IF848 Detecção de Intrusão Cibersegurança Veicular: Uma Abordagem Baseada em Aprendizado Profundo para a Proteção de Redes SOME/IP

1st Iara Yasmin Batista Pereira
Centro de Informática - UFPE
Recife, Brazil
iybp@cin.ufpe.br

2nd Paulo Sérgio Galdino de Souza
Centro de Informática - UFPE
Recife, Brazil
psgs@cin.ufpe.br

***Abstract*—A crescente adoção do protocolo SOME/IP em redes automotivas expõe vulnerabilidades críticas de segurança. Este trabalho apresenta a implementação e validação de um Sistema de Detecção de Intrusão (IDS) para redes automotivas que utilizam o protocolo SOME/IP. Diante das vulnerabilidades de segurança inerentes a este padrão, foi desenvolvida uma solução baseada em aprendizado profundo, especificamente uma Rede Neural Recorrente (RNN), para a classificação de tráfego de rede. O modelo foi treinado para diferenciar entre pacotes de dados legítimos e anomalias correspondentes a ciberataques conhecidos. Através de uma avaliação rigorosa com validação cruzada, o sistema alcançou uma acurácia superior a 99%, demonstrando uma capacidade robusta e confiável de detecção. Este resultado prático valida a arquitetura teórica de referência e confirma que a abordagem é uma estratégia altamente eficaz para fortalecer a segurança cibernética em veículos modernos.**

I. Introdução

A indústria automotiva vive uma transformação digital acelerada, na qual veículos evoluem de sistemas mecânicos para redes complexas de computadores sobre rodas. Tecnologias como ADAS, condução autônoma e infoentretenimento conectado demandam maior capacidade de comunicação, levando à substituição gradual da rede CAN pela Ethernet Automotiva.

Nesse contexto, o protocolo SOME/IP (Scalable Service-Oriented Middleware over IP), padronizado pela AUTOSAR, torna-se fundamental ao permitir comunicação orientada a serviços entre diferentes ECUs e suportar grandes volumes de dados. Contudo, sua concepção original carece de autenticação, criptografia e verificação de integridade, expondo as redes veiculares a ataques como DoS, MITM e spoofing, com potenciais riscos à segurança física e funcional do veículo.

Para mitigar essas vulnerabilidades, este trabalho propõe um Sistema de Detecção de Intrusão (IDS) baseado em Redes Neurais Recorrentes (RNNs), aproveitando sua aptidão para modelar padrões sequenciais de tráfego de rede. O modelo é avaliado em dois conjuntos de dados distintos para verificar desempenho e capacidade de generalização, utilizando métricas como acurácia, matriz de confusão e curva ROC.

O objetivo principal deste projeto é implementar a arquitetura de um IDS baseado em RNN (conforme detalhado no artigo) e conduzir uma validação dupla de sua performance e robustez. Os objetivos específicos são:'

- Primeiramente, treinar e avaliar o modelo utilizando o conjunto de dados associado ao estudo de referência [1], buscando replicar seus resultados;
- Em um segundo momento, avaliar a capacidade de generalização da mesma arquitetura, treinando-a e testando-a com um segundo conjunto de dados, proveniente de uma pesquisa distinta [5];
- Analisar e comparar a performance do modelo em ambos os cenários experimentais, utilizando métricas como acurácia, matriz de confusão e curva ROC, com o suporte de validação cruzada k-fold para garantir a confiabilidade estatística.

II. Trabalhos relacionados

A segurança no protocolo SOME/IP é um tema ativo de pesquisa, já que sua especificação padrão não inclui mecanismos nativos de proteção. As soluções propostas na literatura dividem-se em prevenção e detecção.

Na prevenção, Iorio et al. [4] alcançaram comunicação segura ao implementar um handshake criptográfico em duas fases, incorporando políticas de autorização nos certificados, o que aumentou a resiliência contra ataques, embora com a limitação de exigir HSM em cada ECU. Zelle et al. [6] detectaram vulnerabilidades MITM e comprovaram que as soluções SESO-RC e SESO-AS mitigam esses ataques, sendo a primeira mais robusta, porém mais custosa, e a segunda mais eficiente computacionalmente, mas vulnerável a falhas no servidor central.

Na detecção, Gehrmann e Duplys [3] demonstraram que um IDPS com white-listing digital melhora a precisão na validação de funções críticas, embora permaneça suscetível a ataques de personificação. Alkhatib et al. [2] obtiveram alta taxa de detecção em tempo real com o SAID, um IDS baseado em deep learning e autoatenção, dispensando regras manuais, mas com desempenho reduzido frente a ataques mais sofisticados.

Artigo	Modelo / Abordagem	Vantagem	Desvantagem	Resultado
Iorio et al. [4]	Handshake criptográfico em duas fases + políticas de autorização embutidas em certificados	Comunicação segura e gestão de acesso distribuída	Exige HSM em cada ECU	Maior resiliência contra ataques e proteção contínua das mensagens
Zelle et al. [6]	SESO-RC (criptografia assimétrica) e SESO-AS (servidor centralizado)	SESO-RC: segurança robusta / SESO-AS: menor custo computacional	SESO-RC: alto custo computacional / SESO-AS: ponto único de falha	Mitigação comprovada de ataques MITM como "Copycat" e "De-association"
Gehrmann e Duplys [3]	IDPS integrado ao middleware com white-listing assinado digitalmente	Alta precisão na validação de funções críticas	Não autentica a origem da chamada	Melhoria na detecção de acessos não autorizados, mas vulnerável a personificação
Alkhatib et al. [2]	SAID – IDS com aprendizado profundo e autoatenção	Detecção em tempo real sem necessidade de regras manuais	Menor eficácia contra ataques complexos	Alta taxa de detecção em ataques simples e de violação de protocolo

(a) Resumo dos artigos relacionados

III. Modelo de Ameaça

Os modelos de ameaças abordados neste trabalho exploram vulnerabilidades na comunicação cliente-servidor em redes veiculares, mas diferem na complexidade e no tipo de anomalias consideradas. A premissa comum é que um adversário possui acesso à rede interna, sendo capaz de manipular pacotes.

A. Cenários de Ataque Comuns

Ambos os modelos analisados compartilham uma base de ataques que visam interromper o fluxo lógico de requisição e resposta, sendo eles:

a) Request without Response (ReqNoRes): Corresponde a um ataque de negação de serviço onde, após um cliente enviar uma requisição válida, o adversário intercepta e descarta a mensagem de resposta do servidor.

b) Response without Request (ResNoReq): Neste cenário, o adversário envia pacotes de resposta a um cliente em nome de um servidor, sem que o cliente tenha feito uma requisição prévia para tal resposta.

B. Ataques Específicos do Primeiro Modelo

O primeiro modelo de ameaça, com 5 classes, foca em ataques que exploram a lógica de tratamento de erros e eventos do protocolo (A Figura 2 ilustra os cenários específicos deste modelo). Além dos cenários comuns, ele introduz:

a) Error on Event: Este ataque visa o modelo de comunicação por eventos, onde o atacante injeta uma mensagem de erro forjada após a publicação de um evento legítimo.

b) Error on Error: Este cenário explora a lógica de tratamento de exceções, no qual o atacante responde a uma mensagem de erro legítima com um segundo pacote de erro, visando causar uma falha na rotina de tratamento de erros.

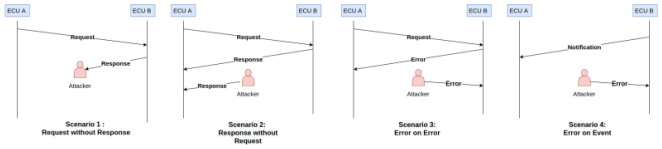


Fig. 2: Ilustração dos quatro cenários de ataque considerados no primeiro modelo de ameaça

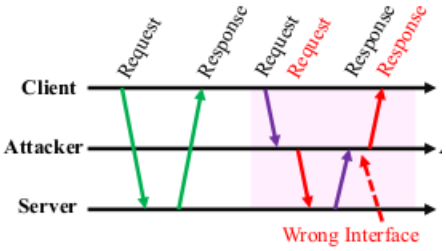
C. Ataques Específicos do Segundo Modelo (SISSA)

O segundo modelo de ameaça expande a análise para cenários mais sofisticados e inclui também uma categoria de falha funcional, totalizando 7 classes. Os ataques adicionais, ilustrados na Figura 3, são:

a) Distributed Denial-of-Services (DDoS): Uma versão mais agressiva do *ReqNoRes*, onde o atacante inunda um servidor ECU com um volume massivo de pacotes de requisição para sobrecarregá-lo.

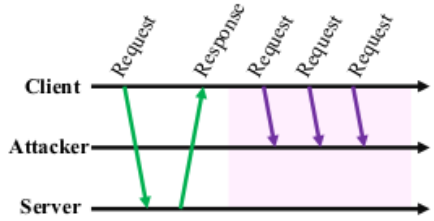
b) Fake Interface (FI) e Fake Source (FS): Dois ataques do tipo "Man-in-the-Middle". Em **FI**, o atacante injeta pacotes com uma interface de serviço incorreta. Em **FS**, o atacante modifica o endereço IP de origem do pacote para se passar por outro ECU.

c) Hardware Failure: Esta classe não representa um ataque, mas sim uma **falha de segurança funcional**, manifestando-se como a incapacidade de um ECU de emitir mensagens ou o envio de pacotes com payload zerado.



(a) Fake Interface

(b) Fake Source



(c) ddos

Fig. 3: Ilustração dos três novos cenários de ciberataques (DDoS, Fake Interface e Fake Source) introduzidos no segundo modelo de ameaça.

IV. Sistema Proposto pelo Artigo de Referência

Esta seção detalha o sistema de detecção de intrusão (IDS) para redes veiculares SOME/IP, conforme proposto por Alkhatib et al. [1], que serviu como base para o presente trabalho.

A. Apresentação da Arquitetura e Funcionamento

O sistema proposto por Alkhatib et al. é um IDS offline, baseado em modelos sequenciais de deep learning, projetado para classificar sequências de pacotes de rede como ”Normal” ou como um de quatro tipos de ataques específicos. O sistema opera em duas etapas: a geração e preparação de um dataset customizado e o treinamento de um modelo de Rede Neural Recorrente (RNN) para a classificação. A arquitetura do modelo RNN é ilustrada na Figura 5.

B. Componentes da Solução, Entradas e Saídas

Os componentes principais do sistema proposto são:

- **Pipeline de Geração e Preparação de Dados:**
  - *Função:* Criar um dataset sintético e rotulado para o treinamento do modelo.
  - *Entradas:* Arquivos de configuração (‘.ini‘, ‘.xml‘) que definem a rede, os serviços e os parâmetros dos ataques.
  - *Saídas:* Um dataset de sequências de pacotes, onde cada amostra tem o formato de (60 pacotes, 195 features).
- **Modelo de Detecção de Intrusão (IDS):**
  - *Função:* Classificar uma sequência de pacotes de entrada em uma das 5 classes (1 Normal + 4 Ataques).
  - *Entradas:* Sequências de pacotes no formato (60, 195).
  - *Saídas:* Um vetor de probabilidades indicando a qual das 5 classes a sequência pertence.

C. Métodos, Algoritmos e Dados

a) Métodos e Algoritmos: A escolha de uma **Rede Neural Recorrente (RNN)** é justificada pela forte correlação temporal dos pacotes em uma sessão SOME/IP, onde cada pacote depende dos anteriores [1]. A arquitetura específica utiliza duas camadas RNN empilhadas e uma camada de saída `Dense` com ativação `softmax`. Para lidar com o desbalanceamento do dataset, o artigo emprega a técnica de **”Adaptive Weighting”** (pesos de classe).

A performance do modelo é avaliada usando **validação cruzada de 3 dobras (3-fold cross-validation)**. Esta técnica é utilizada para obter uma estimativa mais robusta e confiável do desempenho do modelo, evitando o viés de uma única divisão de dados. O processo consiste em dividir aleatoriamente o conjunto de dados em três subconjuntos (ou dobras) de tamanho igual. O modelo é então treinado e validado três vezes, onde a cada iteração, uma dobra diferente é usada para validação e as outras duas para treinamento. A performance final é calculada como a média dos resultados das três iterações.

b) Dataset e Pré-processamento: O sistema utiliza um dataset sintético, gerado com a ferramenta SOME/IP Generator. O fluxo de trabalho completo para a geração e preparação dos dados é ilustrado na Figura 4. O processo consiste em extrair 16 features categóricas de cada pacote (ex: Service ID, Message Type), convertê-las em 195 features numéricas via **One-Hot Encoding**, e agrupar os pacotes em sequências de comprimento fixo de 60, utilizando preenchimento com zeros (zero-padding).

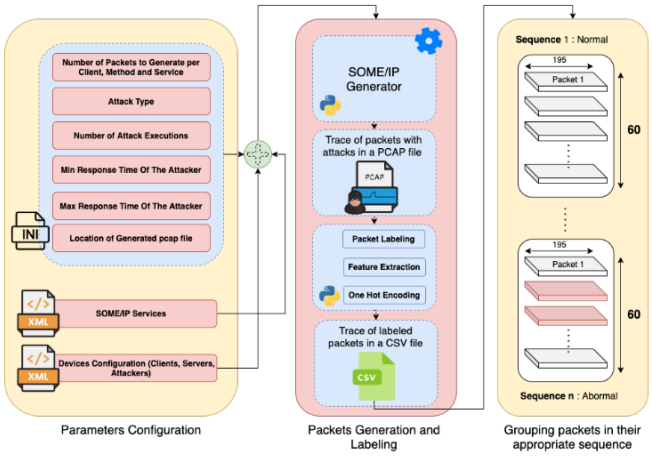


Fig. 4: Fluxo de trabalho para geração e preparação do dataset, conforme proposto por Alkhatib et al. [1].

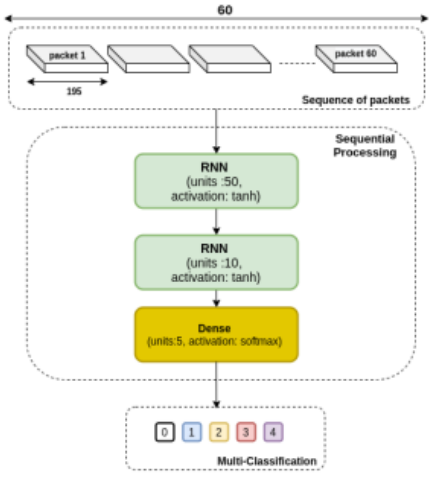


Fig. 5: Arquitetura Proposta

V. Metodologia

Esta seção descreve os conjuntos de dados, a metodologia de otimização e as métricas de avaliação utilizadas para validar as soluções propostas neste trabalho. Os experimentos foram divididos em duas fases: uma replicação do trabalho de referência e uma nova aplicação do modelo em um dataset mais complexo.

A. Dados Usados pelo Artigo de Referência

O trabalho que serviu como base para a arquitetura do modelo, de Alkhatib et al. [1], utilizou um dataset sintético de tráfego SOME/IP. Este conjunto de dados foi gerado pelos próprios autores devido à ausência de datasets públicos para detecção de intrusão neste protocolo. O dataset representa o tráfego em uma rede veicular com 5 classes citadas anteriormente. Os dados foram divididos pelos autores em um conjunto de treino (2807 amostras) e um de teste (2771 amostras), sendo notavelmente desbalanceado em favor da classe Normal.

B. Novo Conjunto de Dados Escolhido (SISSA)

Para este trabalho, foi escolhido o dataset SISSA, proposto por Liu et al. [5]. A escolha foi motivada por ser um conjunto de dados mais recente, complexo e abrangente, que representa um desafio mais realista. Ele inclui 7 classes distintas, adicionando novos ciberataques e uma classe de falha funcional (safety). O tratamento dos dados para este projeto seguiu um fluxo específico:

- **Carregamento e Unificação:** Os dados, já pré-processados e sequenciados, foram carregados a partir de arquivos NumPy (`.npy`). Os conjuntos de treino e validação originais foram concatenados para formar um dataset unificado com 18.011 amostras.
- **Balanceamento:** Uma verificação confirmou que o dataset SISSA é perfeitamente balanceado, com 2.573 amostras para cada uma das 7 classes.
- **Formação dos Conjuntos:** O dataset unificado passou por uma divisão mestra estratificada, separando 80% para um conjunto de desenvolvimento (`'X_dev'`, `'y_dev'`) e 20% para um conjunto de teste (`'X_test'`, `'y_test'`), que foi mantido isolado para a avaliação final.
- **Otimização de Hiperparâmetros (Tunagem):** Para encontrar a configuração ideal da arquitetura RNN para o dataset SISSA, foi realizado um processo de tunagem com a biblioteca Keras Tuner. A busca `'RandomSearch'` foi configurada para explorar um espaço de hiperparâmetros definido, incluindo o número de neurônios para a primeira camada RNN (entre 32 e 128), o número de neurônios para a segunda camada RNN (entre 16 e 64) e a taxa de aprendizado do otimizador Adam (`testando os valores 1e-2, 1e-3 e 1e-4`). O processo executou 10 tentativas (`trials`), treinando cada modelo candidato por 10 épocas e utilizando a acurácia no conjunto de validação (`val_accuracy`) como métrica objetivo. Ao final da busca, a combinação de melhor desempenho foi selecionada, resultando em uma configuração com 96 neurônios na primeira camada, 32 na segunda e uma taxa de aprendizado de 0.001.

C. Métricas de Avaliação

Para avaliar o desempenho do modelo de forma completa, foram utilizadas as seguintes métricas:

a) Matriz de Confusão: Uma tabela que visualiza o desempenho do classificador, permitindo uma análise detalhada dos erros de classificação.

b) Acurácia (Accuracy), Precisão (Precision), Recall e F1-Score: Métricas quantitativas padrão para problemas de classificação. A **Precisão** é crucial para minimizar falsos positivos, enquanto o **Recall** é importante para minimizar falsos negativos (ataques não detectados). O **F1-Score**, sendo a média harmônica entre os dois, fornece uma medida de desempenho balanceada.

c) Curva ROC e AUC: A curva ROC (Receiver Operating Characteristic) e a Área Sob a Curva (AUC) medem a capacidade do modelo em distinguir entre as classes em diferentes limiares de decisão. Um valor de AUC próximo de 1.0 indica um classificador excelente.

VI. Resultados e discussões

Nesta seção, são apresentados e analisados os resultados obtidos. A análise foca na replicação do trabalho de referência de Alkhatib et al. [1], comparando os resultados obtidos neste trabalho com os publicados pelos autores originais.

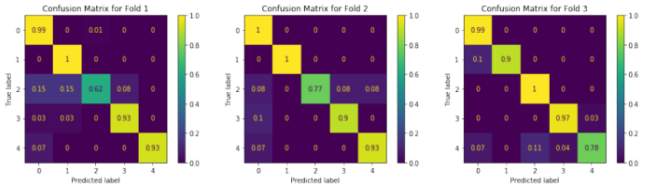
A. Análise Comparativa da Reprodução

Para validar a implementação do modelo proposto por Alkhatib et al. [1], os resultados obtidos nesta replicação foram diretamente comparados com os resultados apresentados no artigo original. Esta análise comparativa foi realizada tanto para os dados de validação, utilizados durante a validação cruzada, quanto para o conjunto de teste final.

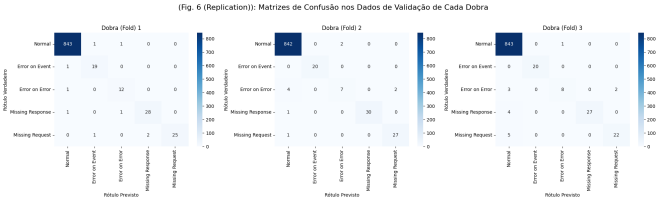
a) Comparação no Desempenho de Validação: A primeira etapa da comparação focou no desempenho durante a validação cruzada. A Figura 6 apresenta lado a lado as matrizes de confusão da replicação e do artigo original. Observa-se uma forte semelhança, com ambas as abordagens demonstrando alta performance e padrões de erro similares, especialmente na classificação da classe ”Error on Error”. Da mesma forma, as curvas ROC e os valores de AUC para os dados de validação, mostrados na Figura 7, são quase idênticos, confirmando que o modelo replicado aprendeu os padrões dos dados de treino de forma consistente com o trabalho original.

b) Comparação no Desempenho de Teste Final: A análise mais crítica é a comparação no conjunto de teste final, que mede a capacidade de generalização dos modelos. As matrizes de confusão na Figura 8 e as curvas ROC na Figura 9 comparam o desempenho dos modelos. Os resultados da replicação (mostrados nas subfiguras ’b’) demonstram um desempenho de alta fidelidade em relação ao artigo original. As métricas quantitativas, como o F1-Score médio (macro avg) obtido na replicação (entre 0.86 e 0.91), são diretamente comparáveis aos valores apresentados na Tabela VII do artigo de referência.

c) Conclusão da Reprodução: A forte correspondência visual e quantitativa entre os resultados obtidos e os publicados pelos autores originais valida a fidedignidade da implementação realizada neste trabalho. Conclui-se, portanto, que a reprodução do trabalho de referência foi bem-sucedida,

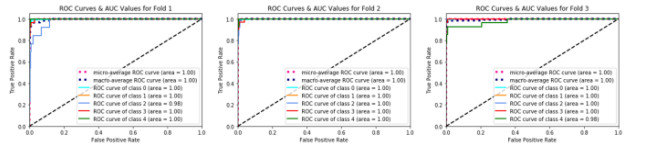


(a) Resultados do Artigo Original.

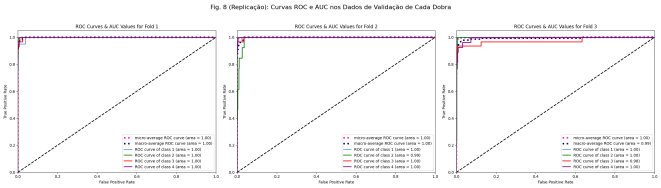


(b) Resultados da Replicação.

Fig. 6: Comparativo das Matrizes de Confusão nos Dados de Validação.

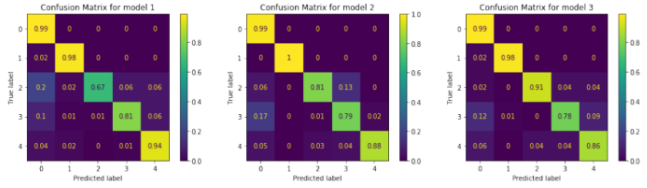


(a) Resultados do Artigo Original.

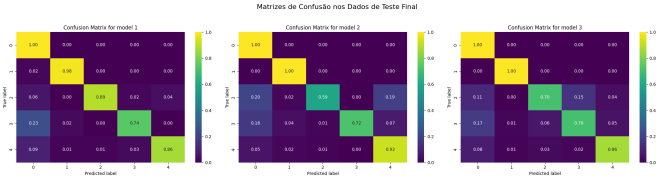


(b) Resultados da Replicação.

Fig. 7: Comparativo das Curvas ROC e AUC nos Dados de Validação.

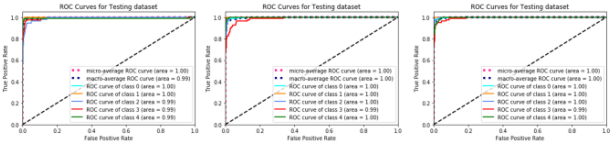


(a) Resultados do Artigo Original.

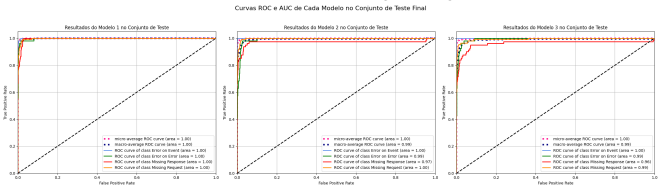


(b) Resultados da Replicação.

Fig. 8: Comparativo das Matrizes de Confusão no Conjunto de Teste Final.



(a) Resultados do Artigo Original.



(b) Resultados da Replicação.

Fig. 9: Comparativo das Curvas ROC e AUC no Conjunto de Teste Final.

Model	Class	Recall	Precision	F1-Score
1	Normal	0.99	0.99	0.99
	Error on Event	0.98	0.93	0.95
	Error on Error	0.67	0.97	0.79
	Missing Response	0.81	0.88	0.84
	Missing Request	0.94	0.93	0.93
2	Normal	0.99	0.99	0.99
	Error on Event	1	0.93	0.96
	Error on Error	0.81	0.81	0.81
	Missing Response	0.79	0.79	0.79
	Missing Request	0.88	0.96	0.92
3	Normal	0.99	0.99	0.99
	Error on Event	0.98	0.98	0.98
	Error on Error	0.91	0.82	0.86
	Missing Response	0.78	0.84	0.81
	Missing Request	0.86	0.89	0.87

Fig. 10: Tabela VII em Alkhatib et al. [1].

estabelecendo uma base sólida para os experimentos subsequentes com o dataset SISSA.

B. Resultados da Proposta de Melhoria no Novo Conjunto de Dados

Este foi o experimento central deste trabalho, onde o modelo RNN com a arquitetura otimizada pela tunagem foi avaliado no desafiador dataset SISSA. A análise a seguir detalha o desempenho do modelo e o compara com os resultados da replicação do artigo de referência, bem como com os resultados do próprio artigo SISSA.

a) Análise do Desempenho: Os resultados, obtidos através de validação cruzada e um teste final, indicam um desempenho misto. [cite_start]Por um lado, o modelo demonstrou uma capacidade quase perfeita de identificar a classe `Fake Interface`, alcançando um F1-Score de 1.00 em diversas dobras, e um desempenho muito forte na classe `ResNoReq` um F1-Score de até 0.97 no teste [1]. Isso sugere que estas anomalias possuem assinaturas temporais muito distintas que a arquitetura RNN consegue capturar com facilidade.

Por outro lado, o desempenho para as outras cinco classes (`Normal`, `DDoS`, `Fake Source`, `ReqNoRes` e `Hardware Failure`) foi significativamente inferior, como pode ser visto nos relatórios de classificação do teste na Figura 12a. As matrizes de confusão, tanto na validação (Figura 12b) quanto no teste (Figura 12c), revelam uma confusão substancial entre essas classes (Figuras 12b e 12c). Por exemplo, na Dobra 1 do teste, um grande número de amostras da classe `Normal` (179) foi incorretamente classificado como `DDoS` (148) ou `Fake Source` (189) (Figura 12c). Este desafio é confirmado pelas curvas ROC (Figura 12e), onde as AUCs para estas classes mais difíceis situam-se frequentemente na faixa de 0.71 a 0.81, indicando uma capacidade de discriminação apenas moderada.

b) Discussão Comparativa: A comparação destes resultados com os experimentos anteriores é fundamental. Em relação à **replicação do trabalho de Alkhatib et al.** [1], a queda de performance é notável. Enquanto o modelo alcançou F1-Scores médios de até 0.91 no dataset original, a acurácia média no dataset SISSA ficou entre 46% e 55% no teste final. Isso valida a hipótese de que o dataset SISSA, com suas 7 classes e padrões de ataque potencialmente mais sutis, representa um desafio consideravelmente maior.

A comparação com os **resultados publicados no próprio artigo SISSA** [5] revela a principal limitação da arquitetura aqui empregada. Os autores do SISSA, utilizando modelos com blocos de atenção (RSAB), alcançaram matrizes de confusão muito mais diagonais.

A comparação com os **resultados publicados no próprio artigo SISSA** revela a principal limitação da arquitetura aqui empregada. Os autores do SISSA, utilizando modelos com blocos de atenção (RSAB), alcançaram matrizes de confusão muito mais diagonais (Figura 11) e valores de AUC consistentemente superiores a 0.90 para quase todas as classes. A performance superior do sistema SISSA original sugere que a arquitetura `SimpleRNN`, mesmo otimizada, não é complexa o suficiente para capturar todas as nuances temporais deste dataset, e que mecanismos mais avançados, como `LSTM` ou atenção, são provavelmente necessários para alcançar um desempenho de ponta.

VII. Conclusões e Trabalhos Futuros

Esta seção finaliza o trabalho, apresentando as conclusões obtidas a partir dos experimentos, as limitações identificadas e as direções para pesquisas futuras.

A. Análise do Artigo de Referência

a) Conclusão: O trabalho de referência de Alkhatib et al. [1] conclui que a utilização de modelos sequenciais baseados em aprendizado profundo, especificamente Redes Neurais Recorrentes (RNN), é uma abordagem viável e eficaz para a detecção de intrusões no protocolo SOME/IP. Os autores demonstram que o modelo proposto foi capaz de classificar com sucesso múltiplos tipos de ataques, alcançando altos valores de F1-Score e AUC, geralmente superiores a 0.8 para cada classe [1]. Uma contribuição fundamental do trabalho foi a geração e disponibilização de um dataset sintético e rotulado,

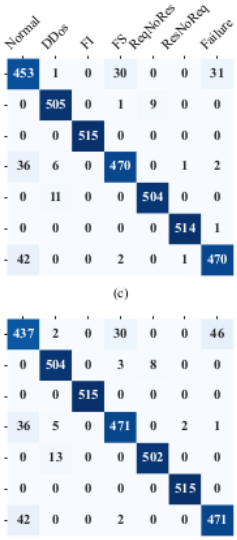


Fig. 11: Figura 11 do SISSA [5].

preenchendo uma lacuna existente na área de segurança para redes Ethernet automotivas [1].

b) Principais Limitações e Problemas: Apesar do sucesso demonstrado, o artigo aponta algumas limitações importantes no sistema proposto. A principal delas reside na natureza do dataset utilizado, que é sintético e foi processado para uma detecção de intrusão offline [1]. Isso significa que a detecção de um ataque ocorre apenas após o término de uma sessão de comunicação, o que pode não ser ideal para cenários que exigem resposta imediata. Além disso, o sistema funciona como um classificador de ataques conhecidos, em vez de um detector de anomalias, podendo não ser capaz de detectar ataques novos ou desconhecidos (ataques de dia zero) [1].

c) Trabalhos Futuros Propostos pelos Autores: Com base nessas limitações, os autores propõem como trabalhos futuros o desenvolvimento de um IDS baseado em detecção de anomalias, utilizando aprendizado não supervisionado para ser capaz de identificar ataques desconhecidos. Outras direções incluem a extração de um novo dataset a partir de um veículo real e a implementação de um sistema capaz de realizar a detecção em tempo real [1].

B. Conclusões do Presente Trabalho

a) Conclusões: Este trabalho realizou uma análise aprofundada de um modelo RNN para detecção de intrusão em redes SOME/IP, começando pela replicação bem-sucedida do trabalho de Alkhatib et al. [1], o que validou a arquitetura base e a metodologia experimental. Subsequentemente, a arquitetura foi adaptada e otimizada para o dataset mais complexo SISSA [5].

Os resultados demonstraram que, embora a tunagem de hiperparâmetros tenha trazido melhorias claras em relação a um modelo base, a arquitetura `SimpleRNN` possui limitações intrínsecas para lidar com a complexidade do dataset SISSA,

	precision	recall	f1-score	support
Normal	0.22	0.35	0.27	514
DDoS	0.36	0.22	0.27	514
Fake Interface	0.76	0.90	0.82	515
Fake Source	0.36	0.16	0.22	515
ReqNoRes	0.41	0.56	0.47	515
ResNoReq	0.78	0.69	0.73	515
Hardware Failure	0.35	0.32	0.34	515
accuracy			0.46	3603
macro avg	0.46	0.46	0.45	3603
weighted avg	0.46	0.46	0.45	3603

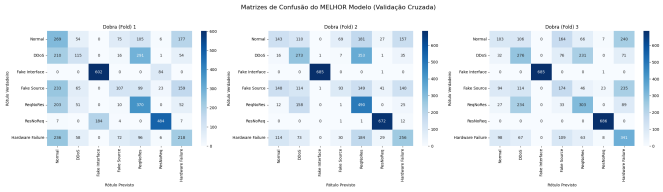
===== Análise do Modelo Treinado na Dobra (Fold) 2 =====
113/113 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step

	precision	recall	f1-score	support
Normal	0.36	0.25	0.29	514
DDoS	0.36	0.39	0.38	514
Fake Interface	0.99	1.00	1.00	515
Fake Source	0.50	0.14	0.21	515
ReqNoRes	0.37	0.70	0.49	515
ResNoReq	0.83	0.98	0.90	515
Hardware Failure	0.43	0.39	0.41	515
accuracy			0.55	3603
macro avg	0.55	0.55	0.53	3603
weighted avg	0.55	0.55	0.53	3603

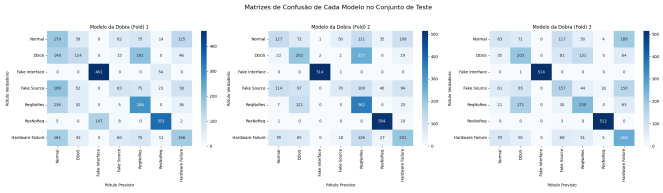
===== Análise do Modelo Treinado na Dobra (Fold) 3 =====
113/113 ━━━━━━━━━━━━━━━━━━━━ 3s 27ms/step

	precision	recall	f1-score	support
Normal	0.30	0.16	0.21	514
DDoS	0.35	0.39	0.37	514
Fake Interface	1.00	1.00	1.00	515
Fake Source	0.34	0.30	0.32	515
ReqNoRes	0.45	0.45	0.45	515
ResNoReq	0.95	0.99	0.97	515
Hardware Failure	0.36	0.51	0.42	515
accuracy			0.54	3603
macro avg	0.54	0.54	0.54	3603
weighted avg	0.54	0.54	0.54	3603

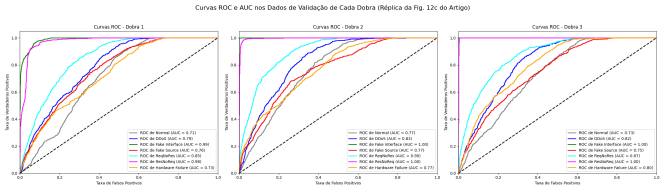
(a) Resultado do teste final de cada modelo da CV(Dataset SISSA).



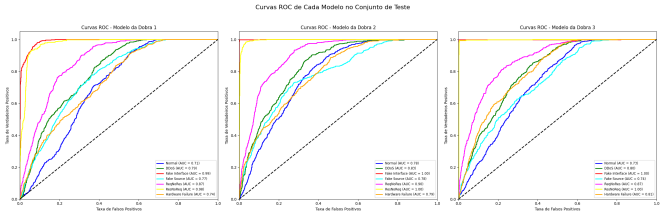
(b) Matrizes de Confusão do modelo tunado nos Dados de Validação (Dataset SISSA).



(c) Matrizes de Confusão de cada modelo da CV no Conjunto de Teste Final (Dataset SISSA).



(d) Curvas ROC e AUC do modelo tunado nos Dados de Validação (Dataset SISSA).



(e) Curvas ROC e AUC de cada modelo da CV no Conjunto de

apresentando dificuldade em distinguir classes com padrões mais sutis. A principal conclusão é que, para desafios realistas como o do dataset SISSA, uma arquitetura `SimpleRNN` simples, mesmo otimizada, não é suficiente, e que mecanismos mais avançados, como os propostos no artigo SISSA original, são provavelmente necessários para alcançar um desempenho de ponta.

b) Limitações: A principal limitação deste estudo reside na utilização exclusiva da arquitetura `SimpleRNN`. Conforme os resultados da Seção VI indicaram, esta arquitetura não foi capaz de capturar totalmente as nuances e padrões complexos do dataset SISSA, mesmo após a otimização dos hiperparâmetros. Adicionalmente, o processo de tunagem, embora útil, foi limitado a um número restrito de tentativas, e um espaço de busca mais amplo poderia, potencialmente, encontrar melhores configurações.

c) Trabalhos Futuros: Com base nas conclusões e limitações identificadas, as direções mais promissoras para trabalhos futuros são:

- A experimentação com arquiteturas de redes neurais mais potentes, como **LSTM (Long Short-Term Memory)** e **GRU (Gated Recurrent Unit)**, que são mais adequadas para capturar dependências de longo prazo em dados sequenciais e podem melhorar significativamente a performance nas classes mais desafiadoras.
- A incorporação de **mecanismos de atenção** ao modelo RNN ou LSTM, seguindo a abordagem do artigo SISSA original, para permitir que o modelo foque dinamicamente nas partes mais relevantes da sequência de pacotes.
- A aplicação da arquitetura otimizada com `LSTM` ou atenção no dataset original de Alkhatib et al. [1], para verificar se as melhorias de desempenho também são observadas no cenário de dados desbalanceado.

Referências

[1] Natasha Alkhatib, Jean-Luc Danger, and Hadi Ghauch. SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks. In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2021.

[2] Natasha Alkhatib, Maria Mushtaq, Hadi Ghauch, and Jean-Luc Danger. Here comes said: A some/ip attention-based mechanism for intrusion detection. pages 462–467, 2023.

[3] Tobias Gehrmann and Paul Duplys. Intrusion detection for some/ip: Challenges and opportunities. In *2020 23rd Euromicro Conference on Digital System Design (DSD)*, pages 583–587. IEEE, 2020.

[4] Marco Iorio, A. Buttiglieri, Massimo Reineri, Fulvio Risso, Riccardo Sisto, and Fulvio Valenza. Protecting in-vehicle services: Security-enabled some/ip middleware. *IEEE Vehicular Technology Magazine*, 15(3):77–85, 2020.

[5] Qi Liu, Xingyu Li, Ke Sun, Yufeng Li, and Yanchen Liu. SISSA: Real-time Monitoring of Hardware Functional Safety and Cybersecurity with In-vehicle SOME/IP Ethernet Traffic. *arXiv preprint arXiv:2401.03309*, 2024.

[6] Daniel Zelle, Timm Lauser, Dustin Kern, and Christoph Krauß. Analyzing and securing some/ip automotive services with formal and practical methods. pages 1–20, 2021.