precisão de 92.11%, mostrando ser superior em comparação aos resultados com outras configurações de entrada testadas para este algoritmo.

O algoritmo HC com AL, ao usar atributos do problema como entrada, obteve melhores resultados de precisão, alcançando 91.63%, comparativamente superiores aos obtidos com outros tipos de entrada nesta mesma configuração do algoritmo.

Por fim, o algoritmo HC com CL, quando utilizou atributos do problema como entrada, apresentou uma precisão de 92.43%, superando os resultados de precisão obtidos com outras configurações de entrada deste algoritmo.

Em síntese, quando o analista de segurança faz a melhor escolha de grupo de casos para reutilizar planos de resposta em novos incidentes, o uso combinado de técnicas de CBR e clustering em qualquer configuração testada melhora significativamente a precisão do sistema. Esse resultado supera o que é obtido com o uso exclusivo de CBR

Figura 32 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Precisão* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

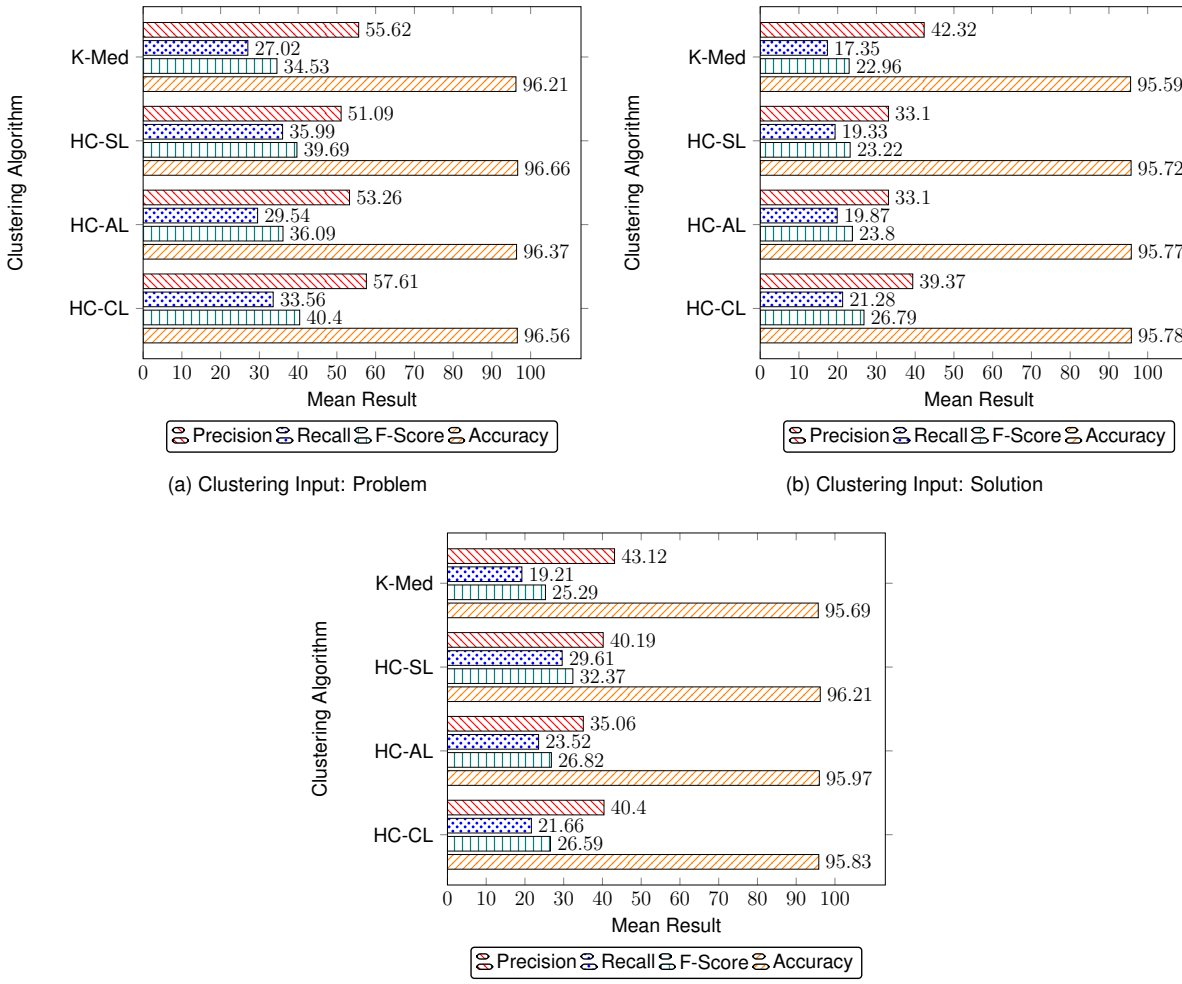
(experimento 2).

No experimento com **novos incidentes** e o emprego dos algoritmos de CBR e clustering, considerando que o analista de segurança tenha realizado a **escolha do pior grupo** de casos formado, os resultados de precisão obtidos são os seguintes.

Nas configurações testadas dos algoritmos de clustering (K-Med e HC com SL, AL e CL), utilizando diferentes tipos de entrada (atributos do problema, solução e problema/solução), a precisão do sistema ficou abaixo daquela alcançada apenas com o uso de técnicas de CBR, que foi de 75.2% com um LR de 75%. Estes resultados são apresentados na Figura 33.

De acordo com os resultados obtidos nestes experimentos, com a escolha do pior grupo sendo realizada pelo analista de segurança para responder aos novos incidentes, nenhuma das técnicas de clustering testadas permitiu melhorar os resultados obtidos com o uso de CBR somente. Ou seja, o emprego de técnicas de clustering não permite melhorar

Figura 33 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Precisão* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

os resultados de precisão do sistema quando um grupo de casos com grande número de casos irrelevantes para a resposta do problema é reusado.

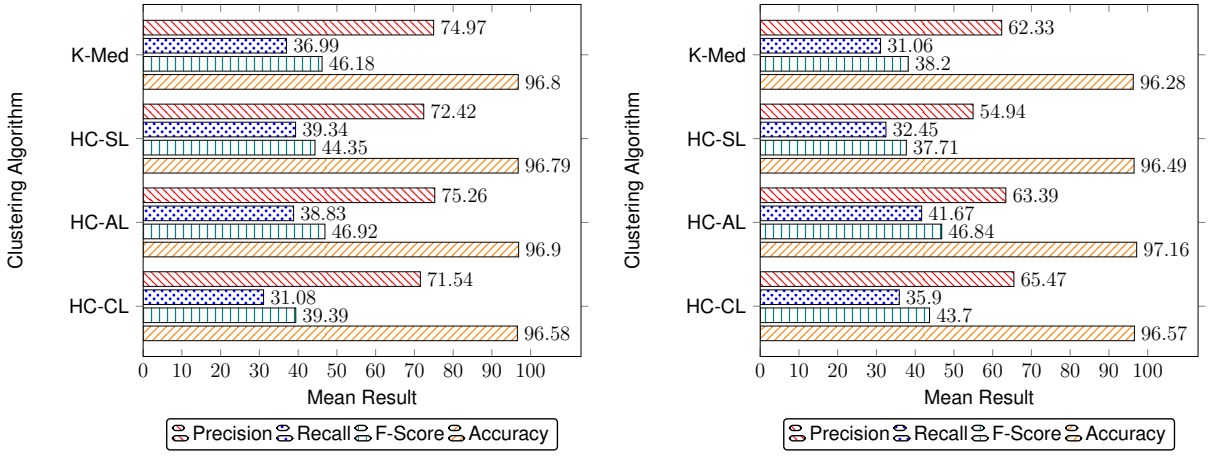
Nos experimentos de validação cruzada e na escolha do pior grupo de casos, os resultados menos insatisfatórios de precisão surgiram quando os algoritmos de clustering utilizaram atributos do problema como entrada, sendo menos desfavoráveis do que os resultados com atributos da solução ou problema/solução. Notavelmente, o uso do algoritmo HC com CL e atributos do problema obteve resultados menos piores neste contexto, diferindo da validação cruzada, onde o K-Med foi o menos insatisfatório. Esse resultado foi 17.59% inferior ao obtido exclusivamente com o uso de técnicas de CBR.

Caso os usuários optem por reutilizar planos de resposta armazenados em um grupo de casos que não corresponda adequadamente ao incidente em análise, é provável que os resultados de precisão alcançados pelo sistema sejam inferiores aos obtidos mediante a aplicação exclusiva das técnicas de CBR.

No experimento que envolveu **novos incidentes**, utilizando tanto os algoritmos de CBR quanto de clustering, e considerando a **escolha aleatória de um grupo** de casos pelo analista de segurança, é possível analisar os resultados de precisão obtidos.

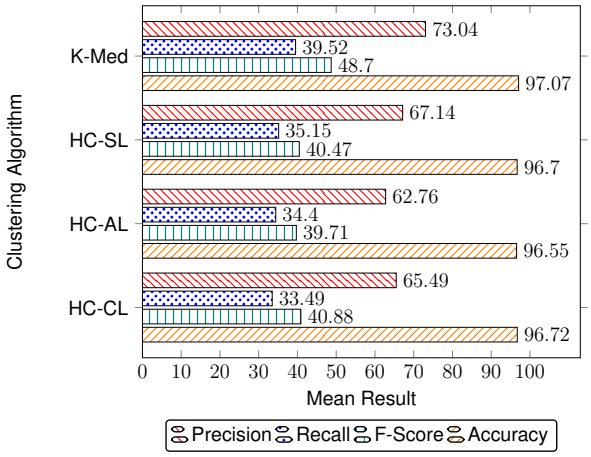
Neste experimento com novos incidentes e escolha aleatória de um grupo de casos, o algoritmo HC com AL, utilizando atributos do problema, obteve um resultado de precisão de 75.26%, o que é ligeiramente superior ao alcançado usando somente técnicas de CBR. O desempenho utilizando atributos do problema foi melhor quando comparado ao uso de atributos da solução ou problema/solução para todas as configurações de algoritmos de clustering testadas. A Figura 34 apresenta os resultados obtidos neste experimento.

Figura 34 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Precisão* da recomendação do sistema como métrica de análise de uma escolha **aleatória** de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution



(c) Clustering Input: Problem/Solution

Fonte: Autor.

Neste experimento envolvendo novos incidentes e a escolha aleatória de um grupo de casos, observou-se que certas configurações dos algoritmos de clustering testados, incluindo K-Med e HC com SL e CL, utilizando atributos do problema, assim como as configurações de todos os algoritmos de clustering com atributos da solução ou problema/solução

como entrada, resultaram em precisão inferior àquela obtida pelo uso exclusivo de técnicas de CBR.

Em resumo, se um analista de segurança realizar uma escolha aleatória de um grupo de casos formado pela execução do algoritmo de clustering HC com AL, utilizando atributos do problema como entrada, a combinação das técnicas de CBR e clustering permite obter resultados de precisão que são equivalentes àqueles alcançados pelo uso exclusivo de técnicas de CBR.

8.6 DISCUSSÃO DOS RESULTADOS DOS EXPERIMENTOS

Diversas abordagens foram exploradas e testadas para a análise da questão proposta: “Como melhorar a recuperação de planos de resposta a incidentes de segurança cibernética a serem reusados na resposta a novos incidentes?”. Neste contexto, foram explorados algoritmos de clustering K-Med e HC (com critérios de ligação SL, AL e CL) para analisar, organizar e apresentar para os analistas de segurança os grupos intrínsecos aos casos recuperados como resultados da execução de consultas CBR no sistema.

Um aspecto relevante para as análises descritas neste trabalho pode ser pontuado. A variedade e amplitude de casos relevantes recuperados pelos algoritmos testados, expressas pela métrica recall, desempenham um papel também relevante no contexto da investigação e resposta a incidentes. Considerando a amplitude de cenários contextuais e de detecção considerados na investigação de incidentes, é relevante que analistas de segurança possam recuperar e analisar situações variadas de incidentes e respostas armazenadas na base de casos do sistema, permitindo melhor compreender as complexidades de cada incidente. Entretanto, ao direcionar o foco dos experimentos desenvolvidos para a pronta resposta de incidentes (isto é, em situações de crise associadas à ocorrência de incidentes de segurança, a pronta resposta é primordial), as análises de precisão do sistema foram priorizadas neste trabalho.

A implementação e a avaliação de diferentes estratégias permitiram uma análise comparativa, destacando a interação entre algoritmos de clustering e o impacto resultante na eficácia do sistema em oferecer recomendações relevantes para resposta a novos problemas. Nas análises apresentadas, o valor do LR de 75% foi definido experimentalmente. Isso possibilitou uma análise comparativa entre as técnicas propostas neste trabalho: i) recuperação utilizando CBR somente; e ii) agrupamento de recomendações de incidentes recuperados para consultas CBR, utilizando atributos do a) problema, b) solução ou c) problema/solução.

Motivado pelo que é proposto em Sadaf e Alam (2012) e Cocea e Magoulas (2012), este trabalho também investiga o agrupamento dos resultados de consultas, possibilitando a descoberta de estruturas de grupos em listas de casos recuperados. Entre outras razões,

as consultas CBR executadas por analistas podem, em muitas situações, retornar em uma lista de casos bastante variada. Isso geralmente ocorre quando os dados do incidente atual, utilizados como entrada na formação de consultas CBR, são imprecisos. Dentre os motivos para consultas imprecisas, alguns podem ser descritos: a dificuldade inerente à análise do contexto do incidente; a falta de informações nos relatórios de incidentes; informações pouco específicas para descrever o contexto do incidente, entre outros. Assim, este trabalho demonstra que o agrupamento dos resultados das consultas é especialmente útil sempre que as consultas CBR originalmente formuladas contenham informações vagas ou ambíguas. Por meio de algoritmos de clustering, abordagens que reduzam o espaço de análise de listas de casos recuperados como resposta de consultas CBR são bem-vindas no processo de recuperação e reúso de planos de respostas a incidentes.

A análise de um grupo de recomendações, em comparação com a avaliação de todos os resultados de uma consulta CBR, oferece uma abordagem mais acessível para os analistas de segurança. Ao agrupar as recomendações com base em atributos específicos do problema, solução ou problema/solução, o analista pode melhor visualizar e compreender padrões e tendências mais claramente. Neste caso, este usuário pode focalizar a análise de planos de resolução em um subgrupo de casos recuperados pelo sistema. Essa estratégia pode simplificar o processo de tomada de decisão, permitindo que o analista concentre sua atenção nos grupos mais relevantes e significativos para construir planos de resposta para novos incidentes. Neste caso, esses analistas de segurança podem utilizar os grupos obtidos para simplificar o processo de análise dos planos de resposta recomendados pelo sistema, com base na análise de apenas um grupo de casos que pode ser selecionado, por exemplo.

O uso de consultas CBR para obter recomendações de planos de tratamento por si só apresenta resultados relevantes neste problema de aplicação, tal como descrito em Guerra et al. (2023). Considerando o uso de recomendações obtido por resultados provenientes apenas de consultas CBR, empregando um LR de 75%, sem refinamento com base em clustering, uma precisão de 81.62% e acurácia de 98% no experimento 1 (Validação Cruzada) foram obtidas. No experimento 3 (novos incidentes), os resultados de acurácia e precisão foram de 75.20% e 97.82%, respectivamente, para o emprego de técnicas de CBR somente.

No experimento com novos incidentes, uma queda de 6.42% na precisão (75.20%) pôde ser observada nos experimentos com técnicas de CBR somente, em contraste com os resultados do experimento de validação cruzada (81.62%) baseados nestas mesmas técnicas. Estes valores foram obtidos com base no LR de 75%. Esse declínio da precisão é um resultado esperado, uma vez que o sistema CBR é desafiado por cenários provenientes de novos incidentes. Isso evidencia a importância de continuamente atualizar e adaptar a base de casos do sistema CBR para melhorar sua capacidade de generalização e competência do sistema em resolver problemas novos.

Os resultados obtidos nos experimentos de validação cruzada e os resultados alcançados nos experimentos com novos incidentes podem ser analisados e comparados.

A escolha pelos usuários do sistema dos melhores grupos de casos formados permite que a combinação de CBR e clustering alcance melhores resultados de precisão que aqueles obtidos pelo uso de técnicas de CBR somente, em todas as configurações de algoritmos de clustering e atributos do modelo de casos usados como entrada nas execuções destes algoritmos. Contudo, os melhores resultados de precisão podem ser obtidos com o uso de atributos do problema e de atributos do problema/solução. Comparando as diferentes configurações de algoritmos testadas nos experimentos de validação cruzada e com novos incidentes, os algoritmos que obtiveram os melhores resultados foram o algoritmo HC com AL usando atributos do problema como entrada e o algoritmo K-Med usando atributos do problema/solução como entrada.

Quando usuários do sistema selecionam os piores grupos de casos, a combinação de CBR e clustering tende a gerar resultados de precisão inferiores aos alcançados com o uso exclusivo de técnicas de CBR, em todas as configurações dos algoritmos de clustering e uso de atributos do modelo de casos como entrada para as execuções destes algoritmos. No entanto, as configurações que utilizam atributos do problema tendem a mostrar resultados menos insatisfatórios entre as configurações de testes executados nos experimentos de validação cruzada e novos incidentes.

Um padrão observado nos experimentos indica que o uso exclusivo de atributos da solução como entrada para algoritmos de clustering resulta em desempenho inferior que outras configurações de entradas testadas. Essa tendência foi constante tanto nos experimentos de validação cruzada quanto nos experimentos empregando novos incidentes.

Quando os usuários do sistema escolhem aleatoriamente grupos de casos para focalizar a resposta de novos incidentes, a combinação de CBR e clustering pode levar a melhorias na precisão em comparação ao uso isolado de casos recuperados via técnicas de CBR. Nos experimentos de validação cruzada, melhores resultados de precisão foram obtidos com os algoritmos K-Med, HC com AL e CL, todos utilizando atributos do problema como entrada. Nos testes com novos incidentes, o algoritmo HC com AL, também usando atributos do problema, apresentou os melhores resultados. Assim, selecionar atributos do problema nas execuções de certos algoritmos de clustering pode melhorar a precisão do sistema, mesmo quando as escolhas dos grupos de casos pelos analistas de segurança são feitas de maneira aleatória.

8.7 CONSIDERAÇÕES FINAIS

A realização de experimentos focados na validação cruzada e nos novos incidentes, sem a inclusão direta de usuários no processo de validação, permite avaliar de maneira

sistemática a performance de algoritmos testados sob diferentes condições. A ausência de usuários reais no processo de validação, embora seja limitada por não capturar a interação humana e as experiências destes analistas de segurança, permite minimizar o impacto de fatores externos que podem introduzir ruídos nos resultados. No entanto, é importante reconhecer que essa abordagem possui limitações, particularmente no que diz respeito à compreensão da usabilidade e aplicabilidade prática dos algoritmos testados em cenários de respostas a incidentes. Abordagens experimentais que envolvem usuários demandam tempo de uso do sistema e disponibilidade de analistas de segurança, aspectos que não puderam ser resolvidos durante o desenvolvimento deste trabalho.

Ao longo deste capítulo, foram delineados elementos para a compreensão do contexto e metodologia adotados neste estudo. Em consonância com a questão de pesquisa que norteia este trabalho, destaca-se a seleção e aplicação de métricas de avaliação. Essas métricas foram escolhidas para medir a eficácia do sistema em oferecer recomendações reusáveis para construir respostas para problemas de segurança.

Na abordagem que envolve o uso de técnicas de CBR apenas, sem refinamento com base na execução de algoritmos de clustering, considerando as métricas de precisão e acurácia avaliadas nos experimentos 1 (Validação Cruzada) e 3 (Novos Incidentes), uma configuração equilibrada envolve usar o limiar de recuperação de 75%, uma vez que são recuperados em média entre 12 e 14 casos da base de casos, além de demonstrar uma maior acurácia associada ao uso exclusivo de técnicas de CBR (experimento 1).

Avaliando a abordagem que emprega o uso de clustering para refinamento de consultas, a configuração do sistema que permite obter uma maior precisão é o uso de atributos do problema ou problema/solução como critério de entrada para os algoritmos de clustering. De forma geral, o algoritmo K-Med apresenta melhores resultados nos experimentos 2 e 4 ao selecionar o melhor grupo de acordo com valores de precisão computados.

A abordagem de refinamento de consultas CBR, incorporando o uso de algoritmos de clustering, é uma estratégia relevante explorada nos experimentos desenvolvidos, tanto na validação cruzada quanto no uso de novos incidentes. Um aspecto relevante dessa investigação envolveu a experimentação com diferentes conjuntos de atributos utilizados como critérios de entrada para os algoritmos de agrupamento. Além disso, algoritmos de clustering distintos foram testados para avaliar os resultados de refinamento de consultas CBR.

Os resultados de precisão, associados a respostas obtidas para consultas CBR executadas, podem ser aprimorados através do refinamento que incorpora o uso de algoritmos de clustering e considerando a seleção de um grupo para análise e reúso de planos de resposta a incidentes. Essa abordagem é especialmente relevante quando o usuário seleciona para a análise os casos pertencentes ao melhor grupo resultante do agrupamento. Este método de seleção pode potencializar a relevância das recomendações apresentadas para o tratamento de novos incidentes. Por outro lado, a escolha de

um grupo de forma aleatória por analistas de segurança demonstra que, mesmo nesse cenário menos ideal, os resultados de precisão e acurácia permanecem consistentes com aqueles obtidos sem o refinamento baseado em computações de clusters. Isso sugere que uma seleção aleatória entre os grupos resultantes do agrupamento não compromete significativamente o processo de análise, mantendo a precisão e acurácia em patamares comparáveis à abordagem sem uso do agrupamento.

Neste contexto, é importante notar que a precisão do sistema pode sofrer uma redução caso o usuário selecione o pior grupo formado no processo de agrupamento. Isso enfatiza a importância de que analistas de segurança realizem uma análise preliminar dos grupos formados e das recomendações/incidentes presentes nos grupos apresentados. Apesar dessas variações na precisão conforme o tipo de grupo selecionado (melhor, pior ou aleatório) por esses analistas de segurança, a acurácia do sistema avaliada nos experimentos desenvolvidos demonstra uma estabilidade notável, mantendo-se em torno de 97% tanto em cenários com o refinamento de consultas CBR, por meio de algoritmos de clustering, quanto apenas empregando técnicas de CBR.

9 CONCLUSÕES

A cibersegurança tem ganho amplo destaque científico e empresarial devido à crescente evolução e sofisticação dos ataques. Para acompanhar esse cenário, ferramentas que empregam IA têm proporcionado um aprimoramento significativo dos procedimentos que auxiliam na manutenção de um ambiente cibernético protegido em organizações. A resposta a incidentes de segurança cibernética envolve vários destes procedimentos que estão relacionados à erradicação de ataques e/ou à mitigação de incidentes, tal como citado por Kaur, Gabrijelčič e Klobučar (2023): análises dos ambientes afetados, vulnerabilidades exploradas e o planejamento de procedimentos que devem ser executados para mitigar incidentes.

Este trabalho focaliza a resposta a incidentes de segurança cibernética com base na integração de técnicas distintas da IA, especificamente CBR e clustering, que são apoiadas por tarefas de aquisição e representação de conhecimento sustentadas pelo emprego de ontologias de aplicação. A investigação conduzida neste trabalho visa desenvolver e avaliar métodos para apoiar o desenvolvimento de melhores estratégias de resposta a incidentes dentro de organizações. O problema de pesquisa abordado questiona como a combinação dessas técnicas computacionais pode apoiar o reúso de planos de resposta a incidentes, padronizando e organizando a resposta a esses incidentes. Com base nestes objetivos, o trabalho procura não apenas investigar a aplicabilidade e integração de CBR e clustering, mas também apresentar métodos que facilitem a seleção e reúso de planos de resposta para tratar novos incidentes.

A falta de padronização nas ações e planos de tratamento para incidentes de segurança cibernética constitui um obstáculo significativo para a aplicação eficiente de técnicas computacionais. Este cenário dificulta o processo de identificação e seleção de respostas aplicáveis e reusáveis a novos incidentes e ameaças. Entre outros motivos, a diversidade e constante falta de padronização de descrições de planos de resposta dificultam uma análise sistemática de experiências passadas de resposta a incidentes. Tal diversidade e inconsistência nas informações relativas ao tratamento de incidentes restringem a capacidade das organizações em reusar procedimentos provenientes de casos passados como resposta a novos incidentes. Assim, este trabalho aborda soluções que tratam essas deficiências, incentivando uma padronização mais ampla nas representações e computações realizadas em experiências de resposta a incidentes cibernéticos, o que pode melhor apoiar a análise e reúso de planos de resposta para estes incidentes.

Um aspecto relevante investigado neste trabalho está associado à seleção e ao reúso de procedimentos na resposta a novos incidentes de segurança cibernética. O funcionamento de sistemas CBR apresenta resultados relevantes que apoiam o desenvolvimento destas tarefas, especialmente quando incidentes com contextos similares estão

presentes na base de casos e são recuperados como resultado de consultas executadas nestes sistemas. Isso facilita que analistas de segurança reusem planos de resposta, visto que é mais simples analisar um conjunto pequeno e homogêneo de experiências de resposta a incidentes recuperados. Contudo, essa situação pode não ocorrer na prática, já que fatores como a falta de informações sobre o contexto e a detecção de incidentes são problemas comuns que dificultam a formação de consultas CBR precisas para os novos problemas que precisam ser resolvidos. Neste caso, existem cenários de resposta a incidentes onde o sistema CBR pode recuperar um conjunto não homogêneo (ou mesmo grande) de casos da base de casos, dificultando o processo de análise e reúso conduzido pelo analista de segurança. Como investigado neste trabalho, a integração de técnicas de CBR e clustering surge como uma estratégia promissora para apoiar analistas de segurança nas tarefas de seleção e reúso de procedimentos de resposta a incidentes.

Este trabalho apresenta contribuições relevantes para a área de pesquisa e desenvolvimento de sistemas voltados para a segurança cibernética, destacando-se pela proposta de um método que integra CBR e clustering. Mais ainda, a modelagem e emprego de uma nova ontologia de aplicação para apoiar a aquisição e representação de planos de resposta a incidentes constitui uma abordagem relevante para esta área por si só. A ontologia desenvolvida também permite o desenvolvimento e a implementação de melhores computações de similaridade requeridas para a análise/comparação de planos de resposta para incidentes por analistas de segurança. Além disso, o emprego de algoritmos de clustering para organizar planos de resposta recuperados por consultas CBR representa um avanço para essa área de pesquisa, pois permite que analistas de segurança selecionem as soluções mais reusáveis para a resposta de novos incidentes. Essas técnicas usadas em conjunto contribuem para o processo de captura de procedimentos de resposta a incidentes, além de ampliar as possibilidades de reúso de planos com base na identificação de diferentes grupos de casos de solução de problemas. Evidências dessas contribuições já podem ser identificadas em uma publicação científica (GUERRA et al., 2023) resultante do trabalho de pesquisa realizado.

Assim como desenvolvido nesta pesquisa, os experimentos de validação cruzada (Experimento 1) e com novos incidentes (Experimento 3), baseados unicamente na análise de consultas CBR sem o auxílio de algoritmos de clustering, por si só apresentaram resultados de precisão e acurácia relevantes para a resolução do problema de aplicação investigado. Os experimentos que incorporaram o refinamento das consultas CBR através do uso de clustering, tanto nos testes de validação cruzada (Experimento 2) quanto com novos incidentes (Experimento 4), os quais foram configurados com um limiar de recuperação de 75% e 2 grupos de casos, permitiram obter resultados ainda melhores que aqueles obtidos com o uso de técnicas de CBR somente. Especificamente, ao focar as análises em escolhas de melhores grupos formados, a precisão do sistema aumentou cerca de 10% em relação aos experimentos sem este tipo de refinamento de resultados de consul-

tas, enquanto a acurácia do sistema manteve valores altos e estáveis, independente da escolha do tipo de grupo usado nesta análise. Quando um grupo formado é escolhido aleatoriamente, a precisão e acurácia se mantêm comparáveis aos experimentos sem o emprego deste refinamento de consultas executado pelo emprego de algoritmos de clustering. Entretanto, a seleção de piores grupos de casos por analistas de segurança geralmente resulta em uma redução da precisão do sistema, reflexo da aglomeração dos casos menos similares ou discrepantes (possivelmente *outliers*) entre si nestes grupos selecionados para análise e reúso de planos de resposta a incidentes.

Em geral, os resultados experimentais obtidos neste trabalho indicam que o refinamento de recomendações geradas por consultas CBR pode aumentar a precisão do sistema, desde que os usuários sejam capazes de analisar os dois grupos formados e selecionar o melhor grupo, conforme as necessidades de solução de problemas do incidente corrente. Quando um grupo aleatório é escolhido por esses analistas, os resultados são comparáveis aos obtidos sem o uso deste refinamento baseado em clustering, mostrando a robustez do método original baseado em técnicas de CBR. Contudo, uma queda na precisão é notada quando o analista escolhe reusar os planos de resposta de um grupo pior, formado pela execução dos algoritmos de clustering. Isso ressalta como o refinamento com base em algoritmos de clustering pode otimizar a análise e o reúso de recomendações na resposta a incidentes, embora a escolha de grupos de casos formados, realizada pelo analista, tenha um papel significativo nesta análise.

Os resultados experimentais da abordagem proposta nesta pesquisa são promissores e indicam que a proposta pode ser estendida a outros contextos. Contudo, devidos ajustes na estrutura dos casos, na ontologia para resposta a incidentes de segurança cibernética, nos atributos empregados no agrupamento ou ainda nos algoritmos usados, podem ser necessárias, para refletir um novo contexto, por exemplo, para resposta a incidentes de uma outra organização. Essa é uma característica esperada a medida que se busca obter uma consolidação de experiências de respostas a incidentes cada vez mais especializadas e precisas na base de casos. Esse distanciamento de uma representação muito simplificada ou geral do contexto (ambiente) do incidente e das ações tomadas como resposta a este, representa um trade-off entre uma abordagem que mantém uma estrutura geral, aplicável a diferentes organizações sem muitas adaptações, em contrapartida ao uso de estratégias e informações cada vez mais específicas, que exigem um maior trabalho de adaptação, mas que na prática apresentará um conhecimento específico e concreto, melhor auxiliando o analista de segurança na resposta a incidentes.

Ao considerar oportunidades para trabalhos futuros, a investigação de diferentes valores de relevância, neste caso representados por pesos, para as variáveis usadas nas medidas de similaridade computadas, tanto pelas técnicas de CBR quanto pelas técnicas de clustering, pode revelar formas ainda mais eficazes de implementação das propostas apresentadas nesta dissertação. Além disso, o uso de diferentes algoritmos de clustering e

combinações de critérios de agrupamento podem ser ainda mais explorados. Experimentos com foco em incidentes caracterizados pela escassez de informações também podem ser conduzidos em trabalhos futuros. A introdução de análises de recomendações apresentadas para uma ou mais consultas CBR, com base em múltiplos grupos resultantes dos algoritmos de clustering, também se apresenta como uma proposta relevante a ser explorada. Essas são oportunidades para o avanço das ferramentas de IA no contexto da segurança cibernética, mirando uma integração mais eficiente e confiável dessas tecnologias requeridas para apoiar os processos de solução de problemas de profissionais de segurança cibernética.

REFERÊNCIAS

AAMODT, A.; PLAZA, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. **AI communications**, v. 7, n. 1, p. 39–59, 1994.

ALAVIZADEH, H. et al. A survey on threat situation awareness systems: Framework, techniques, and insights. **arXiv preprint arXiv:2110.15747**, p. 25, 2021.

ANSON, S. **Applied incident response**. Indianapolis: John Wiley & Sons, 2020. 464 p.

APPLEBAUM, A. et al. Playbook oriented cyber response. In: IEEE. **2018 National Cyber Summit (NCS)**. Huntsville, 2018. p. 8–15.

AWS Samples. **AWS Incident Response Playbooks**. 2023. GitHub repository. Acesso em: 12 fev. 2023. Disponível em: <https://github.com/aws-samples/aws-incident-response-playbooks/tree/0d9a1c0f7ad68fb2c1b2d86be8914f2069492e21/runbooks>.

BARCELOS, F. A. **Um processo de suporte e tomada de decisão no tratamento de incidentes de segurança**. 2020. 88 p. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Santa Maria, Santa Maria, 2020.

BARNUM, S. Standardizing cyber threat intelligence information with the structured threat information expression (stix). **Mitre Corporation**, v. 11, p. 1–22, 2012.

ÇAKMAKÇI, S. D. et al. A framework for intelligent ddos attack detection and response using siem and ontology. In: IEEE. **2021 IEEE International Conference on Communications Workshops (ICC Workshops)**. Montreal, QC, Canada, 2021. p. 1–6.

CALERO, C.; RUIZ, F.; PIATTINI, M. **Ontologies for software engineering and software technology**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. 340 p.

CARIAS, J. F. et al. Systematic approach to cyber resilience operationalization in smes. **IEEE access**, IEEE, v. 8, p. 174200–174221, 2020.

CHEN, J. et al. The application of cluster analysis method in case-based reasoning system. In: **26th International Conference on Geoinformatics**. Kunming: IEEE, 2018. p. 1–4.

CHEN, L. et al. Multi-objective reinforcement learning approach for trip recommendation. **Expert Systems with Applications**, Elsevier, v. 226, p. 120145, 2023.

COCEA, M.; MAGOULAS, G. D. User behaviour-driven group formation through case-based reasoning and clustering. **Expert systems with applications**, v. 39, n. 10, p. 8756–8768, 2012.

CORREA, C. et al. Intelligent decision support for cybersecurity incident response teams: autonomic architecture and mitigation search. In: SPRINGER. **International Conference on Risks and Security of Internet and Systems**. Ames, 2022. p. 91–107.

Counteractive Security Inc. **Incident Response Plan Template Playbooks**. 2023. GitHub repository. Acesso em: 08 jan. 2023. Disponível em: <https://github.com/counteractive/incident-response-plan-template/tree/master/playbooks>.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics**. Minneapolis, Minnesota, United States: Association for Computational Linguistics, 2019. v. 1: Long and Short Papers, p. 4171–4186.

EHRLINGER, L.; WÖSS, W. Towards a definition of knowledge graphs. **SEMANTiCS (Posters, Demos, SuCCESS)**, v. 48, n. 1-4, p. 2, 2016.

EVERITT, B. S. et al. **Cluster Analysis**. 5. ed. Chichester: John Wiley & Sons, Ltd, 2011. 71-110 p.

GRIGORESCU, O. et al. Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques. **Algorithms**, MDPI, v. 15, n. 9, p. 314, 2022.

GUARINO, N. Formal ontology, conceptual analysis and knowledge representation. **International Journal of Human-Computer Studies**, v. 43, n. 5, p. 625–640, 1995.

GUERRA, P. A. C. et al. An artificial intelligence framework for the representation and reuse of cybersecurity incident resolution knowledge. In: **Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing**. La Paz, Bolivia: Association for Computing Machinery, 2023. (LADC '23), p. 136–145.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. **Journal of intelligent information systems**, Springer, v. 17, n. 2, p. 107–145, 2001.

HUSÁK, M.; ČERMÁK, M. Sok: Applications and challenges of using recommender systems in cybersecurity incident handling and response. In: **Proceedings of the 17th International Conference on Availability, Reliability and Security**. Vienna, Austria: ACM, 2022. p. 1–10.

JAKUS, G. et al. **Concepts, ontologies, and knowledge representation**. New York: Springer New York, 2013. 67 p.

KAUR, R.; GABRIJELČIČ, D.; KLOBUČAR, T. Artificial intelligence for cybersecurity: Literature review and future research directions. **Information Fusion**, Elsevier, p. 101804, 2023.

LAHCEN, R. A. M. et al. Review and insight on the behavioral aspects of cybersecurity. **Cybersecurity**, Springer, v. 3, p. 1–18, 2020.

LIU, H. et al. Joint representation learning for multi-modal transportation recommendation. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Hawaii, United States: AAAI, 2019. v. 33, n. 01, p. 1036–1043.

LIVITCKAIA, K. et al. "optimal": an ontology for patient adherence modeling in physical activity domain. **BMC medical informatics and decision making**, Springer, v. 19, n. 1, p. 92, 1–15, 2019.

MANSOUL, A.; ATMANI, B. Clustering to enhance case-based reasoning. In: **Modelling and Implementation of Complex Systems: Proceedings of the 4th International Symposium (MISC 2016)**. Constantine, Algeria: Springer International Publishing, 2016. p. 137–151.

MANTARAS, R. L. D. et al. Retrieval, reuse, revision and retention in case-based reasoning. **The Knowledge Engineering Review**, v. 20, n. 3, p. 215–240, 2005.

MARTÍNEZ, R. **Incident response with threat intelligence: Practical insights into developing an incident response capability through intelligence-based threat hunting**. Birmingham: Packt Publishing, 2022. 446 p.

Meir Wahnon. **Awesome Incident Response Playbooks**. 2023. GitHub repository. Acesso em: 06 jan. 2023. Disponível em: <https://github.com/meirwah/awesome-incident-response#playbooks>.

Mozilla. **Web Security Guidelines**. 2023. Online. Acesso em: 11 jan. 2023. Disponível em: <https://infosec.mozilla.org/guidelines/web_security>.

MÜLLER, G.; BERGMANN, R. A cluster-based approach to improve similarity-based retrieval for process-oriented case-based reasoning. In: **Proceedings of the Twenty-first European Conference on Artificial Intelligence (ECAI 2014)**. Prague, Czech Republic: IOS Press, 2014. v. 263, p. 639–644.

MUNDIE, D. A. et al. An incident management ontology. In: **9th International Conference on Semantic Technologies for Intelligence, Defense, and Security - STIDS 2014**. Fairfax, Virginia, United States: CEUR-WS.org, 2014. p. 62–71.

NUNES, R. C. et al. A case-based reasoning approach for the cybersecurity incident recording and resolution. **International Journal of Software Engineering and Knowledge Engineering**, v. 29, n. 11n12, p. 1607–1627, 2019.

ONTAÑÓN, S. An overview of distance and similarity functions for structured data. **Artificial Intelligence Review**, Springer, v. 53, n. 7, p. 5309–5351, 2020.

ONWUBIKO, C. Cocoa: An ontology for cybersecurity operations centre analysis process. In: **2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)**. Glasgow: IEEE, 2018. p. 1–8.

OpenAI. **ChatGPT: Optimizing Language Models for Dialogue**. 2023. Software available from OpenAI. Acesso em: 10 out. 2023. Disponível em: <https://openai.com/chatgpt>.

PREUVENEERS, D.; JOOSEN, W. An ontology-based cybersecurity framework for ai-enabled systems and applications. **Future Internet**, MDPI, v. 16, n. 3, p. 69, 2024.

RAINS, T. **Cybersecurity threats, malware trends, and strategies: Discover risk mitigation strategies for modern threats in your organization**. 2. ed. Birmingham: Packt Publishing, 2023. 584 p.

RICCI, F.; ROKACH, L.; SHAPIRA, B. (Ed.). **Recommender Systems Handbook, Third Edition**. 3. ed. New York, NY, United States: Springer US, 2022. 1060 p.

RICHTER, M. M.; WEBER, R. O. **Case-Based Reasoning: A Textbook**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 17-40 p.

ROKACH, L.; MAIMON, O. **Data mining and knowledge discovery handbook**. New York: Springer US, 2005. 321-352 p.

SADAF, K.; ALAM, M. Web search result clustering-a review. **International Journal of Computer Science and Engineering Survey**, v. 3, n. 4, p. 85, 2012.

SÁNCHEZ-ZAS, C. et al. Ontology-based approach to real-time risk management and cyber-situational awareness. **Future Generation Computer Systems**, Elsevier, v. 141, p. 462–472, 2023.

SCHOENBORN, J. M.; ALTHOFF, K.-D. A multi-agent case-based reasoning intrusion detection system prototype. In: SPRINGER. **International Conference on Case-Based Reasoning (ICCBR 2023)**. Aberdeen, Scotland, United Kingdom, 2023. p. 359–374.

SCHREIBER, A. T. et al. Book. **Knowledge engineering and management: the CommonKADS methodology**. London, England: MIT press, 2000. 472 p.

SEGOVIA-AGUAS, J.; JIMÉNEZ, S.; JONSSON, A. Generalized planning with positive and negative examples. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. New York, United States: AAAI, 2020. v. 34, n. 06, p. 9949–9956.

STAAB, S.; STUDER, R. **Handbook on ontologies**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 811 p.

STROM, B. E. et al. Mitre att&ck: Design and philosophy. In: **Technical report**. McLean, Virginia, United States: The MITRE Corporation, 2020. p. 46. Publicado originalmente em julho de 2018. Revisado em março de 2020.

SUN, N. et al. Data-driven cybersecurity incident prediction: A survey. **IEEE communications surveys & tutorials**, IEEE, v. 21, n. 2, p. 1744–1772, 2018.

SYED, R. Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. **Information & Management**, v. 57, n. 6, p. 103334, 2020.

TAKAHASHI, T.; KADOBAYASHI, Y. Reference ontology for cybersecurity operational information. **The Computer Journal**, v. 58, n. 10, p. 2297–2312, 2015.

TAKAHASHI, T.; LANDFIELD, K.; KADOBAYASHI, Y. **An Incident Object Description Exchange Format (IODEF) Extension for Structured Cybersecurity Information**. RFC Editor, 2014. RFC 7203. (Request for Comments, 7203). Acesso em 09 fev. 2023. Disponível em: <https://www.rfc-editor.org/info/rfc7203>.

WALTERMIRE, D.; SCARFONE, K. **Guide to using vulnerability naming schemes**. Gaithersburg, 2011. Special Publication (NIST SP) - 800-51 Rev 1, 13 p. Acesso em: 09 fev. 2023. Disponível em: <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=907934>.

WUNSCH, D.; XU, R. **Clustering**. New Jersey: John Wiley & Sons, 2009. 368 p.

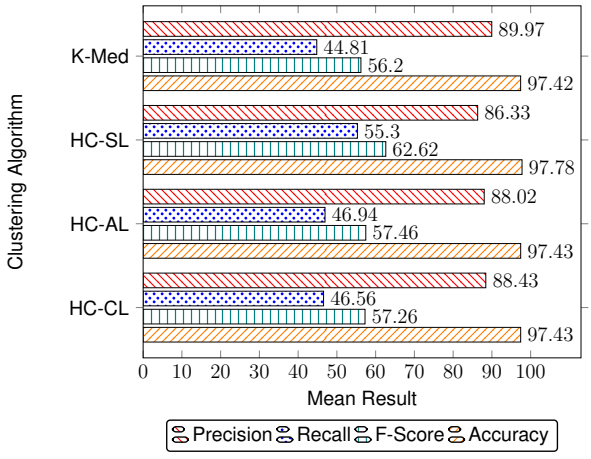
ZHANG, H.; YANG, J. A case retrieval strategy for traffic congestion based on cluster analysis. **Mathematical Problems in Engineering**, Hindawi, v. 2022, p. 5234230, 2022.

ZHANG, Z. et al. Artificial intelligence in cyber security: research advances, challenges, and opportunities. **Artificial Intelligence Review**, Springer, v. 55, n. 2, p. 1029–1053, 2022.

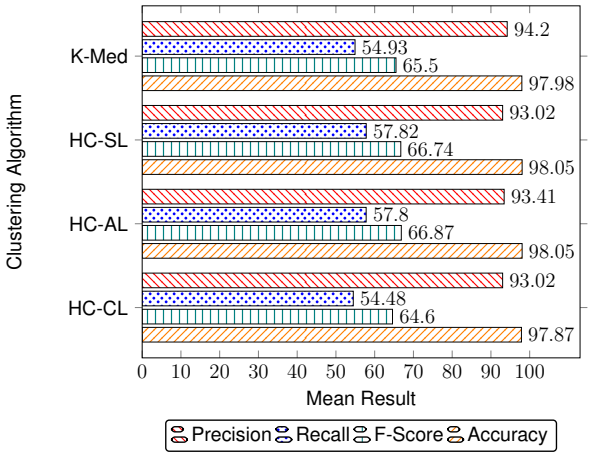
ZHU, G.-N. et al. An integrated feature selection and cluster analysis techniques for case-based reasoning. **Engineering Applications of Artificial Intelligence**, v. 39, p. 14–22, 2015.

APÊNDICE A – GRÁFICOS COMPLEMENTARES

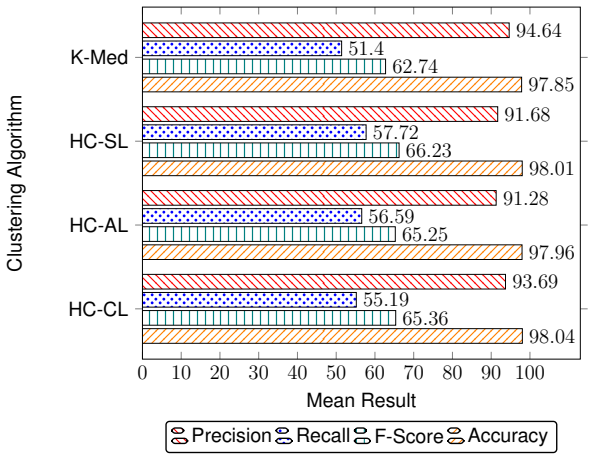
Figura 35 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *Recall* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



(a) Clustering Input: Problem



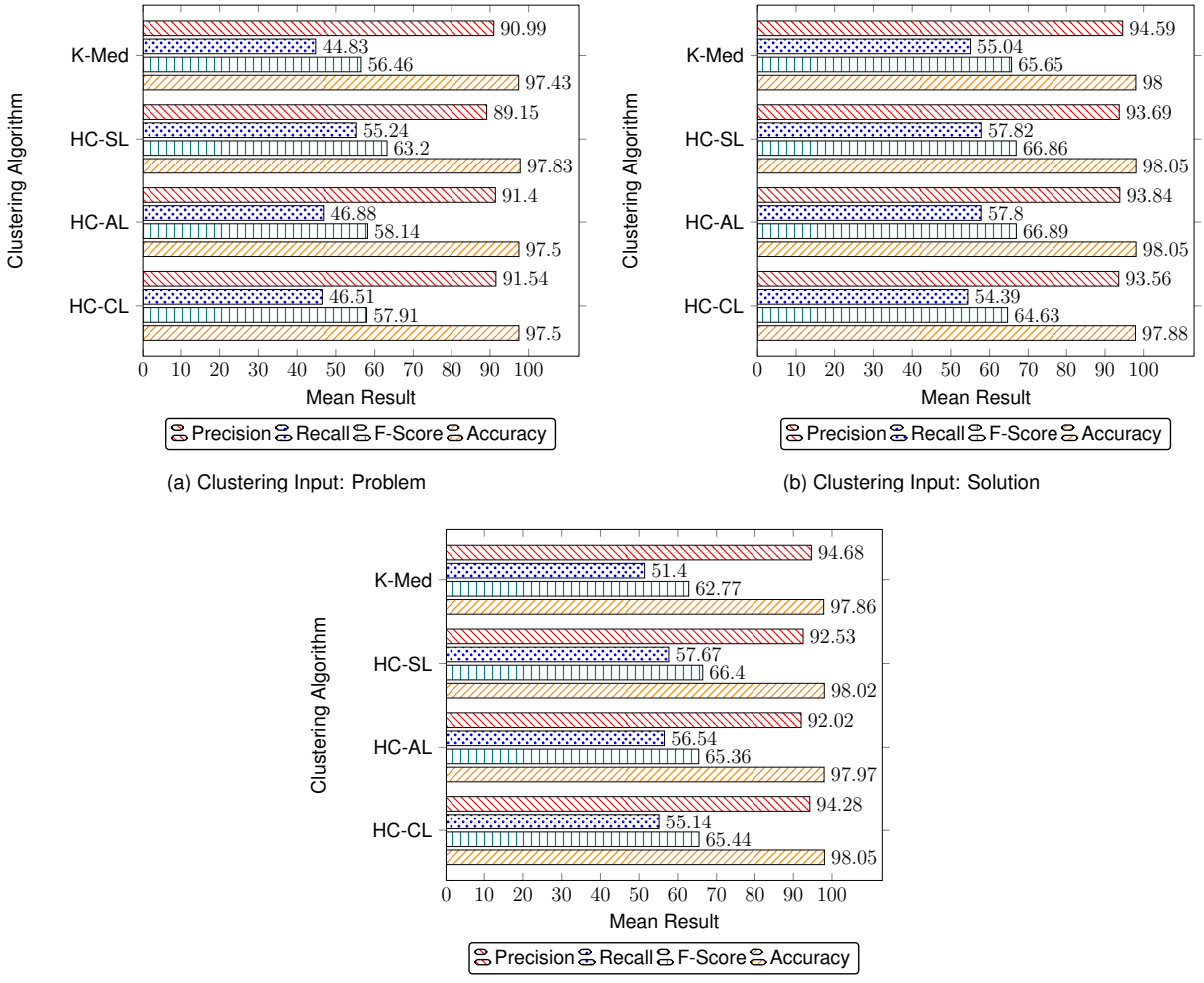
(b) Clustering Input: Solution



(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 36 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *F-Score* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



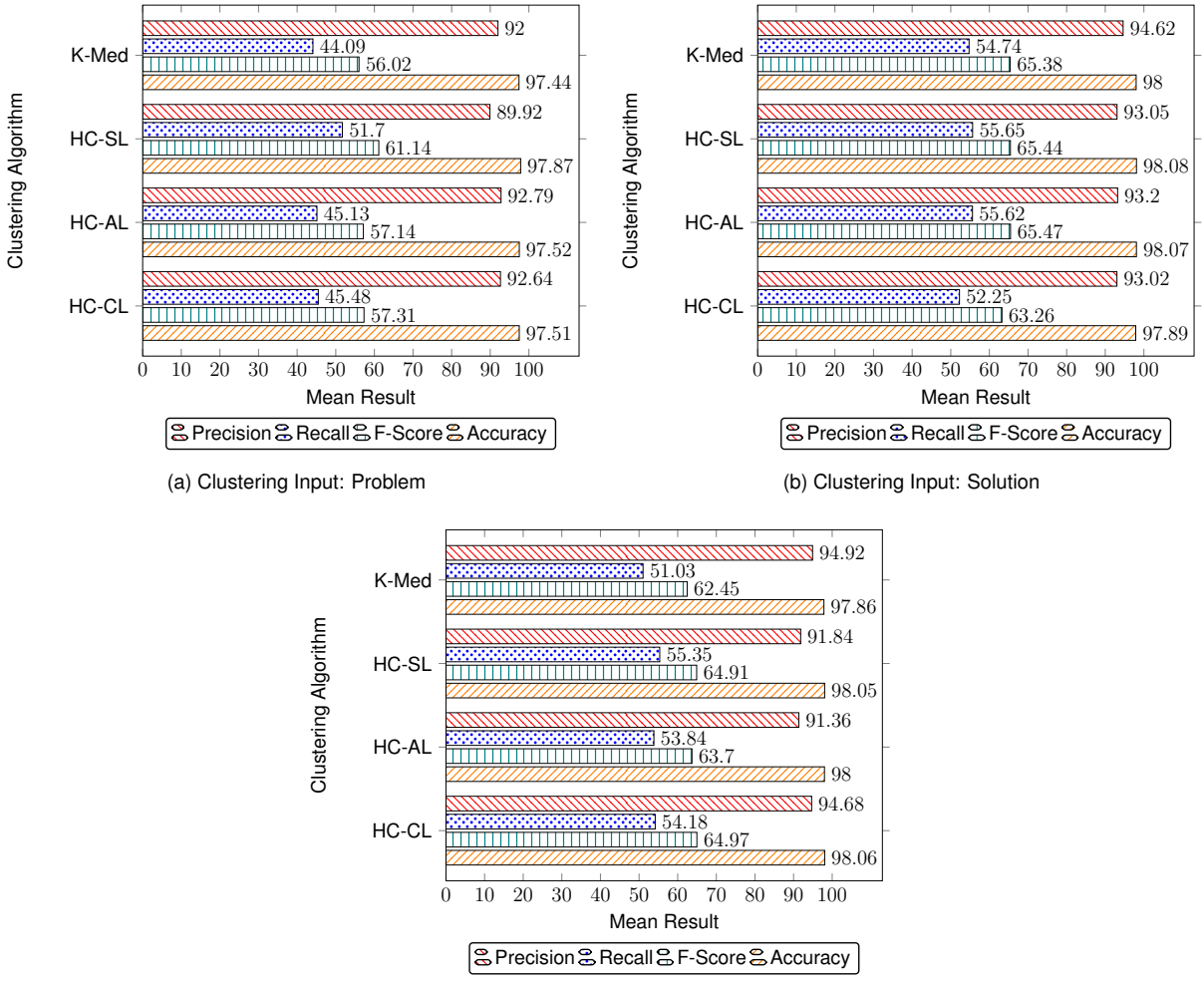
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 37 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Validação Cruzada - *Acurácia* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



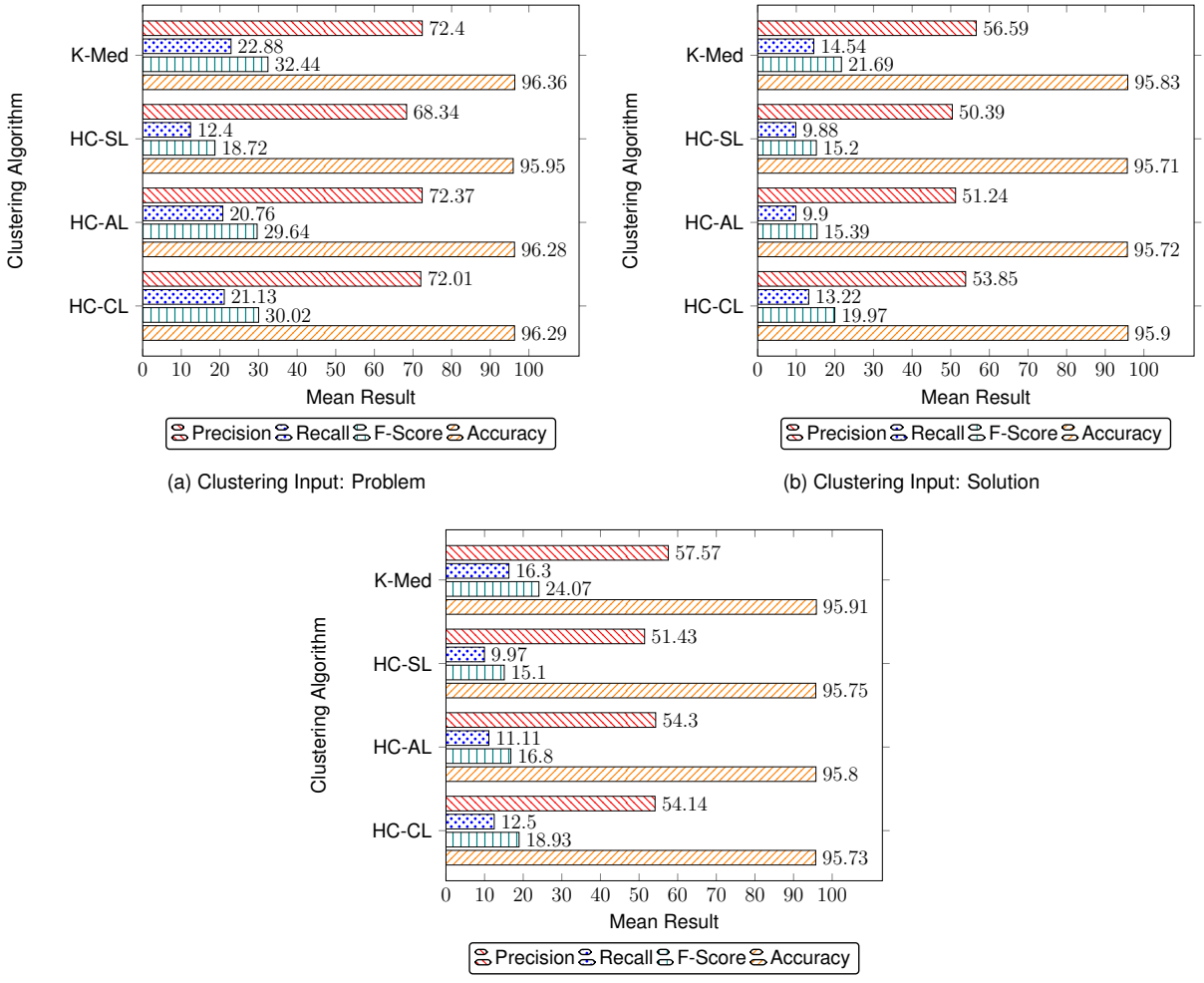
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 38 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Recall* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



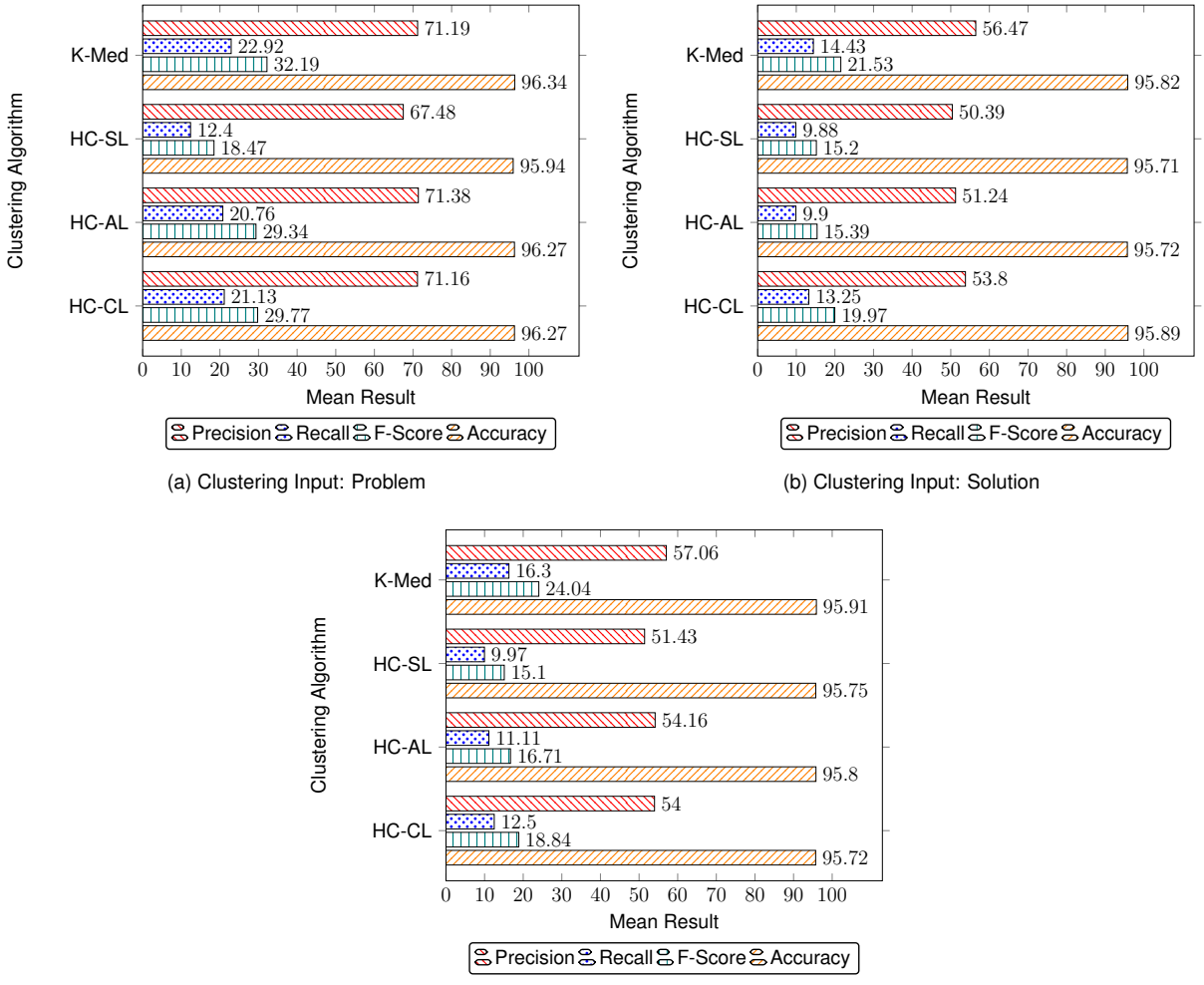
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 39 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *F-Score* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



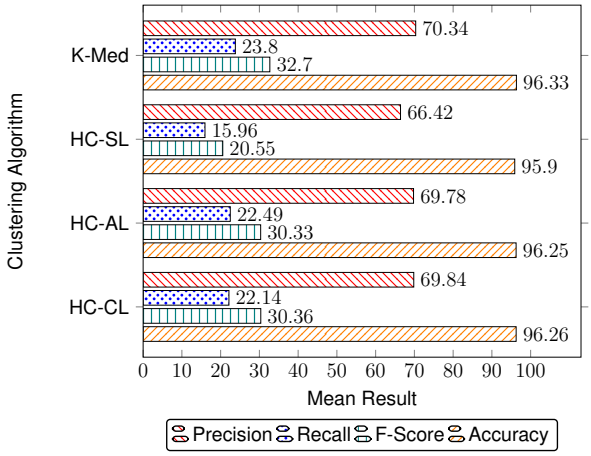
(a) Clustering Input: Problem

(b) Clustering Input: Solution

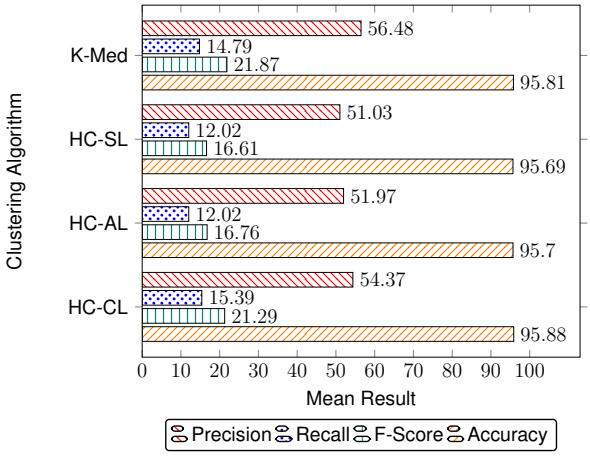
(c) Clustering Input: Problem/Solution

Fonte: Autor.

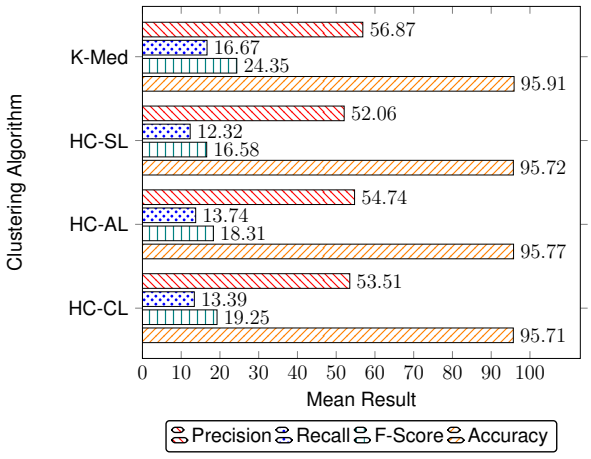
Figura 40 – Experimento 2: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Acurácia* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



(a) Clustering Input: Problem



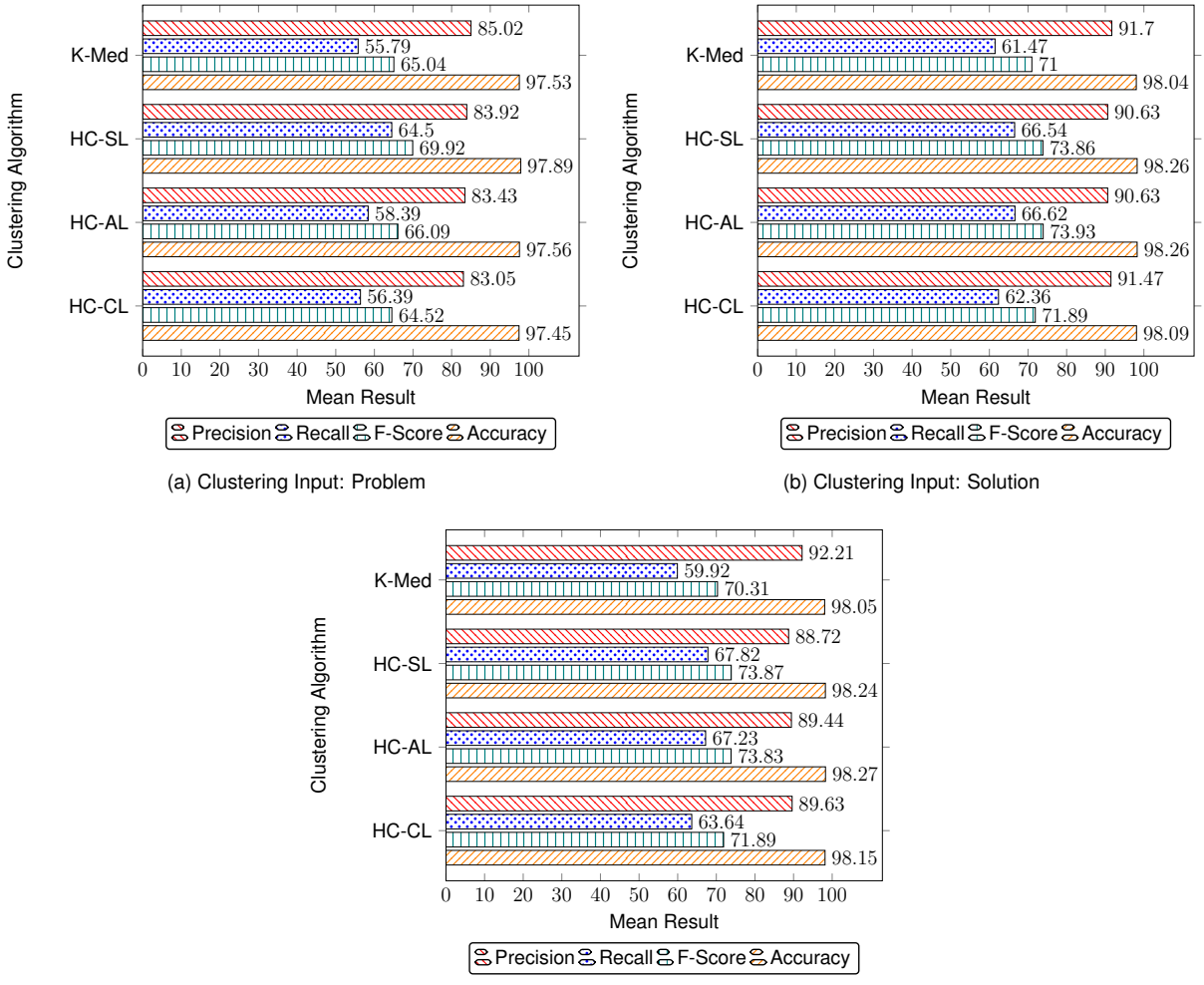
(b) Clustering Input: Solution



(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 41 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Recall* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



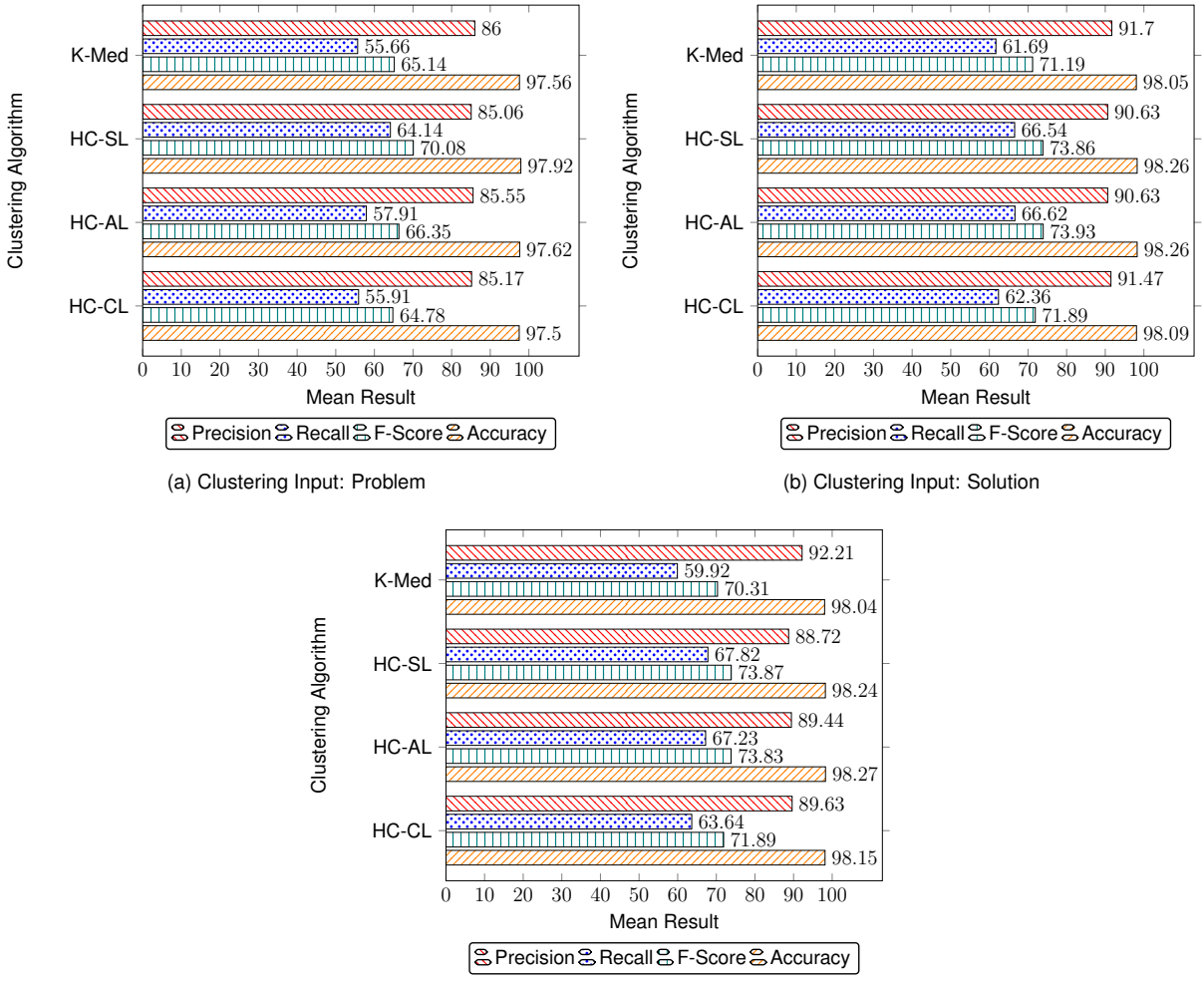
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 42 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *F-Score* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



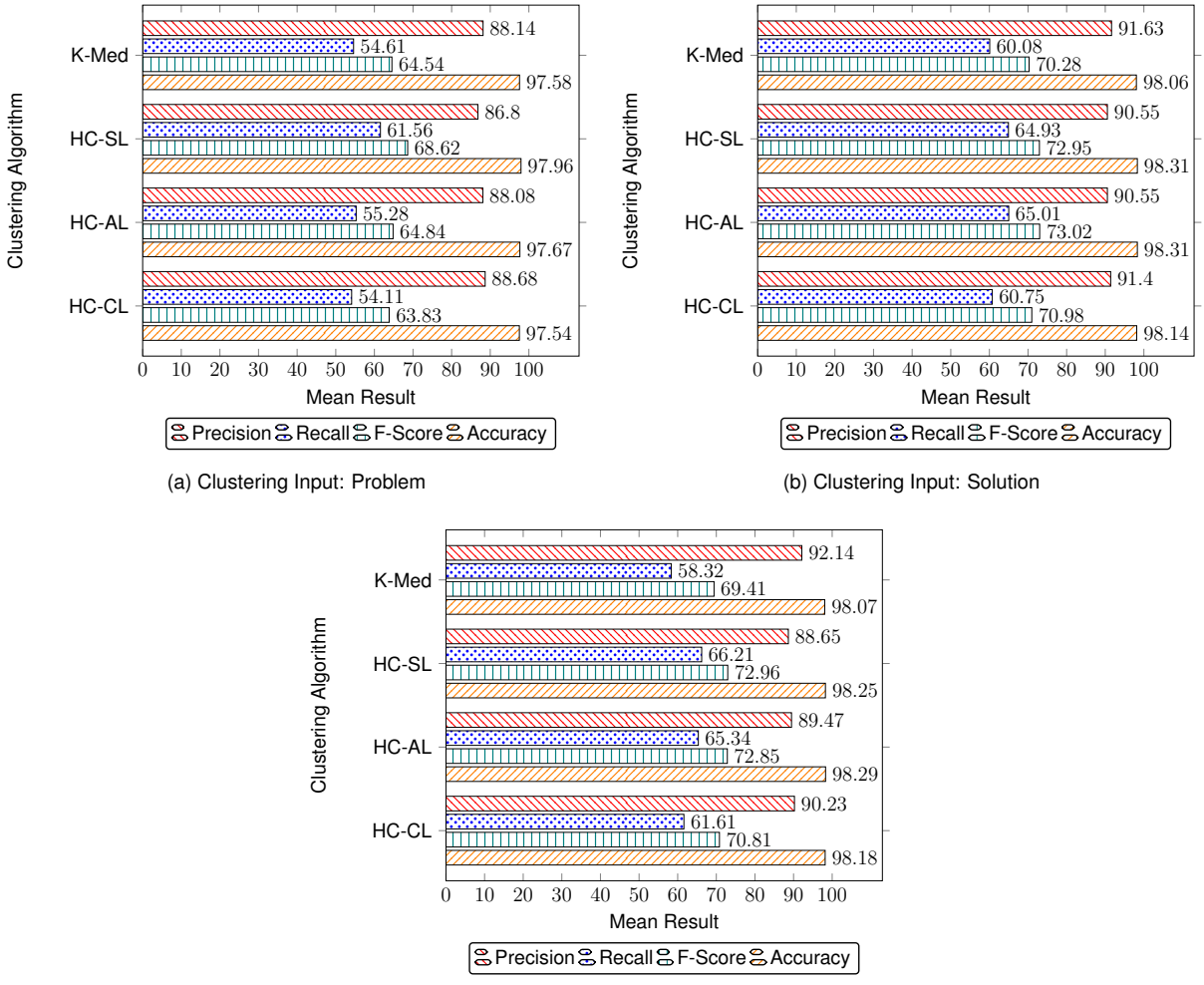
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 43 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Acurácia* da recomendação do sistema como métrica de análise da **melhor** escolha de um cluster de casos.



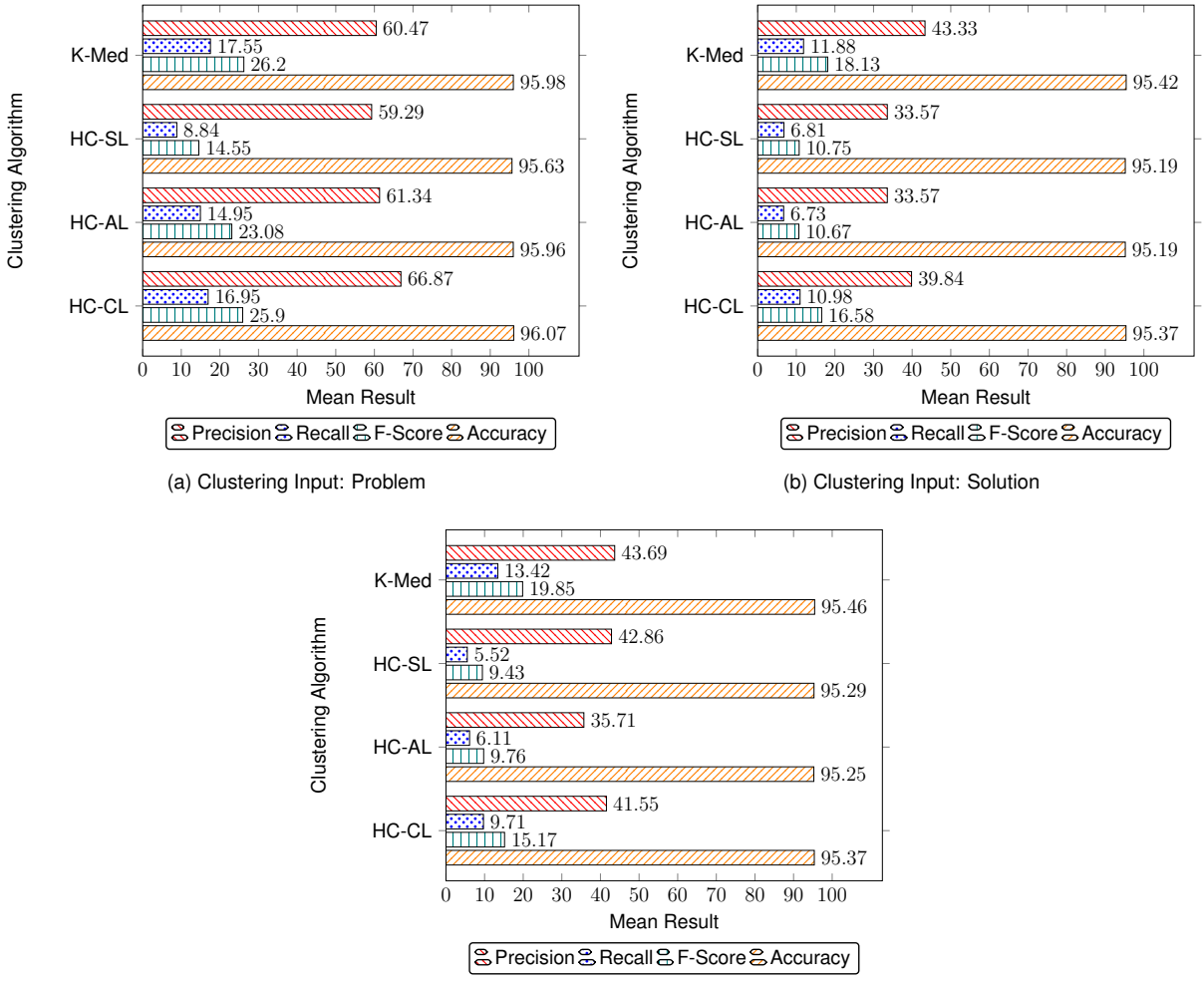
(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 44 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Recall* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



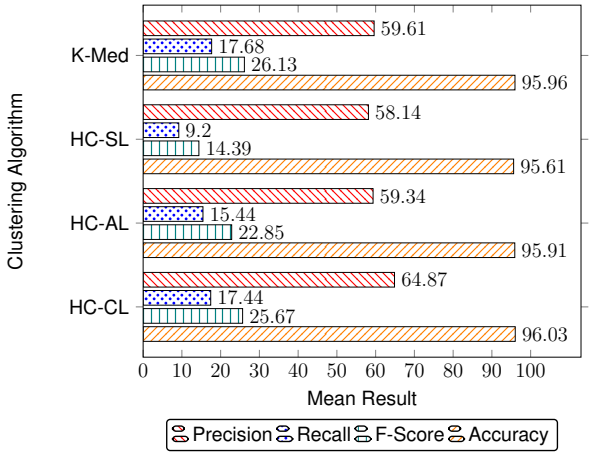
(a) Clustering Input: Problem

(b) Clustering Input: Solution

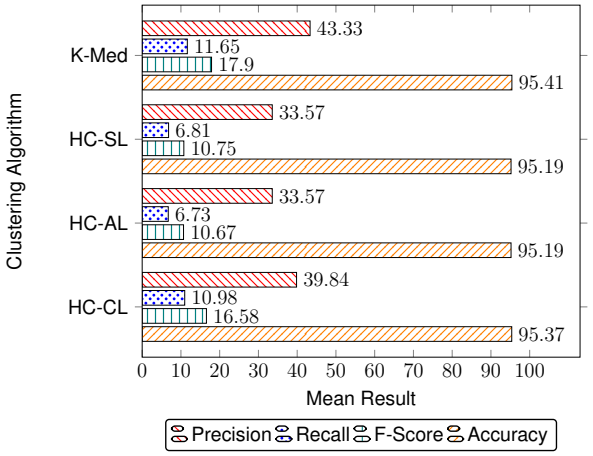
(c) Clustering Input: Problem/Solution

Fonte: Autor.

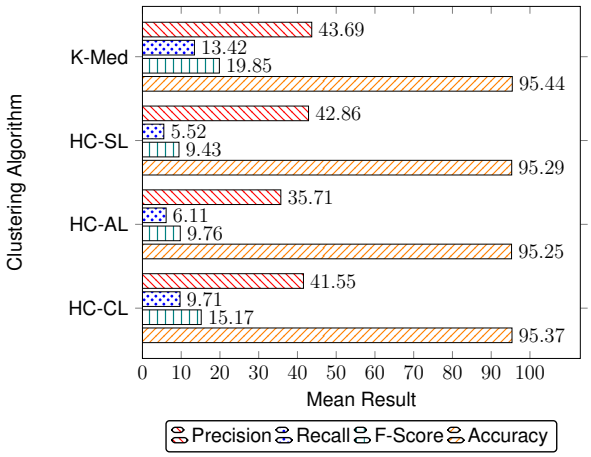
Figura 45 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *F-Score* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



(a) Clustering Input: Problem



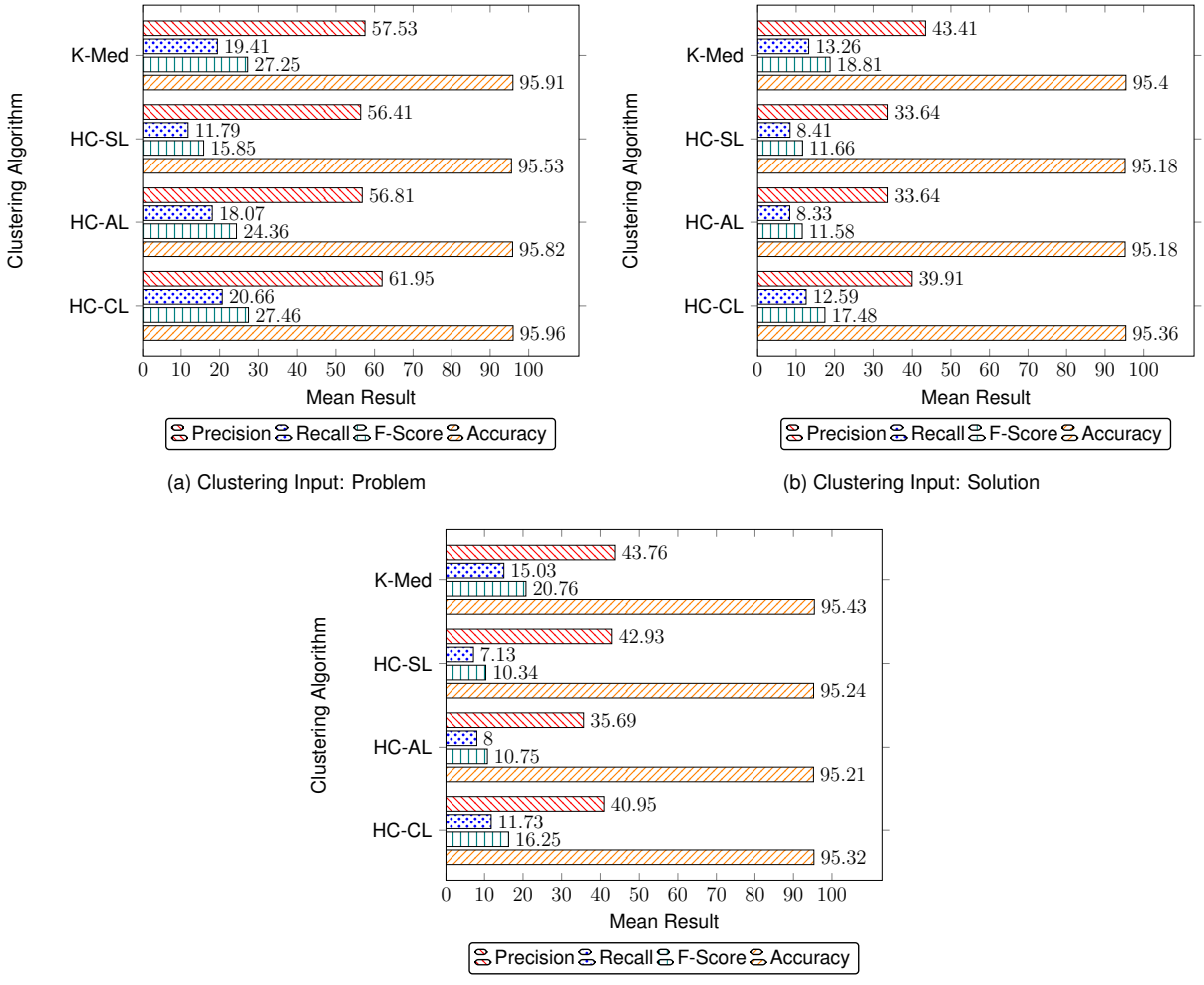
(b) Clustering Input: Solution



(c) Clustering Input: Problem/Solution

Fonte: Autor.

Figura 46 – Experimento 4: Resultados de recomendação de planos de resposta a incidentes utilizando técnicas de CBR e clustering, LR de 75% - Novos Incidentes - *Acurácia* da recomendação do sistema como métrica de análise da **pior** escolha de um cluster de casos.



(a) Clustering Input: Problem

(b) Clustering Input: Solution

(c) Clustering Input: Problem/Solution

Fonte: Autor.

APÊNDICE B – MAPEAMENTO DE ATRIBUTOS E PROPRIEDADES DE INCIDENTES POR CATEGORIA CONFORME OS FORMATOS IODEF E STIX

Tabela 9 – Mapeamento de atributos e propriedades da categoria *Copyright Infringement* conforme os formatos IODEF e STIX.

Copyright Infringement				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
AttackPhase	URL	external-reference	source_name/url	claim_url
File	FileName/ FileProperties	File	name/extensions	file_title
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
Discovery	description	Infrastructure	description	detected_on _device
Software	description	Software	type/name/version	software
File	FileSize	File	size	file_size

Fonte: Autor.

Tabela 10 – Mapeamento de atributos e propriedades da categoria *Invasion Attempts/Vulnerabilities Exploitation* conforme os formatos IODEF e STIX.

Invasion Attempts/Vulnerabilities Exploitation				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
Service	port	network-traffic	type/src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
RelatedActivity	URL	external-reference	source_name/url	compromised_url
AttackPhase	URL	external-reference	source_name/url	explored_url
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
Service	Application	Software	type/name/version	affected_service
Discovery	description	Infrastructure	description	detected_on _device
Software	description	Software	type/name/version	software

Fonte: Autor.

Tabela 11 – Mapeamento de atributos e propriedades da categoria *Web Attacks* conforme os formatos IODEF e STIX.

Web Attacks				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
Service	port	network-traffic	type/src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
RelatedActivity	URL	external-reference	source_name/url	compromised_url
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 12 – Mapeamento de atributos e propriedades da categoria *Malware* conforme os formatos IODEF e STIX.

Malware				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
Service	port	network-traffic	type/src_port	origin_ports
RelatedActivity	URL	external-reference	source_name/url	malicious_url
Reference	ReferenceName	Malware	malware_types	malware
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
System	OperatingSystem	Malware	operating _system_refs	operative_system
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 13 – Mapeamento de atributos e propriedades da categoria *Spam* conforme os formatos IODEF e STIX.

Spam				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	ServiceName	network-traffic	type/protocols	protocol
AttackPhase	URL	external-reference	source_name/url	claim_url
Reference	ReferenceName	Malware	malware_types	malware
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 14 – Mapeamento de atributos e propriedades da categoria *Scan* conforme os formatos IODEF e STIX.

Scan				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	Portlist	network-traffic	type/dst_port	scanned_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
AttackPhase	URL	external- reference	source_name/url	explored_url
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 15 – Mapeamento de atributos e propriedades da categoria *Defacement* conforme os formatos IODEF e STIX.

Defacement				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	port	network-traffic	type/src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
RelatedActivity	URL	external- reference	source_name/url	compromised_url
Discovery	description	Infrastructure	description	detected_on _device
System	OperatingSystem	Malware	operating _system_refs	operative_system

Fonte: Autor.

Tabela 16 – Mapeamento de atributos e propriedades da categoria *Information Leak* conforme os formatos IODEF e STIX.

Information Leak				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	port	network-traffic	type/src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
AttackPhase	URL	external-reference	source_name/url	published_leak_url
RelatedActivity	URL	external-reference	source_name/url	compromised_url
Discovery	description	Infrastructure	description	detected_on _device
File	FileName/ FileProperties	File	name/extensions	file_title

Fonte: Autor.

Tabela 17 – Mapeamento de atributos e propriedades da categoria *Availability Attack* conforme os formatos IODEF e STIX.

Availability Attack				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	port	network-traffic	type/src_port	origin_ports
Service	ServiceName	network-traffic	type/protocols	protocol
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Reference	ReferenceName	Malware	malware_types	malware
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
Service	Application	Software	type/name/version	affected_service
AttackPhase	URL	external-reference	source_name/url	explored_url
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 18 – Mapeamento de atributos e propriedades da categoria *Phishing* conforme os formatos IODEF e STIX.

Phishing				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	type/value	ip_origin
DomainData	Name	domain-name	type/value	origin_hostname
Service	port	network-traffic	type/src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	type/name	explored_failure
Contact	irt/abuse	Identity	type/name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	type/ infrastructure _types	device_type
Service	ServiceName	Infrastructure	type/name/ description	main_service
RelatedActivity	URL	external- reference	source_name/url	compromised_url
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.

Tabela 19 – Mapeamento de atributos e propriedades da categoria *Asset Breach* conforme os formatos IODEF e STIX.

Asset Breach				
Classes IODEF V2	**Atributos IODEF V2**	**Objetos STIX V2.1**	**Propriedades STIX V2.1**	**Atributos Propostos**
Address	ipv4-addr/ ipv6-addr	IPv4 Address/ IPv6 Address	value	ip_origin
DomainData	Name	domain-name	value	origin_hostname
Service	port	network-traffic	src_port	origin_ports
Method Class	sci:Vulnerability	Vulnerability	name	explored_failure
Contact	irt/abuse	Identity	name	device_responsible
NodeRole	category/client/ server-public/ client-mobile	Infrastructure	infrastructure _types	device_type
Service	ServiceName	Infrastructure	name	main_service
AttackPhase	URL	external- reference	url	explored_url
RelatedActivity	URL	url	value	compromised_url
Discovery	description	Infrastructure	description	detected_on _device

Fonte: Autor.