CANEDA-IDS: Uma Arquitetura Orientada a Eventos para Detecção de Intrusão em Redes CAN

Felipe N. Dresch[1], **Felipe H. Scherer** [1], **Silvio E. Quincozes**[1], **Diego Kreutz**[1]

[1]LEA, Universidade Federal do Pampa (UNIPAMPA) – Alegrete, Brasil.

{felipedresch, felipescherer}.aluno@unipampa.edu.br

{silvioquincozes, diegokreutz}@unipampa.edu.br

Abstract. *In this work, we present an event-driven architecture for intrusion detection in vehicular CAN (Controller Area Network) networks. The effectiveness of the proposed architecture was evaluated using the ATAM (Architecture Tradeoff Analysis Method), which allowed for the identification of trade-offs between different quality attributes. Additionally, we conducted a compliance analysis with the ISO/IEC 25010 quality standards. The results indicate that the proposed architecture contributes to enhancing the security of vehicular CAN networks. Finally, the compliance analysis revealed that the architecture meets the critical requirements for robustness, adaptability, and reliability.*

Resumo. *Neste trabalho, apresentamos uma arquitetura orientada a eventos para a detecção de intrusões em redes veiculares CAN (Controller Area Network). A eficácia da arquitetura proposta foi avaliada utilizando o método ATAM (Architecture Tradeoff Analysis Method), que permitiu a identificação de trade-offs entre diferentes atributos de qualidade. Além disso, realizamos uma análise de conformidade com os padrões de qualidade da ISO/IEC 25010. Os resultados indicam que a arquitetura proposta contribui para aumentar a segurança das redes CAN veiculares. Por fim, a análise de conformidade demonstrou que a arquitetura atende aos requisitos críticos de robustez, adaptabilidade e confiabilidade.*

1. Introdução

A segurança em redes automotivas tem ganhado relevância à medida que veículos modernos integram cada vez mais sistemas eletrônicos e conectividade [Buscemi et al. 2023]. Dentre as tecnologias amplamente utilizadas, o protocolo de comunicação CAN (*Controller Area Network*) se destaca por sua eficiência e simplicidade, sendo uma escolha predominante em sistemas embarcados veiculares. No entanto, a ausência de mecanismos de segurança intrínsecos, como autenticação e criptografia, expõe essas redes a uma variedade de ataques cibernéticos, como *spoofing* de mensagens e ataques de negação de serviço (*DoS*). Essas vulnerabilidades não só colocam em risco a integridade dos sistemas veiculares, mas também comprometem diretamente a segurança dos passageiros.

Sistemas de Detecção de Intrusões (*Intrusion Detection Systems* - IDSs) são essenciais para a identificação de comportamentos anômalos em redes CAN [Lokman et al. 2019]. No entanto, muitos dos sistemas existentes enfrentam desafios em termos de escalabilidade, latência e precisão, especialmente em ambientes veiculares, onde respostas rápidas e confiáveis são fundamentais [Rajapaksha et al. 2023].

Este trabalho busca contribuir para minimizar os desafios enfrentados por IDSs em redes CAN através de uma arquitetura orientada a eventos utilizando o padrão *Publish-Subscribe* para redes CAN veiculares. A proposta é inovadora ao integrar essa arquitetura com uma avaliação baseada no método ATAM (*Architecture Tradeoff Analysis Method*) [Kazman et al. 2000], visando identificar *trade-offs* e melhorar a robustez do sistema. Além disso, é realizada também uma análise da qualidade da arquitetura conforme os padrões estabelecidos pela ISO/IEC 25010, assegurando a conformidade com os requisitos críticos de segurança, desempenho e escalabilidade.

Os objetivos principais deste estudo incluem a proposta arquitetural de um IDS distribuído e colaborativo para redes CAN, além da avaliação dessa arquitetura com base no método ATAM. O sistema proposto visa garantir uma resposta eficaz a ameaças cibernéticas em tempo real, possibilitando um IDS robusto e que não dependa do contexto de apenas um veículo para o seu aprendizado, mas que dispõe de uma rede heterogênea e rica em dados para contribuir com tal ação.

2. Fundamentação Teórica

2.1. *Controller Area Network* (CAN)

CAN é um protocolo de comunicação projetado para permitir a troca de dados entre dispositivos eletrônicos sem a necessidade de um controlador central, facilitando a comunicação entre microcontroladores em sistemas embarcados. Originalmente desenvolvida para aplicações automotivas, a rede CAN foi projetada para funcionar com rapidez e baixos custos [Desai et al. 2013]. Dessa forma, são frequentemente aplicadas em ambientes que exigem alta integridade de dados e comunicação em tempo real, como nos sistemas veiculares.

Apesar de sua robustez técnica para comunicação de dados, as redes CAN não possuem mecanismos intrínsecos para segurança cibernética (*e.g.,* autenticação ou criptografia de mensagens) [Lokman et al. 2019], tornando-as vulneráveis a diversas formas de ataque (*e.g., spoofing, Dos, replay e injeção*) [Lundberg et al. 2022]. A ausência de mecanismos de segurança possibilita que invasores manipulem a rede e prejudiquem o funcionamento do veículo. Essas vulnerabilidades representam um desafio significativo e, a partir disso, a utilização de IDSs neste tipo de rede se popularizou.

2.2. Arquitetura Orientada a Eventos

A arquitetura Orientada a Eventos é um modelo de design de software em que a execução e o fluxo do sistema são determinados por eventos (*e.g.,* mensagens CAN), que representam mudanças de estado ou condições específicas ocorrendo dentro ou fora do sistema [Reselman 2021]. Neste modelo, os componentes do sistema são projetados para reagir a esses eventos de maneira assíncrona, permitindo que o sistema seja altamente responsivo e adaptável a mudanças dinâmicas no ambiente de execução [Lazzari and Farias 2023]. Esta arquitetura comporta alguns padrões, como o *Publish-Subscribe* (*Pub-Sub*) [Eugster et al. 2003], frequentemente utilizado em padrões como o MQTT *(Message Queuing Telemetry Transport)* [Quincozes et al. 2019], que organiza a comunicação entre componentes de forma assíncrona e descentralizada. Nesse modelo, os publicadores (*publishers*) enviam mensagens ou eventos a um sistema intermediário sem conhecer os destinatários, enquanto os assinantes (*subscribers*) se registram para receber eventos de interesse, sem necessidade de conhecer a fonte dos eventos.

2.3. ATAM

Para garantir a qualidade de arquiteturas, diversas metodologias já foram propostas, como a *Architecture Tradeoff Analysis Method* (ATAM) [Kazman et al. 2000], que é uma metodologia formal desenvolvida para a avaliação de arquiteturas de software, visando identificar e gerenciar *trade-offs* relacionados a atributos de qualidade de sistemas complexos [Clements et al. 2001] através de uma estrutura sistemática para a avaliação dos compromissos entre diferentes atributos de qualidade. Com esta metodologia é possível identificar como decisões arquiteturais afetam os atributos de qualidade, como desempenho, segurança, e fornecer recomendações para otimizar esses atributos [Kazman et al. 1998]. O ATAM é especialmente útil em contextos onde múltiplos e possivelmente conflitantes requisitos devem ser equilibrados, como sistemas críticos para a segurança (*i.e.,* IDSs).

O ATAM é composto por uma série de etapas estruturadas que incluem: Preparação e Planejamento, onde se definem os objetivos; Apresentação da arquitetura; Identificação de Cenários de Uso, que são situações representativas que a arquitetura deve suportar; Análise de Atributos de Qualidade, que se dá por meio de uma avaliação dos atributos críticos da arquitetura e como ela os comportará; Identificação de *trade-offs*, que implica nos atributos de qualidade; Geração de Requisitos Arquiteturais, feita com base nas identificações das etapas anteriores e, finalmente, o Relatório da Avaliação, etapa onde se discute a analisa os resultados encontrados durante o processo de avaliação.

3. Trabalhos Relacionados

A literatura atual apresenta uma significativa carência no que tange as temáticas abordadas neste trabalho. Apesar disso, podemos elencar estudos que abordam um ou mais destes pontos de maneira individualizada.

Os autores de [Zhou et al. 2020] propuseram um IDS distribuído para redes veiculares baseado em um arcabouço de detecção colaborativa. Através da análise do comportamento dinâmico dos nós (veículos) presentes na rede, o arcabouço proposto consegue identificar comportamentos nocivos. O trabalho se assemelha ao que é proposto no presente trabalho de maneira conceitual apenas, no sentido de implementar um IDS distribuído e colaborativo para redes veiculares. No entanto, a sua aplicação prática abrange as redes veiculares externas e não trabalha o protocolo CAN especificamente.

O trabalho [Hindy et al. 2020] também apresenta uma proposta similar a pretendida neste estudo através da criação e avaliação de um IDS para redes IoT *(Internet of Things)* que utilizam o protocolo MQTT. Os autores comparam diferentes técnicas de aprendizado de máquina, abrangendo 6 tipos de algortimos distintos (*K-Nearest Neighbors, Support Vector Machine, Decision Trees, Random Forest, Naive Baye* e Regressão Logística), além de analisar quatro cenários de ataque. De maneira bastante similar, utilizando o mesmo *dataset* e analisando também as mesmas seis classes de algoritmos, os autores de [Khan et al. 2021] propõem um sistema de detecção de intrusões baseado em técnicas de aprendizado profundo *(Deep Learning - DL)* para identificar e mitigar ameaças presentes no protocolo MQTT dentro do contexto de IoT. No entanto, este trabalho possui o diferencial de incluir um modelo de rede neural profunda *Deep Neural Network - DNN* em suas avaliações, tendo sido este o modelo que apresentou os melhores resultados. Apesar de ambos serem importantes contribuições, o contexto de IoT difere significativamente do que é abordado aqui.

O estudo [Lee et al. 2009] apresenta uma análise detalhada da arquitetura de um sistema que gerencia transações de cartões de crédito online e, para isso, utiliza o Método ATAM para identificar riscos, propor estratégias de mitigação e sugerir melhorias na arquitetura do sistema. O estudo conclui que, apesar das dificuldades em derivar árvores de utilidade de qualidade ou abordagens arquitetônicas devido à falta de requisitos não funcionais predefinidos, a aplicação do ATAM e a modificação do processo permitiram avaliar e melhorar a arquitetura do sistema. Convergentemente, os autores de [Putrama et al. 2017] apresentam a criação de um sistema centralizado que integra dados de várias aplicações existentes em uma instituição educacional e os autores concluem que o uso do ATAM foi valioso para descobrir as forças e fraquezas das opções de arquitetura comparáveis.

4. Proposta

Esta seção descreve a metodologia proposta para a arquitetura de um IDS em redes veiculares do tipo CAN utilizando uma abordagem orientada a eventos com o padrão *Publish-Subscribe*. A metodologia detalha a estrutura e os componentes para a arquitetura, visando garantir a eficácia na detecção de intrusões e o suprimento de todos os requisitos levantados. Dessa forma, busca-se alcançar uma modelagem satisfatória. Além disso, como parte do processo, é realizada uma avaliação ATAM visando identificar *trade-offs* e possíveis falhas. Ainda, é realizada uma checagem de requisitos baseada na ISO/IEC 25010 que define padrões para a qualidade de software.

4.1. Requisitos da Arquitetura

Com base nas necessidades identificas para o sistema foram levantados os requisitos computados na Tabela 1, com suas descrições e prioridades. A natureza do sistema depende da troca de informações entre diversos atores e, portanto, existem muitas partes interdependentes. Dessa forma, é de vital importância o funcionamento adequado de todos estes componentes e torna-se inviável a priorização de apenas uma fração deles.

Tabela 1. Levantamento de Requisitos Funcionais (RF)

Requisitos	Descrição	Prioridade
RF01	O sistema deve ser capaz de detectar ataques	Alta
RF02	O sistema deve ser capaz de receber e realizar a troca do modelo de classificação	Alta
RF03	O sistema deve ser capaz de armazenar as mensagens malignas e postá-las no respectivo tópico	Alta
RF04	O sistema deve ser capaz de receber novas mensagens e realizar o treinamento do novo modelo	Alta
RF05	O sistema deve ser capaz de despachar as mensagens corretamente para os respectivos módulos	Alta

Além disso, o sistema deve atender aos requisitos não funcionais identificados e descritos na Tabela 2, a fim de assegurar sua eficácia e eficiência. Existem algumas exigências do sistema que devem ser priorizadas em primeiro momento, à exemplo do RFN01, referente à latência de detecção. É de suma importância que a detecção de intrusões seja feita em um tempo adequado, para que os eventuais mecanismos de proteção

(externos ao sistema proposto) possuam tempo hábil para realizar as suas ações. Além disso, a necessidade de escalabilidade (RFN06) deve ser sempre considerada, pois o sistema deve ser capaz de abarcar o recebimento de uma crescente quantidade de dados veiculares. Tal necessidade possui ligação direta com o RFN03, cuja capacidade de armazenamento exigida refere-se ao mínimo necessário para o sistema funcionar. O último requisito de alta prioridade refere-se à confiabilidade e disponibilidade (RFN07), pois o sistema deve estar sempre online para receber e enviar os dados exigidos.

Em relação aos requisitos com prioridade média, a manutenibilidade (RFN04), embora uma questão relevante, não configura uma necessidade primordial para o bom funcionamento do sistema. Da mesma forma, a vazão dos dados (RFN05) também não é uma questão crítica, pois o sistema ainda atende ao seu propósito nos casos em que a transmissão de envios e recebimentos esteja mais lenta devido ao alto tráfego de rede e qualquer informação que não for transmitida naquele momento pode ser feita no futuro. Este mesmo pensamento levou à atribuição de prioridade baixa à latência de publicação ou subscrição (RFN02), tendo em vista que essa comunicação não precisa acontecer de maneira instantânea. É preciso levar em consideração a velocidade oscilante e a baixa confiabilidade das redes móveis.

Tabela 2. Levantamento de Requisitos Não-Funcionais

Requisitos	**Descrição**	**Prioridade**
RFN01	Latência de Detecção: $\leq$ 1ms	Alta
RFN02	Latência de Publicação/Subscrição: $\leq$ 1min	Baixa
RFN03	Armazenamento mínimo de 100 GB	Alta
RFN04	Manutenibilidade	Média
RFN05	Alta Vazão	Média
RFN06	Escalabilidade	Alta
RFN07	Confiabilidade e Disponibilidade	Alta

4.2. Proposta de Arquitetura Orientada a Eventos

A arquitetura proposta é fundamentada nos requisitos descritos na Seção 4.1. Conforme ilustrado na Figura 1, a arquitetura do IDS é composta por três componentes principais: Coletor, Processador e Despachante de Eventos, cada um com suas respectivas funções. Além disso, utiliza tópicos específicos no *broker*. O tópico *can/messages* é responsável por receber as mensagens CAN maliciosas, que serão processadas pelo servidor, o qual realizará o treinamento de um novo modelo de detecção aprimorado, a ser publicado no tópico *can/fitmodel*. Essa abordagem se beneficia da comprovada eficiência do algoritmo XGBoost em tarefas de classificação, além de sua capacidade de converter o modelo treinado em um arquivo JSON, facilitando seu compartilhamento via *broker* [Dhaliwal et al. 2018]. Com isso, sempre que houver uma conexão disponível, o IDS receberá o modelo validado e o atualizará em seu módulo de detecção, desde que o novo modelo apresente um desempenho superior ao anterior, avaliado por métricas como *F1-Score*, *recall* e precisão, entre outras.

A arquitetura proposta é composta pelos seguintes componentes: o Coletor de Eventos *(Event Collector)*, responsável pela captura dos eventos transmitidos na rede

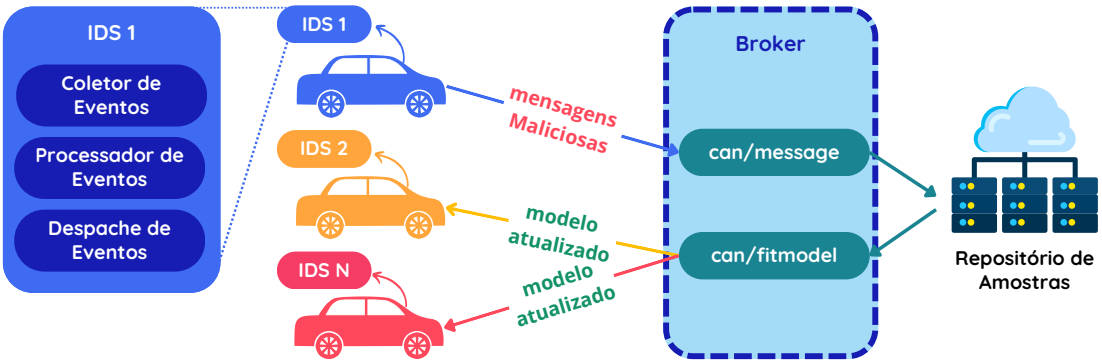

Figura 1. Modelagem da Arquitetura Proposta

CAN; o Despachante de Eventos *(Event Dispatcher)*, encarregado de encaminhar os eventos capturados para os módulos de processamento adequados, garantindo a entrega ordenada e em tempo hábil dos eventos aos componentes responsáveis; e, por fim, o Processador de Eventos *(Event Processor)*, que agrupa todos os módulos de processamento e tem a função de tratar os eventos recebidos, assegurando a execução adequada dos processos necessários.

O Processador de Eventos inclui dois módulos principais: o Módulo de Detecção de Intrusões, que utiliza o algoritmo XGBoost para identificar padrões anômalos nas mensagens CAN, sendo responsável por analisar os dados e detectar comportamentos que desviam do padrão esperado; e o Módulo de Armazenamento, encarregado de armazenar as mensagens CAN transmitidas e, quando apropriado, publicá-las no tópico *can/messages*.

4.3. Fluxo de Dados do IDS

A arquitetura proposta segue o seguinte fluxo de dados: i) **Captura de Eventos:** O Coletor de Eventos monitora e captura as mensagens e sinais da rede CAN; ii) **Despacho de Eventos:** Os eventos capturados são encaminhados pelo Despachante de Eventos para os módulos de processamento ou, quando identificado um fluxo benigno, redirecionados novamente para a rede. iii) **Processamento de Eventos:** Os Processadores de Eventos executam suas funções específicas (detecção de anomalias, armazenamento), podendo gerar novos eventos com base na análise realizada.

A metodologia proposta apresenta uma arquitetura estruturada e flexível para a implementação de um IDS eficiente em redes CAN veiculares, promovendo maior segurança e adaptabilidade a futuros avanços tecnológicos. Além disso, a abordagem distribuída permite que os IDSs compartilhem mensagens maliciosas identificadas, possibilitando o aprimoramento contínuo de outros sistemas IDS interconectados.

5. Aplicação do Método ATAM

Com o intuito de maximizar os benefícios do método de avaliação utilizado, este trabalho adapta as etapas do método ATAM para uma maior coesão com os objetivos propostos.

5.1. Preparação e Planejamento

Como parte fundamental da primeira etapa do método ATAM, que envolve preparação e planejamento, é imprescindível definir claramente os objetivos do sistema. Essa definição

é crucial para garantir que a proposta seja plenamente compreendida por todas as partes interessadas (*stakeholders*). Com isso em mente, foram estabelecidos os seguintes objetivos do sistema: i) **Segurança e Proteção:** Detectar e mitigar ataques cibernéticos em redes CAN veiculares, assegurando a segurança dos passageiros e a integridade do sistema; ii) **Eficiência Operacional:** Proporcionar uma detecção rápida e precisa de intrusões, sem comprometer a latência ou o *throughput* da rede; iii) **Adaptabilidade e Escalabilidade:** Permitir a atualização contínua e a fácil manutenção do sistema IDS, garantindo sua capacidade de adaptação a novas ameaças e tecnologias emergentes; iv) **Confiabilidade e Disponibilidade:** Garantir que o sistema IDS esteja sempre disponível e operacional, assegurando a detecção ininterrupta de ameaças.

Além disso, é fundamental identificar as partes interessadas no sistema. Esse processo envolve reconhecer e compreender todos os indivíduos ou entidades que possuem algum interesse ou exercem influência sobre o funcionamento do sistema. A etapa de identificação permitiu reconhecer as seguintes partes interessadas: i) **Fabricantes de Veículos:** Interessados na implementação de um sistema de segurança eficaz e confiável em seus produtos; ii) **Engenheiros de Segurança:** Responsáveis por monitorar e responder a ameaças de segurança na rede CAN; iii) **Desenvolvedores de Software:** Encarregados de implementar e realizar a manutenção contínua do sistema IDS; iv) **Proprietários de Veículos:** Beneficiários diretos da segurança aprimorada proporcionada pelo IDS.

5.2. Identificação de Cenários de Uso

A identificação dos cenários é essencial para avaliar a capacidade da arquitetura em lidar com diferentes demandas e situações. Para essa análise, foram definidos três cenários de uso principais. Esses cenários permitem compreender como a arquitetura se comporta em distintos contextos e como responde a diferentes exigências, garantindo que o sistema seja capaz de se adaptar às necessidades especificadas.

Tabela 3. Cenários de Uso

Cenário	Descrição	Estimulação	Resposta Esperada
Detecção de Intrusão	O sistema deve detectar um ataque de injeção de mensagens na rede CAN em tempo real.	O atacante injeta uma série de mensagens maliciosas na rede CAN.	O Módulo identifica o ataque com alta precisão e baixa latência.
Atualização do Modelo de Intrusão	O sistema deve ser capaz de receber e atualizar o modelo de detecção com base em novos dados de treinamento.	Novos dados de treinamento são disponibilizados no tópico can/messages para modelagem.	O sistema processa os novos dados e publica o modelo atualizado no tópico can/fitmodel.
Escalabilidade e Alta Carga de Eventos	O sistema deve operar eficientemente sob alta carga de eventos, garantindo *throughput* e baixa latência.	Um grande número de eventos é gerado na rede CAN.	O sistema processa todos os eventos sem degradação significativa no desempenho.

5.3. Análise dos Atributos de Qualidade

Nesta etapa, são identificados os atributos de qualidade essenciais para a arquitetura e avalia-se o nível de suporte que ela oferece para atender a esses atributos. Entre os principais atributos de qualidade considerados, estão:

- **Desempenho:** A capacidade do sistema de processar eventos em tempo real, mantendo baixa latência e alto *throughput*.

- **Segurança:** A eficácia do sistema em detectar ataques com precisão, minimizando a ocorrência de falsos positivos e negativos.
- **Manutenibilidade:** A facilidade com que o sistema pode ser atualizado, seja para incorporar novos modelos de detecção ou para corrigir possíveis falhas.
- **Escalabilidade:** A capacidade do sistema de lidar com um aumento na carga de eventos sem comprometer o desempenho.

5.4. Avaliação dos *Trade-Offs*

Com base nos atributos de qualidade, é importante avaliar e identificar os possíveis *trade-offs* entre eles. Essa análise ajuda a entender como diferentes atributos podem impactar uns aos outros e permite otimizar a arquitetura de forma a equilibrá-los adequadamente.

- **Desempenho vs. Segurança:** A adoção de algoritmos de detecção mais sofisticados pode melhorar a segurança, mas pode também introduzir latência adicional, afetando o desempenho.
- **Escalabilidade vs. Manutenibilidade:** Aumentar a escalabilidade do sistema pode exigir uma arquitetura mais complexa, o que pode dificultar a manutenção.
- **Confiabilidade vs. Desempenho:** Garantir alta disponibilidade pode requerer a utilização de recursos redundantes, o que pode aumentar a sobrecarga de processamento e impactar o desempenho geral do sistema.

5.5. Geração de Requisitos Arquiteturais

Com base na análise de *trade-offs*, aplica-se a etapa de geração de requisitos arquiteturais, na qual são identificados requisitos adicionais ou modificações que otimizam os atributos de qualidade. Nessa etapa, foram identificados três requisitos principais:

- **Requisito 1:** Implementar mecanismos de cache para melhorar o desempenho sem comprometer a precisão na detecção de intrusões.
- **Requisito 2:** Adicionar mecanismos de redundância para aumentar a confiabilidade, assegurando que falhas em componentes individuais não prejudiquem a operação global do sistema.
- **Requisito 3:** Utilizar técnicas de compressão para reduzir a sobrecarga de comunicação durante a transmissão de modelos atualizados.

6. Análise Qualitativa Baseada na ISO/IEC 25010

Para avaliar a qualidade da arquitetura proposta, foi realizada uma análise qualitativa com base em um *check-list*, visando assegurar o cumprimento das normas de qualidade estabelecidas pela ISO/IEC 25010. A seguir, destacam-se algumas particularidades da proposta.

Primeiramente, a modificabilidade pode representar um desafio, dado que o sistema apresenta certa complexidade e muitas partes interdependentes. Devido a essa mesma complexidade, a testabilidade completa de ponta a ponta também se revela uma tarefa desafiadora. Outro ponto a considerar é a instalabilidade, que pode ser limitada, já que cada veículo precisará de hardware compatível para conexão ao sistema. Por fim, a facilidade de uso pode ser um fator restritivo, pois o sistema exige conhecimento técnico especializado para sua operação adequada.

Tabela 4. *Check-list* baseado na ISO/IEC 25010

(a) *Check-list*

Categoria	Atendido
Funcionalidade	
Adequação Funcional	●
Precisão	●
Interoperabilidade	●
Conformidade	●
Desempenho e Eficiência	
Tempo de Resposta	●
Utilização de Recursos	●
Capacidade	●
Manutenibilidade	
Analisabilidade	●
Modificabilidade	◐
Testabilidade	◐

(b) *Check-list* (Cont.)

Categoria	Atendido
Portabilidade	
Adaptabilidade	●
Instalabilidade	◐
Substituibilidade	●
Segurança	
Confidencialidade	●
Integridade	●
Disponibilidade	●
Usabilidade	
Facilidade de Uso	○
Aprendizado	●
Operabilidade	●

7. Avaliação e Discussão

Nesta seção, foram analisados os resultados da aplicação do método ATAM e da avaliação qualitativa com base na norma ISO/IEC 25010, visando a arquitetura proposta para um sistema de detecção de intrusão em redes CAN. A aplicação do ATAM permitiu identificar importantes *trade-offs* entre os atributos de qualidade, como desempenho, segurança, escalabilidade e manutenibilidade. Por exemplo, o uso de algoritmos avançados de detecção de intrusão melhora a segurança, mas pode aumentar a latência, afetando o desempenho. Da mesma forma, a necessidade de alta escalabilidade pode tornar a arquitetura mais complexa, dificultando sua manutenção. O equilíbrio entre confiabilidade e desempenho também se mostrou desafiador, uma vez que garantir alta disponibilidade exige recursos redundantes, o que pode aumentar oprocessamento e comprometer o desempenho.

A análise qualitativa com base na ISO/IEC 25010 mostrou que a arquitetura atende bem a várias categorias de qualidade, como funcionalidade, desempenho, manutenibilidade e segurança. Ela se revelou adequada ao ambiente de redes CAN veiculares, oferecendo detecção eficaz de intrusões, baixa latência e alto *throughput*. A adaptabilidade e escalabilidade do sistema foram confirmadas, aspectos essenciais para enfrentar novas ameaças e incorporar avanços tecnológicos. No entanto, a análise apontou a necessidade de melhorar a interoperabilidade entre os componentes e implementar mecanismos de atualização robustos para garantir a proteção contínua.

Para trabalhos futuros, é fundamental que a implementação da arquitetura seja acompanhada de uma nova análise, sob um viés mais realista, a fim de validar os aprimoramentos propostos e garantir que as soluções desenvolvidas sejam adequadas ao ambiente operacional, levando em consideração variáveis e condições práticas que podem influenciar o desempenho e a segurança.

Referências

Buscemi, A., Turcanu, I., Castignani, G., Panchenko, A., Engel, T., and Shin, K. G. (2023). A survey on controller area network reverse engineering. *IEEE Communications Surveys & Tutorials*.

Clements, P. C., Kazman, R., and Klein, M. (2001). Evaluating software architectures.

Desai, M., Shetty, R., Padte, V., Parulekar, M., and Ramrajkar, S. (2013). Controller area network for intelligent vehicular systems. In *2013 International Conference on Advances in Technology and Engineering (ICATE)*, pages 1–6. IEEE.

Dhaliwal, S. S., Nahid, A.-A., and Abbas, R. (2018). Effective intrusion detection system using xgboost. *Information*, 9(7).

Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131.

Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., and Bellekens, X. (2020). Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset). In *International networking conference*, pages 73–84. Springer.

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., and Carriere, J. (1998). The architecture tradeoff analysis method. In *Proceedings. Fourth IEEE International Conference on Engineering of Complex Computer Systems (Cat. No.98EX193)*, pages 68–78.

Kazman, R., Klein, M., and Clements, P. (2000). *ATAM: Method for architecture evaluation*. Carnegie Mellon University, Software Engineering Institute Pittsburgh, PA.

Khan, M. A., Khan, M. A., Jan, S. U., Ahmad, J., Jamal, S. S., Shah, A. A., Pitropakis, N., and Buchanan, W. J. (2021). A deep learning-based intrusion detection system for mqtt enabled iot. *Sensors*, 21(21):7016.

Lazzari, L. and Farias, K. (2023). Uncovering the hidden potential of event-driven architecture: A research agenda.

Lee, J., Kang, S., Chun, H., Park, B., and Lim, C. (2009). Analysis of van-core system architecture- a case study of applying the atam. In *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, pages 358–363.

Lokman, S.-F., Othman, A. T., and Abu-Bakar, M.-H. (2019). Intrusion detection system for automotive controller area network (can) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–17.

Lundberg, H., Mowla, N. I., Abedin, S. F., Thar, K., Mahmood, A., Gidlund, M., and Raza, S. (2022). Experimental analysis of trustworthy in-vehicle intrusion detection system using explainable artificial intelligence (xai). *IEEE Access*, 10:102831–102841.

Putrama, I. M., Dermawan, K. T., Dantes, G. R., and Aryanto, K. Y. E. (2017). Architectural evaluation of data center system using architecture tradeoff analysis method (atam): A case study. In *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*.

Quincozes, S., Emilio, T., and Kazienko, J. (2019). MQTT protocol: fundamentals, tools and future directions. *IEEE Latin America Transactions*, 17(09):1439–1448.

Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., and Cheah, M. (2023). Ai-based intrusion detection systems for in-vehicle networks: A survey. *ACM Computing Surveys*, 55(11):1–40.

Reselman, B. (2021). Architectural messaging patterns: an illustrated guide. Accessed on August, 2024.

Zhou, M., Han, L., Lu, H., and Fu, C. (2020). Distributed collaborative intrusion detection system for vehicular ad hoc networks based on invariant. *Computer Networks*, 172:107174.