INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAIBA
COORDENAÇÃO DO CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA ELÉTRICA

RAYLLE CORDEIRO DA NÓBREGA

USO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE DADOS SENSÍVEIS DE VEÍCULOS CONECTADOS

JOÃO PESSOA
2024

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Coordenação do curso superior de tecnologia em sistemas para internet

USO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE DADOS SENSÍVEIS DE VEÍCULOS CONECTADOS

Projeto Final de Curso submetido à Coordenação do Curso Superior de Bacharelado em Engenharia Elétrica do Instituto Federal da Paraíba como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica

Orientador: Patric Lacouth da Silva
Coordenador(a) do Curso: Gilvan Vieira de Andrade Junior

João Pessoa
2024

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *campus* João Pessoa

N754u Nóbrega, Raylle Cordeiro da.
Uso de aprendizado de máquina para classificação de dados sensíveis de veículos conectados / Raylle Cordeiro da Nóbrega. – 2024.
62 f. : il.

TCC (Graduação em Engenharia Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB / Coordenação de Engenharia Elétrica.
Orientador: Prof. Patric Lacouth da Silva.
.

1. Segurança de dados. 2. Algoritmo de aprendizado de máquina. 3. Redes neurais. I. Título.

CDU 004.056

Bibliotecária responsável Josinete Nóbrega de Araújo – CRB15/116

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

FOLHA DE APROVAÇÃO

RAYLLE CORDEIRO DA NÓBREGA

20172610014

"USO DE APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE DADOS SENSÍVEIS DE VEÍCULOS CONECTADOS"

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Superior de Bacharelado em Engenharia Elétrica do Instituto Federal da Paraíba, como parte dos requisitos para a obtenção do grau de Engenheira Eletricista.

Trabalho aprovado pela banca examinadora em 08 de outubro de 2024.

BANCA EXAMINADORA:

(assinaturas eletrônicas via SUAP)

Dr. Patric Lacouth da Silva

IFPB (Orientador)

Dr. Alexandre Fonseca D'Andrea

IFPB (Examinador Interno)

Dr. Lincoln Machado de Araújo

IFPB (Examinador Interno)

Documento assinado eletronicamente por:

- **Patric Lacouth da Silva**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 16/10/2024 09:45:54.
- **Alexandre Fonseca D Andrea**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 17/10/2024 08:46:37.
- **Lincoln Machado de Araujo**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 17/10/2024 13:29:00.

Este documento foi emitido pelo SUAP em 16/10/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 620531
Verificador: fbaeacf280
Código de Autenticação:

Agradecimentos

Gostaria de expressar minha gratidão a todos que estiveram ao meu lado ao longo desta jornada. Sou muito grata pelas amizades que fiz, pelas pessoas conheci e pelas oportunidades que essa graduação me proporcionou. Aos amigos, que tornaram essa trajetória mais leve, divertida e menos solitária, meu muito obrigada.

Agradeço à minha família, em especial a minha mainha, a minha mãe e aos meus tios, por todo o apoio. Também agradeço a Angelo, pela compreensão e companheirismo em todos os momentos.

Sou grata aos professores e tutores que cruzaram meu caminho e compartilharam conhecimento e experiência de forma generosa. Agradeço especialmente ao meu orientador, Patric Lacouth, por toda orientação, paciência e conselhos valiosos, que foram fundamentais para a escrita deste trabalho. Também gostaria de agradecer ao professor Alexandre D'Andrea por, desde o início da graduação, acreditar em mim e me guiar com seus conselhos.

Aos demais docentes e à equipe do time de *AVD Classification* da *Ford Motor Company*, em especial a Pablo, Uziel e Lucas, pela confiança, suporte e colaboração ao longo desta pesquisa, meu sincero agradecimento. A ajuda de todos foi essencial para o realização deste trabalho.

“Não entre em pânico“

(Douglas Adams)

Resumo

Este trabalho propõe a aplicação de algoritmos de aprendizado de máquina para a classificação de dados sensíveis em veículos conectados. Com o aumento do volume e da complexidade dessas informações, é essencial garantir a proteção adequada dos dados trafegados nos veículos. Utilizando um conjunto de dados obtido pela empresa *Ford Motor Company*, especializada no setor automotivo, a solução desenvolvida permite classificar informações com uma boa precisão, assegurando que dados de maior criticidade sejam tratados com o nível de segurança apropriado.

A implementação de um sistema de classificação automática melhora a eficiência do processo e reduz a probabilidade de erros em comparação com métodos manuais, permitindo que os modelos aprendam padrões complexos e tornem o sistema mais robusto. Assim, este trabalho destaca a importância da proteção de dados sensíveis, e o uso de modelos de aprendizado de máquina para a automatização deste processo, contribuindo para a segurança da informação e a privacidade dos usuários no setor automotivo.

Palavras-chaves: Aprendizagem de máquina. Redes Neurais. Veículos Conectados. Ciência de Dados. Segurança de Dados.

Resumo

This article proposes the application of machine learning algorithms to classify sensitive data in connected vehicles. As the volume and complexity of this information increases, it is essential to ensure that the data transmitted in vehicles is properly protected. Using a data set obtained by the company *Ford Motor Company*, which specializes in the automotive sector, the solution developed makes it possible to classify information with good accuracy, ensuring that the most critical data is treated with the appropriate level of security.

Implementing an automatic classification system improves the efficiency of the process and reduces the likelihood of errors compared to manual methods, allowing the models to learn complex patterns and making the system more robust. In other words, this work points out the importance of protecting confidential data and the use of machine learning models to automate this process, contributing to information security and user privacy in the automotive sector.

Keywords: Machine learning. Neural Networks. Connected Vehicles. Data Science. Data Security.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CVD	Connected Vehicle Data (Dados de Veículos Conectados)
DID	Diagnostic Identifier (Identificador de Dados de Diagnóstico)
CAN	Controller Area Network
UDS	Unified Diagnostic Services (Serviço de Diagnóstico Unificado)
ECU	Eletronic Control Unit (Unidade de Controle Eletrônico)
SGD	Stochastic Gradient Descent (Método do Gradiente Estocástico)
PLN	Processamento de Linguagem Natura
SAE	Sociedade de Engenheiros Automotivos
MDX	Multidimensional Expressions

Lista de Códigos

Os códigos utilizados nesse trabalho configuram propriedade intelectual da empresa *Ford Motor Company* e não podem ser compartilhados.

Sumário

1 Introdução

A Internet de Veículos (IoV) surgiu como uma das aplicações da IoT (Internet of Things) e uma evolução das Redes Veiculares Ad-hoc (VANETs). Suas características são mais abrangentes no que se refere à disponibilização de serviços, compartilhamento de dados e segurança das aplicações no contexto de Sistemas Inteligentes de Transporte (ITS -Intelligent Transportation Systems). (QUEIROZ et al., 2023)

De acordo com (PORTER; HEPPELMANN, 2015), os produtos inteligentes possuem três componentes principais, sendo eles no contexto dos veículos elétricos:

• Componentes fisicos: Referem-se as peças mecânicas e eletrônicas propriamente ditas: motor, pneu, bateria etc.

• Componentes inteligentes: Referem-se aos captores, microprocessadores, memorias, comandos, softwares, interfaces homem-maquina etc. No caso automotivo, pode-se citar como exemplos a unidade de injeção eletrônica, o sistema de frenagem ABS, o limpador de para-brisa automático acoplado ao sensor de chuva, o sistema multimídia etc.;

• Componentes de conectividade: Referem-se a componentes que permitam a comunicação com ou sem fio com o produto, como e o caso de antenas, modens, protocolos de comunicação etc.

Ainda, segundo (PORTER; HEPPELMANN, 2015), os produtos conectados são aqueles que suportam a troca de dados entre o produto, o usuário, o fabricante, outros produtos, e ainda possíveis fontes externas.

A chegada massiva dos componentes de conectividade no mundo automotivo e que tem modificado o panorama e amplificado as possibilidades para esta área. A conectividade automotiva abre também a porta para novos modelos de negócios, como: gestão inteligente de veículos comunitários e de locação, formatos inovadores de seguros, tecnologias "*big data*", entre outros. (SUGAYAMA; NEGRELLI, 2016)

Esta explosão na quantidade de informações tem elevado a importância do aprendizado a partir de dados a um patamar extremamente elevado. (JUNIOR, 2017). É cada vez maior o numero de cientistas, decisores e governantes que consideram a analise massiva de dados como uma grande alavanca para promover melhoras na qualidade de vida e proporcionar um grande crescimento econômico a sociedade. (EDWARDS, 2014)

Esses dados podem ser usados para diversas finalidades, inclusive para melhorar a segurança, reduzir o congestionamento, aprimorar a experiência de dirigir e fornecer novos serviços a motoristas e passageiros. Alguns dos dados que podem ser coletados por veículos conectados incluem informações sobre o desempenho do veículo, ambiente,

tráfego, localização, comportamento do motorista e diagnósticos veicular. (KUMAR; ZHU; DADAM, 2023). Em outras palavras, dados sensíveis também transitam na rede veicular, exigindo robustas medidas de segurança e privacidade para proteger essas informações.

O veículo conectado, inserido neste contexto, abre ainda outra porta de acesso à rede industrial, que deve, por sua vez, também receber proteção.[...] O fato de usar os meios de comunicação padrões dos veículos para a conexão destes com a indústria, pode potencializar uma vulnerabilidade adicional em relação aos meios de produção e ao produto, uma vez que a rede e protocolos de comunicação serão de conhecimento público. Esse fato exigira estratégias para inserir segurança na comunicação entre o veículo e a indústria, mesmo sendo esta transmitida por um protocolo e meios físicos conhecidos. (SUGAYAMA; NEGRELLI, 2016)

Considerando a importância da proteção dos dados contra vazamentos e ações de indivíduos mal-intencionados, assim como a grande quantidade de informações que trafegam na rede veicular, é essencial que camadas de segurança adequadas sejam aplicadas a cada tipo de dado. Exemplos notórios de vazamentos de dados, como o incidente da *Cambridge Analytica*[1], onde informações pessoais de milhões de usuários do *Facebook* foram coletadas sem consentimento para influenciar campanhas políticas, e o vazamento de dados da *Volkswagen*[2], que expôs informações sensíveis de clientes e veículos em um ataque cibernético, ressaltam as severas consequências que podem advir da exposição de dados.

Neste contexto, este trabalho estuda um sistema de classificação automática dos dados coletados nos veículos conectados, levando em conta o tipo de dado e as premissas de privacidade estabelecidas por uma equipe jurídica e pela legislação vigente no país de coleta. Para otimizar o processo e torná-lo mais eficiente, confiável e seguro, foi empregado o uso de aprendizado de máquina. Essa técnica, que permite que sistemas aprendam com dados, é fundamental para categorizar automaticamente diferentes tipos de informações, garantindo que dados sensíveis sejam gerenciados de acordo com as normas legais e regulatórias. Através da análise de padrões, o aprendizado de máquina oferece uma maneira ágil e eficaz de lidar com a crescente complexidade das informações geradas pelos veículos conectados.

[1] <https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election>
[2] <https://edition.cnn.com/2021/06/11/cars/vw-audi-hack-customer-information/index.html>

2 Fundamentação Teórica

2.1 Tipos de dados

Nos veículos, diversos tipos de dados são trocados entre os módulos, desempenhando funções como enviar solicitações de operação, notificações de serviço, status do veículo, localização, entre outros. Entretanto, neste trabalho, focaremos especificamente nos DIDs e nos sinais CAN.

2.1.1 DID (*Diagnostic Identifier*) - Identificador de Dados de Diagnóstico

O Identificador de Dados de Diagnóstico (DID) é um identificador exclusivo usado para representar um item de dados específico ou um grupo de itens de dados na Unidade de Controle Eletrônico (ECU) de um veículo. Esses itens de dados podem incluir coisas como leituras de sensores, posições de atuadores e códigos de falhas de diagnóstico (DTCs). O DID é usado para acessar e manipular esses itens de dados por meio do protocolo UDS (*Unified Diagnostic Services*), que permite a comunicação entre as ferramentas de diagnóstico e o computador de bordo do veículo. (TECH, 2023)

Como o DID se trata de um identificador de dados, ele precisa da posição de memória que ocupa dentro da ECU, sendo esse um número de identificação hexadecimal, conhecidos como *DID Number*, um identificador de sinal e a sua descrição. Conforme a ISO14229, o DID pode assumir valores diferentes de acordo com o seu uso, com isso alguns DIDs são de uso reservado para finalidades legais, de manufatura ou para uso da própria SAE (Sociedade de Engenheiros Automotivos) enquanto outros simplesmente indicam informações sobre os módulos dos veiculos que eles trafegam. Os DIDs podem possuir parâmetros internos (filhos) cujos valores podem ser diferentes do DID principal (pai). Na figura 1 é possível observar o DID 0x8012 e seus sete parâmetros.

Figura 1 – Exemplo de DID: 0x8012

Diagnostic Specification (Part II)	DID Number (Hex)	DID Name	Parameter Number	Parameter Name	Unit	Description
DS1U5T-14G371-CA	0x8012	GPS Information Threshold deactivated				
DS1U5T-14G371-CA	0x8012	GPS Information	1	Altitude	m	
DS1U5T-14G371-CA	0x8012	GPS Information	2	Map Matched Latitude	Undefined / Not Used	
DS1U5T-14G371-CA	0x8012	GPS Information Threshold	3	Map Matched Longitude	Undefined / Not Used	
DS1U5T-14G371-CA	0x8012	GPS Information	4	GPS Fix		0x01:Not available; 0x02:No fix; 0x03:2D fix; 0x04:3D fix; 0x05:Antenna not properly connected;
DS1U5T-14G371-CA	0x8012	GPS Information	5	GPS speed	m/s	
DS1U5T-14G371-CA	0x8012	GPS Information	6	Heading	Deg	
DS1U5T-14G371-CA	0x8012	GPS Information	5	GPS speed	m/s	

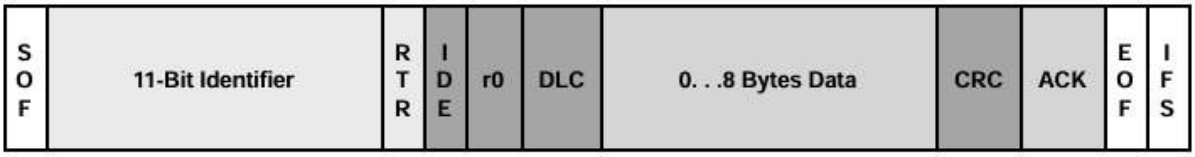
FONTE: Autoria Própria

2.1.2 Sinais CAN (*Controller Area Network)*

A CAN é um barramento de comunicação serial definido pela Organização Internacional de Padronização (ISO) originalmente desenvolvido para o setor automotivo para substituir o complexo chicote de fiação por um barramento de dois fios. A especificação exige taxas de sinalização de até 1 Mbps, alta imunidade a interferências elétricas elétrica e uma capacidade de autodiagnosticar e reparar erros de dados. Esses recursos levaram à popularidade da rede CAN em vários setores, incluindo automotivo, marítimo, médico, manufatura e aeroespacial. (HPL, 2002)

A comunicação em uma rede CAN é feita por um barramento *broadcast*, baseado em uma topologia em linha com um barramento linear, em que um numero de ECUs, ou Centrais Eletrônicas de Controle, são conectadas via interface CAN (CARVALHO; CAMPOS, 2018), seguindo o padrão abaixo para o quadro de dados padrão:

Figura 2 – Quadro de Dados CAN Padrão



FONTE: Transmissão de mensagens e gerenciamento de erros em uma rede can automotiva.

O quadro de dados do protocolo CAN pode ser subdividido em sete partes, sendo elas: campo de início do quadro de dados, campo de arbitragem, campo de controle, campo de dados, campo CRC, campo ACK e o campo de fim do *frame*. De acordo com (SOUZA; CAMPOS, 2017), essas camadas podem ser definidas da seguinte forma:

- **IDE** (*Identifier Extension*): bit identificador de extensão dominante (IDE) - especifica que uma mensagem CAN padrão (sem extensões no identificador) está sendo transmitida;
- **r0**: bit reservado para possíveis modificações futuras;
- **DLC** (D*ata Length Code*) : esse campo de comprimentos de mensagens (4 bits) indica o número de bytes de dados que está sendo transmitido;
- **Data**: é a mensagem que se deseja transmitir, podendo chegar a 8 bytes (64bits);
- **CRC** (*Cyclic Redundancy Check* ou Verificação de Redundância Cíclica): é um método para detecção de erros;
- **ACK** (*Acknowledge Error Check* ou Confirmação da Checagem de Erro): cada nó que recebe uma mensagem precisa sobrescrever esse campo na mensagem original

com um bit dominante, indicando que a mensagem recebida está livre de erro. Se um nó detectar um erro, ele descarta a mensagem e pede ao nó transmissor que a repita. Dessa forma,cada nó confirma a integridade da mensagem. ACK tem 2 bits, sendo o primeiro a confirmação ,e o segundo, um espaço para o próximo campo;

- **EOF** (*End of Frame*): esse campo de 7 bits indica o fim do quadro CAN e desabilita o preenchimento de bits, indicando um erro de preenchimento quando dominante **IFS** (*Interframe Space*): o espaçamento entre quadros contém o tempo requerido entre dois quadros

2.1.3 Categoria dos Dados

A classificação dos dados se dá de acordo com o seu nível de criticidade, em ordem crescente, respeitando as categorias abaixo:

1. **Dados Veiculares (*Vehicle Data*)** reflete informações sobre o funcionamento do veículo, como estado da trava das portas ou janelas.
2. **Dados do motorista (*Driver Data*):** reflete informações sobre o estilo de direção do motorista, como ângulo do pedal do acelerador.
3. **Dados de Geolocalização (*Geolocation*):** consiste nas informações de localização, como GPS.
4. **Identificador Indireto (*Indirect Identifier*):** reflete informações que podem levar a identificação do motorista, ainda que de forma indireta, como perfil utilizado no veículo.
5. **Identificador Direto (*Direct Identifier*):** reflete informações que levam a identificação direta do motorista, ou passageiro, como nome, telefone e afins.
6. ***Blocklist***: consiste em dados que não podem ser classificados, seja por ambiguidade do sinal ou limitação legislação. Como exemplo desse dado pode ser citado os dados de localização da China, que não podem ser coletados de acordo com a exigência legal do país.

2.2 Aprendizado de Máquina

2.2.1 PLN - Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é uma área da computação que tem como objetivo extrair representações e significados mais completos de textos livres escritos em linguagem natural. (INDURKHYA; DAMERAU, 2010)

Algoritmos de PLN, portanto, buscam gerar representações matemáticas significativas para elementos textuais, treinados a partir de conjuntos de dados (corpora) representativos do problema a ser processado. Essas representações, por sua vez, tendem a capturar características essenciais de linguagem, como morfologia, sintaxe e, em especial, semântica. (MIKOLOV et al., 2013)

2.2.2 Indicadores de Desempenho

Medir o desempenho de algo consiste em mensurar ações, onde medição é o processo de quantificar e as ações conduzem ao desempenho. Um "indicador de desempenho" pode ser definido como a métrica usada para quantificar a eficiência e/ou eficácia de uma ação (NEELY; GREGORY; PLATTS, 1999). Assim, para que os indicadores de desempenho sejam eficazes, é necessário que eles sejam escolhidos de forma criteriosa e alinhados aos objetivos estratégicos da organização ou projeto. Além disso, é importante que eles sejam mensuráveis, confiáveis e relevantes (SILVA; LIMA, 2015)

Para avaliar o desempenho do algoritmo utilizado, são observados parâmetros como acurácia, precisão, sensibilidade, *F-score* e matriz de confusão. Esses indicadores refletem a performance do algoritmo utilizado e permitem a escolha da abordagem mais adequada com base na necessidade específica de cada projeto. Portanto, é crucial considerar esses fatores durante a seleção e o desenvolvimento do modelo de aprendizado de máquina. Neste trabalho, os parâmetros de acurácia e precisão foram os mais determinantes na avaliação e escolha do modelo utilizado. De acordo com (MARIANO; XAVIER, 2021), os parâmetros em questão podem ser definidos como:

- **Acurácia (*Accuracy*)**: avalia o percentual de acertos, ou seja, ela pode ser obtida pela razão entre a quantidade de acertos e o total de entradas:

$$\text{Acurácia} = \frac{\text{Total de acertos}}{\text{Total de itens}} = \frac{VP + VN}{N} \tag{2.1}$$

- **Precisão (*Precision*)**: é uma métrica que avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Soma de todos os valores positivos}} = \frac{VP}{VP + FP} \tag{2.2}$$

- **Sensibilidade (*Recall*)**: também chamado de revocação, avalia a capacidade do método de detectar com sucesso resultados classificados como positivos.

$$\text{Sensibilidade} = \frac{\text{Verdadeiros Positivos}}{\text{Soma de Verdadeiros Positivos e Falsos Negativos}} = \frac{VP}{VP + FN} \tag{2.3}$$

- ***F-score* ou f1**: é uma média harmônica calculada com base na precisão e na revocação. Ela pode ser obtida com base na equação:

$$f1 = 2 * \frac{\text{precisão} * \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}} \tag{2.4}$$

- **Perda de Entropia Cruzada *(Cross-Entropy Loss)***: também conhecida como perda de log, é uma métrica usada no aprendizado de máquina para medir o desempenho de um modelo de classificação. Seu valor varia de 0 a 1, sendo que o menor valor é melhor. Um valor ideal seria 0. O objetivo de um otimizador encarregado de treinar um modelo de classificação com perda de entropia cruzada seria obter o modelo o mais próximo possível de 0. (GEEKSFORGEEKS, 2024)

- **Matriz de confusão (*Confusion Matrix)***: também conhecida como tabela de contingência, permite uma melhor visualização da performance do algoritmo em termos de distinguir os resultados apresentados pela classe. A representação dos resultados é presentada em uma matriz de confusão onde os valores da diagonal principal indicam as instâncias classificadas corretamente e a diagonal secundária as instâncias classificadas incorretamente (SOUZA, 2015). Essa relação pode ser observada na tabela 1, onde as células em verde representam as predições corretas e as em vermelho indicam as predições incorretas.

Tabela 1 – Matriz de Confusão

		Valor Predito	
		A	**B**
Real	**A**	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	B	Falso Positivo (FP)	Verdadeiro Negativo (VN)

FONTE: Autoria própria.

Conforme (FERRARI; SILVA, 2017), em problemas de classificação binária, predições podem ter quatro possíveis classes:

- **Verdadeiro positivo (VP):** quando o método diz que a classe é positiva e, ao verificar a resposta, vê-se que a classe era realmente positiva;

- **Verdadeiro negativo (VN):** quando o método diz que a classe é negativa e, ao verificar a resposta, vê-se que a classe era realmente negativa;

- **Falso positivo (FP):** quando o método diz que a classe é positiva, mas ao verificar a resposta, vê-se que a classe era negativa;

- **Falso negativo (FN):** quando o método diz que a classe é negativa, mas ao verificar a resposta, vê-se que a classe era positiva;

2.2.3 Algoritmos Utilizados

2.2.3.1 Classificador Bayesiano Simples (Naive-Bayes)

O Naive Bayes faz parte de uma família de algoritmos de aprendizado generativo, o que significa que ele busca modelar a distribuição de inputs de uma determinada classe ou categoria. Ao contrário de classificadores discriminativos, como regressão logística, ele não aprende quais características são mais importantes para diferenciar entre classes. (IBM, 2023b)

Não existe apenas um tipo de classificador Naïve Bayes. Os tipos mais populares diferem com base nas distribuições dos valores dos recursos, o tipo utilizado nesse projeto foi o Multinomial, este tipo de classificador Naïve Bayes assume que os recursos são de distribuições multinomiais. Essa variante é útil ao utilizar dados discretos, como contagens de frequência, e é geralmente aplicada em casos de uso de processamento de linguagem natural, como classificação de spam. (IBM, 2023b)

2.2.3.2 Método do Gradiente Estocástico - *Stochastic Gradient Descent* (SGD)

O método do gradiente estocástico é uma variante do método de gradiente descendente que nada mais é que um algoritmo comumente utilizando para treinar modelos de aprendizagem de maquina e redes neurais, o qual treina minimizando erros entre resultados previstos e reais. (IBM, 2024a)

Os dados de treinamento ajudam esses modelos a aprender ao longo do tempo, o método gradiente descendente atua como um barômetro, medindo sua precisão a cada iteração de atualizações de parâmetros. Até que a função esteja próxima ou igual a zero o modelo continuará a ajustar seus parâmetros para produzir o menor erro possível. (IBM, 2024a)

De forma simplificada o algoritmo é utilizado para encontrar o mínimo de uma função, geralmente a função de custo, que mensura o quão bem o modelo se ajusta aos dados.No método estocástico cada iteração o algoritmo utiliza apenas uma amostra aleatória dos dados para calcular o gradiente ao invés de todo o conjunto de dados.

Já o método de gradiente estocástico (SDG) combina os dois conceitos supracitados utilizando uma amostra aleatória a cada iteração para atualizar os parâmetros na direção oposta ao gradiente. Sendo este uma variação do método gradiente descendente abordando a ineficiência computacional do método tradicional em especial em grandes conjuntos de dados onde o custo computacional por iteração é reduzido uma vez que ele não exige o processamento de todo conjunto de dados como os demais desta forma aumentando significativamente a velocidade e precisão computacional.(MISHRA, 2023)

2.2.3.3 Regressão Logística

A regressão logística é uma técnica de análise de dados que usa matemática para encontrar as relações entre dois fatores de dados. Em seguida, essa relação é usada para prever o valor de um desses fatores com base no outro. A previsão geralmente tem um número finito de resultados, como sim ou não.(Amazon Web Services, 2023)

A regressão logística estima a probabilidade de um evento ocorrer como o votar ou não baseando-se em um determinado conjunto de dados de variáveis independentes. [...] Este tipo de modelo estatístico (também conhecido como modelo logit) é frequentemente usado para classificação e análise preditiva. Como o resultado é uma probabilidade, a variável dependente é limitada entre 0 e 1. Na regressão logística, uma transformação logit é aplicada nas probabilidades — ou seja, a probabilidade de sucesso dividida pela probabilidade de falha. Isso também é comumente conhecido como log odds, ou logaritmo natural das probabilidades.(IBM, 2024b)

Em resumo, a regressão logística é uma técnica estatística que prevê a probabilidade de uma amostra pertencer a uma classe específica, facilitando a análise e a tomada de decisões

2.2.3.4 Rede Neural Convolucional - *Convolutional Neural Network* (CNN)

Redes Neurais Convolucionais (CNN) consiste em um subconjunto do aprendizado de máquina utilizadas com mais frequência para tarefas de classificação e visão computacional. Cada nó conecta-se a outro e tem peso e um limite associados. Se a saída de qualquer nó individual estiver acima do valor de limiar especificado, esse nó será ativado, enviando dados para a próxima camada da rede. Caso contrário, nenhum dado será passado para a próxima camada da rede (IBM, 2023a).

Embora tenham sido inicialmente desenvolvidas para o processamento de imagens, as Redes Neurais Convolucionais também tem se mostrado eficazes na classificação de texto conforme (CAMACHO, 2019), a classificação de texto será realizada após a tokenização e tratamento das palavras para que se convertam em vetores, uma vez que as camadas convolucionais operam de maneira espacial. Os *kernels* convolucionais, então, atuam sobre esses vetores, capturando características e padrões nas sequências de palavras, assim, permitindo a classificação textual.

2.3 Classificação dos Dados

A classificação dos dados é essencial para garantir a proteção das informações transmitidas, assegurando que os níveis de segurança adequados sejam aplicados a cada tipo de dado. Isso é crucial para evitar o vazamento de informações sensíveis do motorista

e demais ocupantes do veículo, como senhas, endereços e outros dados pessoais.

A classificação de texto pode ser executada de diversas formas. Ela pode usar da técnica *Bag of Words, Naive Bayes*, Aprendizado discriminativo, Regressão Logística e Redes Neurais, por exemplo (GARCIA; RAMOS, 2023). De acordo com (EISENSTEIN, 2018), o uso de redes neurais é a abordagem predominante para a classificação não linear no processamento de linguagem natural hoje.

3 Metodologia

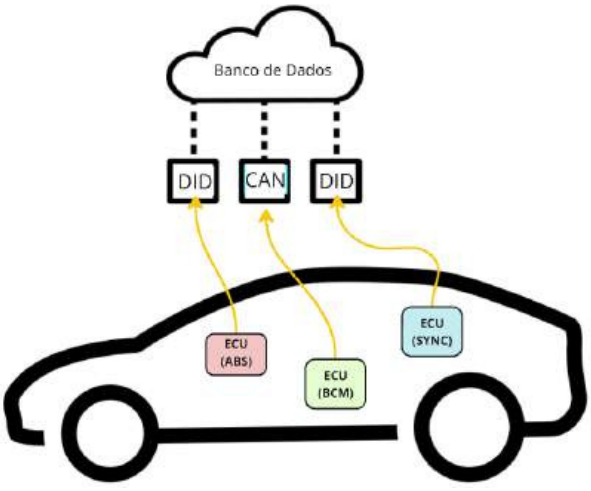
Os dados utilizados nesse estudo foram cedidos pela empresa automotiva *Ford Motor Company* através da equipe de *AVD Classification*.

3.1 Coleta e Acesso aos Dados

É fundamental ressaltar que os Dados dos Veículos Conectados (CVD) só podem ser coletados com a autorização do usuário para compartilhamento com a montadora. Além disso, é importante destacar que, independentemente da autorização, a coleta e o uso interno de qualquer sinal—seja ele DID, CAN ou outro—só podem ocorrer se sua categoria for previamente conhecida. Assim, a classificação dos dados deve ocorrer antes da coleta.

Outro ponto a ser considerado é que os dados analisados para essa classificação consistem em definições, e não nas informações reais coletadas dos veículos. Os dados que circulam entre as *ECUs*, ou seja, entre os diversos módulos do veículo, podem ser coletados e armazenados em um banco de dados hospedado na *Ford Cloud*, conforme a figura 3.

Figura 3 – Coleta de Dados Veicular



FONTE: Autoria própria.

Ao criar um sinal, como DID ou CAN, é necessário atribuir a ele uma série de informações específicas, como um identificador único, um nome, uma descrição e os possíveis valores que o sinal pode assumir. Esses dados estão contidos nos arquivos de Especificação de Diagnóstico, ou *Part II Spec*, de cada ECU (Unidade de Controle Eletrônico) do veículo, e são essenciais para a classificação.

Desta forma, algoritmo de classificação se conecta ao servidor de dados utilizando a biblioteca *Pyodbc* e inicia a montagem das tabelas com as informações necessárias para a classificação, variando conforme o tipo de sinal a ser tratado. Essas informações são obtidas a partir de arquivos MDX, contendo as especificações técnicas (Part II Spec) de cada módulo, e convertidas para tabelas armazenadas no *Hadoop*. No final, é tida uma tabela para a classificação de DIDs e outra para os sinais CAN, como visto nas figuras 4 e 5. Informações como nome do sinal, sua descrição e, caso tenha, unidade são utilizadas para a classificação dos mesmos.

Figura 4 – Tabela de DIDs para classificação

	DID ID	DID Number (Hex)	DID Name	Parameter Number	Parameter Name	Unit	Detail
0	841241	0xDE00	Config Block DE00	1	Smart DSP		Indicates that a Smart DSP node is available i...
1	841241	0xDE00	Config Block DE00	2	AAM		Indicates that an AAM node is available in the...
2	841241	0xDE00	Config Block DE00	3	SDARS		Indicates that an SDARS node is available in t...
3	841241	0xDE00	Config Block DE00	4	RSEM		Indicates that an RSEM node is available in th...
4	841241	0xDE00	Config Block DE00	5	PDC HMI		Indicates PDC HMI is On/Off Off : 0x00 On :...
...	...	...	...	...	...	...	...
850520	1619122	0xFD02	EEPROM Data	8	Block_8		
850521	1619122	0xFD02	EEPROM Data	9	Block_9		
850522	1619122	0xFD02	EEPROM Data	10	Block_10		
850523	1619122	0xFD02	EEPROM Data	11	Block_11		
850524	1619122	0xFD02	EEPROM Data				

850525 rows × 7 columns

FONTE: Autoria própria.

Figura 5 – Tabela de sinais CAN para classificação

	CAN ID	Signal Name	Detail	Unit
0	13996	PhonCallZone6_D_Rq	Signal requesting to switch to active phone ca...	
1	13020	RgstrVertObr_An_Actl	Aim Status of Righthand Outboard Register Vert...	degrees
2	14315	VehAudioMdeCtl_D_Stat2	The Digital Signal Processeing Amplifier modul...	
3	12718	PersDgtlKey2_D_Stat	Index number of the digital key associated to ...	
4	13956	MrorAutoSavDrv_D_Stat	Signal to indicate that changes have occurred ...	
...	...	...	...	...
140	13980	OvrhdLampR2Left_B_Rq	Request for Second Row Left Lamp On/OFF	
141	14634	SeatMemPosDrv_D_Stat	Memory position for Driver seat	
142	15214	VehElEffAvg_No2_Dsply	Average Watt hours per km based on vehicle his...	watt*hour / kilometer
143	12796	TsrChime_D_Rq	Traffic Sign Recognition Chime	
144	13614	VdsSftyDrv_D_Rq	The safety driver to take over the driving task.	

FONTE: Autoria própria.

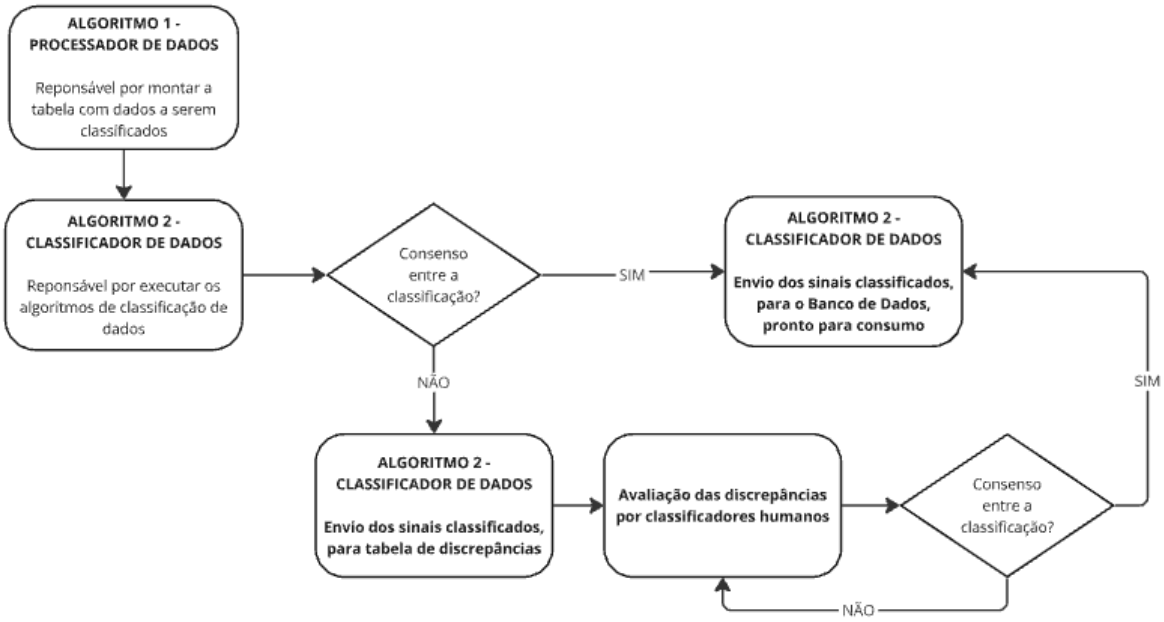
3.2 Algoritmo de Classificação

A classificação dos dados era inicialmente feita de forma manual por uma equipe de quatro engenheiros especializados. Essa abordagem envolvia uma análise aprofundada de

cada sinal, com base em documentos, experiência técnica e discussões entre os avaliadores. No entanto, a subjetividade nas interpretações, a necessidade de consenso e o volume crescente de dados tornavam o método lento e propenso a erros e redundâncias. Para otimizar essa tarefa e garantir maior consistência na classificação, foi decidida a implementação de algoritmos de aprendizado de máquina. A base de dados utilizada para treinar e testar esses algoritmos foi construída a partir dos dados classificados manualmente, servindo como referência para o desenvolvimento dos modelos.

A introdução da classificação automática de dados, incluindo DIDs e sinais CAN, trouxe melhorias significativas. Esse processo, descrito na figura 6, é realizado semanalmente utilizando dois algoritmos distintos: o primeiro, denominado Processador de Dados, é responsável por organizar a tabela a ser classificada. Já o segundo, Classificador de Dados, efetua a classificação dos sinais e os armazena no banco de dados, facilitando o acesso e a utilização das informações.Atualmente, o sistema de classificação automática utiliza três modelos independentes que avaliam o mesmo sinal e determinam uma classificação final apenas se houver consenso entre eles. Caso contrário, o dado é encaminhado para uma tabela de discrepâncias, onde é revisado e classificado manualmente por um classificador humano, sendo utilizado no próximo ciclo de aprendizado dos modelos utilizados. Essa abordagem reduz erros e aumenta a precisão da classificação, melhorando a eficiência geral do processo.

Figura 6 – Fluxo de Execução para Classificação Automática dos Dados



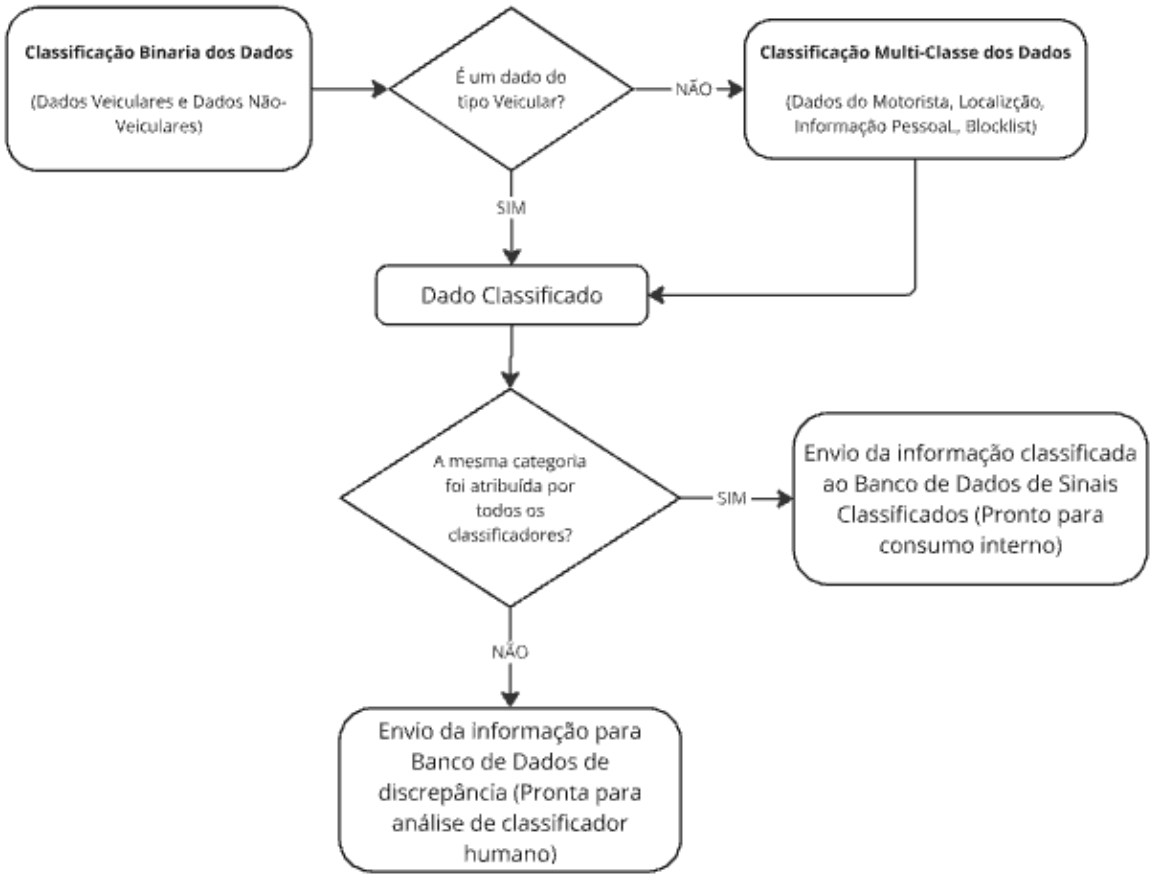
FONTE: Autoria própria.

Os modelos utilizados incluem uma Rede Neural Convolucional (*CNN*) e dois modelos de aprendizado de máquina. A estratégia empregada consiste em uma classificação

binária inicial entre dados veiculares e não veiculares, seguida por uma multi-classificação, totalizando três modelos para a classificação binária e três para a classificação multi-classe. Esses modelos têm sido essenciais para aprimorar a precisão e a consistência da classificação, superando as limitações do processo manual anterior.

Para a classificação foi adotada uma Rede Neural Convolucional empregando as bibliotecas *Keras* e *TensorFlow*. Como os dados são categorizados de acordo com a sua criticidade, sendo os dados não veiculares mais críticos, a classificação é dividida em duas etapas. Na primeira etapa temos uma classificação binária que separa os dados veiculares do não veiculares e uma segunda etapa de multi-classificação, que entre os dados não veiculares os classifica nas demais categorias. Para validar a categorização, foi desenvolvido um algoritmo em *Python*, denominado internamente pela equipe de Regras. Devido à baixa flexibilidade desse algoritmo em relação ao aumento dos dados, a validação passou a ser realizada por meio de dois modelos de aprendizado de máquina: um conjunto de dois modelos para a classificação binária e outro conjunto de dois modelos para a classificação multi categoria, sendo eles. (*Naive Bayes*, SDG e Regressão Linear). A classificação é considerada correta quando há concordância entre pelo menos dois dos três algoritmos utilizados. Nesse cenário, a classificação é registrada no banco de sinais classificados e disponibilizada para consumo interno. Se ocorrer uma divergência, o sinal é encaminhado para a tabela de divergências e será revisado e classificado manualmente por um engenheiro capacitado. Esse processo é descrito na figura 7

Figura 7 – Estratégia para Classificação Automática dos Dados

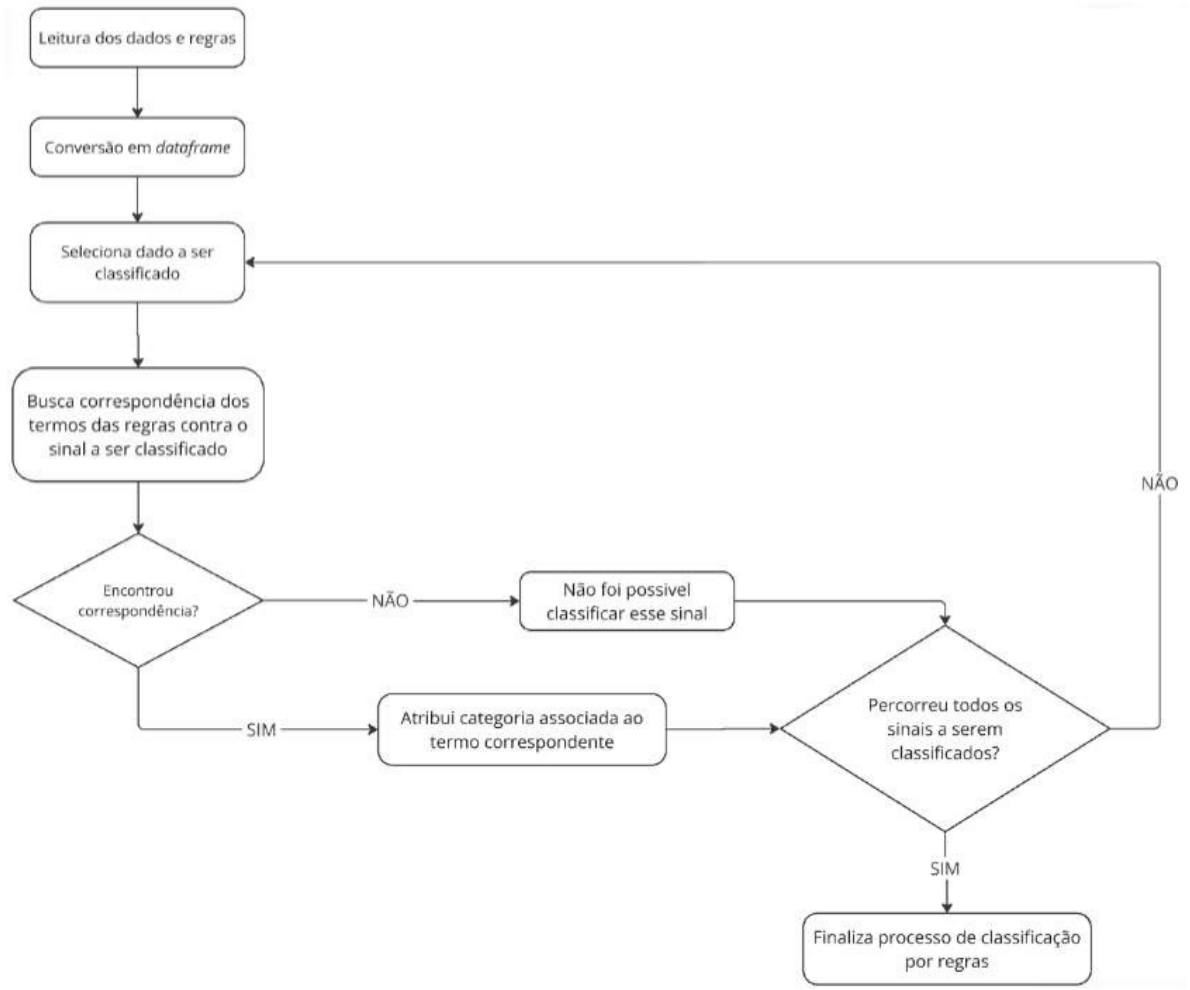


FONTE: Autoria própria.

3.2.1 Algoritmo de Regras

A primeira forma de classificação para os DIDs foi através de um *script* que buscava por correspondências de termos em uma tabela de regras pré-definidas, seguindo a estratégia observada na figura 8. Nesse contexto, a abordagem baseada em regras foi combinada com a classificação realizada por uma Rede Neural Convolucional (CNN), proporcionando uma validação adicional e aumentando a confiança nas classificações obtidas.

Figura 8 – Fluxo de Classificação por Regras



FONTE: Autoria própria.

No entanto, as Regras apresentam algumas limitações, como rigidez em relação a erros de escrita e abreviações desconhecidas, o que pode levar a previsões incorretas. Além disso, a classificação é influenciada pela ordem em que cada regra aparece no arquivo CSV, assim, as regras correspondentes as categorias mais criticas precisam estar no inicio da tabela, que pode ser vista na figura 9. A manutenção e a atualização desse arquivo também se mostrou desafiadora devido ao aumento contínuo dos dados a serem classificados, o que favoreceu a adoção de modelos de aprendizado de máquina como alternativa para validação dos dados.

Figura 9 – Arquivo de Regras

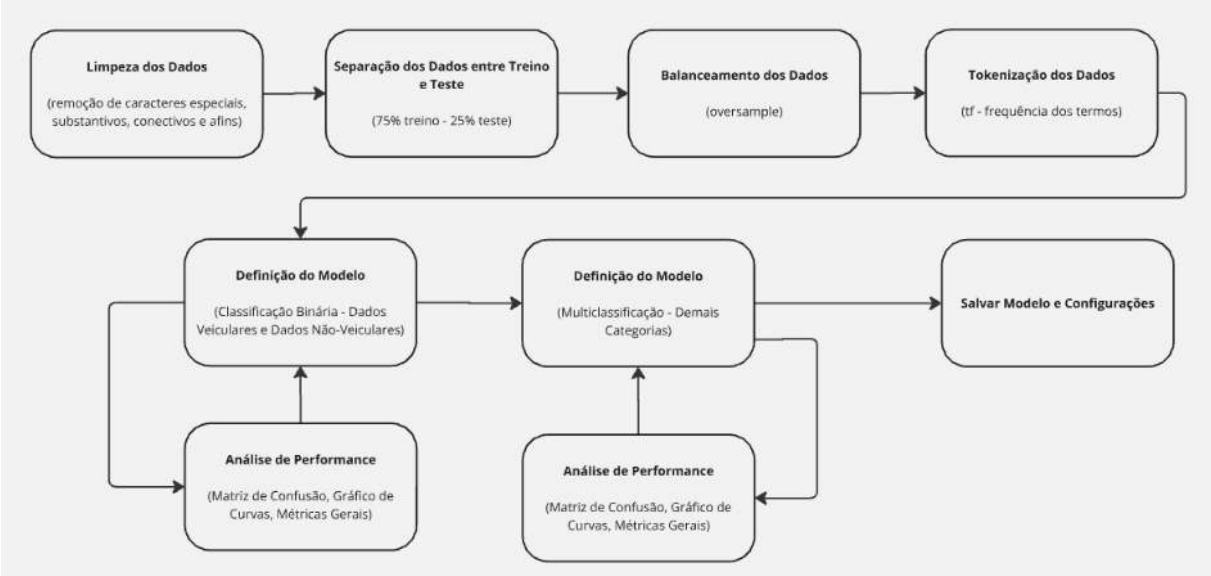
	A	B
1	Keywords	Classification
2	Inhibit	Driver Data
3	Blocked	Driver Data
4	Aborted	Driver Data
5	Fail	Driver Data
6	Deactivated	Driver Data
7	Commanded	Driver Data
8	Latitude	Geolocation
9	Longitude	Geolocation
10	Driver Application	Driver Data
11	Driver Selectio	Driver Data
12	Driver Release	Driver Data
13	Driver Actions	Driver Data
14	Driver Initiated	Driver Data
15	Pedal Application	Driver Data
16	Select	Driver Data
17	Opened	Driver Data
18	Depress	Driver Data
19	Fasten	Driver Data
20	Change of Mind	Driver Data
21	Pressed	Driver Data
22	Push	Driver Data
23	Over Speed	Driver Data
24	Overspeed	Driver Data
25	Over_Speed	Driver Data
26	Push	Driver Data
27	Brake Pedal	Driver Data

FONTE: Autoria própria.

3.2.2 Treinamento dos Modelos de Aprendizado Computacional

O modelo de aprendizado de máquina utiliza como dados de treino um *dataset* com definições reais dos dados classificados por um time de humanos. O primeiro passo para o treinamento é a limpeza dos dados, onde todas as palavras são mudadas para caixa baixa, conectivos e substantivos são retirado e as palavras, no caso da CNN, são *tokenizadas*. Os dados são balanceados, já que devido a natureza dos Veículos Conectados, a maioria das informações trafegadas consiste em Dados Veiculares. Esse processo respeita os passos da figura 10.

Figura 10 – Fluxo de treino para os modelos de Aprendizagem de Máquina



FONTE: Autoria própria.

Para a CNN, é necessário definir alguns parâmetros de configuração, como o tamanho da camada densa de entrada, taxa de aprendizado, número de amostras (*batch*), número de épocas ou *epochs*. Durante o treinamento do modelo, ocorre a intercalação entre camadas densas, com um certo número de neurônios havendo a distribuição de probabilidade sobre as classes, e de *dropout*, que desativa aleatoriamente uma fração dos neurônios durante o treinamento para evitar *overfitting*. O otimizador Adam (*Adaptive Moment Estimation*) também é utilizado para ajustar a taxa de aprendizado para cada parâmetro.

Como mencionado anteriormente, o processo de treinamento é o mesmo para os DIDs e sinais CAN, contudo, devido às peculiaridades de cada tipo de sinal, os modelos apresentam algumas diferenças, com adaptações específicas que visam maximizar a eficácia de cada um.

3.2.3 DIDs

Devido à possibilidade de um mesmo DID (Identificador de Dados de Diagnóstico) estar presente em mais de uma *Part II Spec*, torna-se imprescindível a remoção de duplicatas durante o pré-processamento dos dados. A tabela 2 mostra a relação entre a quantidade de DIDs para cada categoria, sendo 95% dos dados pertencentes a classe veicular e 4,27% pertencendo as demais categorias, representando uma proporção de 1:22 entre os dados. Desta forma, é necessário realizar o balanceamento dos dados para poder iniciar-se o treino.

Tabela 2 – Proporção entre os dados para DIDs (Pré balanceamento)

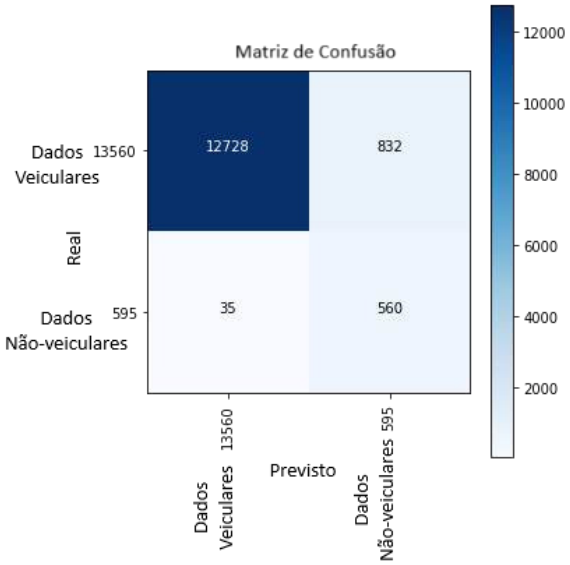
Categoria	**Quantidade (und)**	**Quantidade (%)**
Dados Veiculares (0)	54606	95,73
Dados do Motorista (1)	1853	3.25
Geolocalização (2)	24	0.04
Identificador Indireto (3)	335	0.59
Identificador Direto (4)	6	0.01
Identificador Direto Elevado (5)	4	0.01
Blocklist (6)	223	0.39

FONTE: Autoria própria.

Conforme mencionado, um DID pode conter parâmetros internos (DIDs filhos), cuja classificação determina a categoria do DID principal. Assim, cada parâmetro é classificado individualmente, e a categoria mais crítica atribuída a esses parâmetros define a categoria do DID principal. Para a tarefa de classificação, foram avaliados diversos algoritmos de aprendizado de máquina, como Rede Neural Convulacional (CNN), *Random Forest, SGD* e *Naive Bayes*.

Desta forma, para avaliar o desempenho dos algoritmos classificação dos DIDs, utilizamos uma combinação de métricas de desempenho e análises gráficas. A seguir, são apresentados os principais resultados de treinamento, inciando pela CNN:

Figura 11 – Matriz de Confusão (CNN) - DIDs



FONTE: Autoria própria.

Na figura 11, a matriz de confusão mostra a distribuição de previsões corretas e incorretas. Observa-se que o modelo tem um bom desempenho ao identificar os dados veiculares, com poucos erros de classificação, classificando 35 dos 560 sinais não-veiculares com uma categoria menor que a real.

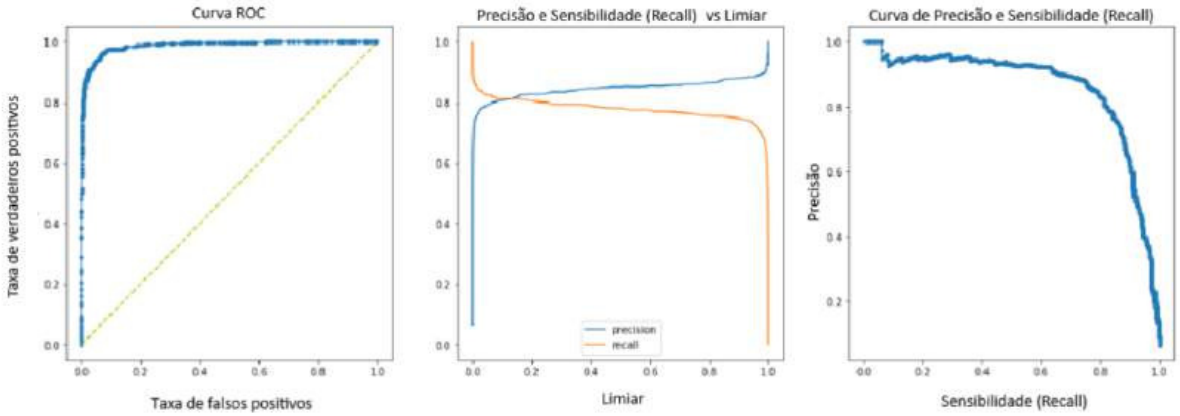
Tabela 3 – Métricas de Desempenho CNN (DIDs)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Dados veiculares (0)	0.94	1.00	0.94	0.97
Dados não-veiculares (1)		0.40	0.94	0.56

FONTE: Autoria própria.

Conforme apresentado na tabela 3, o modelo demonstrou um ótimo desempenho para a classe de dados veiculares, com precisão e *recall* (sensibilidade) muito elevados. A detecção de dados não-veiculares também foi satisfatória, com um *recall* alto, indicando que o modelo conseguiu identificar bem esses exemplos. No entanto, a precisão para essa classe foi baixa, resultando em muitos falsos positivos e, consequentemente, um *F1-score* inferior. Outro ponto relevante é o desequilíbrio entre as classes, uma vez que o volume de dados veiculares trafegado é significativamente maior que o de dados não-veiculares entre os módulos dos veículos conectados. Por essa razão, a acurácia foi considerada secundária, já que pode ser uma métrica enganosa em cenários de desbalanceamento de classes.

Figura 12 – Curvas CNN - DID



FONTE: Autoria própria.

Na figura 12, a curva ROC mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para diferentes limiares de decisão. A área sob a curva (AUC) reflete a habilidade do modelo em distinguir entre as classes, com um desempenho global de 0.987 para a classificação binária.

Tabela 4 – Métricas de Desempenho Classificação Binária - Naive Bayes (DIDs

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Dados Veiculares	0.95	0.97	0.94	0.95
Dados Não-Veiculares		0.93	0.96	0.94

FONTE: Autoria própria.

O modelo Naive Bayes apresenta um desempenho robusto na classificação entre dados veiculares e dados não-veiculares, como visto na tabela 4, com uma acurácia de 95% em ambas as categorias. A precisão é alta, com 97% para dados veiculares e 93% para dados não-veiculares, indicando uma baixa taxa de falsos positivos. O *recall* também é satisfatório, com 94% para dados veiculares e 96% para dados não-veiculares, mostrando que a maioria dos casos relevantes está sendo identificada. O F1-Score, que reflete um equilíbrio entre precisão e *recall*, é de 0.95 para dados veiculares e 0.94 para dados não-veiculares. Em resumo, o modelo é eficaz e confiável para a tarefa de classificação, embora ajustes possam ser necessários ao longo do tempo.

Tabela 5 – Métricas de Desempenho Classificação Binária - SGD (DIDs)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Dados Veiculares	0.94	0.94	0.99	0.97
Dados Não-Veiculares		0.94	0.46	0.61

FONTE: Autoria própria.

Como pode ser visto na tabela 5, o O desempenho da implementação em SGD foi satisfatória ao identificar Dados Veiculares, com precisão (94,72%) e *recall* (99,74%),

resultando em um F1-score de 97,17%. No entanto, seu desempenho em identificar Dados Não-Veiculares é inferior, com *recall* de 46,05%, sugerindo que ele tem dificuldade em reconhecer corretamente essa classe, apesar de manter uma boa precisão de 94,72%.

Para a multiclassificação, foi treinado um modelo de Rede Neural Convulacional, um de *SGD* e *Random Forest*. O desempenho da CNN pode ser visto na tabela 6

Tabela 6 – Métricas de Desempenho - Multiclassificação CNN (DIDs)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Identificador Indireto	0.97	0.988	0.988	0.988
Dados do Motorista		0.999	0.999	0.999
Identificador Direto		0.500	0.500	0.500
Geolocalização		0.833	0.833	0.833
Blocklist		0.997	0.997	0.997

FONTE: Autoria própria.

Tabela 7 – Métricas de Desempenho para Multiclassifição - SGD (DIDs)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Identificador Indireto	0.98	0.98	0.98	0.98
Dados do Motorista		0.99	0.99	0.99
Identificador Direto		0.97	0.97	0.97
Blocklist		0.90	0.90	0.90

FONTE: Autoria própria.

Já o SGD é um algoritmo de otimização que ajusta os pesos do modelo de forma iterativa, utilizando um subconjunto aleatório dos dados a cada iteração. Isso permite uma atualização mais rápida e eficiente, especialmente em grandes conjuntos de dados. No contexto deste modelo, e analisando a tabela 7, o modelo é capaz de encontrar um ótimo local na função de perda. As métricas de precisão e *recall* também refletem a eficácia do modelo, com uma precisão de 0.97 para dados veiculares e 0.93 para dados não-veiculares, sugerindo sua eficácia em minimizar os falsos positivos, ao mesmo tempo que mantém uma boa taxa de identificação correta dos casos relevantes.

Tabela 8 – Métricas de Desempenho para Multiclassifição - *Random Forest* (DIDs)

Tipo de Dados	**Acurácia**	**Precisão**	***Recall***	***F1-Score***
Identificador Indireto	0.97	0.95	0.92	0.94
Dados do Motorista		0.99	0.98	0.98
Identificador Direto		0.72	0.88	0.80
Geolocalização		0.60	0.42	0.50
Blocklist		0.91	0.95	0.93

O modelo de *Random Forest* apresenta alta acurácia geral de 97,65%, com bom desempenho, ilustrado na tabela 8 em identificar Dados do Motorista e *Blocklists*, com

F1-scores acima de 90%. Contudo, o desempenho em classes como Geolocalização e Identificador Elevado Direto é inferior, sendo que a última não foi identificada corretamente, resultando em um F1-score de 0%, o que se torna preocupante dado a criticidade dessa categoria.

3.2.4 Sinais CAN

Figura 13 – Can Signals Dataset.

	ID	Signal Name	Detail	Unit	AVD Category
0	3	EngAoutAntiShuf_Tq_Rq	Anti-shuffle torque offset request. Add this o...	newton*meter	Vehicle Data
1	7	GboxTotN_Rt_Actl	Actual gearbox speed ratio (including gear and...	unitless	Vehicle Data
2	11	Eng_Tq_FflAdd	Feed forward additive torque request from Feed...	newton*meter	Vehicle Data
3	12	EngMde_D_Rq	Requested engine mode that includes:\n0 = Off\...		Vehicle Data
4	13	EngAout_N_Rq	Target engine speed from VSC.\nThis engine spe...	rpm	Vehicle Data
...	...	...	...	...	...
9352	14844	HcmPersSrvc_D_Res	Response for personalization profile setting S...		Driver Data
9353	14854	FogLghtRearButtn_B_Rq	Request signal to On/Off Rear Fog Lights throu...		Driver Data
9354	14855	FogLghtFrontButtn_B_Rq	Request signal to On/Off Front Fog Lights thro...		Driver Data
9372	14974	TrlrIdActv_No_Rq	This signal is a request to activate trailer I...	unitless	Vehicle Data
9373	14994	TrlrCamraMsgTxt_D_Rq	Message text signal used for displaying Traile...		Vehicle Data

8641 rows × 5 columns

FONTE: Autoria própria.

Como observado abaixo, na tabela 9, o *dataset* de sinais CAN é bem desbalanceado, havendo quase 5 vezes mais dados do tipo veiculares que dados do tipo não-veiculares. Para lidar com esse desbalanceamento, as categorias são substituídas por números, com os dados veiculares são representados por 0 e os dados não-veiculares por 1.

Tabela 9 – Distribuição dos Tipos de Dado - sinais CAN

Tipo do Dado	**Quantidade (und)**	**Quantidade (%)**
Dados Veiculares (0)	6067	83.02%
Dados Não-Veiculares (1)	1241	16.98%

FONTE: Autoria própria.

Após isso, esses dados são balanceados através de técnicas de *oversample*, resultando na proporção de 1:1 entre os dados veiculares e não-veiculares, conforme ilustrado na tabela 10.

Tabela 10 – Proporção entre dados veiculares e dados não-veiculares para sinais CAN (Pós balanceamento)

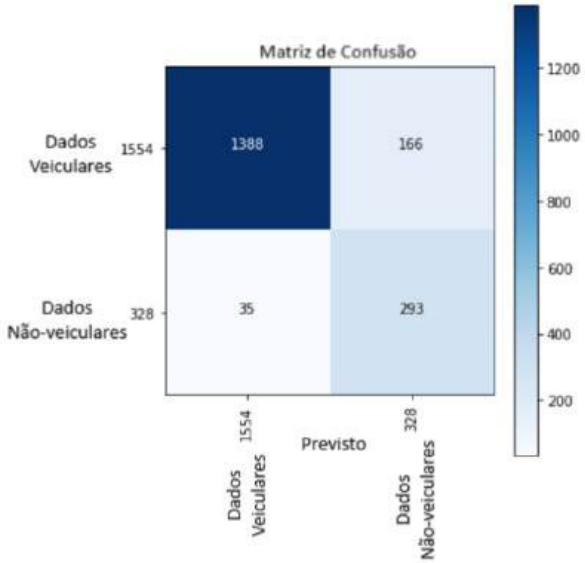
Tipo do Dado	**Quantidade (und)**	**Quantidade (%)**
Dados Veiculares (0)	4549	50%
Dados Não-Veiculares (1)	4549	50%

FONTE: Autoria própria.

Em seguida, os dados foram divididos entre treino e teste, utilizando-se a função train_test_split da biblioteca *sklearn*. Para a classificação desse sinal, foi realizado o treinamento de uma Rede Neural Convulacional (CNN) e modelos de aprendizado tais quais SGD, Naive Bayes, Regressão Logistica, XGBoost e Random Forest.

Iniciando pelo modelo de classificação binário, a CNN teve o seguinte desempenho durante o treinamento: Além disso, as curvas ROC e AUC foram empregadas para avaliar o desempenho global dos modelos, com foco principal na CNN. Abaixo, são apresentados os resultados obtidos para a CNN, seguidos de uma comparação com outros modelos testados, como Naive Bayes, SGD e Regressão Logística e outros.

Figura 14 – Matriz de Confusão (CNN) - Sinais CAN

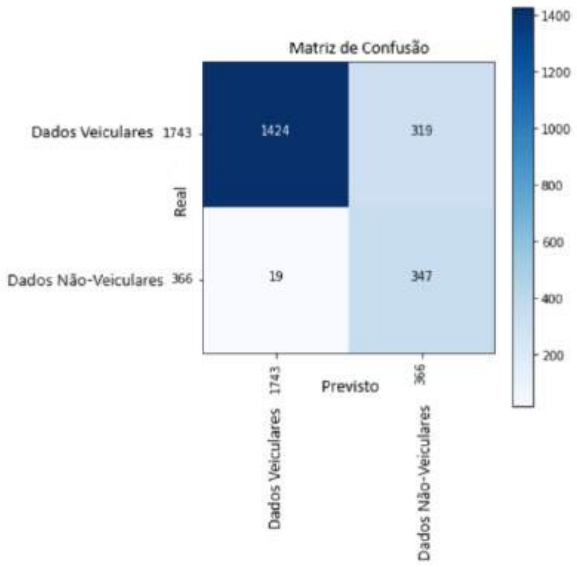


FONTE: Autoria própria.

Na figura 14, a matriz de confusão mostra a distribuição de previsões corretas e incorretas. Observa-se que o modelo tem um bom desempenho principalmente ao identificar os dados veiculares, com poucos erros de classificação.

Além da CNN, dois outros modelos foram aplicados para a classificação dos sinais CAN, sendo eles o *Naive Bayes* e o *SGD*, seguindo a mesma abordagem de realizar primeiro uma classificação binária, seguida pela classificação multiclasse para prever as diferentes categorias que os dados podem assumir. Para o primeiro estágio, os algoritmos Naive Bayes, SGD e *Random Forest* foram analisados, levando aos seguintes resultados:

Figura 15 – Matriz de Confusão para Classificação Binária - *Naive Bayes* - Sinais CAN



FONTE: Autoria própria.

O modelo de *Naive Bayes* apresentou a distribuição mostrada na matriz de confusão da figura 15. Nela, é possível observar que o algoritmo classificou corretamente a grande maioria dos dados da classe não-veicular, errando apenas 19 de 366 casos. No entanto, o modelo apresentou um número maior de erros na classificação dos dados veiculares. Considerando o contexto da aplicação, é preferível que os erros resultem na atribuição de uma categoria superior à real, em vez do oposto, a fim de evitar a subproteção desses dados.

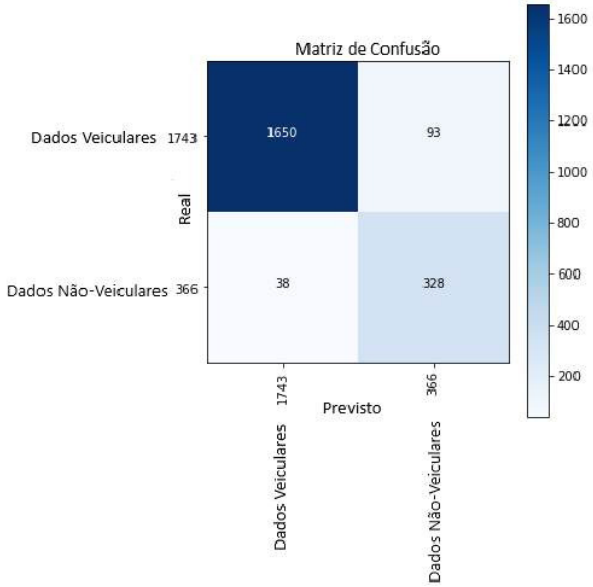
Tabela 11 – Métricas de Desempenho Classificação Binária - Naive Bayes (sinais CAN)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Dados veiculares (0)	0.84	0.99	0.82	0.89
Dados não-veiculares (1)		0.52	0.95	0.67

FONTE: Autoria própria.

O próximo modelo utilizado para a classificação binária foi o SGD, cuja matriz está representada na figura 16. Essa matriz revela uma taxa de erro relativamente baixa nas classificações, com 1.650 dados veiculares corretamente identificados e apenas 93 classificados erroneamente. Para os dados não-veiculares, 328 foram corretamente identificados, enquanto 38 apresentaram erros. Esses resultados indicam que o modelo tem uma boa capacidade de distinguir entre as duas categorias.

Figura 16 – Matriz de Confusão para Classificação Binária - SGD - Sinais CAN



FONTE: Autoria própria.

A tabela 12, indica que o modelo acertou 90% dos dados não-veiculares, um resultado crucial para evitar a subproteção, pois um alto recall ajuda a garantir que dados importantes não sejam deixados de lado. Além disso, com uma precisão de 0.98, a maioria das classificações positivas para dados veiculares foi correta, o que demonstra um bom equilíbrio no desempenho do modelo, como mostra o F1-score. Por outro lado, a precisão para dados não-veiculares é de 0.78, o que significa que, embora uma parte considerável das classificações positivas esteja correta, ainda existem falsos positivos. Isso sugere que o modelo pode confundir alguns dados não-veiculares com veiculares. Entretanto, como mencionado anteriormente, é preferível que, em caso de erro, os sinais recebam um nível maior de proteção do que o necessário, em vez do contrário.

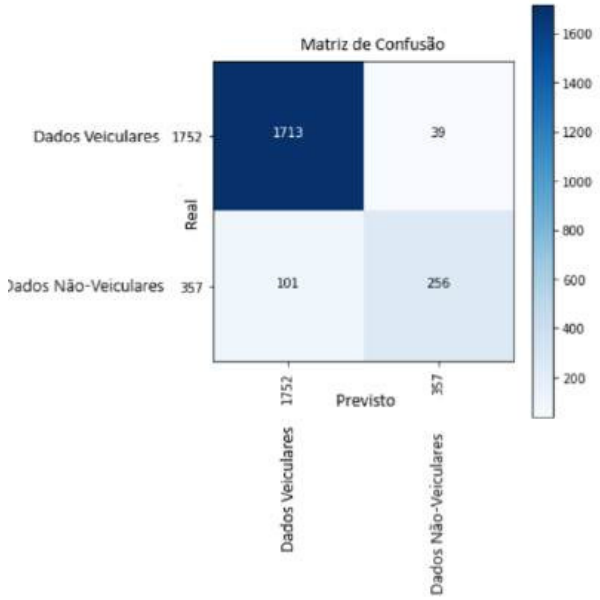
Tabela 12 – Métricas de Desempenho Classificação Binária - SGD (sinais CAN)

Tipo de Dado	Acurácia	Precisão	Recall	F1-Score
Dados veiculares (0)	0.94	0.98	0.95	0.96
Dados não-veiculares (1)		0.78	0.90	0.83

FONTE: Autoria própria.

Além dos dois modelos citados também foi realizado testes com o *Random Forest*, que consiste em uma combinação de árvores de predição, de modo que cada árvore depende dos valores de um vetor aleatório amostrado de forma independente e com a mesma distribuição para todas as árvores da floresta. (BREIMAN, 2001)

Figura 17 – Matriz de Confusão para Classificação Binária - *Random Forest* - Sinais CAN



FONTE: Autoria própria.

De acordo com a matriz de confusão observada na figura 17, o *Random Forest* teve um bom resultado na classificação de dados veiculares mas o mesmo não se estende a classificação de "Dados Não-Veiculares", apresentando uma grande taxa de erro.

Tabela 13 – Métricas de Desempenho Classificação Binária - *Random Forest* (sinais CAN)

Tipo de Dado	Acurácia	Precisão	Recall	F1-Score
Dados veiculares (0)	0.93	0.94	0.98	0.96
Dados não-veiculares (1)		0.87	0.72	0.79

FONTE: Autoria própria.

É possível observar que, nesse modelo, apesar de ter uma acurácia alta, a precisão, *recall* e *f1-score* são mais baixos para a classificação de dados não-veiculares.

Tabela 14 – Métricas de Desempenho CNN (sinais CAN)

Tipo de Dado	Acurácia	Precisão	Recall	F1-Score
Dados do Motorista	0.97	0.98	0.98	0.98
Geolocalização		0.92	0.92	0.92
Identificador Indireto		1.00	0.88	0.93
Blocklist		0.96	0.98	0.97

FONTE: Autoria própria.

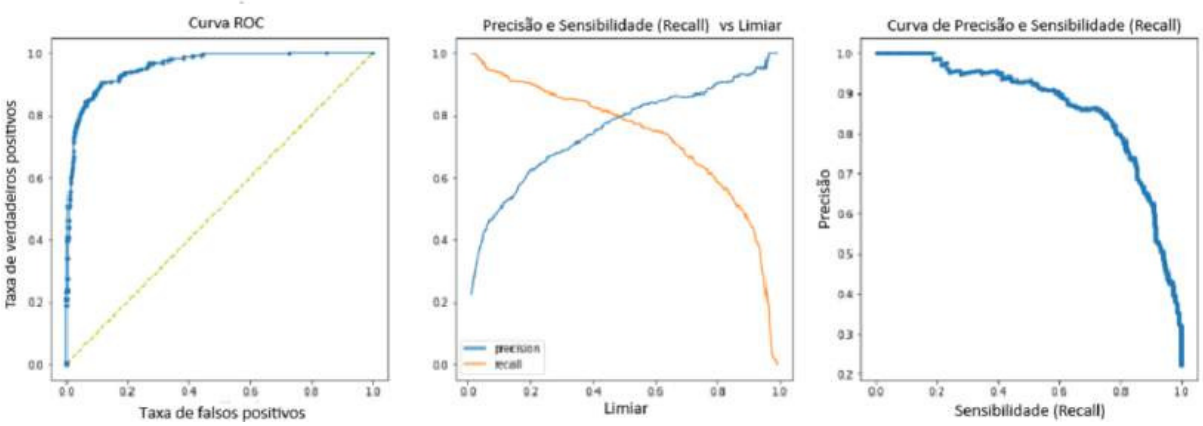
De acordo com a tabela 14, as predições da multiclassificação para a CNN alcançaram uma precisão de 0,98, indicando que a maioria das previsões feitas para os dados veiculares foram corretas. O valor de *recall* de 0.89 demonstra que o modelo conseguiu

identificar a maior parte dos exemplos reais dessa classe. Além disso, o *F1-Score* de 0,93 reflete um bom equilíbrio entre precisão e *recall*.

Semelhante ao que aconteceu na classificação dos DIDs, para os dados não veiculares, o modelo apresentou uma precisão inferior, de 0.64, sugerindo uma quantidade maior de falsos positivos. No entanto, o *recall* foi elevado (0.89), indicando que o modelo identificou a maioria dos exemplos reais dessa classe. O *F1-Score* de 0,74 evidencia o impacto da menor precisão, apesar do *recall* favorável.

Essa discrepância no desempenho pode ser atribuída ao desbalanceamento das classes, uma vez que existem mais exemplos de dados veiculares, devido à natureza dos veículos conectados, o que beneficiou o modelo nessa categoria em detrimento dos dados não veiculares. Embora a acurácia global do modelo seja alta, ela foi considerada secundária, já que, em contextos com classes desbalanceadas, métricas como precisão e *recall* fornecem uma visão mais abrangente do desempenho do modelo.

Figura 18 – Curvas CNN - Sinais CAN

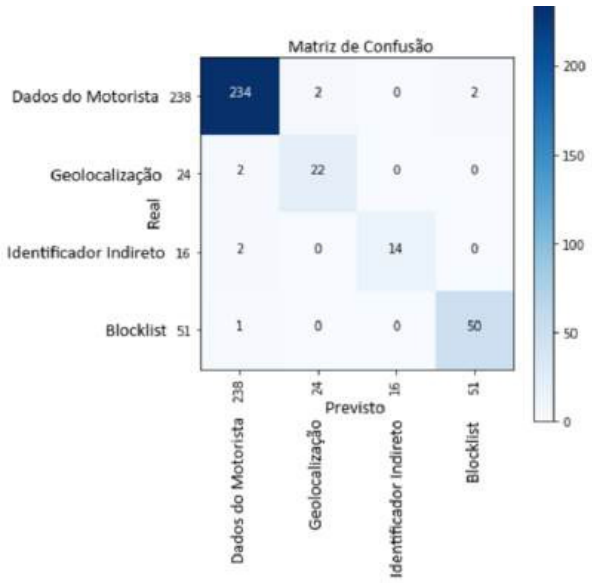


FONTE: Autoria própria.

Na figura 18, a curva ROC mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para diferentes limiares de decisão. A área sob a curva (AUC) reflete a habilidade do modelo em distinguir entre as classes, com um desempenho global de 0.958 para a classificação binária.

Após a classificação binária, o conjunto de dados não-veiculares segue para a etapa de multi-classificação sendo trabalhado os modelos de SGD, Regressão Linear e CNN. Até a implementação desses modelos, testes foram realizados utilizando os modelos citadoe outros, como *XGBoost*.

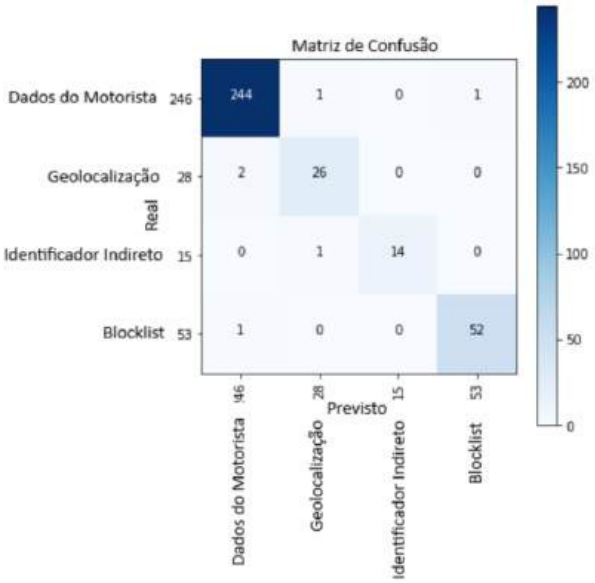
Figura 19 – Matriz de Confusão da CNN para Multiclassificação (Sinais CAN)



FONTE: Autoria própria.

A matriz de confusão evidencia um bom desempenho do modelo CNN na classificação de dados CAN, mostrando que a maioria das previsões foi correta, especialmente nas classes Dados do Motorista (234 acertos em uma amostra de 238) e *Blocklist* (50 acertos em 51 amostras). Os erros estão concentrados em classes de criticidade intermediária, e seu número é relativamente pequeno quando comparado à taxa de acertos.

Figura 20 – Matriz de Confusão Multiclasse - SGD - sinais CAN



FONTE: Autoria própria.

Na figura 20, observa-se que o modelo tem um bom desempenho ao identificar os dados não-veiculares nas categorias de Dado do Motorista, Geolocalização, Identificador

Indireto e *Blocklist*. Isso pode ser verificado pelas métricas da tabela 15, que indicam uma acurácia geral de 0.98. Para os Dados do Motorista, o modelo alcançou uma precisão de 0.99, um *recall* de 0.99 e um *F1-score* de 0.99, demonstrando alta eficácia com poucos erros na classificação. Na categoria de Geolocalização, a precisão e o *recall* foram de 0.93, refletindo um desempenho robusto. O Identificador Indireto apresentou precisão altissíma, embora o *recall* tenha sido um pouco menor, em 0.93, resultando em um *F1-score* de 0.97. Por fim, a categoria *Blocklist* mostrou resultados equilibrados entre os parâmetros analisados. Essas métricas confirmam a capacidade do modelo SGD em classificar dados não-veiculares com precisão e confiabilidade, tornando-o uma escolha adequada para a identificação desses dados.

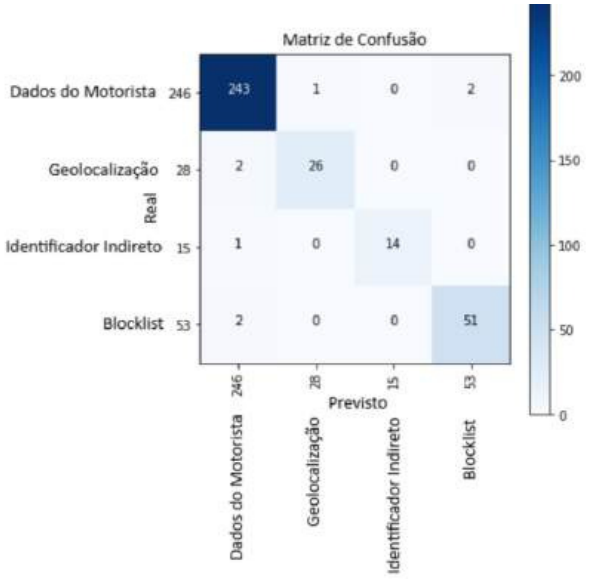
Tabela 15 – Métricas de Desempenho Multiclasse SGD (sinais CAN)

Tipo de Dado	Acurácia	Precisão	Recall	F1-Score
Dados do Motorista	0.98	0.99	0.99	0.99
Geolocalização		0.93	0.93	0.93
Identificador Indireto		1.00	0.93	0.97
Blocklist		0.98	0.98	0.98

FONTE: Autoria própria.

O algoritmo de Regressão Linear implementado para a multiclassificação dos sinais CAN, como mostrado na imagem 21, está funcionando bem em termos de precisão geral, classificando corretamente a maioria dos dados. No entanto, em algumas categorias, ele atribui uma classe com criticidade inferior à esperada. Isso foi observado principalmente nas classes *Blocklist*, Identificador Indireto e Geolocalização, que em alguns casos foram classificadas erroneamente como Dado do Motorista, uma categoria de menor criticidade.

Figura 21 – Matriz de Confusão Multiclasse - Regressão Logistica - sinais CAN



FONTE: Autoria própria.

Pode-se ter uma melhor noção do desempenho desse modelo ao se analisar a tabela 16, que apresenta as métricas de desempenho da Regressão Logística para os sinais CAN. A tabela mostra uma acurácia geral muito alta, de 0,98. As classes Dados do Motorista e Blocklist apresentam resultados bastante equilibrados, com precisão, recall e F1-Score próximos de 0,98, o que indica que o modelo classifica essas categorias de forma consistente. Para a classe Geolocalização, a precisão é de 0,96, mas o recall é um pouco menor, em 0,93, sugerindo que alguns dados dessa classe não estão sendo capturados corretamente. Identificador Indireto tem a maior precisão, 1,00, mas com um recall de 0,93, o que significa que, apesar de o modelo ser muito preciso quando classifica essa categoria, ele não consegue identificar todos os dados pertencentes a ela. Em geral, o desempenho do modelo é muito bom, com apenas algumas dificuldades em identificar completamente as classes de Geolocalização e Identificador Indireto.

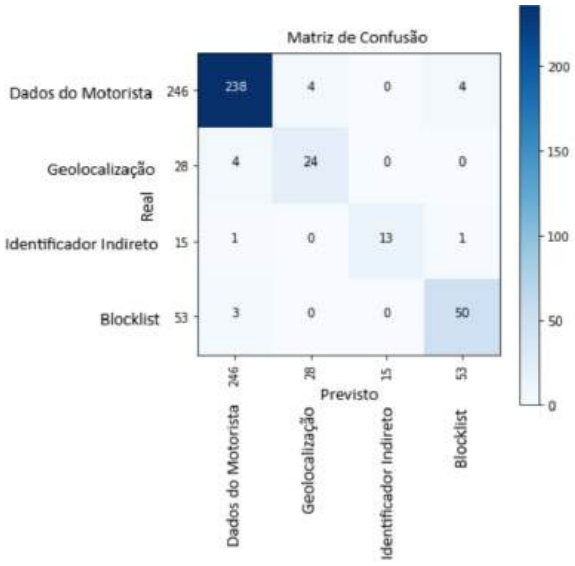
Tabela 16 – Métricas de Desempenho Multiclasse Regressão Logística (sinais CAN)

Tipo de Dado	Acurácia	Precisão	Recall	F1-Score
Dados do Motorista	0.98	0.98	0.99	0.98
Geolocalização		0.96	0.93	0.95
Identificador Indireto		1.00	0.93	0.97
Blocklist		0.96	0.98	0.98

FONTE: Autoria própria.

Outro algoritmo interessante foi considerado para a automatizar a classifição desses dados foi o XGBoost que consiste em um algoritmo de aprendizado de máquina que pertence à categoria de aprendizado de conjunto, especificamente à estrutura de aumento de gradiente. Ele utiliza árvores de decisão como aprendizes básicos e emprega técnicas de regularização para aprimorar a generalização do modelo. (VIDHYA, 2024)

Figura 22 – Matriz de Confusão Multiclasse - *XGBoost* - sinais CAN



FONTE: Autoria própria.

De acordo com a matriz da figura 22, é possível notar que apesar de ter uma boa classificação, esse modelo previu uma quantidade maior de dados incorretos que os demais.

Tabela 17 – Métricas de Desempenho Multiclasse *XGBoost* (sinais CAN)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
Dados do Motorista	0.95	0.97	0.97	0.97
Geolocalização		0.86	0.86	0.86
Identificador Indireto		1.00	0.87	0.93
Blocklist		0.91	0.94	0.93

FONTE: Autoria própria.

4 Resultados

Como resultado deste trabalho, foi implementado um sistema automatizado para a classificação de dados veiculares, trazendo benefícios como a redução do tempo de processamento e maior confiabilidade nos resultados. Os algoritmos empregados operam em conjunto, atribuindo a classificação aos dados quando há concordância entre as predições dos três modelos. Esses algoritmos se dividem em duas categorias principais: classificação binária e multiclassificação. Para a classificação de DIDs e sinais CAN, foram utilizados os modelos descritos na tabela. 18:

Tabela 18 – Modelos utilizados para classificação dos dados (DIDs e Sinais CAN)

Tipo do Sinal	**Classificação Binária**	**Multiclassificação**
DID	CNN	CNN
	Naive Bayes	Random Forest
	SGD	SGD
CAN	CNN	CNN
	SGD	SGD
	Naive Bayes	Regressão Logística

FONTE: Autoria própria.

A escolha do algoritmo considerou a precisão, sensibilidade e acurácia de cada modelo, juntamente com a análise de sua matriz de confusão e Perda de Entropia Cruzada (*Cross Entropy Loss*). Essa avaliação não apenas levou em conta as predições corretas, mas também como o modelo comete erros. Foi considerado preferível que o algoritmo classifique um dado em uma categoria de maior criticidade do que em uma categoria inferior, uma vez que isso minimiza o risco de subproteger informações sensíveis. Isso se deve ao fato de que é mais problemático que um dado receba um nível menor de proteção do que o necessário que o contrario. Já para a escolha dos parâmetros da CNN, as curvas de ROC e AUC também foram consideradas.

Tabela 19 – Métricas de Desempenho para Classificação Binária (DIDs)

Modelo	**Acurácia (%)**	***Recall (%)***	**Perda de Entropia Cruzada**	**AUC (%)**
CNN	97.32	97.33	0.04	99.6
SGD	94.72	94.72	0.17	98.9
Naive Bayes	93.33	97.33	0.43	99.3

FONTE: Autoria própria.

A tabela 19 mostra um comparativo entre os modelos escolhidos para a classificação binária dos DIDs, com isso é possível notar que cada modelo apresenta pontos fortes e fracos, que quando implementados juntos se complementam garantindo uma categorização

robusta dos dados. A CNN apresenta uma acurácia de 97,32% e um *recall* de 97,33%, sendo o modelo mais eficiente, capturando padrões complexos nos dados com alta precisão. Seu valor de *Cross Entropy Loss*, de apenas 0,04, indica que ela faz previsões muito precisas com pouquíssima incerteza. O AUC de 99,6% reflete a boa capacidade desse modelo em separar corretamente as classes.

Por outro lado, o SGD, embora tenha uma acurácia menor (94,72%) e uma *Cross Entropy Loss* maior (0,17), contribui com uma boa capacidade de discriminar entre as classes, como refletido pelo seu AUC de 98,9%. O *recall*, também de 94,72%, assegura que o modelo está corretamente identificando a maior parte dos casos positivos, reforçando a robustez da predição conjunta.

Já o Naive Bayes, apesar de ter a menor acurácia entre os três (93,33%), compensando com um *recall* equivalente ao da CNN (97,33%). Sua *Cross Entropy Loss*, de 0,43, é a mais alta, o que indica que ele faz previsões com maior incerteza. Entretanto, o Naive Bayes contribui significativamente para o sistema colaborativo com seu AUC de 99,3%, mostrando que ele consegue diferenciar as classes de forma eficiente, o que enriquece as decisões quando combinado com os outros dois modelos.

Portanto, quando esses três modelos trabalham em conjunto, eles formam um sistema de predição mais robusto e preciso. A CNN lidera com sua precisão e baixa incerteza nas previsões, o SGD agrega uma discriminação confiável com uma taxa de erro moderada, e o Naive Bayes fortalece o sistema com sua alta capacidade de separar classes, apesar da menor acurácia geral. Juntos, eles otimizam a predição na classificação binária, equilibrando as falhas e vantagens de cada um.

Após isso, os modelos de multiclassificação para DIDs recebem os dados classificados como não-veiculares e performam as suas predições. Para esse tipo de classificação, foi utilizado uma Rede Neural, o SGD e o Random Forest, com os seguintes resultados:

Tabela 20 – Modelos utilizados para classificação dos dados (DID)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
CNN				
Identificador Indireto	0.97	0.988	0.988	0.988
Dados do Motorista		0.999	0.999	0.999
Identificador Direto		0.500	0.500	0.500
Geolocalização		0.833	0.833	0.833
Blocklist		0.997	0.997	0.997
SGD				
Identificador Indireto	0.98	0.98	0.98	0.98
Dados do Motorista		0.99	0.99	0.99
Identificador Direto		0.97	0.97	0.97
Blocklist		0.90	0.90	0.90
Random Forest				
Identificador Indireto	0.97	0.95	0.92	0.94
Dados do Motorista		0.99	0.98	0.98
Identificador Direto		0.72	0.88	0.80
Geolocalização		0.60	0.42	0.50
Blocklist		0.91	0.95	0.93

FONTE: Autoria própria.

A escolha dos modelos CNN, SGD e *Random Forest* foi guiada por sua capacidade de lidar com diferentes características dos dados veiculares. O CNN foi utilizado devido à sua eficácia em detectar padrões complexos, o SGD se mostrou eficiente na otimização rápida e eficiente de grandes volumes de dados, enquanto o *Random Forest* foi selecionado pela sua robustez em cenários de classificação multiclasses, especialmente onde há variação significativa na distribuição das classes. Embora cada modelo tenha apresentado pontos fortes, o *Random Forest* mostrou-se mais equilibrado em termos de precisão geral, apesar de dificuldades em identificar corretamente a categoria de Geolocalização.

A avaliação de desempenho para a classificação dos sinais CAN seguiu os mesmos critérios utilizados para os DIDs, incluindo métricas como precisão, sensibilidade, acurácia e a matriz de confusão. A tabela 21, sintetiza a escolha dos modelos utilizados para a classificação binária desses sinais.

Tabela 21 – Métricas de Desempenho Classificação Binária - sinais CAN

Modelo	**Acurácia**	**Precisão**	**Recall**	**F1-Score**	**Precisão**	**Recall**	**F1-Score**
		Dados Veiculares			**Dados Não-Veiculares**		
CNN	0.95	0.96	0.89	0.93	0.64	0.89	0.74
Naive Bayes	0.84	0.99	0.82	0.89	0.52	0.95	0.67
SGD	0.94	0.94	0.95	0.95	0.78	0.90	0.83
Random Forest	0.93	0.98	0.98	0.96	0.87	0.72	0.79

FONTE: Autoria própria.

Já a tabela 21 compara os modelos de classificação utilizados no estudo. O SGD

se destacou com a maior acurácia (0,94) e um desempenho equilibrado entre as classes, mostrando-se consistente na classificação de dados não-veiculares. O *Naive Bayes* apresentou boa eficácia na identificação de dados não-veiculares, apesar de apresentar uma precisão baixa, ele mostrou um *recall* alto para essa classe. Isso é importante em contextos onde é preferível identificar um dado como não-veicular, mesmo que ocorra uma pequena taxa de falsos positivos. Já o *Random Forest*, apesar de alcançar uma alta precisão para dados não-veiculares, apresentou um *recall* menor, indicando que perdeu algumas classificações importantes.

A CNN, por sua vez, obteve uma acurácia de 0,95, demonstrando um bom desempenho geral, especialmente para dados veiculares. Sua capacidade de capturar padrões complexos justifica sua inclusão, mesmo que a precisão para dados não-veiculares"tenha sido inferior.

Dada a natureza do problema, que envolve relações lineares entre as variáveis, a escolha de modelos mais simples, como *Naive Baye*s e SGD, é apropriada, pois oferece eficiência e rapidez. A CNN é uma boa opção quando há necessidade de identificar padrões complexos, enquanto o SGD se destaca pela consistência e desempenho equilibrado. Portanto, para cenários que priorizam simplicidade e interpretabilidade, a *Naive Bayes* e o SGD são mais indicados, enquanto a CNN é útil quando a complexidade dos dados exige uma abordagem mais robusta.

Os modelos CNN, SGD e Regressão Logística foram escolhidos devido ao seu desempenho robusto em termos de precisão, *recall* e *F1-Score*, conforme observado nas tabelas anteriores. A CNN se destaca por sua capacidade de lidar com dados complexos, especialmente em tarefas de reconhecimento de padrões, enquanto o SGD, um modelo linear, mostrou alta eficácia na classificação de dados não-veiculares. A Regressão Logística complementa essa abordagem, mostrando consistência na classificação correta das categorias com alta acurácia e *F1-Score*. Por outro lado, o *XGBoost*, apesar de apresentar resultados sólidos, obteve métricas inferiores em algumas categorias, como Geolocalização, onde a precisão e o *recall* foram de 86%, o que representa uma queda em relação aos outros modelos. Esse fator torna o *XGBoost* menos adequado para o problema em questão. A tabela 22 mostra um comparativo entre esses dados, de acordo com o modelo e classe.

Tabela 22 – Modelos utilizados para classificação dos dados (CAN)

Tipo de Dado	**Acurácia**	**Precisão**	**Recall**	**F1-Score**
CNN				
Dados do Motorista	0.98	0.98	0.98	0.98
Geolocalização		0.92	0.92	0.92
Identificador Indireto		1.00	0.88	0.93
Blocklist		0.96	0.98	0.97
SGD				
Dados do Motorista	0.99	0.99	0.99	0.99
Geolocalização		0.93	0.93	0.93
Identificador Indireto		1.00	0.93	0.97
Blocklist		0.98	0.98	0.98
Reg. Logística				
Dados do Motorista	0.98	0.98	0.99	0.98
Geolocalização		0.96	0.93	0.95
Identificador Indireto		1.00	0.93	0.97
Blocklist		0.96	0.96	0.96
XGBoost				
Dados do Motorista	0.97	0.97	0.97	0.97
Geolocalização		0.86	0.86	0.86
Identificador Indireto		1.00	0.87	0.93
Blocklist		0.91	0.94	0.93

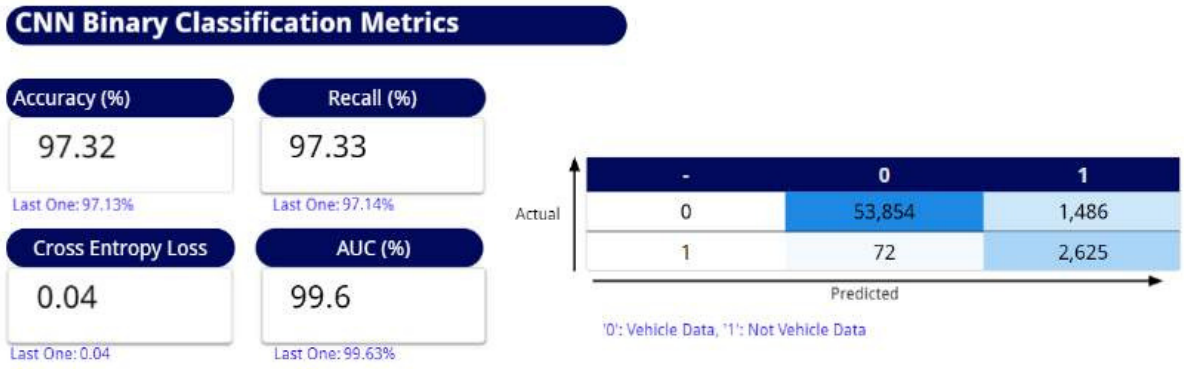
FONTE: Autoria própria.

Desta forma, os modelos escolhidos para a realizar multiclassificação dos sinais CAN conseguem formar um sistema robusto e com bom desempenho, atendendo as necessidades especificas dessa atividade.

Como citado nas sessões anteriores, os *scripts* para classificação automática do DIDs e Sinais CAN são executados semanalmente e continuam demonstrando um bom desempenho diante da criação de novos dados. Esse acompanhamento pode ser feito por gráficos gerados periodicamente dispostos em *dashboard* especifico, exibindo as métricas de cada modelo implementado.

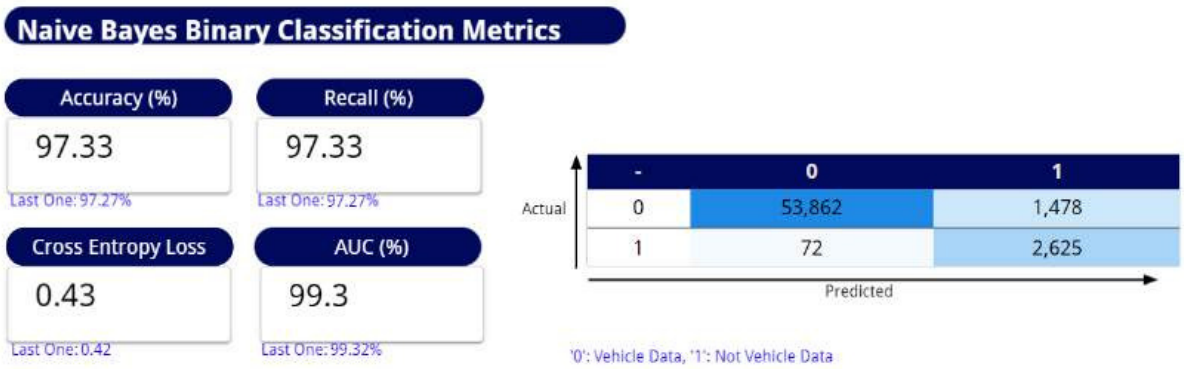
O desempenho dos modelos implementados na composição desse sistema continua se provando satisfatório. As imagens 23, 24 e 25 mostram como os modelos de classificação binária para os DIDs tem performado enquanto as imagens 26, 27, e 28 exibem a performance atual dos modelos de multiclassificação para esse tipo de sinal.

Figura 23 – Métricas de Classificação Binária - CNN (DIDs)



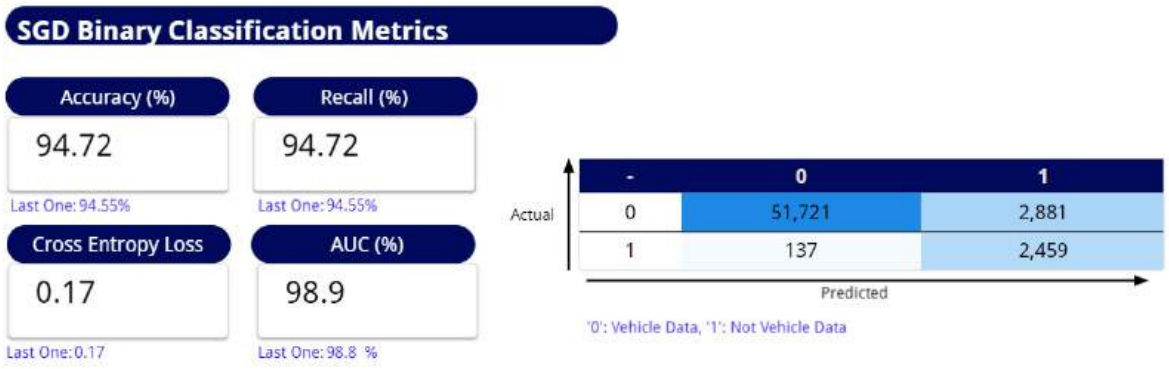
FONTE: AVD Classification Dashboard

Figura 24 – Métricas de Classificação Binária - Naive Bayes (DIDs)



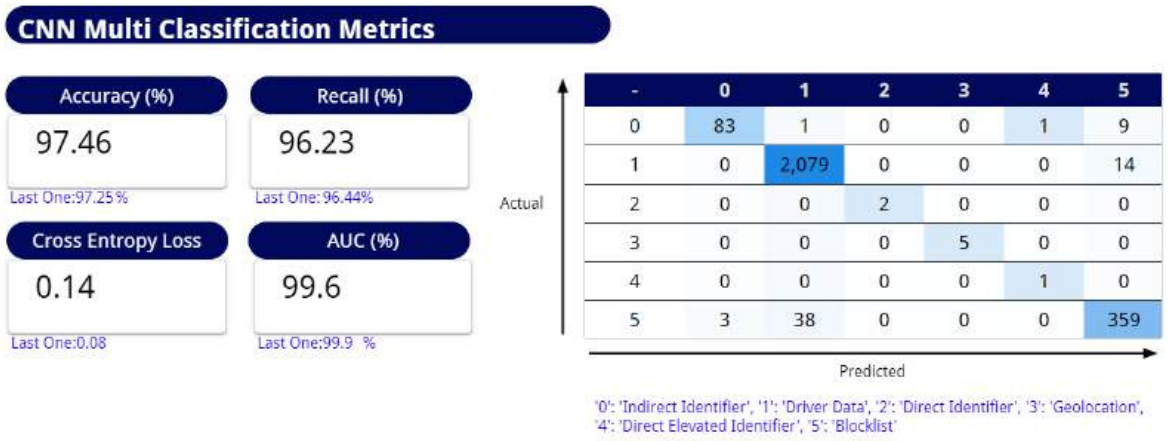
FONTE: AVD Classification Dashboard

Figura 25 – Métricas de Classificação Binária - SGD (DIDs)



FONTE: AVD Classification Dashboard

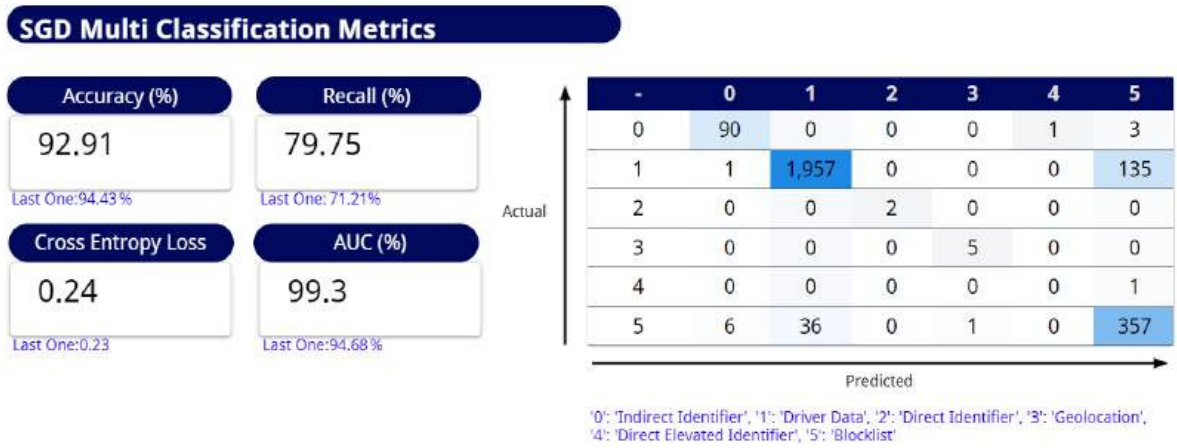
Figura 26 – Métricas de Multiclassificação - CNN (DIDS)



-	0	1	2	3	4	5
0	83	1	0	0	1	9
1	0	2,079	0	0	0	14
2	0	0	2	0	0	0
3	0	0	0	5	0	0
4	0	0	0	0	1	0
5	3	38	0	0	0	359

FONTE: AVD Classification Dashboard

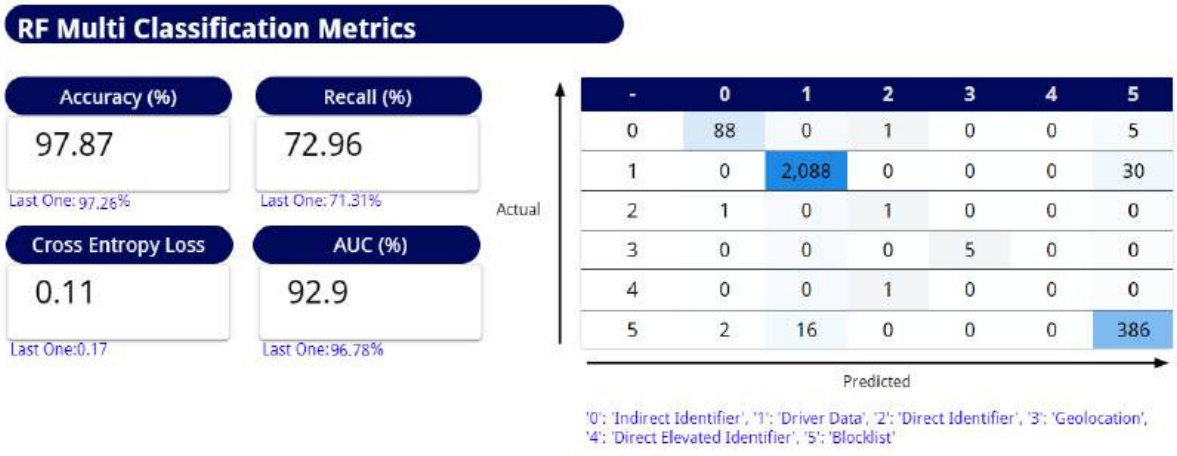
Figura 27 – Métricas de Multiclassificação - SGD (DIDs)



-	0	1	2	3	4	5
0	90	0	0	0	1	3
1	1	1,957	0	0	0	135
2	0	0	2	0	0	0
3	0	0	0	5	0	0
4	0	0	0	0	0	1
5	6	36	0	1	0	357

FONTE: AVD Classification Dashboard

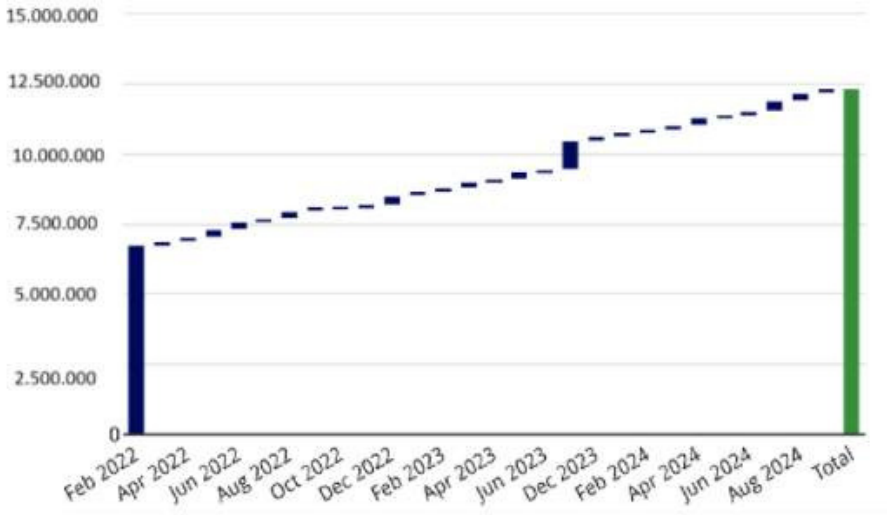
Figura 28 – Metricas de Multiclassificação - Random Forest (DIDs)



-	0	1	2	3	4	5
0	88	0	1	0	0	5
1	0	2,088	0	0	0	30
2	1	0	1	0	0	0
3	0	0	0	5	0	0
4	0	0	1	0	0	0
5	2	16	0	0	0	386

FONTE: AVD Classification Dashboard

Figura 29 – Evolução da classificação para DIDs ao longo do tempo



FONTE: AVD Classification Dashboard

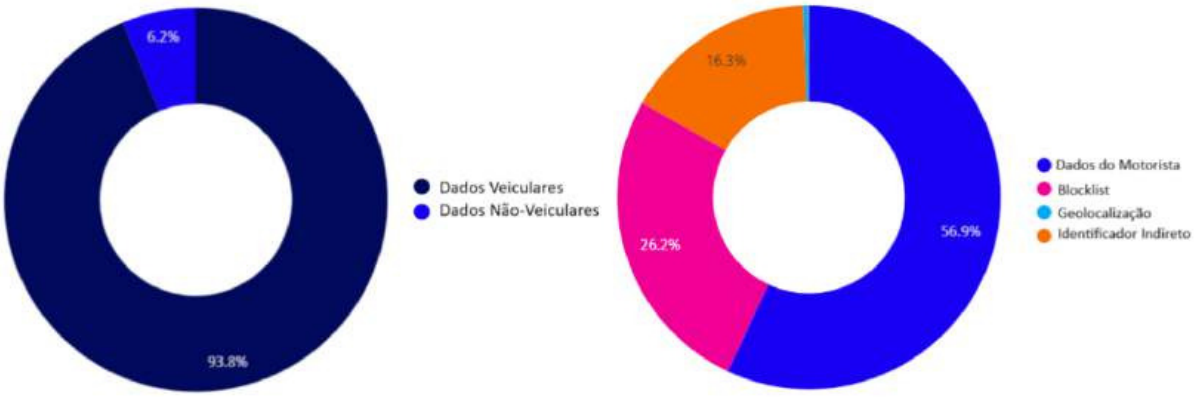
Conforme o gráfico da figura 29, é possível ver que aproximadamente 150.000 DIDs são criados mensalmente, tornando impraticável a classificação manual dessa quantidade de dados mensalmente. Entretanto, com o uso do sistema automatizado, 99.99% dos DIDs existentes já foram classificados, como mostra a figura 30.

Figura 30 – Taxa de Classificação - DIDs

FONTE: AVD Classification Dashboard

A figura 31 também demonstra a distribuição de classificação entre os DIDs, ilustrando a predominância de dados veiculares.

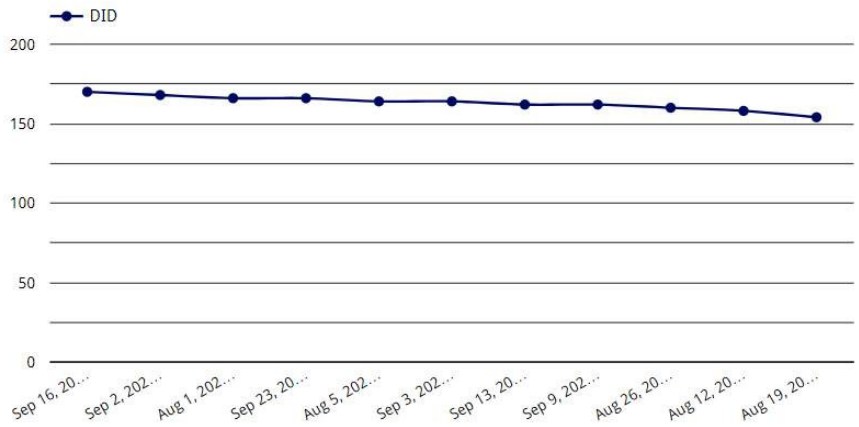
Figura 31 – Distribuição de Classificação Binária e Multiclasse para os dados (DIDs)



FONTE: AVD Classification Dashboard

Entre esses dados também é possível analisar-se os erros cometidos, no qual as inconsistências entre a classificação dos modelos são enviadas para a revisão manual. Isso geralmente ocorre quando mais é detectado que um sinal foi classificado mais de uma vez, a classificação do DID principal não corresponde a do maior parâmetro e afins. Esse parâmetro pode ser visto no gráfico da figura 32.

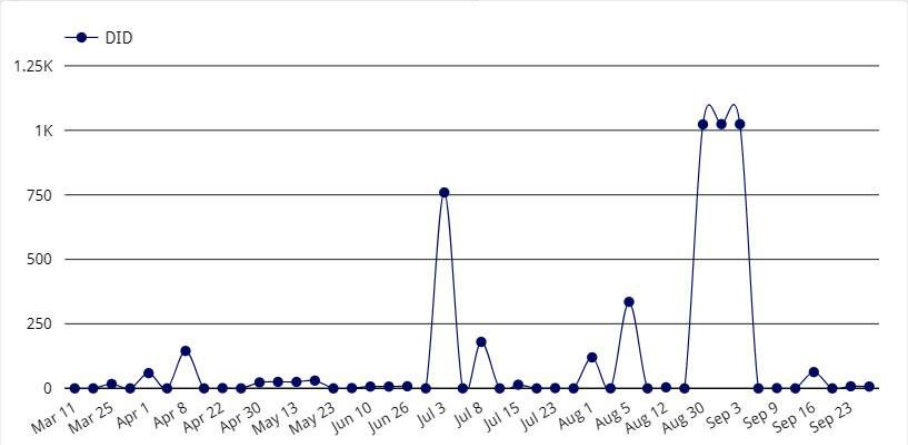
Figura 32 – Relação de Inconsistências para a Classificação Combinada de DIDs



FONTE: AVD Classification Dashboard

Já as divergências são computadas quando os modelos implementados não conseguem chegar a um consenso da categoria para determinado dado, com isso, o sinal é enviado para uma tabela separada - de divergências - para aguardar a classificação manual. O histórico de divergências para a classificação dos DIDs pode ser observado na figura 33

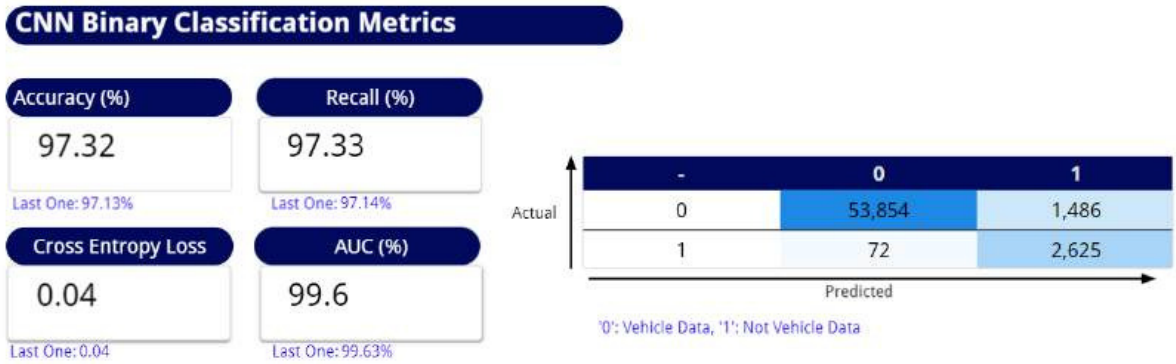
Figura 33 – Histórico de Divergências para a Classificação de DIDs



FONTE: AVD Classification Dashboard

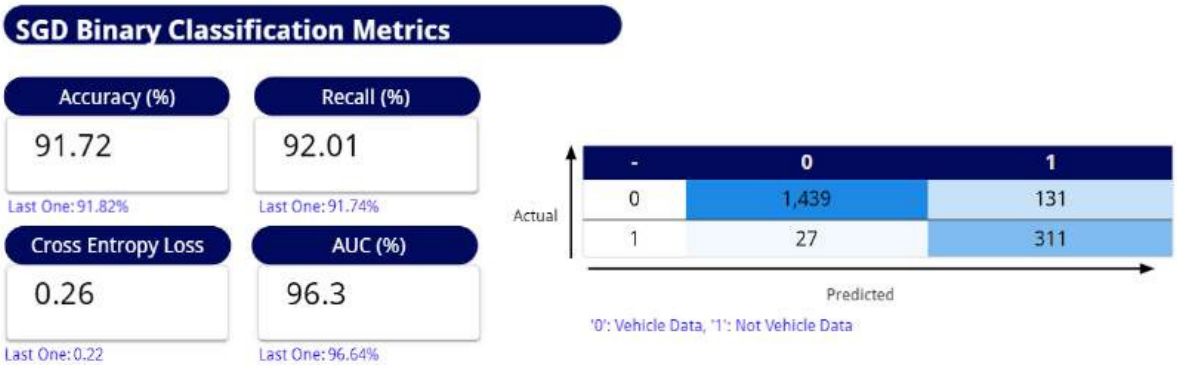
Os modelos adotados para a classificação dos sinais CAN continuam exibindo uma boa performance. O desempenho atual dos modelos de classificação binária pode ser observado nas figuras 34, 35 e 36.

Figura 34 – Métricas de Classificação Binária - CNN (Sinais CAN)



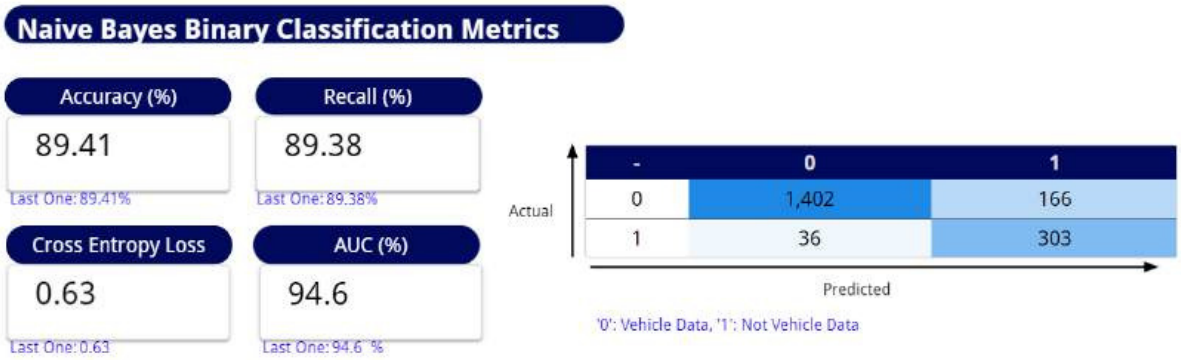
FONTE: AVD Classification Dashboard

Figura 35 – Métricas de Classificação Binária - SGD (Sinais CAN)



FONTE: AVD Classification Dashboard

Figura 36 – Métricas de Classificação Binária - *Naive Bayes* (Sinais CAN)

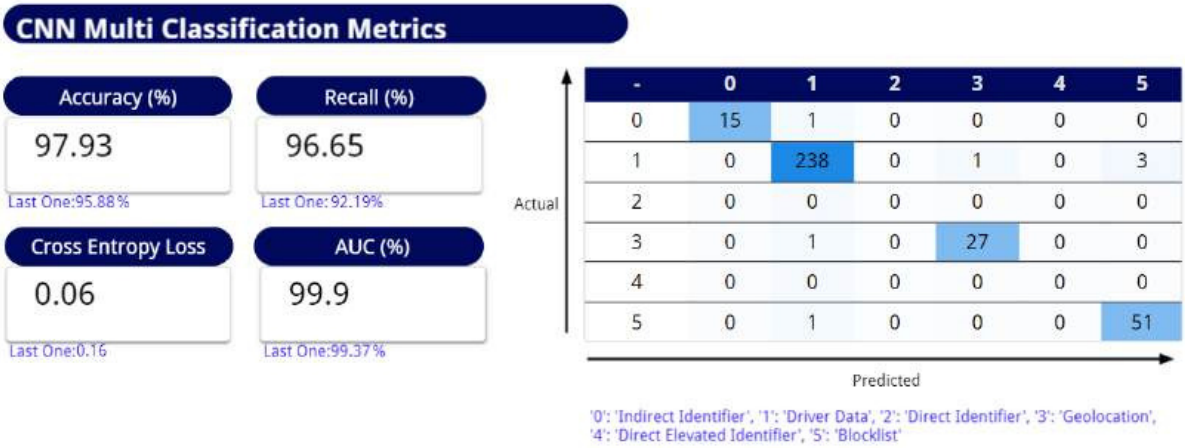


-	0	1
0	1,402	166
1	36	303

FONTE: AVD Classification Dashboard

O desempenho atual dos modelos empregados na multiclassificação podem ser vistos nas figuras 37, 38, e 39.

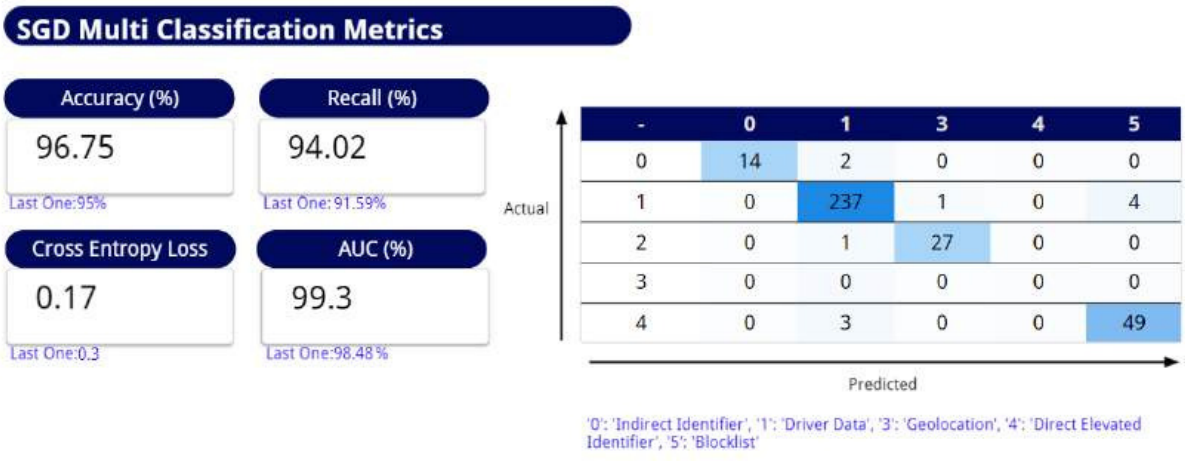
Figura 37 – Métricas de Multiclassificação - CNN (Sinais CAN)



-	0	1	2	3	4	5
0	15	1	0	0	0	0
1	0	238	0	1	0	3
2	0	0	0	0	0	0
3	0	1	0	27	0	0
4	0	0	0	0	0	0
5	0	1	0	0	0	51

FONTE: AVD Classification Dashboard

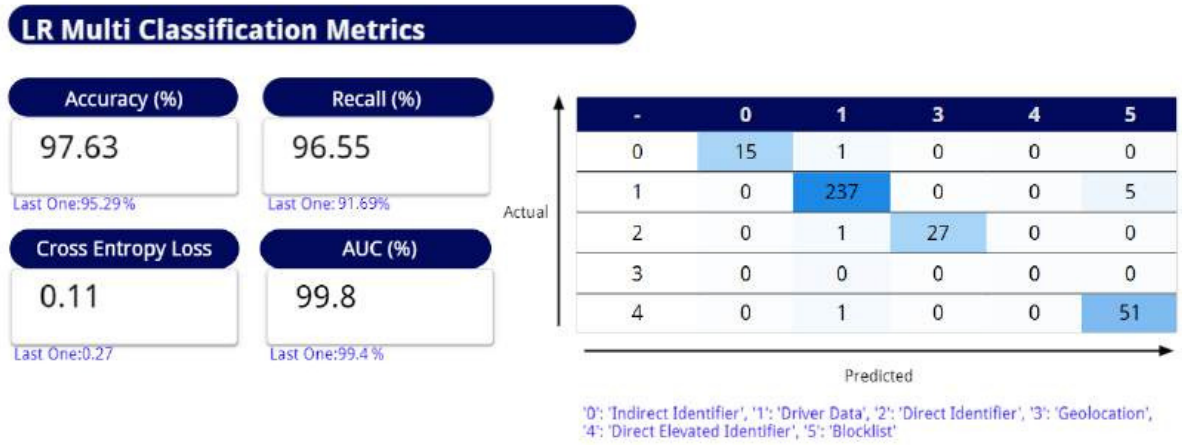
Figura 38 – Métricas de Multiclassificação - SGD (Sinais CAN)



-	0	1	3	4	5
0	14	2	0	0	0
1	0	237	1	0	4
2	0	1	27	0	0
3	0	0	0	0	0
4	0	3	0	0	49

FONTE: AVD Classification Dashboard

Figura 39 – Métricas de Multiclassificação - Regressão Linear (Sinais CAN)

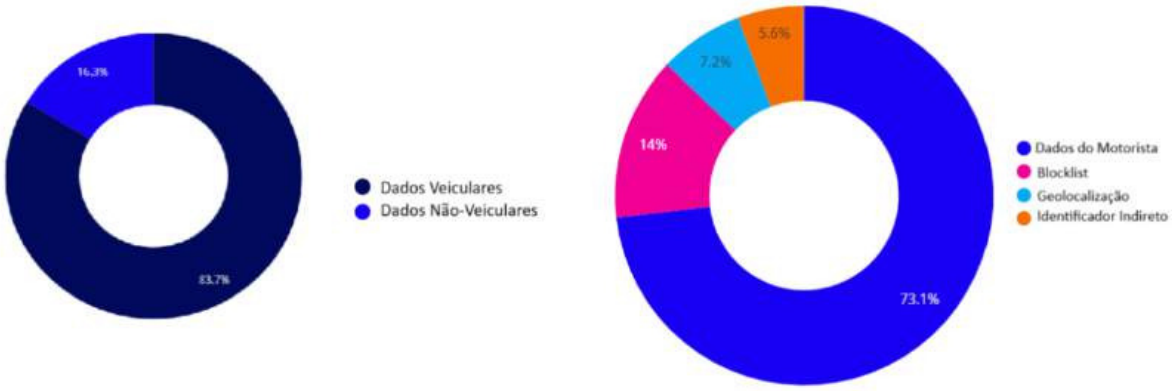


-	0	1	3	4	5
0	15	1	0	0	0
1	0	237	0	0	5
2	0	1	27	0	0
3	0	0	0	0	0
4	0	1	0	0	51

FONTE: AVD Classification Dashboard

A figura 40 também demonstram a distribuição de classificação entre os DIDs, ilustrando a predominância de dados veiculares.

Figura 40 – Distribuição de classificação binária e multiclasse para os dados (Sinais CAN)



FONTE: AVD Classification Dashboard

Na figura 41, é possível observar que a quantidade de dados criados ao longo do ano é bem baixa, especialmente em comparação com os DIDs, com uma média de 15 sinais novos e classificados por mês.

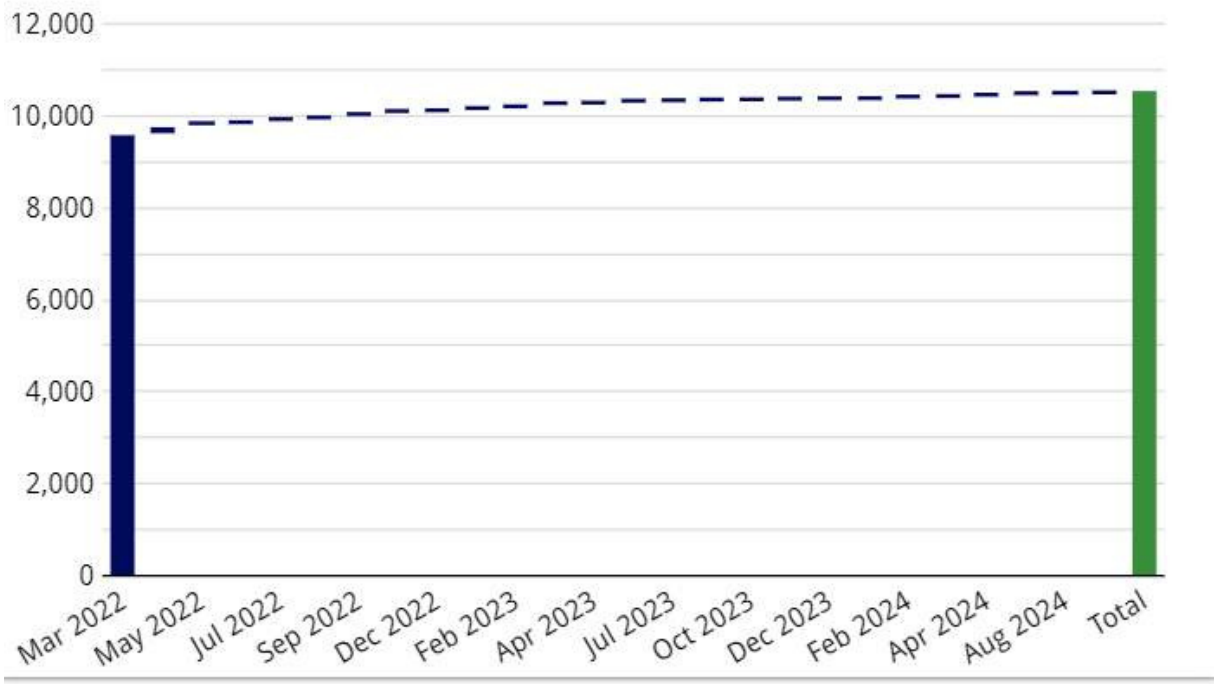
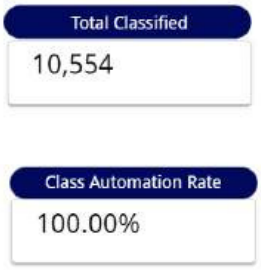


Figura 41 – Evolução da classificação para sinais CAN ao longo do tempo

FONTE: AVD Classification Dashboard

Com isso, a taxa de classificação dos sinais CAN corresponde a 100%, de acordo com o *report*, figura 42 gerado em 24 de Setembro de 2024. Não havendo inconsistências ou divergências para a classificação.

Figura 42 – Relatório Classificação de sinais CAN



FONTE: AVD Classification Dashboard

5 Considerações finais

Este trabalho propôs uma abordagem eficiente para a classificação de dados sensíveis em veículos conectados, utilizando algoritmos de aprendizado de máquina. A principal vantagem do processo automatizado em relação ao método manual anterior está na sua rapidez, precisão e capacidade de lidar com grandes volumes de dados, Enquanto o processo manual demandava muito tempo e estava sujeito a erros humanos e interpretações subjetivas, a automação não só acelerou a análise, mas também garantiu maior consistência nos resultados. Assim, a classificação automática não apenas otimiza o tempo, mas também é capaz de processar a crescente complexidade e quantidade de dados gerados pelos veículos conectados de forma muito mais eficiente, algo que seria impraticável em um processo inteiramente manual.

Um dos principais desafios enfrentados na era dos veículos conectados é a proteção de dados sensíveis contra vazamentos e ataques cibernéticos. O processo desenvolvido neste trabalho foca justamente na classificação precisa desses dados, divididos em categorias com diferentes níveis de criticidade. A classificação correta deles é essencial para garantir que informações mais delicadas, como Identificadores Diretos e dados de Geolocalização, recebam a proteção adequada. A aplicação de algoritmos como *Naive Bayes, Stochastic Gradient Descent (SGD)*, Regressão Logística e Redes Neurais Convolucionais (CNN) trouxe resultados promissores, mostrando-se eficiente na identificação de dados críticos e na mitigação de riscos de exposição indevida, assegurando que as informações fossem corretamente protegidas conforme sua importância.

O uso de múltiplos modelos de classificação também agregou valor ao processo, oferecendo maior segurança ao cruzar as decisões de diferentes algoritmos, proporcionando uma camada extra de segurança para a classificação. A introdução de uma revisão manual em casos de discrepância reforçou ainda mais a confiabilidade do sistema. Além disso, o uso de métricas como acurácia, precisão e *recall* foi essencial para ajustar os modelos e garantir que dados críticos fossem devidamente protegidos, sem o risco de subproteção, resultando em uma solução robusta para classificar e proteger grandes volumes de dados. Isso foi particularmente relevante no caso dos DIDs, que apresentam uma grande variação em seus parâmetros e criticidade, exigindo um processo automatizado que acompanhe o volume de dados e a velocidade com que são gerados.

Com relação a melhorias futuras, há várias possibilidades que podem ser exploradas para melhorar a performance do sistema. Uma delas é o aprimoramento das predições dos modelos por meio de técnicas de *tunagem*, ajustando os hiper parâmetros para otimizar a performance, tornando as predições ainda mais precisas. Outra abordagem interessante

seria a implementação de modelos especializados para diferentes categorias de dados, otimizando o desempenho para tipos específicos de classificação. Além disso, a adoção de um critério de decisão baseado em votação ponderada entre os modelos permitiria que algoritmos com melhor desempenho em determinadas classes tivessem mais influência na classificação final, aumentando ainda mais a confiabilidade e segurança do sistema.

Em suma, este estudo não apenas aprimora a segurança e privacidade dos veículos conectados, mas também abre caminho para futuras inovações na área de proteção de dados. A combinação de aprendizado de máquina e segurança da informação é essencial para garantir que, à medida que o volume de dados cresce, a proteção dessas informações continue sólida. Com as melhorias sugeridas, a abordagem proposta tem o potencial de se expandir para outras áreas, como veículos autônomos e sistemas de *infotainment*. Com isso, espera-se que o setor automotivo continue a evoluir, sempre priorizando a proteção e privacidade dos usuários.

Referências

Amazon Web Services. *What is Logistic Regression?* 2023. Acesso em: 30 set. 2024. Disponível em: <https://aws.amazon.com/pt/what-is/logistic-regression/>. Citado na página 24.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Citado na página 40.

CAMACHO, C. *CNNs for text classification*. 2019. Accessed: [date today]. Disponível em: <https://cezannec.github.io/CNN_Text_Classification/>. Citado na página 24.

CARVALHO, S. M. T. d.; CAMPOS, G. L. Transmissão de mensagens e gerenciamento de erros em uma rede can automotiva. *ForSci.: r.cient. IFMG*, IFMG, Formiga, v. 6, n. 1, p. e00341, jan. 2018. Número de Engine Control. Citado na página 19.

EDWARDS, J. Signal processing leads to new wireless technologies. *IEEE Signal Processing Magazine*, v. 31, n. 5, p. 10–14, 2014. Citado na página 16.

EISENSTEIN, J. *Natural Language Processing*. [S.l.]: MIT Press, 2018. Under contract. Citado na página 25.

FERRARI, D. G.; SILVA, L. N. D. C. *Introdução a mineração de dados*. [S.l.]: Saraiva Educação S.A., 2017. Citado na página 22.

GARCIA, I. B. d. S.; RAMOS, P. C. d. S. Identificação de autoria de contos usando técnicas de processamento de linguagem natural. *Repositório Institucional do Conhecimento - RIC-CPS*, nov 2023. Orientador: Dezani, Henrique. Outros colaboradores: Simonato, Adriano Luís; Ribeiro, Matheus Gonçalves. Editor: 121. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/19745>. Citado na página 25.

GEEKSFORGEEKS. *What is Cross-Entropy Loss Function?* 2024. Acesso em: 2 de outubro de 2024. Tradução minha. Disponível em: <https://www.geeksforgeeks.org/what-is-cross-entropy-loss-function/>. Citado na página 22.

HPL, S. C. Introduction to the controller area network (can). *Application Report SLOA101*, Texas instruments Dallas, TX, USA, p. 1–17, 2002. Citado na página 19.

IBM. *Convolutional Neural Networks*. 2023. Acesso em: 21 set. 2024. Disponível em: <https://www.ibm.com/br-pt/topics/convolutional-neural-networks>. Citado na página 24.

IBM. *O que são classificadores Naïve Bayes?* 2023. [Acesso em: 26 set. 2024]. Disponível em: <https://www.ibm.com/br-pt/topics/naive-bayes>. Citado na página 23.

IBM. *Gradient Descent*. 2024. Accessed: 2024-09-30. Disponível em: <https://www.ibm.com/topics/gradient-descent>. Citado na página 23.

IBM. *Logistic Regression*. 2024. Acesso em: 30 set. 2024. Traduzido do inglês. Disponível em: <https://www.ibm.com/topics/logistic-regression>. Citado na página 24.

INDURKHYA, N.; DAMERAU, F. J. *Handbook of Natural Language Processing*. 2nd. ed. [S.l.]: Chapman & Hall/CRC, 2010. (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series). ISBN 9781420085921. Citado na página 20.

JUNIOR, P. C. D. *Serviços telemáticos em uma rede de transporte público baseados em veículos conectados e dados abertos*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, 2017. Citado na página 16.

KUMAR, V.; ZHU, D.; DADAM, S. Connected vehicle data – prognostics and monetization opportunity. *SAE Technical Paper*, 2023. 2023-01-1685. Citado na página 17.

MARIANO, D.; XAVIER, J. S. r. Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e f-score. *BIOINFO - Revista Brasileira de Bioinformática*, n. 01, jul 2021. Citado na página 21.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: *Proceedings of the International Conference on Learning Representations Workshop*. [S.l.: s.n.], 2013. ICLR Workshop. Citado na página 21.

MISHRA, M. *Stochastic Gradient Descent: A Basic Explanation*. 2023. Accessed: 2024-09-30. Disponível em: <https://mohitmishra786687.medium.com/stochastic-gradient-descent-a-basic-explanation-cbddc63f08e0>. Citado na página 23.

NEELY, A. D.; GREGORY, M. J.; PLATTS, K. W. Performance measurement system design: a literature review and research agenda. *International Journal of Operations and Production Management*, v. 15, n. 4, p. 80–116, 1999. Citado na página 21.

PORTER, M. E.; HEPPELMANN, J. E. How smart, connected products are transforming companies. *Harvard Business Review*, v. 93, n. 10, p. 96–114, 2015. Citado na página 16.

QUEIROZ, A. et al. Autenticação com suporte à computação de borda 5g para a internet de veículos. *Brazilian Journal of Development*, v. 9, n. 5, 2023. ISSN 2525-8761. Citado na página 16.

SILVA, E. H. D. R.; LIMA, E. P. O estudo de indicadores de desempenho sob o enfoque da gestão estratégica organizacional. *GEPROS. Gestão da Produção, Operações e Sistemas*, Bauru, v. 10, n. 3, p. 159–175, jul-sep 2015. Citado na página 21.

SOUZA, A.; CAMPOS, G. Rede can veicular: levantamento bibliográfico e apresentação de conceitos iniciais. *ForSci.: r.cient. IFMG*, IFMG, Formiga, v. 5, n. 1, p. e00234, jan. 2017. Citado na página 19.

SOUZA, M. N. V. d. *Comparação de Algoritmos do Aprendizado de Máquina Aplicados na Mineração de Dados Educacionais*. Dissertação (Mestrado) — Universidade Federal Rural de Pernambuco, Departamento de Estatística e Informática, Recife, dezembro 2015. Citado na página 22.

SUGAYAMA, R.; NEGRELLI, E. Connected vehicle on the way of industry 4.0. In: UNIVERSIDADE TECNOLóGICA FEDERAL DO PARANá - ESPECIALIZAçãO ENGENHARIA AUTOMOTIVA. *SIMEA 2016*. [S.l.], 2016. Citado 2 vezes nas páginas 16 e 17.

TECH, P. *Data Identifiers (DID) of UDS Protocol ISO 14229*. 2023. <https://piembsystech.com/data-identifiers-did-of-uds-protocol-iso-14229/>. Citado na página 18.

VIDHYA, A. *An End to End Guide to Understand the Math Behind XGBoost*. 2024. Accessed: 2024-09-30. Disponível em: <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>. Citado na página 45.

Cópia de documento digital impresso por Josinete Araujo (1218532) em 29/07/2025 11:13.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus João Pessoa - Código INEP: 25096850
Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, João Pessoa (PB)
CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Ostensivo (Público)

Trabalho de Conclusão de Curso

Assunto:	Trabalho de Conclusão de Curso
Assinado por:	Raylle Nobrega
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Raylle Cordeiro da Nobrega**, **ALUNO (20172610014) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA**, em 18/10/2024 12:21:23.

Este documento foi armazenado no SUAP em 18/10/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1283408
Código de Autenticação: 9b0a121021