Pontifícia Universidade católica do rio de janeiro

Redes Neurais Recorrentes e Análise Estatística Multivariada para Previsão e Medição da influência entre as variáveis de Emissões e Consumo de Combustível em Veículos

Caio Coutinho Palmieri

Projeto Final de Graduação

Orientação Prof Paulo Ivson Netto Santos

Centro Técnico Científico - CTC Departamento de Informática

Curso de Graduação em Ciências da Computação

Rio de Janeiro, dezembro de 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Caio Coutinho Palmieri

Redes Neurais Recorrentes e Análise Estatística Multivariada para Previsão e Medição da influência entre as variáveis de Emissões e Consumo de Combustível em Veículos

Projeto Final apresentado ao Curso de Ciência da Computação da PUC-Rio como parte do cumprimento parcial dos requisitos para o grau de Bacharel em Ciência da Computação.

Orientador: Paulo Ivson Netto Santos

Rio de Janeiro

Dezembro de 2024

Agradecimento

Primeiramente, agradeço a Deus pela força e sabedoria concedidas ao longo desta jornada acadêmica.

Gostaria de agradecer à minha mãe , Silvania Palmieri e ao meu pai Alexandre Coutinho Silva por todo o amor, apoio e incentivos incondicionais. Vocês sempre acreditaram em mim.

Ao meu orientador Paulo Ivson, pela orientação, paciência e dedicação ao longo de todo o processo de desenvolvimento deste trabalho.

Agradeço também a todos os professores e colegas de curso que contribuíram, direta ou indiretamente, para minha formação acadêmica. Cada aprendizado e experiência compartilhada foram fundamentais para o meu crescimento pessoal e profissional.

Resumo

Coutinho Palmieri, Caio. Ivson, Paulo. Redes Neurais Recorrentes e Análise Estatística Multivariada para Previsão e Medição da influência entre as variáveis de Emissões e Consumo de Combustível em Veículos. Rio de Janeiro, 2024. 78p. Relatório de Projeto Final de Graduação – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho desenvolve modelos preditivos baseados em Redes Neurais Recorrentes (RNNs) e Long Short-Term Memory (LSTM) para prever emissões de poluentes e consumo de combustível em veículos, utilizando dados históricos como hodômetro e tipo de combustível. As RNNs e LSTMs, por sua capacidade de capturar padrões complexos em séries temporais, são aplicadas para identificar tendências e prever comportamentos futuros, contribuindo para a eficiência energética e a redução de emissões. Além disso, uma análise estatística multivariada com o Random Forest e outros algoritmos, como AdaBoost e Gradient Boost, é realizada para avaliar a influência das variáveis independentes na variável alvo, permitindo identificar fatores críticos que impactam no desempenho veicular. A combinação dessas técnicas de aprendizado de máquina e ciência de dados oferece soluções robustas e inovadoras, promovendo o desenvolvimento sustentável ao enfrentar desafios ambientais e econômicos relacionados à poluição atmosférica e à otimização do setor de transporte.

Palavras-chave

Rede neural Recorrente; consumo; emissão, Análise estatística Multivariada

Abstract

Coutinho Palmieri, Caio. Ivson, Paulo. Recurrent Neural Networks and Multivariate Statistical Analysis for Forecasting and Measuring the Influence Between Emission and Fuel Consumption Variables in Vehicles. Rio de Janeiro, 2024. 78p. Relatório de Projeto Final de Graduação – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

This work develops predictive models based on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) to forecast pollutant emissions and fuel consumption in vehicles, using historical data such as odometer readings and fuel type. RNNs and LSTMs, due to their ability to capture complex patterns in time series, are applied to identify trends and predict future behaviors, contributing to energy efficiency and emission reduction. Additionally, a multivariate statistical analysis with Random Forest and other algorithms, such as AdaBoost and Gradient Boost, is performed to assess the influence of independent variables on the target variable, identifying critical factors that impact vehicle performance. The combination of these machine learning and data science techniques provides robust and innovative solutions, promoting sustainable development by addressing environmental and economic challenges related to air pollution and the optimization of the transportation sector.

Keywords

Recurrent Neural Network; Consumption; Emission; Multivariate Statistical Analysis

Sumário

1. Introdução

Nos últimos anos, a crescente preocupação com a sustentabilidade ambiental e a necessidade imperativa de reduzir as emissões de poluentes têm fomentado um avanço significativo nas pesquisas e nos desenvolvimentos tecnológicos em múltiplos setores [1], dentre esses, destacam a ciência de dados e a inteligência artificial (IA), que fornecem ferramentas para a análise e otimização de processos complexos. Uma abordagem baseada na coleta de dados de veículos, tais como consumo de combustível, emissões de CO2, padrões de condução, entre outros, permite a implementação de estratégias mais precisas e eficazes para a otimização da eficiência dos veículos[2]. Com o aumento das cidades e o crescimento significativo da quantidade de veículos nas ruas, a emissão de poluentes têm se tornado uma das principais fontes de poluição do ar nas cidades [3]. Portanto, a busca por soluções que visem a redução dos lançamentos de poluentes é crucial para o desenvolvimento sustentável.

As emissões de poluentes atmosféricos vem aumentando proporcionalmente conforme o crescimento populacional, como dióxido de carbono (CO2), óxidos de nitrogênio (NOx), monóxido de carbono (CO) e material particulado (MP)[4]. Essas emissões não apenas contribuem para o aquecimento global [5], mas também têm impactos diretos na saúde pública, causando doenças respiratórias e cardiovasculares. Diante desse cenário, a redução das emissões de poluentes de veículos tornou-se uma prioridade global. Atualmente, as indústrias de automóveis passam por desafios de regular a demanda por veículos eficientes e ao mesmo tempo conseguir reduzir a emissão de poluentes no ar[4]. É notório que o governo cria medidas para limitar a quantidade da emissão de poluentes que os veículos podem emitir[7], fazendo com que imponha uma pressão sobre os fabricantes para diminuir a quantidade de gases estufas emitidos, mas devido a complexidade dos sistemas veiculares tornam essa tarefa difícil de ser concluída .

Apesar dos avanços significativos, as soluções atuais para reduzir as emissões e melhorar o consumo de combustível dos veículos ainda apresentam várias limitações. Tecnologias como motores de combustão interna mais eficientes,a criação de veículos híbridos e o desenvolvimento de veículos totalmente elétricos têm sido adotadas globalmente [13]. No entanto, esses métodos frequentemente não são suficientes devido a fatores como alto custo de implementação, infraestrutura inadequada e a lenta substituição da frota de veículos antigos. Além disso, enquanto

os veículos elétricos eliminam as emissões diretas, por outro lado a produção de eletricidade aumenta e ainda depender de fontes não-renováveis de energia, transferindo a poluição para outro setor, podemos citar também a maneira que a bateria do veículo elétrico é descartado, que caso seja descartado de maneira incorreta poderá causar tanto impacto no solo como impacto na poluição do ar[14].

Com o avanço das tecnologias de ciência de dados e inteligência artificial, novas meios surgiram para abordar esses desafios, como por exemplo, Técnicas de aprendizado de máquina, regressão linear, árvores de decisão e redes neurais artificiais esses artifícios permitem a construção de modelos que sejam preditivos que são capazes de identificar ineficiências operacionais nos veículos e podendo prever a emissões de poluentes com maior precisão. Além disso, podemos citar Redes Neurais Profundas (DNNs)[8] e redes neurais recorrentes (RNNs)[8], que oferecem capacidades avançadas para a análise de séries temporais e detecção de padrões complexos. Para que essas técnicas e métodos sejam precisos é necessário fazer uma análise eficiente sobre a emissão de poluentes com uma grande coleta de dados de fontes diferentes, como dados meteorológicos, padrão de condição, entre outros que sejam importantes para análise. Todavia, temos uma quantidade muito grande de dados, que os podem deixar heterogêneos e não estruturados, o que dificulta a análise e a elaboração de ideias que sejam úteis para a resolução do problema. Sendo assim, os métodos tradicionais que necessitam de estatística descritivas e modelos lineares são insuficientes para identificar padrões e lidar com a complexidade desses dados e se faz necessário a implementação de modelos mais complexos.

Diversos estudos na literatura exploraram a aplicação de técnicas avançadas de ciência de dados e inteligência artificial para a análise de emissões e consumo de veículos. Um estudo apresenta uma abordagem inovadora para prever o consumo de combustível em veículos utilizando RNNs. O modelo manipula dados de velocidade, aceleração e inclinação da estrada como entrada para o monitoramento em tempo real dos veículos, para conseguir realizar uma otimização do planejamento de rotas e redução de CO2[15]. O modelo foi testado em caminhões pesados, demonstrando alta precisão na estimativa do consumo de combustível. A pesquisa destaca os benefícios para a gestão de frotas ao melhorar o planejamento de rotas e reduzir despesas operacionais e emissões de poluentes. Ademais, outro tema que pode ser abordado foi um estudo que utilizou Random Forest para desenvolver um modelo preditivo baseado em machine learning para calcular as emissões de poluentes e o consumo de combustível dos veículos em áreas urbanas[16]. Utilizando dados detalhados de tráfego e características dos veículos, sendo elas velocidade, aceleração, tipo de

combustível e condições da estrada, para treinar e aprovar o modelo. O Random Forest[9] se destaca pois é capaz de lidar com grandes volumes de dados divergentes e identificar interações complexas entre as variáveis, tornando uma boa ferramenta para análises multivariadas no contexto automotivo. Em suma,os exemplos citados estão demonstrando que a aplicação de tecnologias de ciência de dados e inteligência artificial na análise de emissões e eficiência de veículos é essencial para enfrentar os problemas ambientais e de saúde pública associados às emissões veiculares.

Portanto, diante do cenário atual, existe a necessidade de desenvolver abordagens que sejam mais integradas e inovadoras para reduzir as emissões e aumentar a eficiência dos veículos. Nesse contexto, a ciência de dados e a inteligência artificial surgem como ferramentas poderosas e indispensáveis. Por meio da aplicação de análises estatísticas multivariadas e redes neurais recorrentes, será possível identificar ineficiências nos veículos, determinar suas causas e prever comportamentos futuros. Essas técnicas permitirão não apenas otimizar os processos existentes, mas também implementar estratégias proativas para reduzir as emissões de poluentes e melhorar o consumo de combustível. Assim, esta pesquisa busca contribuir para um desenvolvimento sustentável, abordando diretamente o problema das emissões e do consumo dos veículos de forma eficaz e inovadora

2 . Objetivo Do Trabalho

O objetivo deste trabalho é desenvolver e aplicar modelos baseados em Redes Neurais Recorrentes (RNNs) e sua variação, Long Short-Term Memory (LSTM), para prever valores futuros de emissões e consumo de combustível em veículos com base em valores passados. As RNNs e LSTMs são eficazes na análise de séries temporais, pois conseguem capturar dependências de longo prazo e padrões complexos nos dados, permitindo uma modelagem precisa das variações e tendências. A proposta inclui a utilização de dados, como hodômetro e tipo de combustível, para criar modelos que forneçam previsões robustas e relevantes para o setor automotivo, contribuindo para a otimização da eficiência veicular e redução das emissões de poluentes.

Além da modelagem preditiva, este projeto realiza uma análise estatística multivariada utilizando o algoritmo Random Forest, AdaBoost, Boruta, Gradient Boost e Mutual Information para avaliar o impacto das variáveis independentes na variável alvo, como o consumo de combustível ou a emissão de CO2. O Random Forest é uma ferramenta

poderosa que permite identificar as variáveis mais relevantes, mesmo em conjuntos de dados complexos e com grande quantidade de interações. Dessa forma, a análise proporciona resultados valiosos sobre quais fatores têm maior influência no desempenho dos veículos, oferecendo informações críticas para a criação de estratégias que visem à melhoria da eficiência energética e à redução de emissões. Contudo, o trabalho combina técnicas avançadas de aprendizado de máquina e análise estatística para abordar um problema de alta relevância ambiental e econômica, promovendo soluções mais precisas e eficazes para o setor de transporte.

3. Trabalhos Relacionados

A previsão de séries temporais é um campo de pesquisa amplamente explorado, especialmente devido à sua aplicação em diversos contextos, como finanças, análise de trajetórias, emissões de poluentes e eficiência de consumo de combustível em veículos. Diversas abordagens têm sido utilizadas, destacando-se o uso de redes neurais recorrentes (RNN) e suas variações, como a Long Short-Term Memory (LSTM), pela sua capacidade de capturar dependências temporais complexas nos dados. Em um estudo feito por Nelson [17] propõem o uso de redes LSTM para a previsão de movimentos de preços no mercado de ações. O estudo demonstra que as redes LSTM são eficazes em identificar padrões temporais, superando modelos tradicionais de previsão financeira. Esse trabalho destaca a importância do uso de modelos de aprendizado profundo em cenários onde os dados apresentam alta variabilidade temporal [17].

Por outro lado, Xue, Huynh, e Reynolds desenvolveram o modelo hierárquico SS-LSTM para a previsão de trajetórias de pedestres[18]. O estudo ressalta que a estrutura hierárquica das redes LSTM pode ser aplicada a dados espaciais e temporais, resultando em previsões mais precisas em cenários dinâmicos e de alta complexidade, como a movimentação de pedestres em ambientes urbanos[18]. No contexto de eficiência energética e controle de emissões,um estudo analisou o desempenho de motores de ignição por centelha utilizando biogás, gás natural e syngas, com diferentes teores de hidrogênio. O estudo foca na eficiência e emissões de poluentes, destacando a relevância de otimizar o consumo de combustível e reduzir a poluição em aplicações industriais, especialmente em contextos como as indústrias de arroz no Brasil [19].

Ademais, podemos citar o estudo que aborda a previsão de emissões de gases de efeito estufa em redes rodoviárias utilizando aprendizado profundo com modelos LSTM. A pesquisa demonstra que as LSTM são capazes de capturar as dependências temporais presentes nos dados de tráfego, possibilitando previsões mais precisas das emissões, o que é fundamental para a criação de políticas de transporte mais sustentáveis [20]. Além disso, Li realiza uma comparação entre modelos estatísticos tradicionais e modelos de aprendizado profundo, como LSTM, na previsão de emissões diárias. Os resultados mostram que os modelos baseados em aprendizado

profundo apresentam um desempenho superior, especialmente em cenários com alta variabilidade temporal, reforçando a importância da adoção dessas técnicas em análises ambientais [21].

Por fim, a comparação entre modelos ARIMA, LSTM e BiLSTM na previsão de séries temporais financeiras evidencia as vantagens das redes LSTM e BiLSTM sobre os modelos tradicionais. O estudo enfatiza a superioridade dos modelos de redes neurais recorrentes na captura de padrões temporais, o que é essencial em ambientes financeiros voláteis [22].

Além das redes neurais, o algoritmo Random Forest tem se destacado em análises relacionadas a emissões de poluentes e consumo de combustível devido à sua robustez e flexibilidade. Um estudo desenvolveu modelos baseados em Random Forest para prever as taxas de emissão de CO_2, CO, NOx e hidrocarbonetos (HC) em veículos de passageiros sob condições reais de condução no Egito[23]. Os modelos demonstraram alta precisão, com mais de 97% da variância explicada, destacando-se como uma ferramenta eficaz para análises ambientais e desenvolvimento de políticas de controle de emissões [23].

Esses trabalhos mostram a ampla aplicabilidade de modelos como RNN, LSTM e Random Forest na previsão de séries temporais e em análises multivariadas, destacando a importância de seu uso em contextos variados, como o financeiro, ambiental e de eficiência energética, contribuindo para avanços significativos nessas áreas.

4 . Fundamentação Teórica

4.1. Aprendizado de Máquina (Machine Learning) e Redes Neurais

Aprendizado de Máquina (Machine Learning - ML) é uma área da inteligência artificial que se dedica ao desenvolvimento de algoritmos capazes de aprender padrões a partir de dados sem serem explicitamente programados. Em vez de definir regras fixas, o ML cria modelos matemáticos que se ajustam aos dados, encontrando relações, padrões e estruturas subjacentes. Dessa forma, o modelo aprende a generalizar e realizar previsões ou classificações em dados não vistos, assumindo que o conjunto de treinamento seja representativo.

Um importante subconjunto do Machine Learning são as Redes Neurais Artificiais (RNAs), inspiradas vagamente em neurônios biológicos. Uma rede neural simples é composta por nós (neurônios) organizados em camadas (entrada, uma ou mais camadas ocultas e saída). Cada conexão entre neurônios possui um peso, e o treinamento consiste em ajustar esses pesos de modo a minimizar o erro entre a saída prevista e a saída real. Esse ajuste é feito por métodos como o gradiente descendente, que interativamente atualizam os pesos com base no erro do modelo.

Redes Neurais Profundas (Deep Neural Networks - DNNs) surgem quando há múltiplas camadas ocultas entre a entrada e a saída. Quanto mais camadas, mais "profunda" é a rede. Redes profundas têm alta capacidade de representação, permitindo modelar relações complexas e extrair características de alto nível diretamente dos dados. Esse aumento de profundidade, combinado com técnicas de regularização e avanços computacionais, resultou no grande sucesso do Deep Learning em diversos domínios (visão computacional, processamento de linguagem, previsão de séries temporais, entre outros).

4.2. Redes Neurais Recorrentes (RNN) e LSTM

Redes Neurais Recorrentes (RNNs) são redes projetadas para lidar com dados sequenciais (como séries temporais), introduzindo conexões recorrentes que alimentam o estado anterior da rede no estado atual. Assim, a RNN armazena uma "memória" do que já foi visto, permitindo que a previsão atual dependa não apenas da entrada atual, mas também do histórico. Por outro lado, vemos que as RNNs tradicionais enfrentam o problema de gradiente de desaparecimento que é uma condição em que o gradiente do modelo se aproxima de zero no treinamento. Quando o gradiente desaparece, a RNN falha em aprender de forma eficaz com os dados de treinamento, resultando em um ajuste insuficiente, dificultando o aprendizado de dependências de longo prazo.

A Long Short-Term Memory (LSTM) é um tipo específico de Rede Neural Recorrente (RNN) concebido para superar as limitações das RNNs tradicionais, principalmente o problema de gradiente de desaparecimento [26], que é uma condição em que o gradiente do modelo se aproxima de zero no treinamento. Quando o gradiente desaparece, a RNN falha em aprender de forma eficaz com os dados de treinamento, resultando em um ajuste insuficiente. Enquanto as RNNs simples são capazes de lidar com dependências temporais de curto prazo, elas tendem a falhar quando o padrão que precisa ser aprendido está muito distante na sequência, dificultando a captura de relações de longo alcance [24]. A LSTM resolve esse problema por meio de um mecanismo interno mais complexo, estruturado em células de memória e "portas" que regulam o fluxo de informações, o que a torna uma das arquiteturas mais robustas para análise de dados sequenciais em diversas aplicações [25].

Uma célula LSTM difere de um neurônio recorrente simples por manter um estado interno adicional, o "estado da célula"(c_t), além do estado oculto tradicional(h_t). Esse estado interno funciona como uma espécie de "canal" pelo qual informações importantes podem fluir ao longo do tempo, com mecanismos explícitos para adicionar, remover ou ler informações [26].

A célula LSTM possui três portas principais (Forget, Input e Output), que são essencialmente unidades sigmoides (e em um caso, combinações de sigmóide e tanh) que decidem o que esquecer, o que armazenar e o que expor da memória interna.

Porta de Esquecimento (Forget Gate): Decide quais informações do estado anterior devem ser descartadas.

Porta de Entrada (Input Gate): Controla quais informações novas entram na célula de memória

Porta de Saída (Output Gate): Determina quais partes do estado interno serão usadas para a saída.

Por que usar a LSTM?

Retenção de Longo Prazo**:** O estado da célula funciona como uma “linha de memória” ao longo do tempo, permitindo a retenção de informações relevantes por muitas etapas[26].

Superando o Gradiente de Desaparecimento**:** Ao permitir que o gradiente flua ao longo do tempo sem desaparecer rapidamente, a LSTM preserva sinais importantes, mesmo após muitas iterações, superando assim as limitações das RNNs simples [24].

Flexibilidade para Dados Temporais Complexos**:** Em problemas de previsão, como emissões de poluentes ou consumo de combustível, cujo padrão pode depender de eventos distantes no passado, a LSTM consegue “lembrar” desses eventos críticos [25].

Isso permite que a LSTM retenha informações relevantes por longos períodos e descarte informações irrelevantes, superando as limitações das RNNs tradicionais. Por essa capacidade, a LSTM é especialmente adequada para prever variáveis em séries temporais complexas, como emissões ou consumo de combustível, influenciados por eventos passados distantes.

4.3. Métricas de Avaliação

Ao treinar um modelo de aprendizado de máquina ou de redes neurais para previsão, é essencial quantificar o quão próximo o modelo está dos valores reais. Essa avaliação é feita por meio de métricas de erro, que indicam o desvio entre as previsões do modelo e os valores reais.

Mean Squared Error (MSE)

A MSE calcula a média dos erros ao quadrado. Ao elevar a diferença ao quadrado, dá-se um peso maior a erros grandes, tornando a MSE sensível a outliers ou a previsões grosseiramente distantes dos valores reais.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Root Mean Squared Error (RMSE)

O RMSE é a raiz quadrada do MSE. Ao tirar a raiz, retornamos à mesma unidade da variável alvo, tornando o RMSE mais interpretável que o MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

Mean Absolute Error (MAE)

A MAE é a média dos valores absolutos das diferenças entre o valor real e o previsto.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|$$

Mean Absolute Percentage Error (MAPE)

O MAPE expressa o erro médio relativo em porcentagem, comparando a magnitude do erro com o valor real.

$$MAPE(\%) = \frac{100\%}{n}\sum_{i=1}^{n}|\frac{y_i - \widehat{y_i}}{y_i}|$$

4.4. Métodos de Seleção de Variáveis e Análise Multivariada

Problemas com dados multivariados (muitas variáveis) exigem ferramentas para identificar quais fatores são mais relevantes na previsão. Além disso, compreender a importância das variáveis auxilia na interpretabilidade dos modelos, especialmente quando se lida com modelos complexos como RNNs ou LSTMs.

4.4.1 Random Forest

O Random Forest[9] é um algoritmo de aprendizado de máquina classificado como um método de ensemble. Ele consiste na criação de múltiplas árvores de decisão durante a fase de treinamento, dividindo a base de dados em subgrupos até que os subgrupos tenham elementos em comum, essa divisão ocorre pelo conceito de impureza e entropia, via índice Gini [9]. A previsão final do modelo é obtida pela média dos resultados gerados por cada uma dessas árvores, proporcionando assim uma maior robustez e precisão nas predições

4.4.2 AdaBoost (Adaptive Boosting)

O método conhecido como AdaBoost [11,12], é mais uma técnica que utiliza Árvores de Decisão, embora sua implementação seja diferente do método do Random Forest. A principal distinção entre AdaBoost e Random Forest reside na complexidade das árvores geradas: enquanto o Random Forest gera uma coleção de Árvores de Decisão que podem ser bastante complexas, o AdaBoost emprega uma série de árvores mais simples e de baixa profundidade. Por ser fundamentado em Árvores de Decisão, o AdaBoost também emprega o conceito de impureza, utilizando o índice de Gini [11,12], para determinar as divisões mais eficazes.

4.4.3 Gradient Boost

O Gradient Boost também é baseado em Árvores de Decisão[12]. Possui uma abordagem para combinar as árvores que é distinta tanto do AdaBoost quanto do Random Forest. O método Gradient Boost[12] utiliza princípios similares de impureza para determinar as divisões mais eficazes. Porém, ao invés de usar o índice Gini, o Gradient Boost emprega o RMSE de Friedman, que representa a raiz do erro quadrático médio.

4.4.4 Boruta

O Boruta[10] é uma metodologia avançada baseada em Random Forests, que tem como objetivo identificar e classificar as características mais relevantes em um modelo preditivo. Esse processo é iterativo ele começa criando 'variáveis ocultas', que são dados aleatórios adicionados ao modelo para assegurar que não exista correlação com a variável de interesse que se pretende analisar. Em cada iteração, um modelo de Random Forest é aplicado para realizar a regressão necessária e prever o resultado da variável alvo. Além disso, a importância de cada característica é avaliada usando o mecanismo interno do Random Forest. Características que apresentam menor importância do que as variáveis ocultas são eliminadas, pois essas

variáveis como são aleatórias não têm relação com o que está sendo analisado. O processo se repete até que restem somente as características mais significativas, até que todas sejam descartadas, ou até que se atinja o limite máximo de iterações estabelec

4.4.5 Mutual Information (MI)

O mutual information é uma medida não negativa que quantifica o grau de dependência entre elas. Um valor igual a zero indica independência entre as variáveis. Essa métrica é útil para estabelecer um ranking de importância das variáveis, ao calcular a informação mútua entre cada variável existente e a variável global que no caso do projeto seria as variáveis de consumo e emissão. Dessa forma, quanto maior a informação mútua entre uma variável explicativa e a variável alvo, maior será a dependência entre elas. Por outro lado, quanto menor a informação mútua, mais independentes serão as variáveis.

Esses métodos permitem a identificação das variáveis mais influentes, ajudando a entender o fenômeno e tornando o treinamento do modelo mais eficiente ao reduzir o conjunto de entradas.

5. Métodos Propostos

5.1 Preparação e Limpeza dos Dados

Cada arquivo com dados de uma determinada placa (veículo) é carregado de uma pasta específica. Os dados são então ordenados cronologicamente pela coluna "DATA TRANSACAO". Essa ordenação é fundamental, pois estamos lidando com um problema de previsão baseado em séries temporais, onde a relação temporal entre as amostras é essencial.

Tratamento de Dados Faltantes:

Inicialmente, é feito um diagnóstico da presença de dados faltantes. Colunas com mais de 95% de valores ausentes são removidas, por não fornecerem informação suficiente. Em seguida, os valores faltantes remanescentes são preenchidos com zero ou as linhas são removidas, dependendo da criticidade da coluna. O objetivo é garantir que os dados estejam completos ou minimamente consistentes, evitando distorções devido a valores ausentes.

Codificação de Variáveis Categóricas:

Todas as colunas do tipo "object" são convertidas para representação numérica por meio de LabelEncoder. Essa abordagem é útil para permitir que as redes neurais trabalhem apenas com valores numéricos, indispensáveis para o treinamento adequado dos modelos. Embora um encoding one-hot ou outros métodos pudessem ser usados, a codificação por inteiro (LabelEncoder) é um primeiro passo simples e eficaz, dada a natureza do problema.

Remoção de Colunas com Baixa Variabilidade:

Colunas com variância muito baixa (próxima de zero) ou sem variabilidade significativa são removidas. Essas colunas tendem a não contribuir para o aprendizado do modelo, já que não fornecem informação discriminante entre amostras. Eliminar tais variáveis ajuda a reduzir a dimensionalidade e o ruído no treinamento.

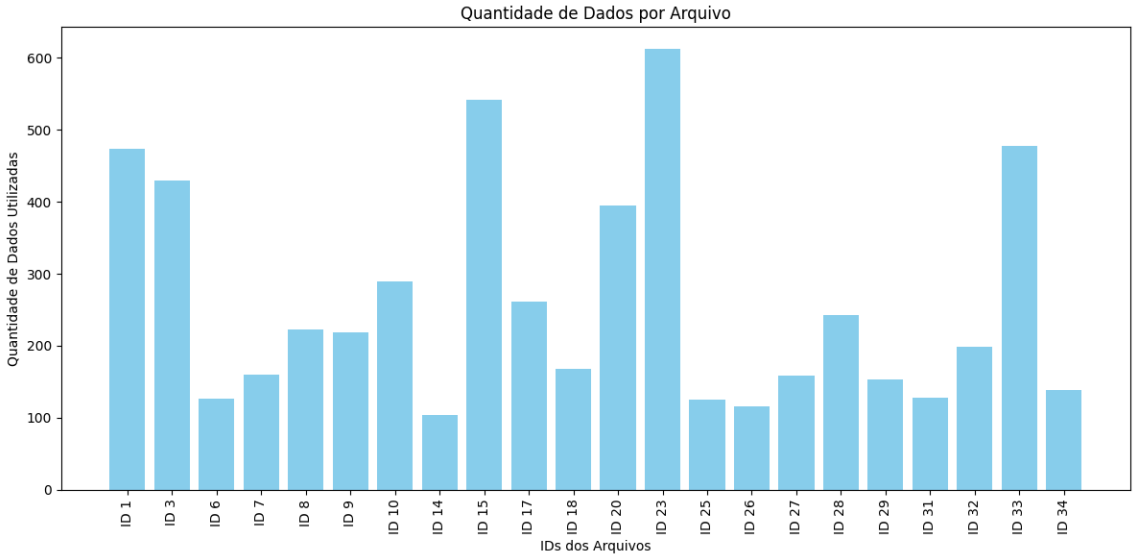
Verificação e Remoção de Colunas sem Variabilidade:

Caso ainda restem colunas com variabilidade nula, elas são removidas. Esse passo reforça o ponto acima, garantindo que apenas variáveis relevantes sejam utilizadas.

Remoção de Colunas de Data após Ordenação:

Qualquer coluna de data/datetime, após ter sido utilizada para ordenação, é removida das features. O objetivo é não fornecer ao modelo diretamente a informação temporal bruta, já que a sequência temporal já é preservada pelo ordenamento e pela criação das janelas de entrada.

Após o tratamento de dados temos o gráfico de quantos dados foram utilizados



5.2 Criação de Sequências Temporais

Para prever o valor futuro da variável alvo, é necessário fornecer ao modelo uma sequência de valores passados. Assim, criamos janelas de tempo (time_steps = 30) onde, dado um conjunto de 30 amostras passadas, o modelo deve prever o valor alvo imediatamente subsequente. Este processo é implementado pela função create_sequences, que transforma a série temporal original em tensores tridimensionais (amostra, passos de tempo, features). Essa abordagem reflete a natureza temporal do problema, permitindo que o modelo aprenda dependências de curto e, potencialmente, de longo prazo entre as variáveis de entrada.

Normalização dos Dados

Antes do treinamento, tanto as features (X) quanto a variável alvo (y) são normalizadas usando MinMaxScaler. A normalização garante que todas as variáveis estejam em escalas comparáveis, o que facilita o treinamento de redes neurais, evitando que variáveis com magnitudes muito diferentes dominem o processo de otimização.

Divisão em Conjuntos de Treino e Teste (Validação Cruzada Temporal)

Ao invés de simplesmente dividir os dados em treino e teste de forma estática, é utilizada a técnica de Time Series Split (validação cruzada para séries temporais). O TimeSeriesSplit cria múltiplos folds, mantendo a ordem cronológica dos dados. A cada fold, uma porção inicial dos dados é usada para treino e uma porção subsequente para teste, permitindo estimar a capacidade de generalização do modelo em vários cenários temporais.

Este procedimento:

Evita leakage temporal: Não se utiliza dados futuros para treinar o modelo que será testado com dados passados.

Fornece Métricas Mais Confiáveis: Ao rodar vários folds, obtém-se uma estimativa mais robusta do desempenho médio do modelo.

No experimento, foi utilizado k=5 folds.

5.3 Modelagem: Arquiteturas RNN e LSTM

Foram desenvolvidos dois modelos de rede neural recorrente

5.3.1 Modelo RNN (SimpleRNN)

Camada de entrada com a dimensão (time_steps, número de features).

Duas camadas SimpleRNN, a primeira com units=100 e a segunda com units=50, ambas com ativação 'tanh'.

Uma camada densa final com apenas um neurônio para prever o valor contínuo da variável alvo.

Otimizador: RMSprop com taxa de aprendizagem de 1e-4.

Função de perda: mean_squared_error.

A RNN simples é uma abordagem inicial. Embora possa capturar dependências temporais de curto prazo, tende a ter dificuldades com dependências mais longas devido ao problema de "vanishing gradients".

5.3.2 Modelo LSTM (Long Short-Term Memory)

Camada de entrada (time_steps, número de features).

Duas camadas LSTM empilhadas, a primeira com units=128 e a segunda com units=64, ambas com ativação 'tanh'.

Uma camada densa final com um neurônio para saída.

Otimizador: Adam com taxa de aprendizagem de 1e-3.

Função de perda: mean_squared_error.

As LSTMs são especialmente desenhadas para lidar com dependências de longo prazo, graças aos seus mecanismos internos de "portas" que controlam o fluxo de informação ao longo do tempo.

5.3.2 Treinamento e Parâmetros

Número de Épocas:

O modelo RNN foi treinado por 100 épocas e o modelo LSTM por 150 épocas. Esses valores foram escolhidos empiricamente, buscando um equilíbrio entre complexidade, capacidade de aprendizagem e tempo de treinamento.

Tamanho do Lote (Batch Size):

Utilizou-se batch_size=32, um valor padrão frequentemente eficaz, oferecendo um bom compromisso entre estabilidade do gradiente e eficiência computacional.

5.4 Validação Interna

Em cada fold do TimeSeriesSplit, parte dos dados servem como validação, permitindo monitorar a perda na validação (val_loss) e evitando o overfitting. Isso garante ajuste fino durante o treinamento e auxilia na compreensão da estabilidade do modelo.

5.4.1 Avaliação e Métricas

Para avaliar o desempenho dos modelos preditivos, foram consideradas as seguintes métricas:

RMSE (Root Mean Squared Error):

Mede o erro quadrático médio, penalizando fortemente grandes discrepâncias entre previsto e observado. É uma métrica intuitiva, pois retorna o erro na mesma unidade da variável alvo.

MAE (Mean Absolute Error):

Mede o erro absoluto médio, fornecendo uma visão mais robusta contra outliers do que o RMSE (embora o clipping já tenha reduzido o impacto destes).

MAPE (Mean Absolute Percentage Error):

Mede o erro relativo médio em porcentagem, o que facilita a interpretação de quão distante, em termos percentuais, as previsões estão do valor real.

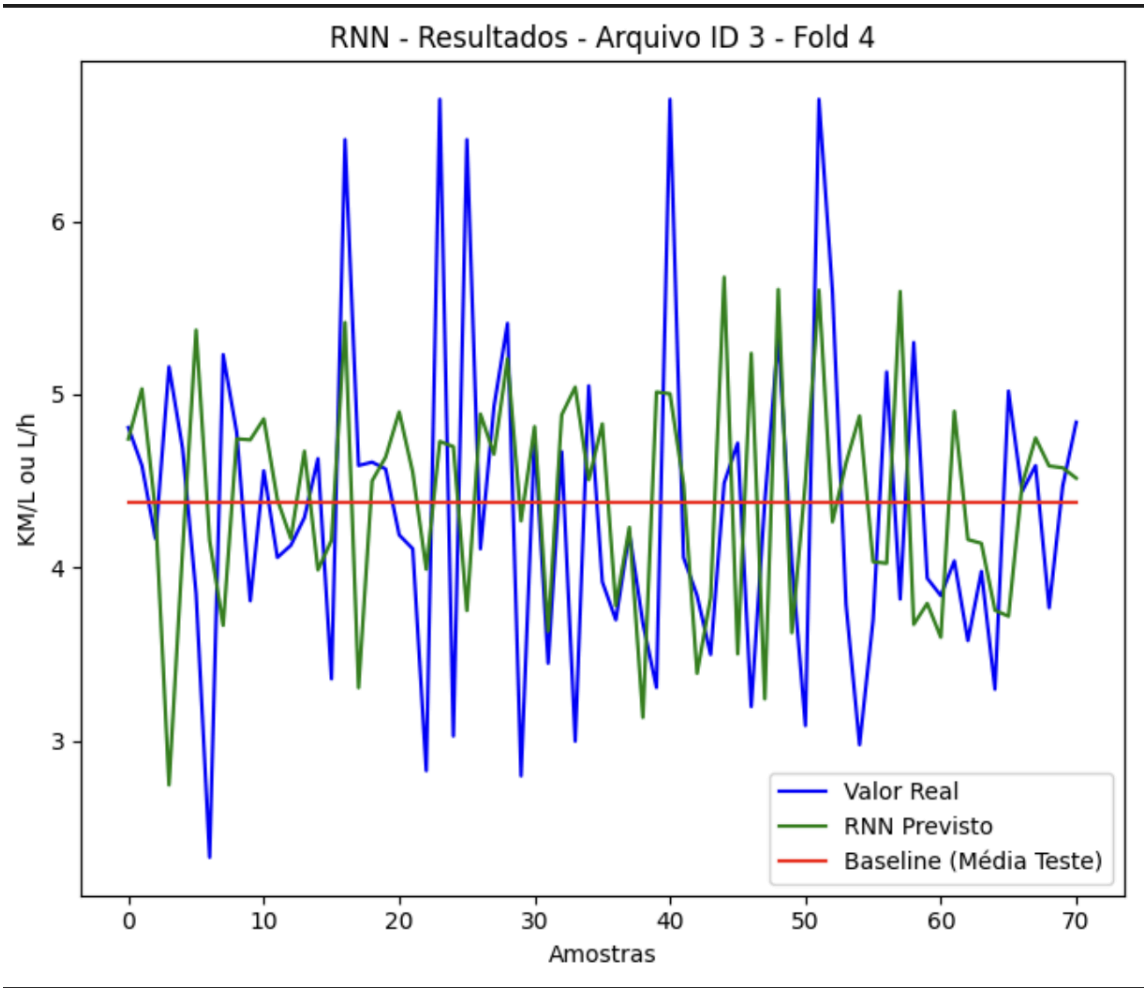
Além disso, utiliza-se um Baseline simples, calculando a média dos valores reais do conjunto de treino e utilizando esta média como previsão para todas as amostras de teste. As métricas do baseline são comparadas com as métricas da RNN e LSTM, avaliando se os modelos complexos ultrapassam uma solução trivial.

5.5 Visualizações

Foram gerados diversos gráficos para análise:

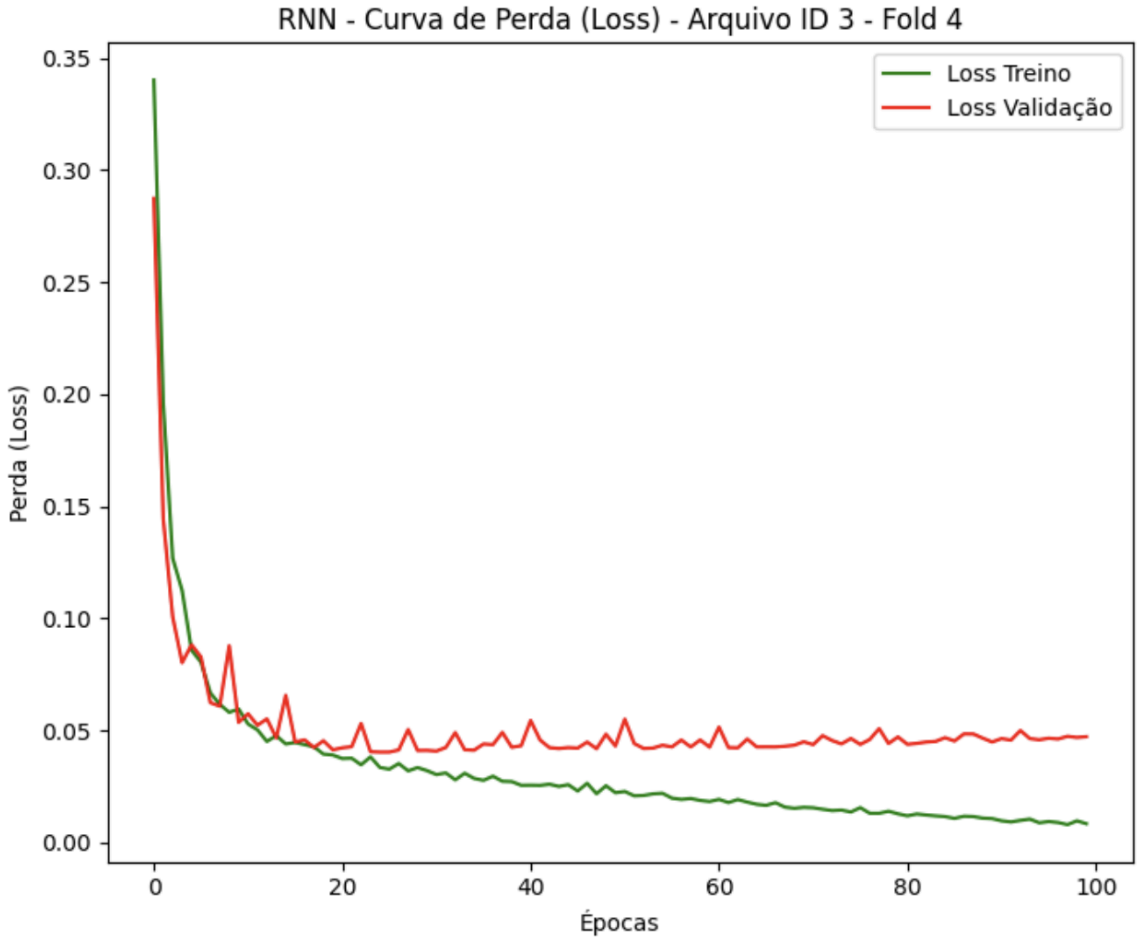
Valores Reais vs. Preditos (Curvas):

Permite avaliar visualmente a aderência das previsões à série real, tanto para o modelo quanto para a baseline.



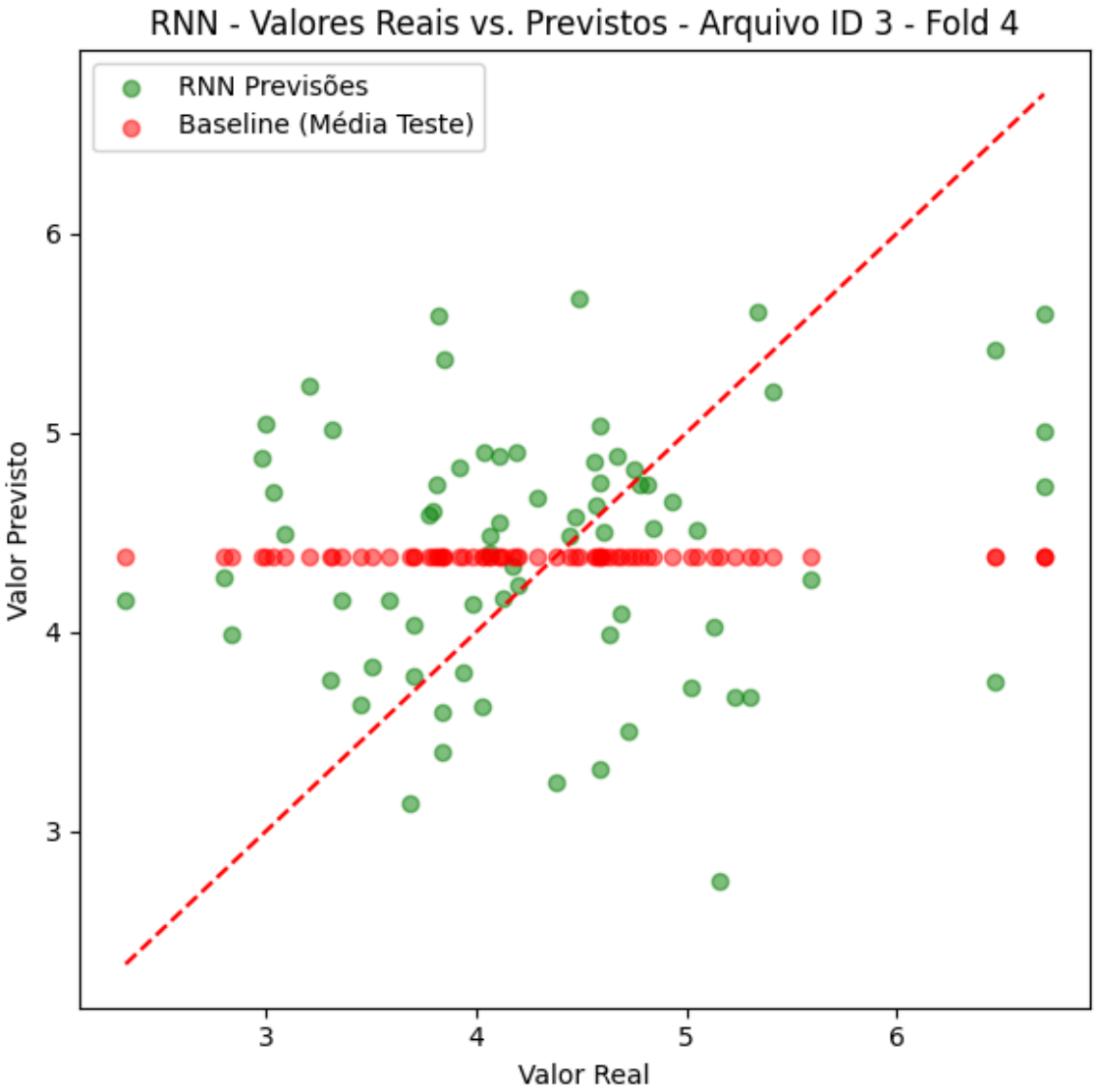
Curvas de Perda (Loss) do Treinamento:

Exibem a evolução da função de perda no conjunto de treino e validação ao longo das épocas, indicando se o modelo está convergindo ou sobreajustando.



Gráficos de Dispersão (Scatter Plot):

Mostram a distribuição dos valores previstos versus os valores reais, ideal para verificar a correlação e a tendência de sub ou superestimativa.



Métricas por Fold:

São mostradas para cada fold do TimeSeriesSplit, bem como suas médias, permitindo compreender a robustez e a estabilidade do modelo ao longo do tempo.

5.6 Métodos de Seleção de Variáveis e Análise Multivariada

A função carrega_dados foi desenvolvida para consolidar dados armazenados em múltiplos arquivos Excel, padronizá-los e realizar ajustes básicos de limpeza e formatação. Abaixo, descrevemos o funcionamento da função em detalhes:

Identificação dos arquivos Excel no diretório: Utiliza-se a biblioteca glob para identificar todos os arquivos .xlsx presentes no diretório especificado. Essa abordagem garante que novos arquivos adicionados sejam automaticamente incluídos no processo.

Leitura e concatenação dos arquivos: Os arquivos identificados são lidos com pd.read_excel, e os dados são concatenados utilizando a função pd.concat. O resultado é um único DataFrame contendo todos os registros.

Padronização de strings: Todas as colunas contendo strings são convertidas para letras maiúsculas. Essa padronização facilita buscas, comparações e análises subsequentes.

Remoção de colunas irrelevantes: Colunas consideradas desnecessárias para a análise, como LITROS e KM RODADOS OU HORAS TRABALHADAS, são removidas.

Resultado final: A função retorna um DataFrame consolidado e padronizado, pronto para o processamento adicional.

2. Codificação de Dados Categóricos

A função codifica_para_numerico converte dados categóricos (strings ou valores não numéricos) em representações numéricas utilizando o método LabelEncoder. Isso é necessário para que algoritmos de aprendizado de máquina possam processar esses dados.

Identificação de colunas categóricas: A função verifica se o tipo de dado de cada coluna é object. Se verdadeiro, considera-se que a coluna contém valores categóricos.

Aplicação do LabelEncoder: Para cada coluna categórica, utiliza-se o LabelEncoder para mapear os valores únicos da coluna para números inteiros. Antes da codificação, os valores são convertidos para strings para evitar erros.

Manutenção de colunas numéricas: Colunas já numéricas ou que não requerem codificação são copiadas diretamente para o novo DataFrame.

Armazenamento de codificadores: Para permitir a decodificação ou reutilização posterior, o LabelEncoder utilizado para cada coluna é armazenado em uma lista.

Resultado final: O DataFrame resultante contém todas as colunas em formato numérico, garantindo compatibilidade com algoritmos analíticos e de aprendizado de máquina.

3. Normalização dos Dados

A função normaliza_dados realiza a normalização dos dados para que todas as variáveis tenham uma escala uniforme, uma etapa essencial para modelos sensíveis à magnitude dos valores.

Remoção de colunas de datas: Colunas com valores do tipo datetime são identificadas e removidas, pois não são adequadas para normalização numérica.

Codificação de colunas categóricas e booleanas:

Colunas categóricas são convertidas para números utilizando o LabelEncoder.

Colunas booleanas são transformadas para inteiros (0 ou 1), garantindo a compatibilidade.

Normalização com StandardScaler: O StandardScaler é utilizado para ajustar os dados para que tenham média zero e desvio padrão igual a 1. Essa técnica é essencial para modelos que dependem da escala dos dados, como regressão logística ou redes neurais.

Conversão para DataFrame: Após a normalização, os dados são convertidos de volta para um DataFrame com as mesmas colunas originais, permitindo um uso mais intuitivo.

Esse tratamento de dados foi realizado em todos os modelos abaixo.

5.6.1 Métodos

Métodos	Abordagem	Vantagens	Limitação	Aplicação Principal	Treinamento	Métricas
Gradient Boosting	Ensemble	Captura relações complexas e não lineares	Sensível a ajustes de hiperparâmetros	Seleção de variáveis e predição simultâneas	GridSearchCV é utilizado para otimizar hiperparâmetros	RMSE, MAE e MAPE.
AdaBoost	Ensemble	Foco em melhorar erros de classificação ou regressão difíceis	Sensível a ruídos nos dados	Seleção incremental de variáveis importantes	GridSearchCV é utilizado para otimizar hiperparâmetros	RMSE, MAE e MAPE.

Boruta	Wrapper	Alta precisão e robustez na seleção de variáveis	Alto custo computacional	Seleção robusta de variáveis	RandomForestRegressor é utilizado para treinar o modelo com as variáveis selecionadas.	RMSE, MAE e MAPE.
Mutual Information	Estatística	Identifica dependências não lineares	Não indica a direção da relação	Análise exploratória inicial	Modelos regressivos como RandomForestRegressor são treinados com as variáveis selecionadas.	RMSE, MAE e MAPE.
Random Forest	Ensemble	Simplicidade e boa performance em dados complexos	Não lida bem com variáveis altamente correlacionadas	Seleção e ranqueamento de variáveis	GridSearchCV é utilizado para otimizar hiperparâmetros,	RMSE, MAE e MAPE.

Ensemble:

Essa abordagem combina múltiplos modelos fracos, como árvores de decisão, para criar um modelo robusto. Métodos baseados em ensemble, como Boosted Models e Random Forest, são eficazes para lidar com grandes volumes de dados e capturar interações complexas entre as variáveis, além de oferecer medidas de importância durante o processo de predição.

Wrapper:

Métodos wrapper, como o Boruta, avaliam o impacto das variáveis diretamente no desempenho de um modelo preditivo. Eles iteram sobre diferentes combinações de variáveis para identificar as mais relevantes, mas têm um custo computacional mais elevado devido ao treinamento repetitivo de modelos.

Estatística:

Métodos estatísticos, como a Mutual Information, analisam relações entre variáveis de forma matemática, identificando padrões e dependências não lineares. Eles são rápidos e eficientes para análises iniciais, mas podem ser limitados por não capturar interações mais complexas ou a direção das relações

6. Análise Dos Resultados

Iremos analisar os resultados obtidos por esses métodos, com uma tabela que mostra a média de notas de todos os teste que foram realizados, primeiro iremos analisar para a variável de consumo

Variável	Nota Média
KM/LITRO OU LITROS/HORA	0.3797
HODÔMETRO OU HORÍMETRO	0.2803
ENDEREÇO	0.0550

Tabela 1 - Ranking Ranking de importância de variáveis para a variável de consumo

Com base nos resultados podemos concluir

KM/LITRO OU LITROS/HORA: Esta variável é relacionada ao consumo de combustível do veículo, sendo um indicador fundamental para ser utilizado. Os modelos identificaram essa variável como a mais importante, pois ela reflete a quantidade de combustível utilizada por quilômetro rodado ou por hora, impactando diretamente a eficiência e os custos operacionais do veículo.

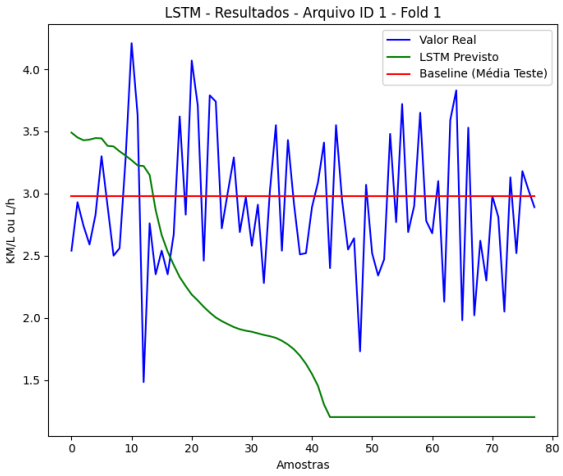
HODÔMETRO OU HORÍMETRO: Indicando a distância total percorrida pelo veículo ou o tempo total de operação, esta variável foi considerada importante em vários modelos. Ela fornece dados históricos sobre o uso do veículo, permitindo análises de desgaste e eficiência ao longo do tempo.

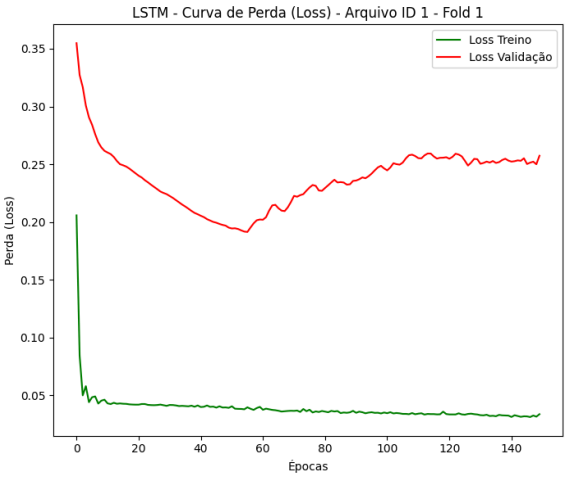
ENDEREÇO: Essa variável geográfica emerge como importante devido à correlação com fatores como topografia, qualidade das estradas e condições de tráfego, que afetam o consumo de combustível. Áreas montanhosas ou com trânsito intenso tendem a ter maior consumo de combustível.

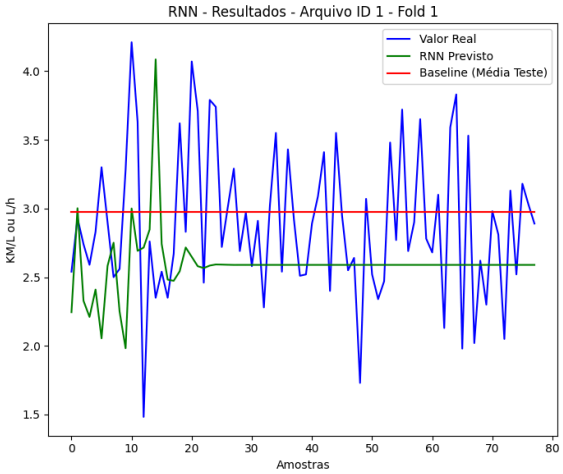
Resultados RNN e LSTM

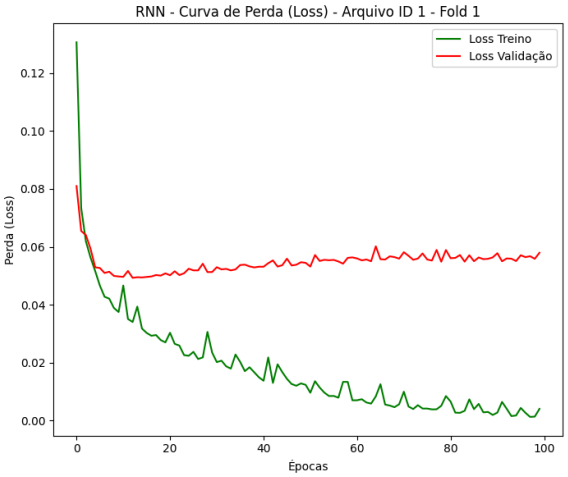
Arquivo ID1

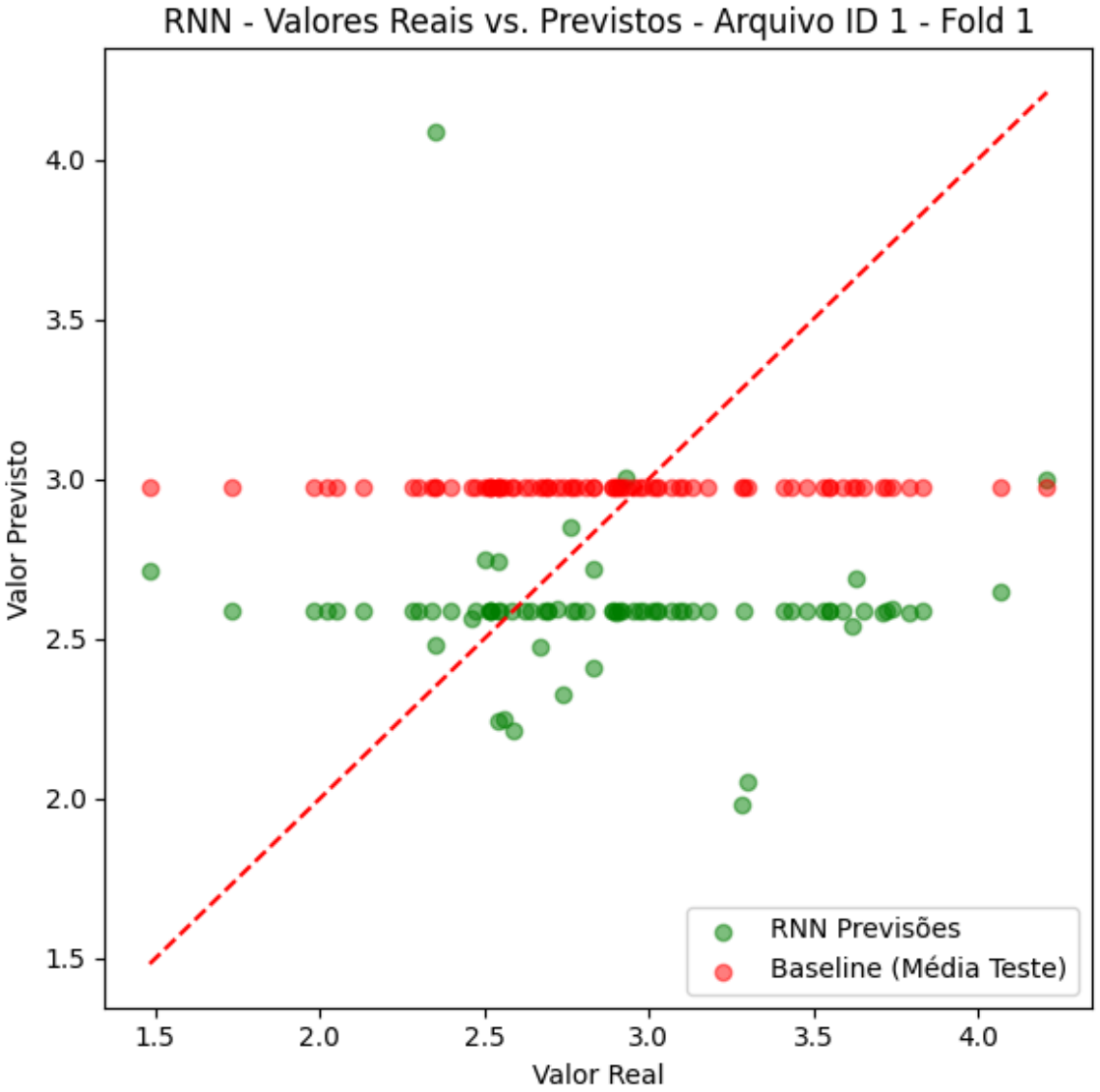
Fold 1

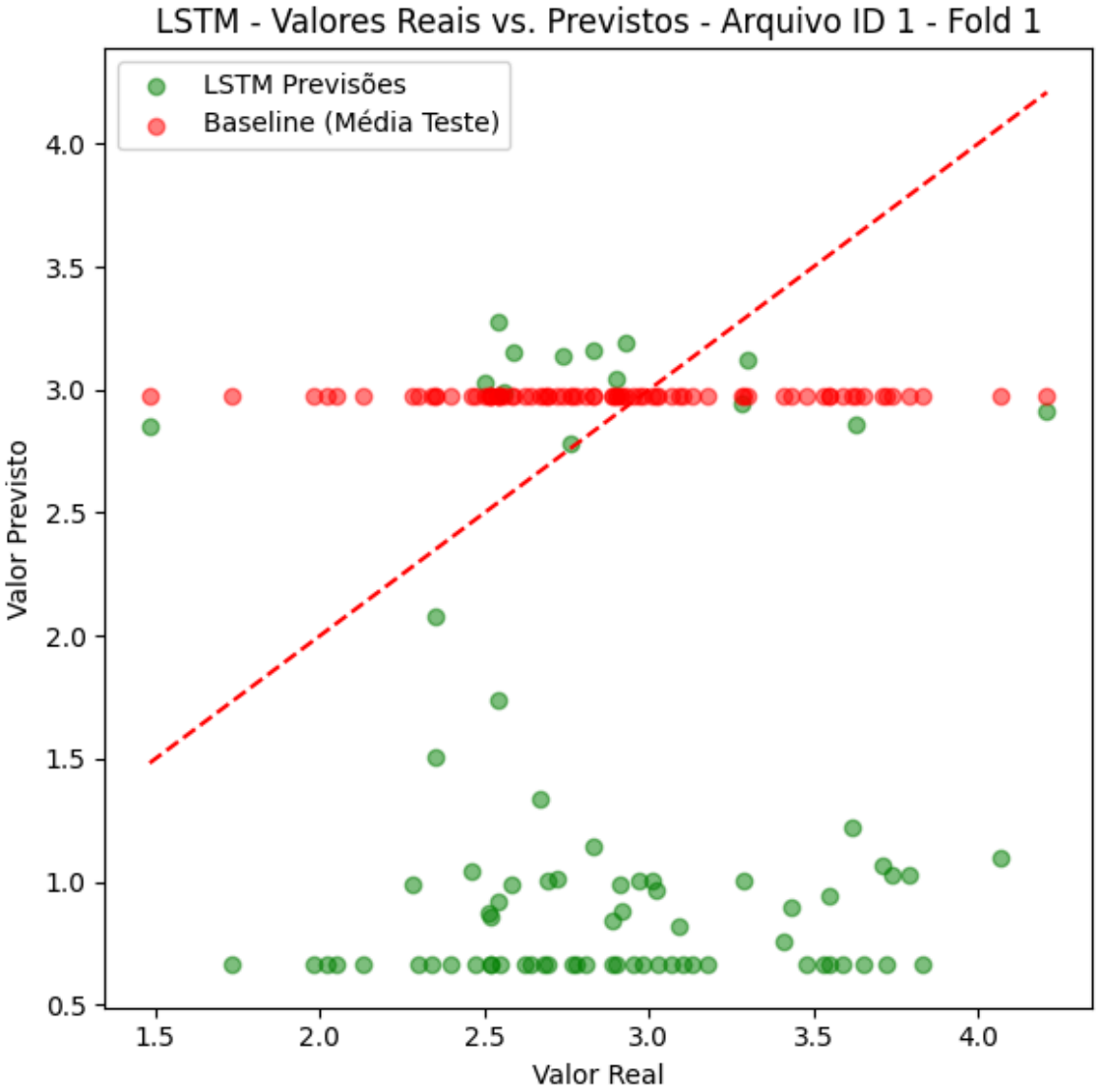

LSTM - Resultados - Arquivo ID 1 - Fold 1
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 1 - Fold 1
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 1 - Fold 1
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 1 - Fold 1
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 1 - Fold 1
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
RNN Previsões
Baseline (Média Teste)

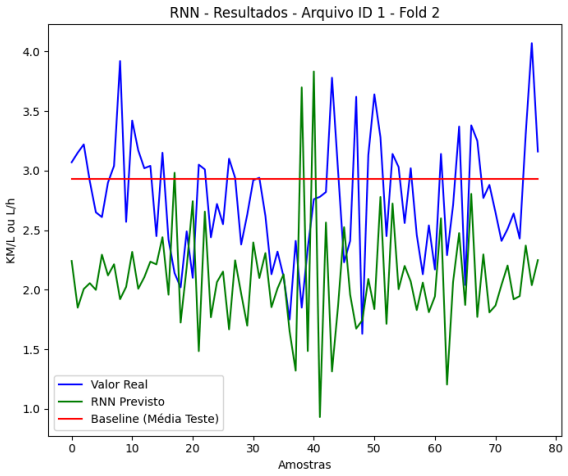


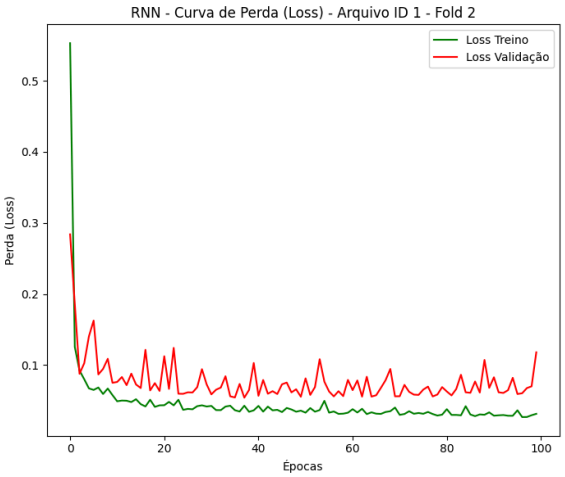
Baseline (Média do Teste) Fold 1 -> RMSE: 0.5426, MAE: 0.4386, MAPE: 16.51%

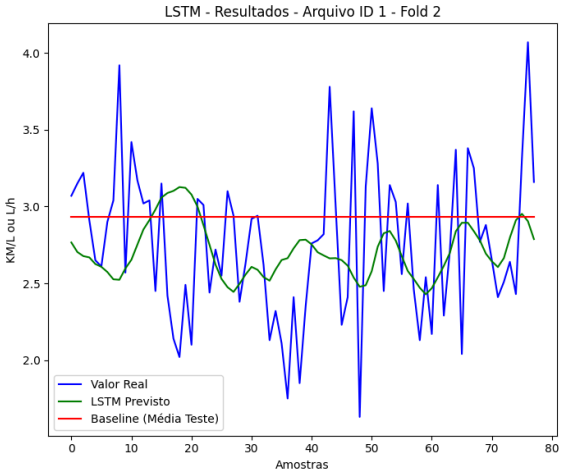
RNN Fold 1 -> RMSE: 0.6595, MAE: 0.5099, MAPE: 17.27%

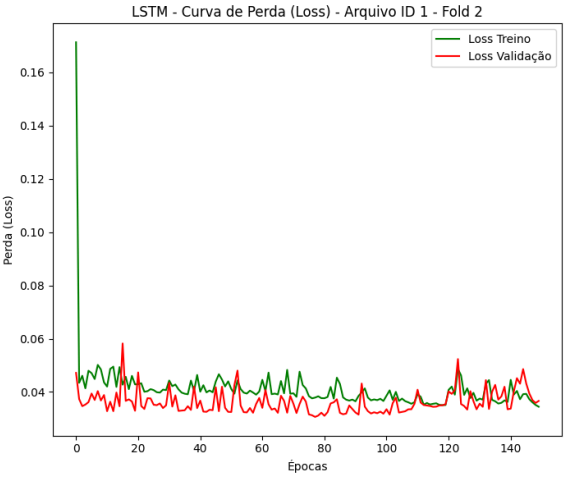
LSTM Fold 1 -> RMSE: 1.3899, MAE: 1.2383, MAPE: 42.52%

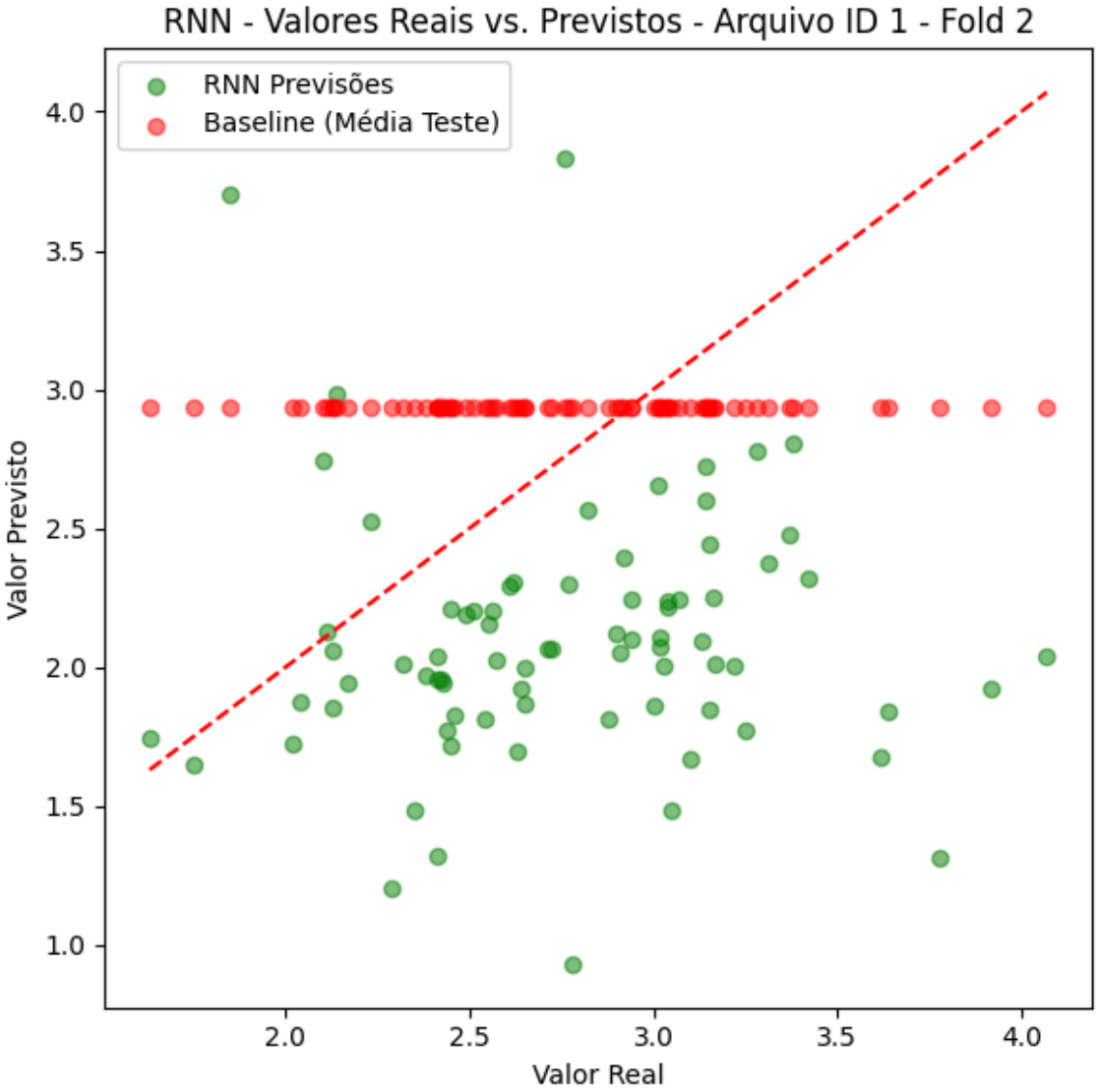
Fold 2

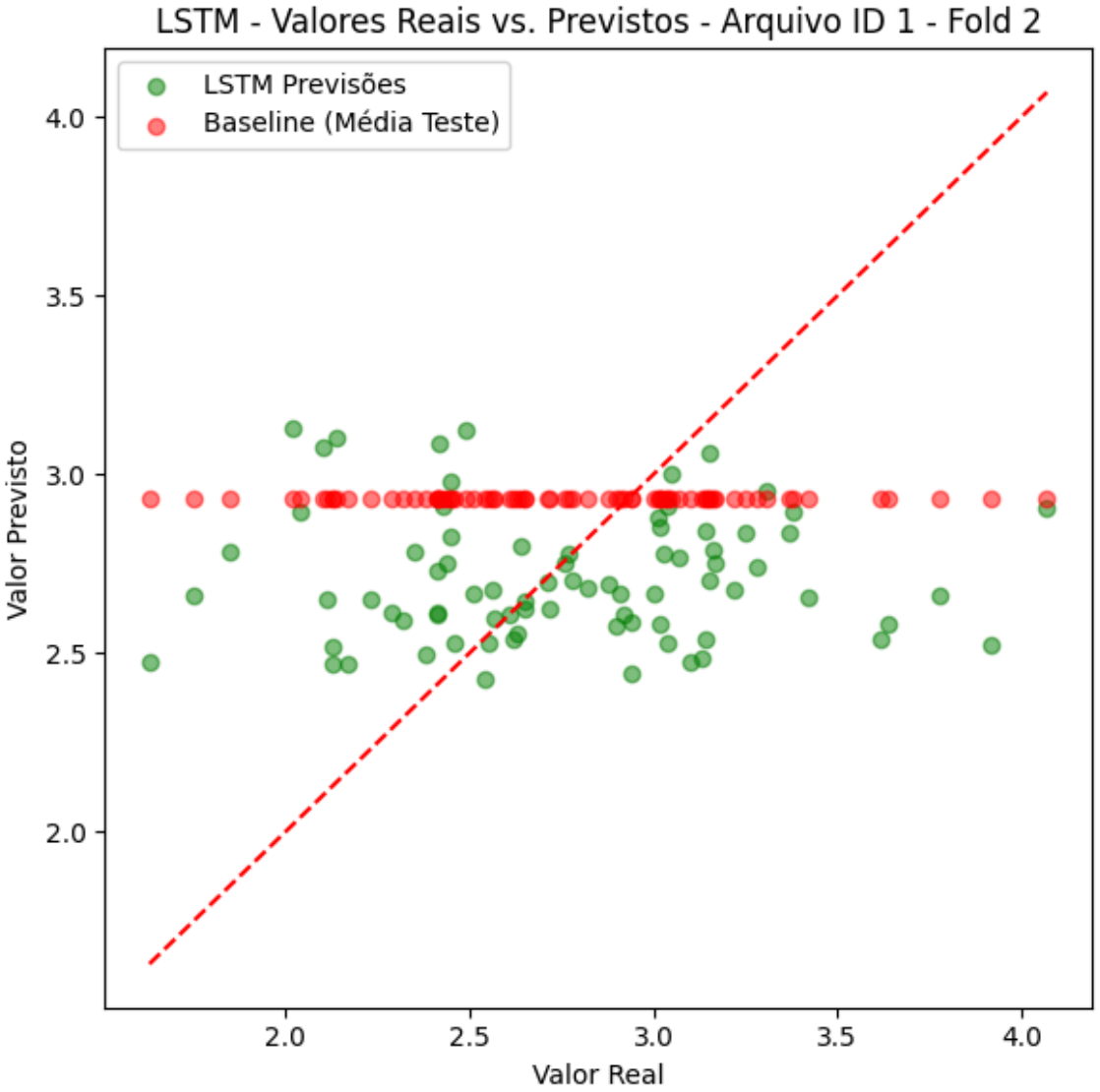

RNN - Resultados - Arquivo ID 1 - Fold 2
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 1 - Fold 2
Perda (Loss)
Épocas
Loss Treino
Loss Validação


LSTM - Resultados - Arquivo ID 1 - Fold 2
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 1 - Fold 2
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 1 - Fold 2
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.0
2.0
2.5
3.0
3.5
4.0

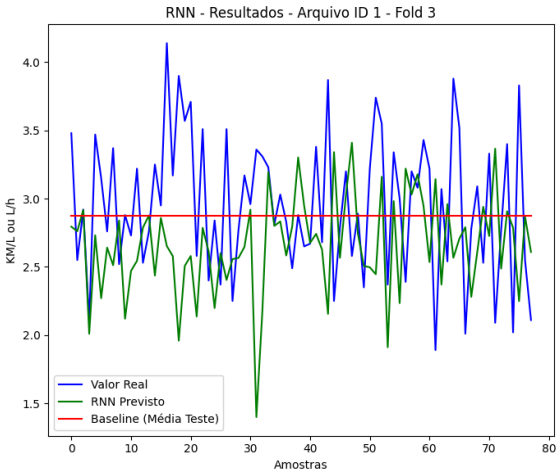


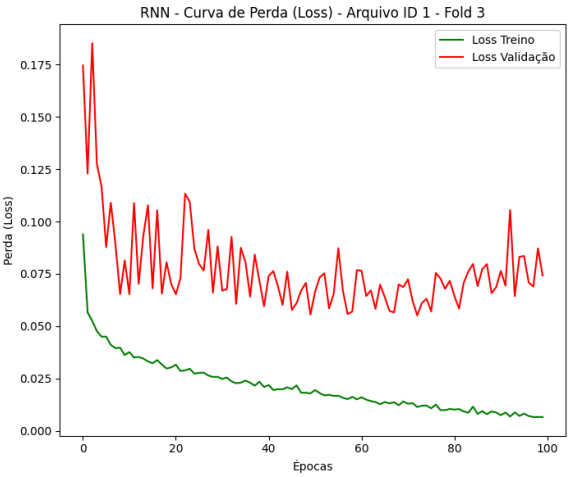
Baseline (Média do Teste) Fold 2 -> RMSE: 0.5206, MAE: 0.4203, MAPE: 16.97%

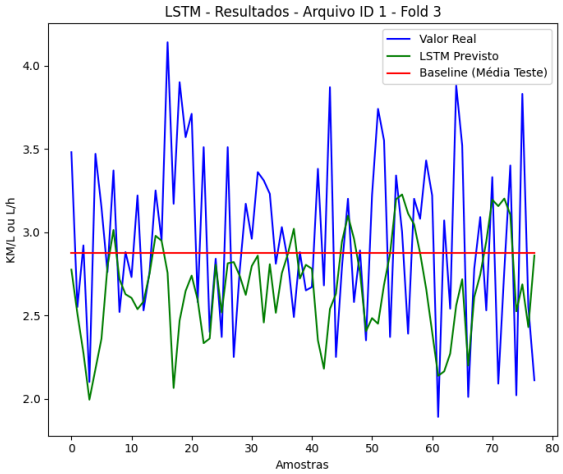
RNN Fold 2 -> RMSE: 0.9413, MAE: 0.7902, MAPE: 27.54%

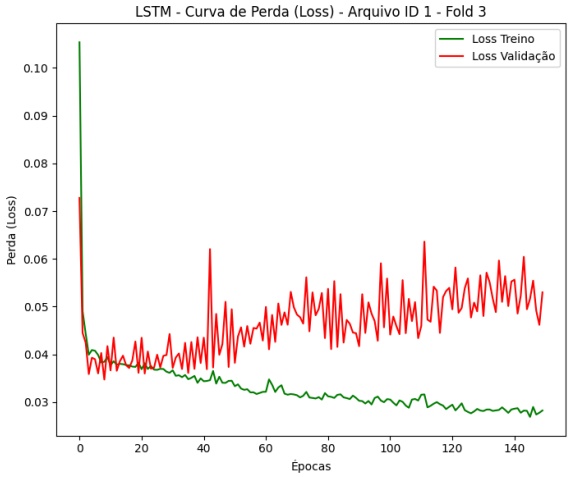
LSTM Fold 2 -> RMSE: 0.5244, MAE: 0.4080, MAPE: 15.36%

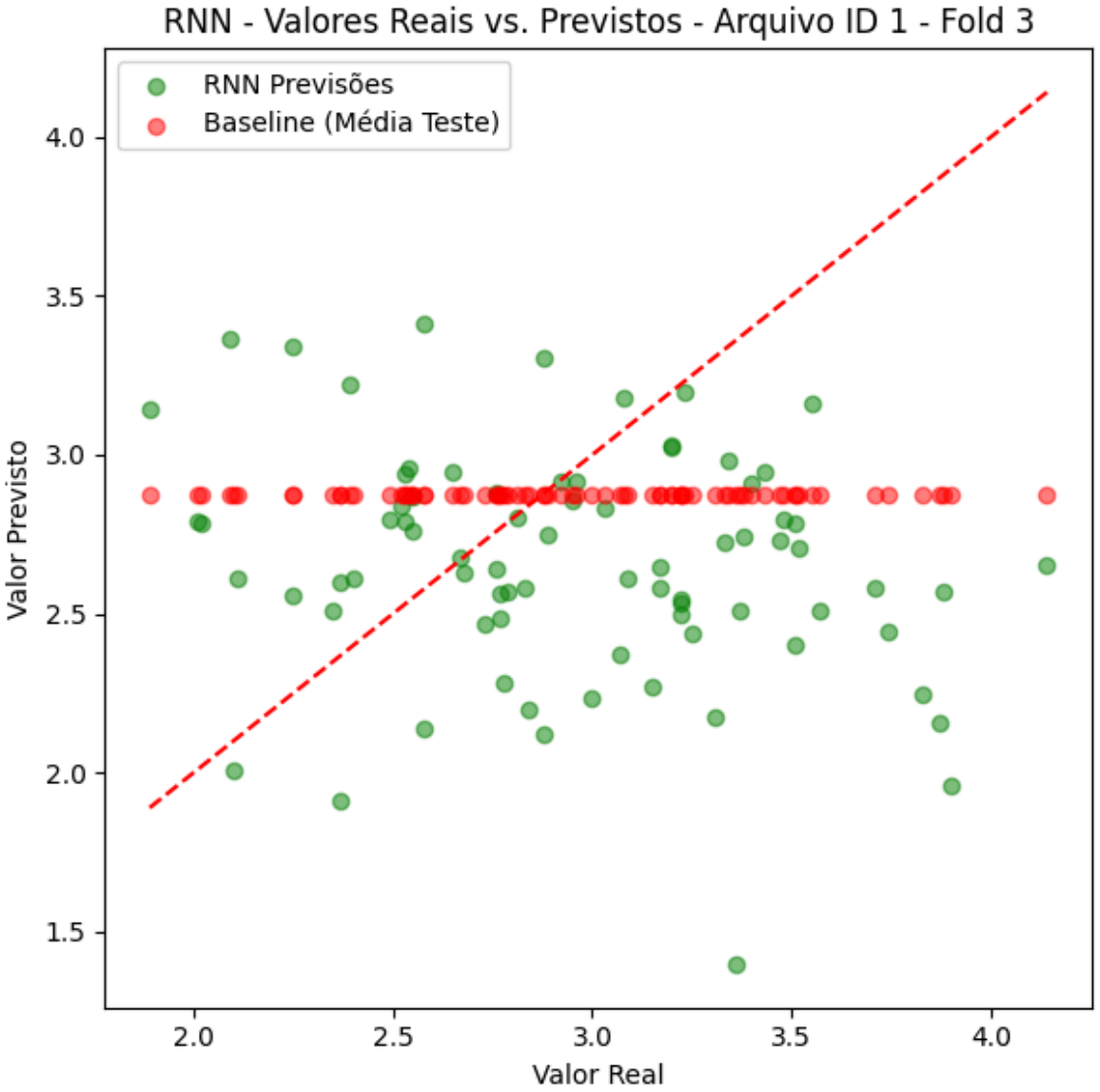
Fold 3

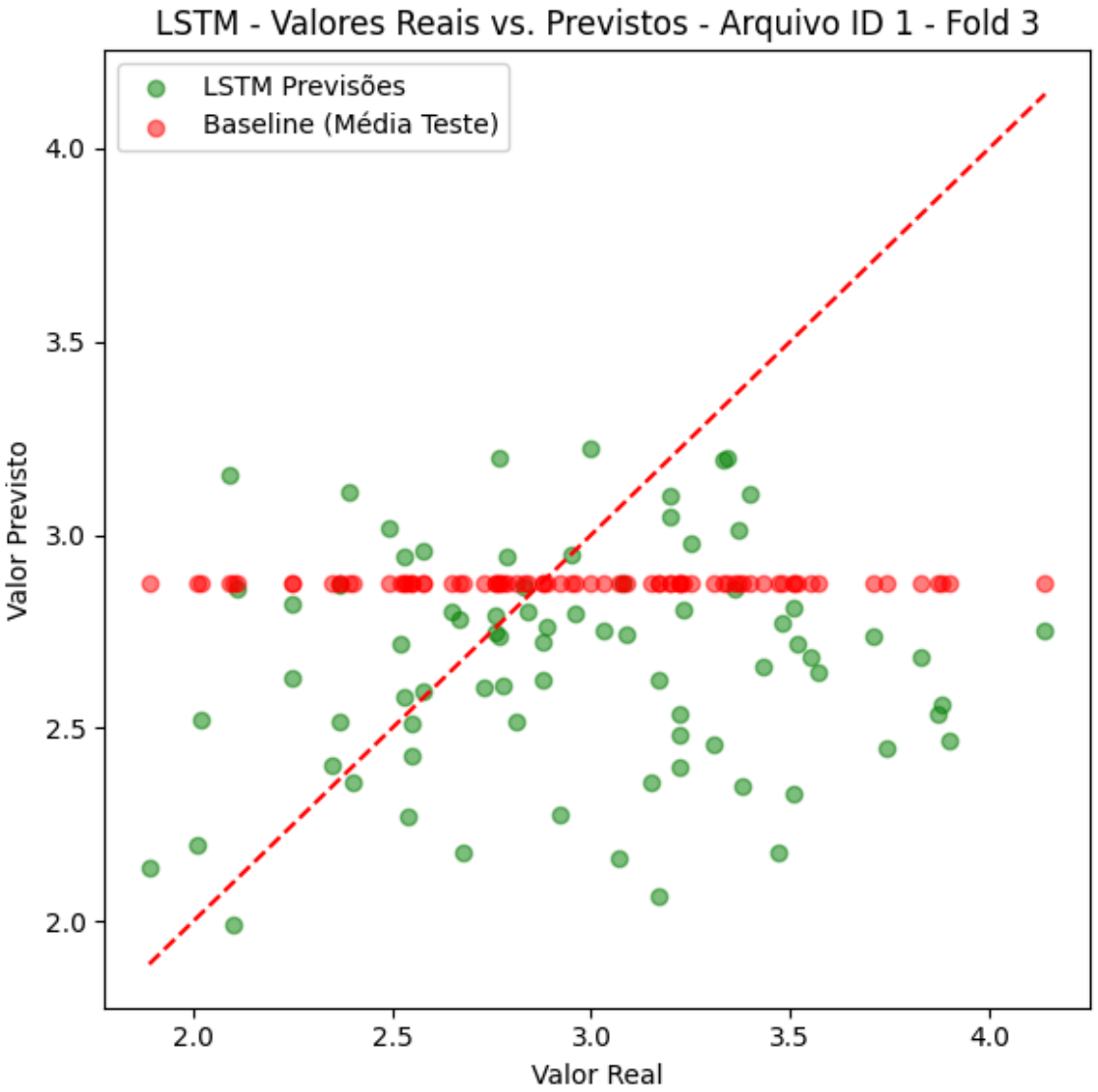

RNN - Resultados - Arquivo ID 1 - Fold 3
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 1 - Fold 3
Perda (Loss)
Épocas
Loss Treino
Loss Validação


LSTM - Resultados - Arquivo ID 1 - Fold 3
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 1 - Fold 3
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 1 - Fold 3
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
2.0
2.5
3.0
3.5
4.0

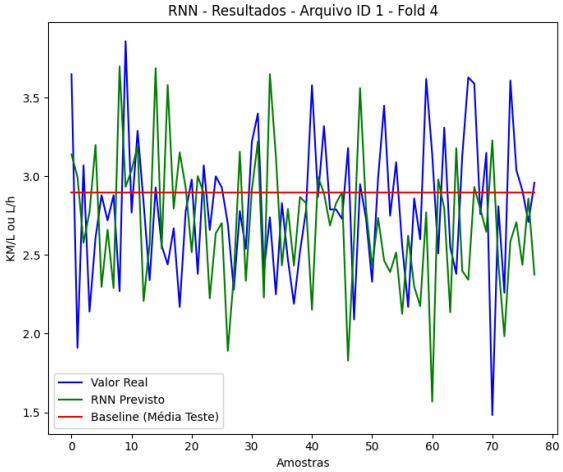


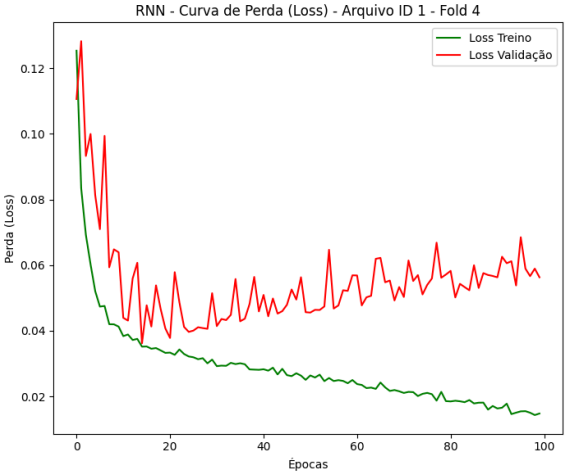
Baseline (Média do Teste) Fold 3 -> RMSE: 0.5136, MAE: 0.4238, MAPE: 14.69%

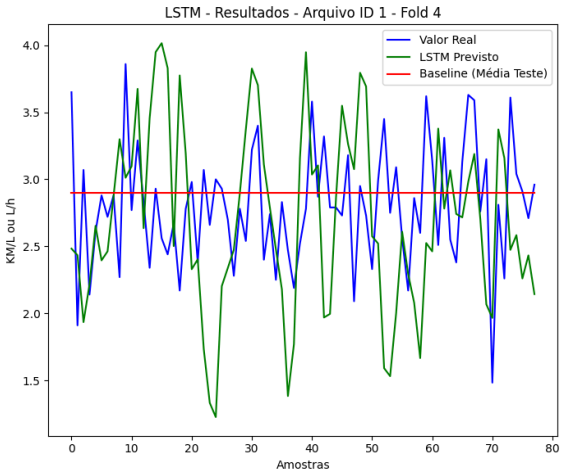
RNN Fold 3 -> RMSE: 0.7463, MAE: 0.5887, MAPE: 19.60%

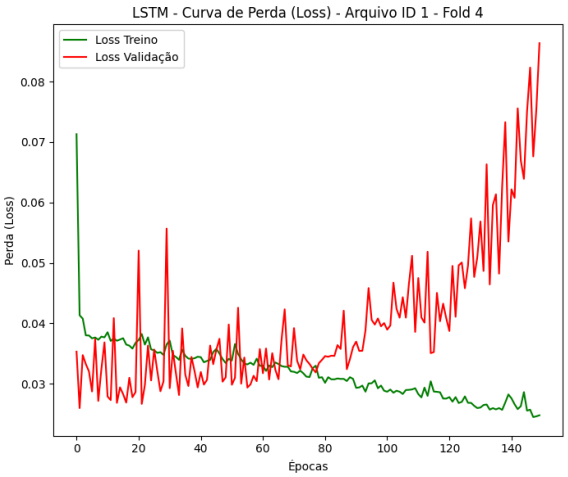
LSTM Fold 3 -> RMSE: 0.6308, MAE: 0.4840, MAPE: 15.64%

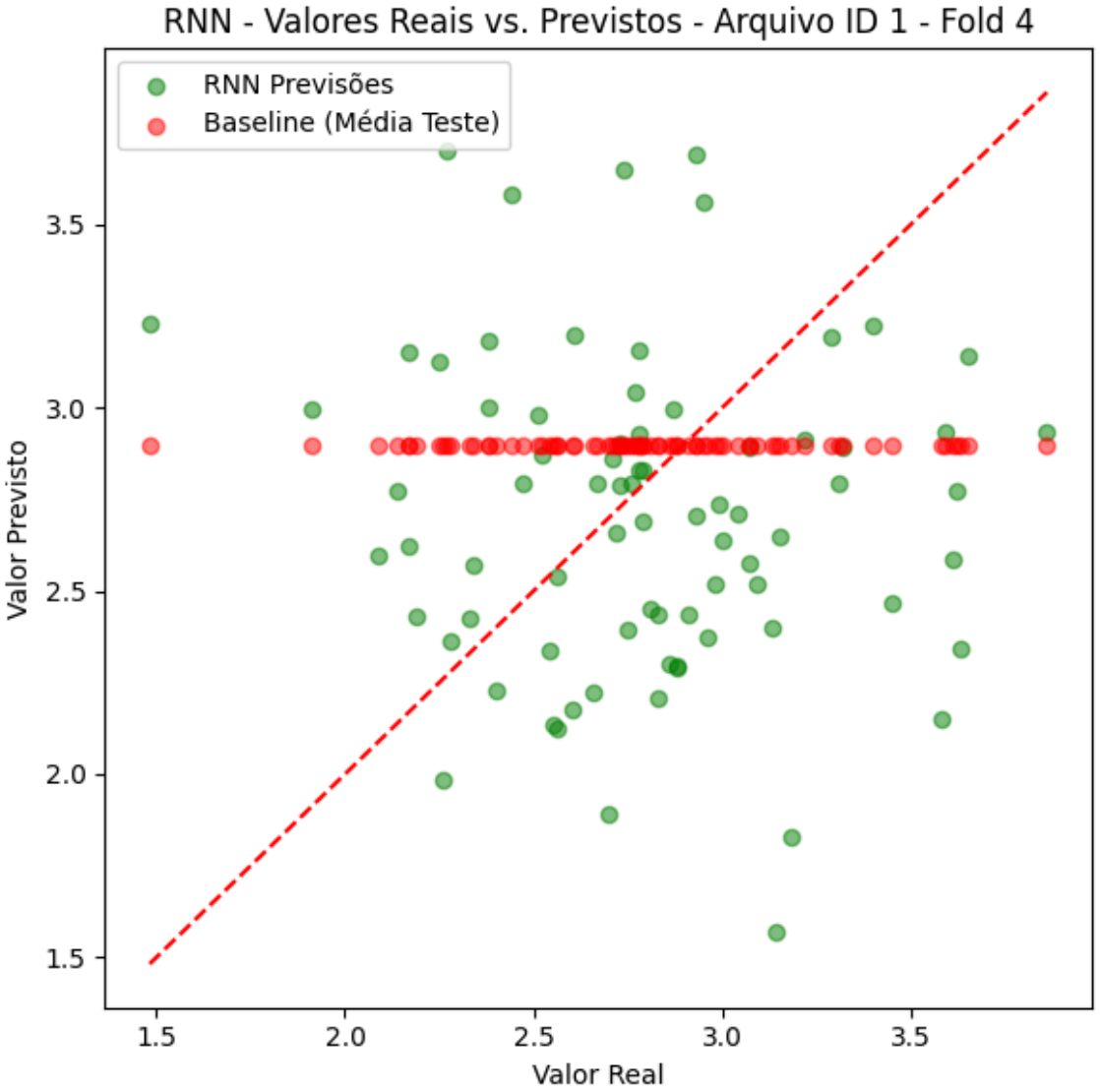
Fold 4

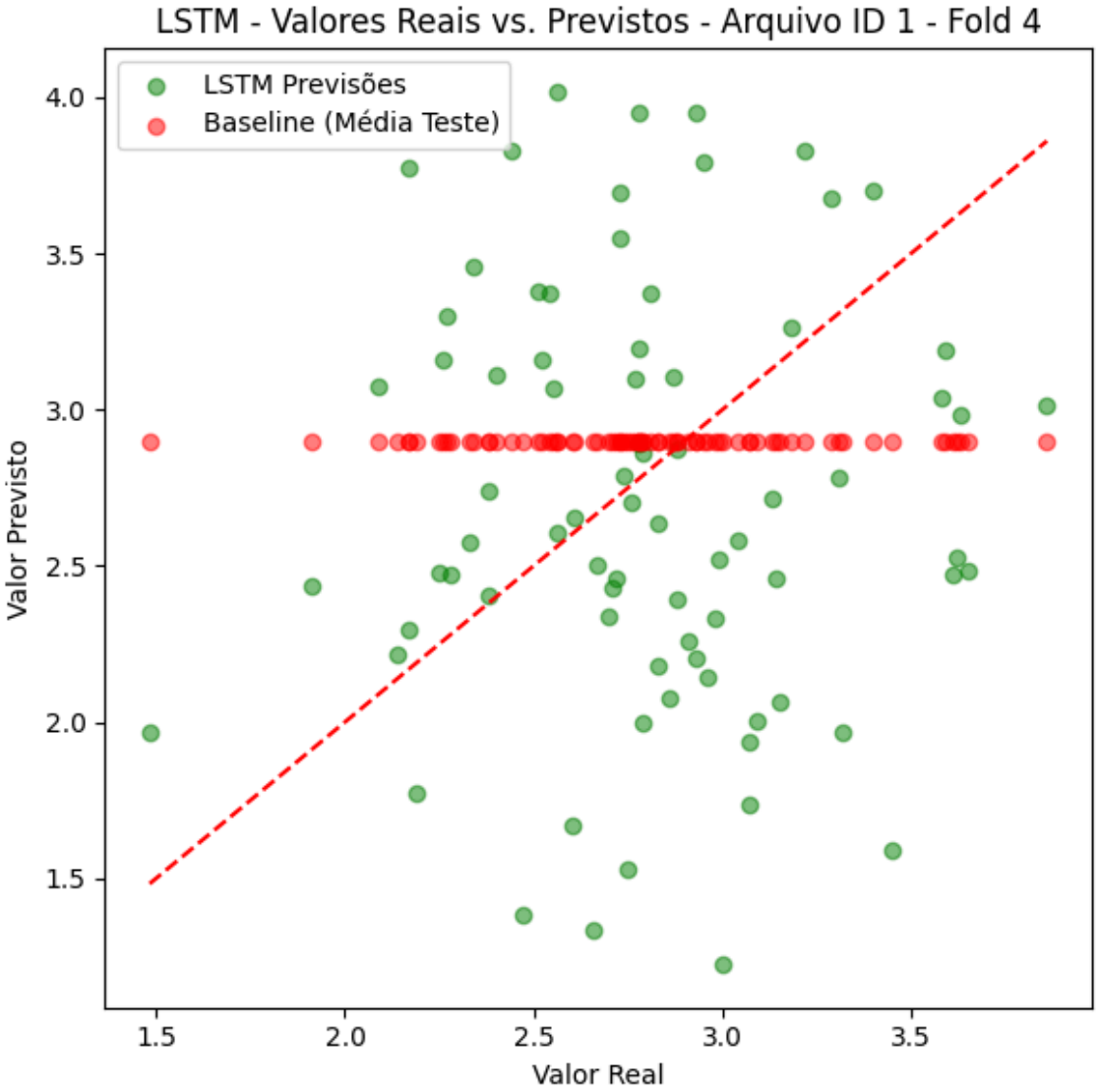

RNN - Resultados - Arquivo ID 1 - Fold 4
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 1 - Fold 4
Perda (Loss)
Épocas
Loss Treino
Loss Validação


LSTM - Resultados - Arquivo ID 1 - Fold 4
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 1 - Fold 4
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 1 - Fold 4
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
1.5
2.0
2.5
3.0
3.5

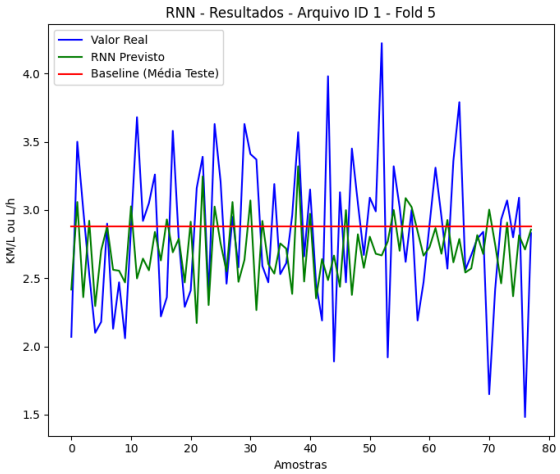


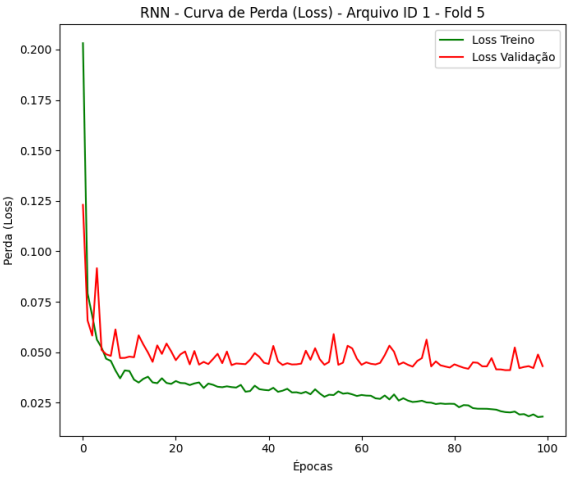
Baseline (Média do Teste) Fold 4 -> RMSE: 0.4545, MAE: 0.3559, MAPE: 13.87%

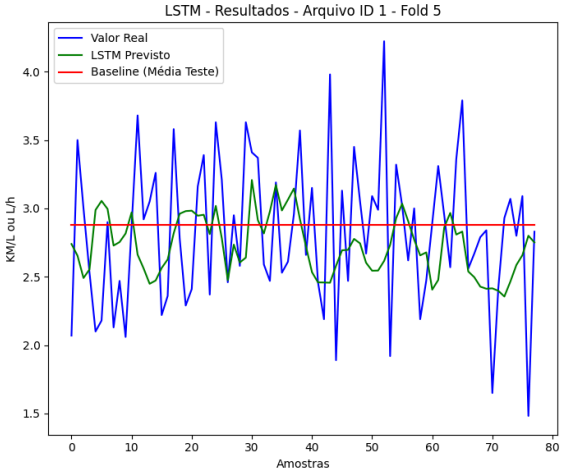
RNN Fold 4 -> RMSE: 0.6501, MAE: 0.5204, MAPE: 19.35%

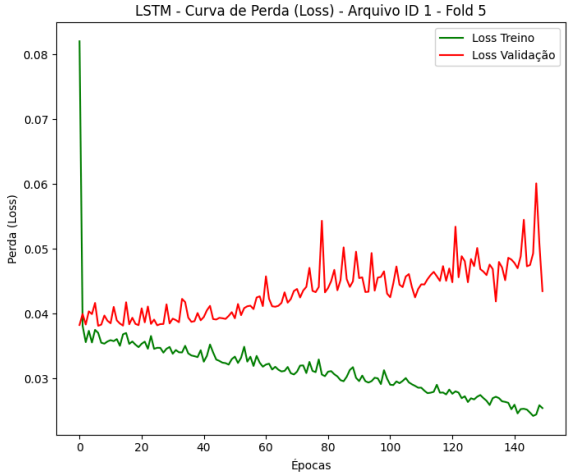
LSTM Fold 4 -> RMSE: 0.8052, MAE: 0.6698, MAPE: 24.15%

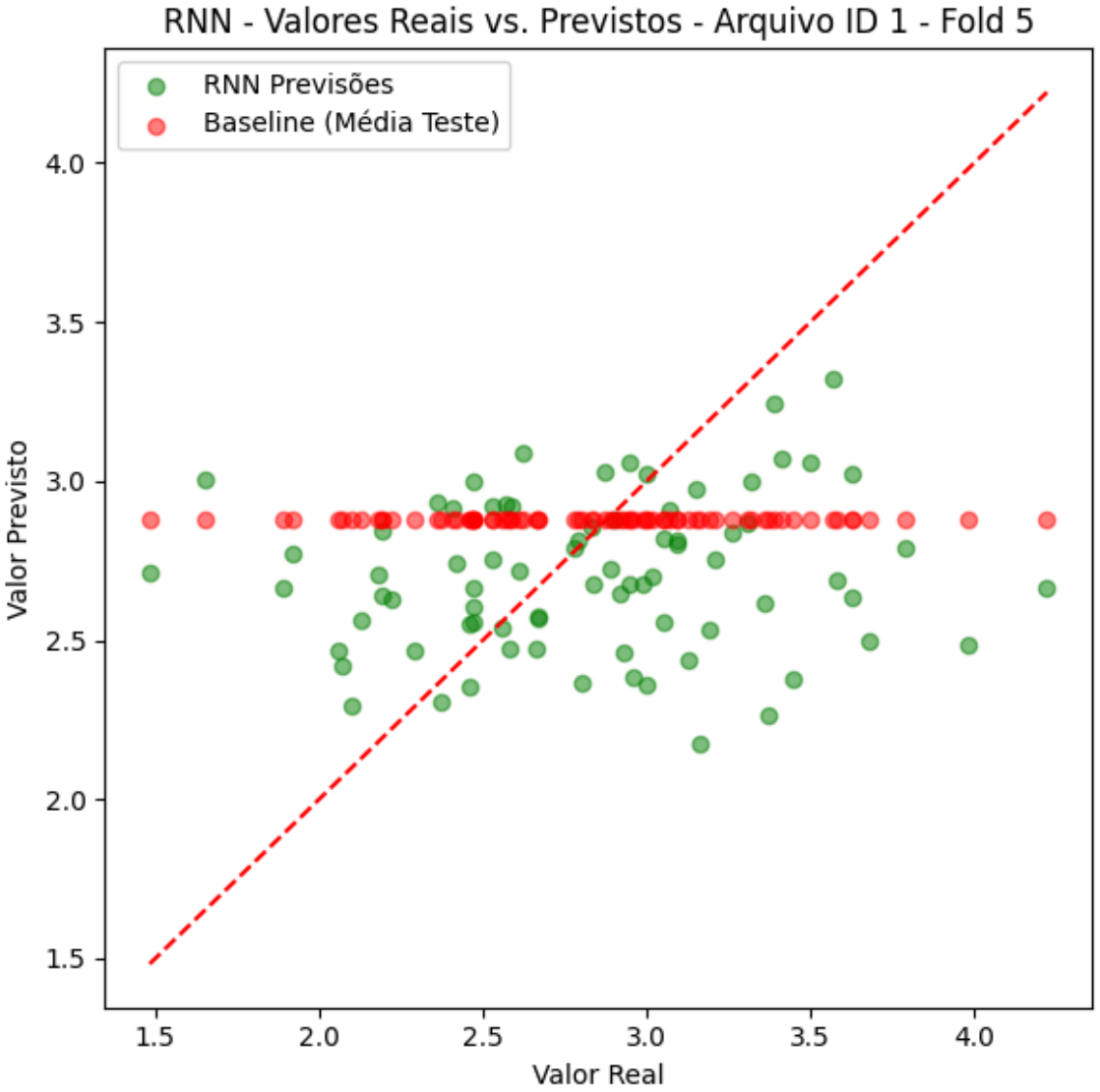
Fold 5

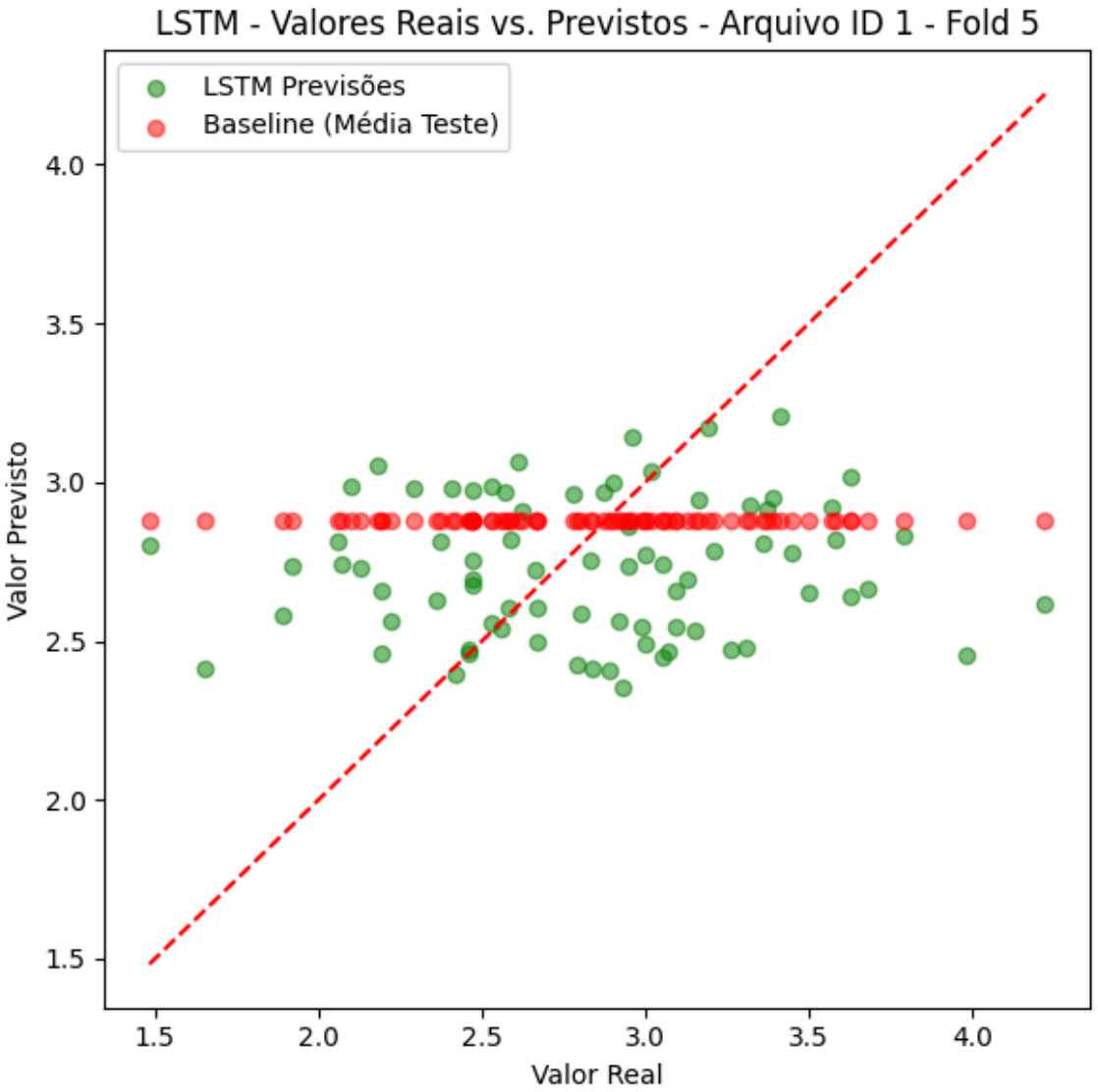

RNN - Resultados - Arquivo ID 1 - Fold 5
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 1 - Fold 5
Loss Treino
Loss Validação
Perda (Loss)
Épocas


LSTM - Resultados - Arquivo ID 1 - Fold 5
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 1 - Fold 5
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 1 - Fold 5
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
1.5
2.0
2.5
3.0
3.5
4.0



Baseline (Média do Teste) Fold 5 -> RMSE: 0.5368, MAE: 0.4317, MAPE: 16.77%

RNN Fold 5 -> RMSE: 0.5689, MAE: 0.4384, MAPE: 16.04%

LSTM Fold 5 -> RMSE: 0.5713, MAE: 0.4608, MAPE: 16.94%

Média de todos os folds do arquivo ID 1

Baseline (Média do Teste) Média sobre 5 folds -> RMSE: 0.5136, MAE: 0.4141, MAPE: 15.76%

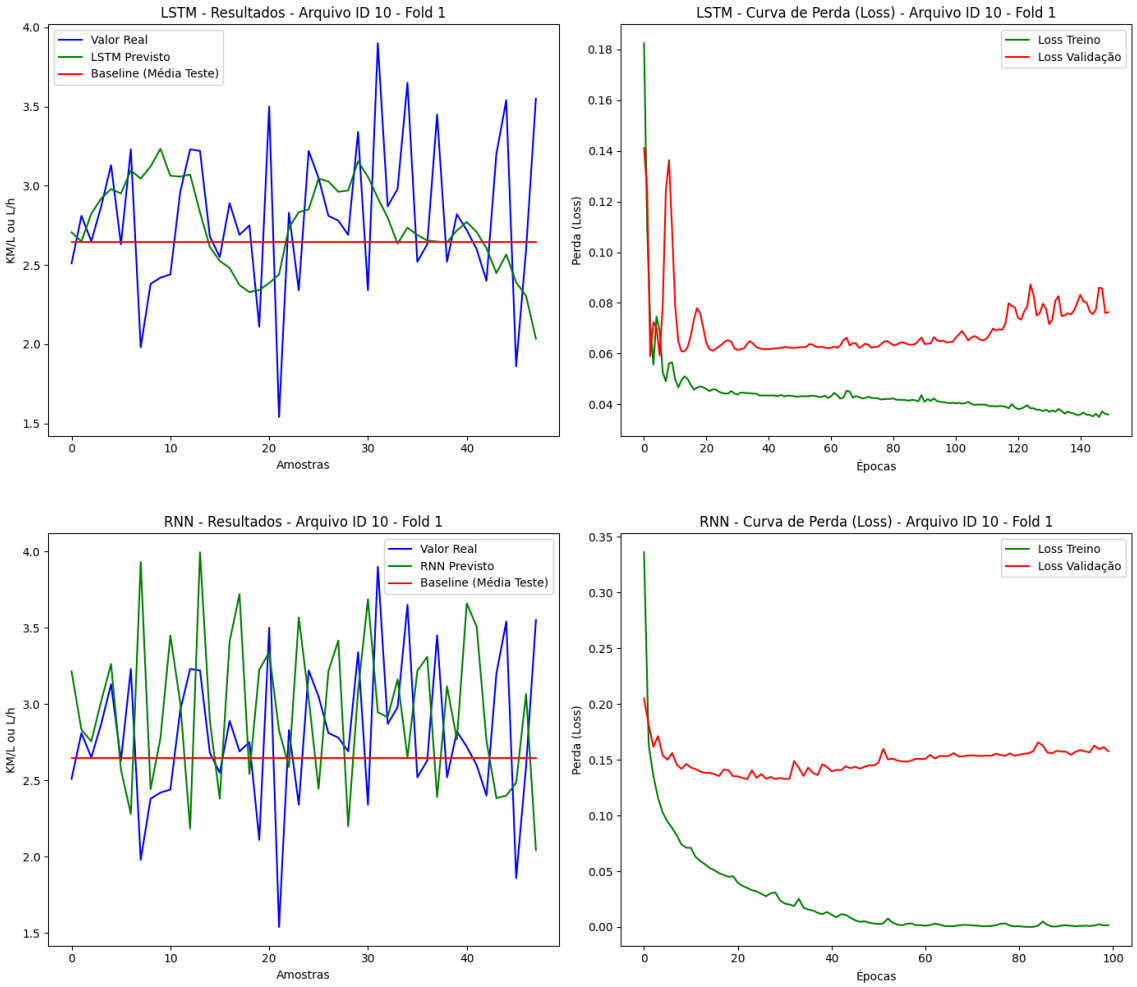
RNN Média sobre 5 folds -> RMSE: 0.7132, MAE: 0.5695, MAPE: 19.96%

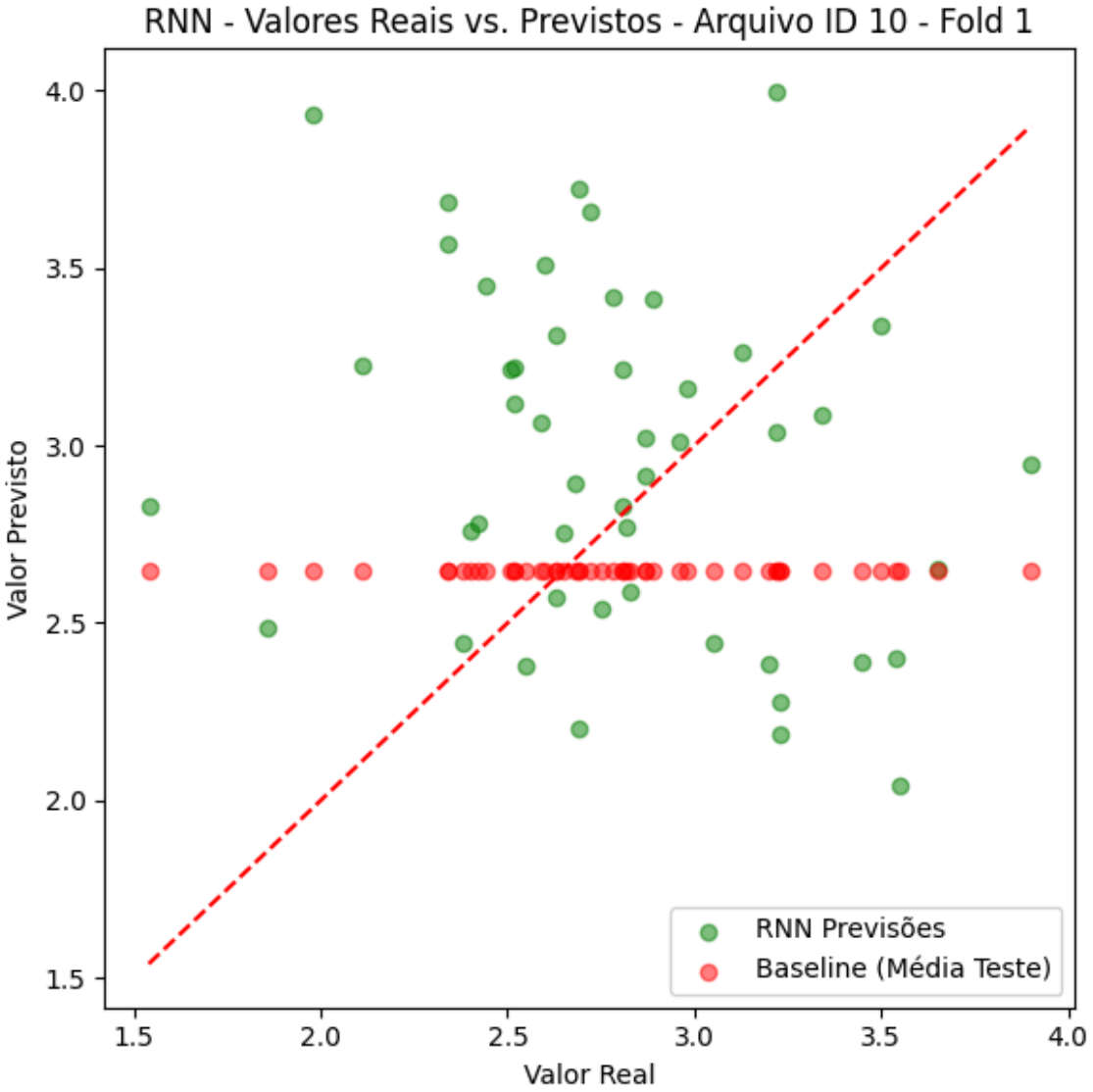
LSTM Média sobre 5 folds -> RMSE: 0.7843, MAE: 0.6522, MAPE: 22.92%

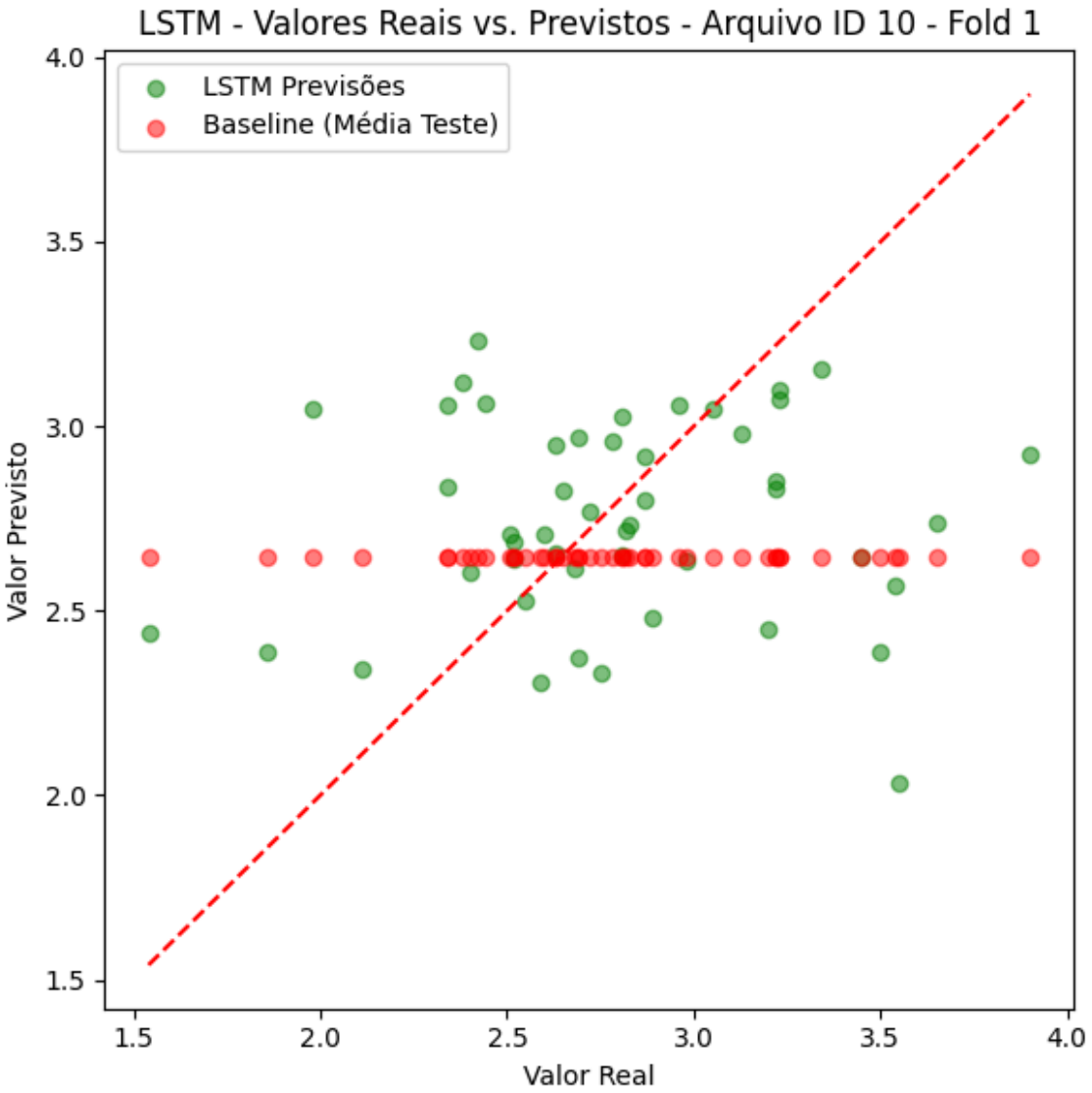
Para o Arquivo ID 1 podemos notar pontos interessantes com a RNN que apresentou desempenho melhor que o LSTM em 4 dos 5 folds. Seu erro médio (RMSE: 0.7132, MAE: 0.5695, MAPE: 19.96%) é menor em comparação com a LSTM. Apesar de não superar o Baseline na média geral, em alguns folds (como Fold 5) a RNN conseguiu apresentar resultados competitivos. Já a LSTM apresentou os piores resultados médios entre os modelos avaliados. O overfitting é evidente nos gráficos de Loss, onde a perda de validação se estabiliza ou cresce, enquanto a perda de treino continua a diminuir. Seu desempenho foi melhor apenas no Fold 2, onde superou as demais abordagens.

Arquivo ID 10

Fold 1




RNN - Valores Reais vs. Previstos - Arquivo ID 10 - Fold 1
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
RNN Previsões
Baseline (Média Teste)

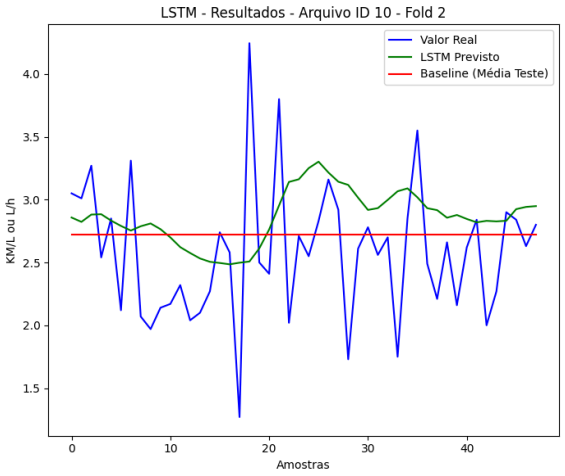


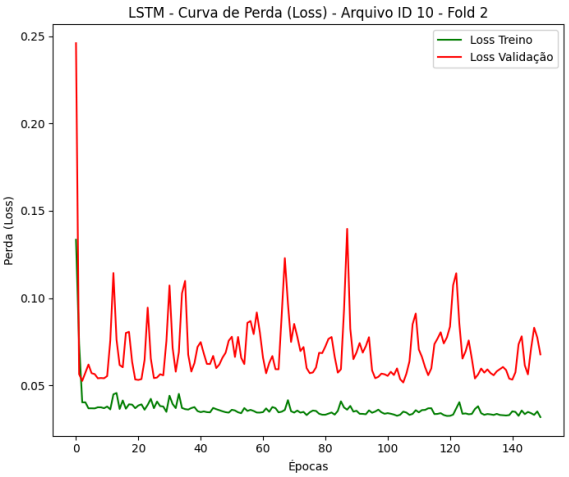
Baseline (Média do Teste) Fold 1 -> RMSE: 0.4931, MAE: 0.3728, MAPE: 13.45%

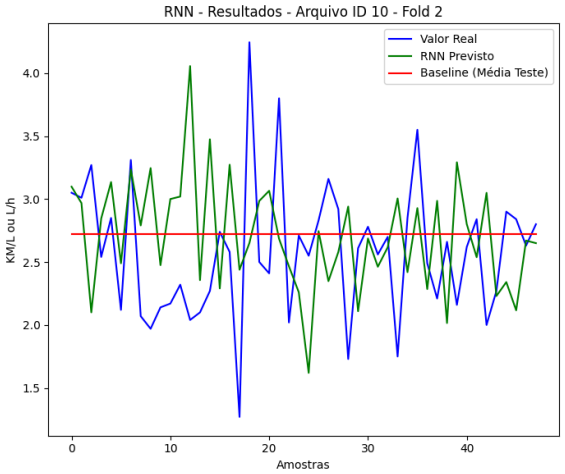
RNN Fold 1 -> RMSE: 0.7664, MAE: 0.6145, MAPE: 23.37%

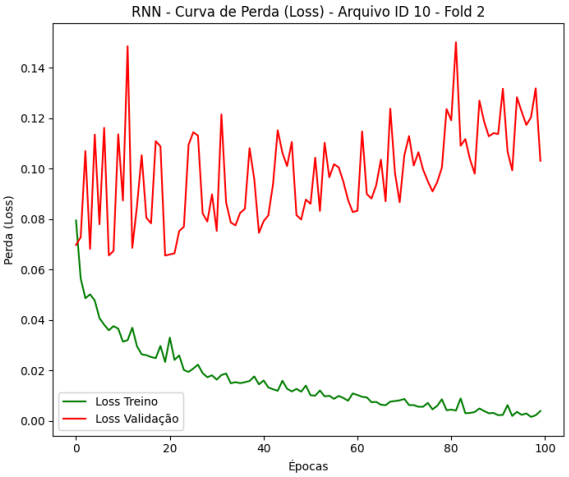
LSTM Fold 1 -> RMSE: 0.5333, MAE: 0.3969, MAPE: 14.62%

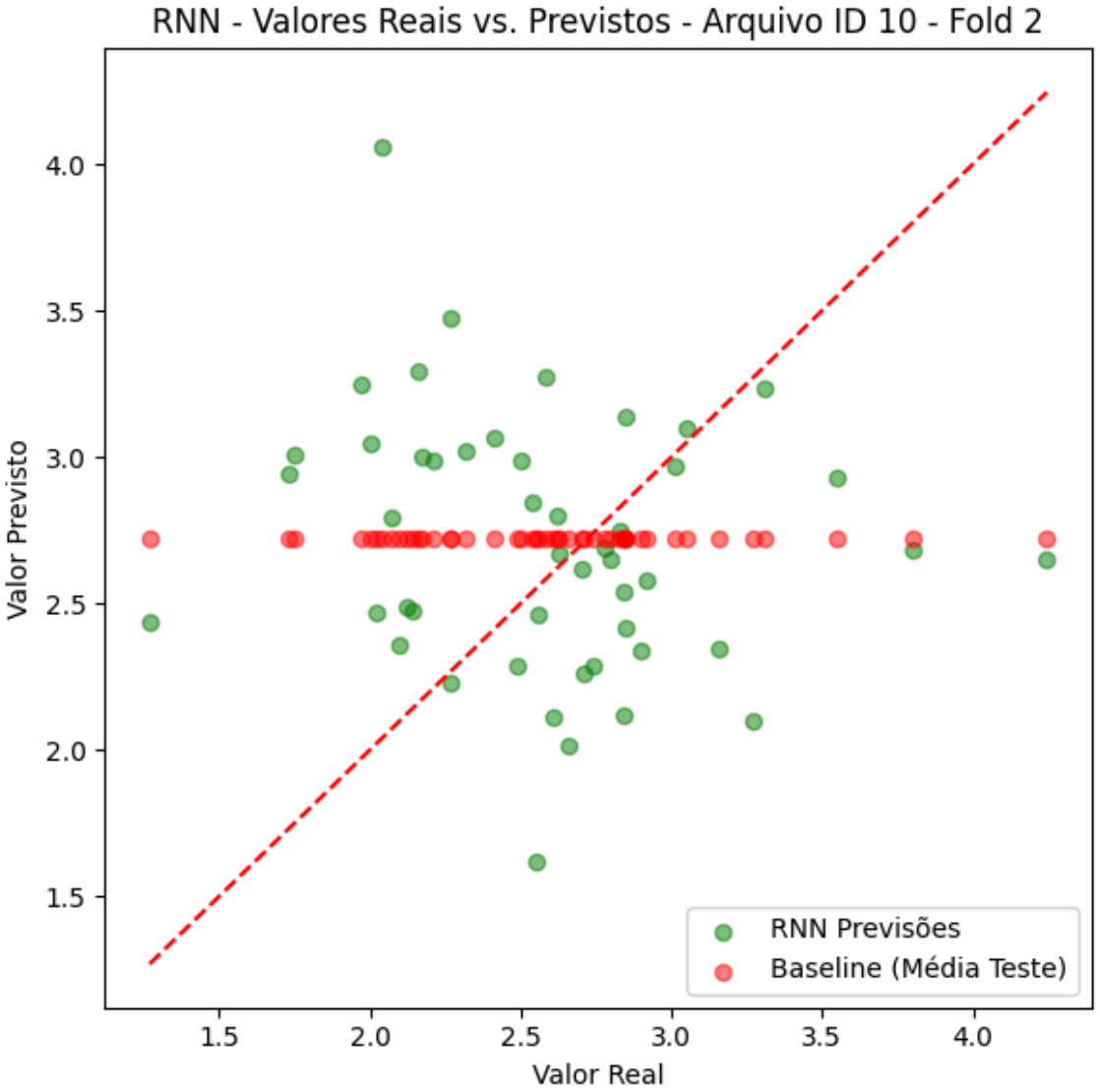
Fold 2

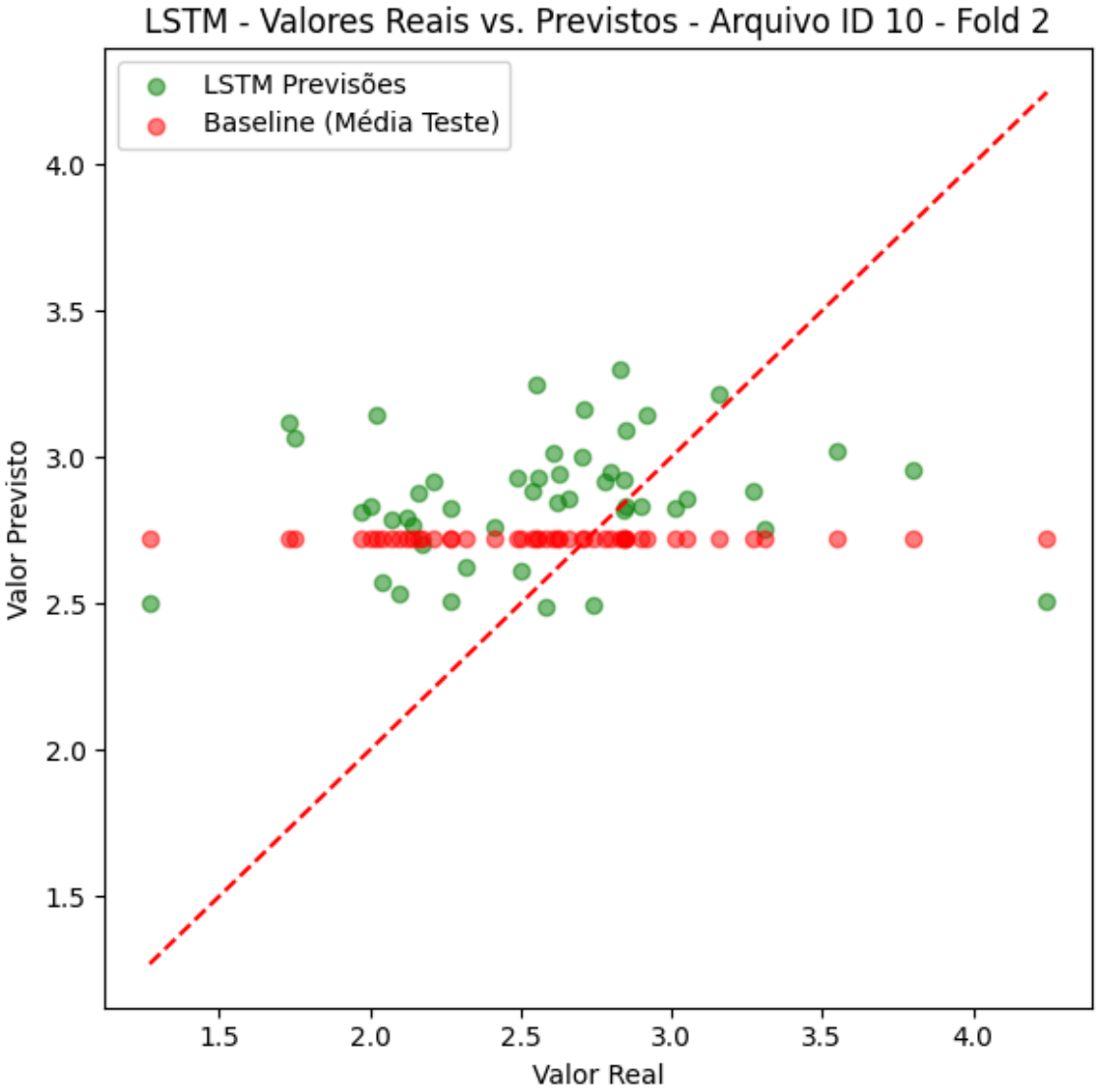

LSTM - Resultados - Arquivo ID 10 - Fold 2
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 10 - Fold 2
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 10 - Fold 2
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 10 - Fold 2
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 10 - Fold 2
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0
RNN Previsões
Baseline (Média Teste)

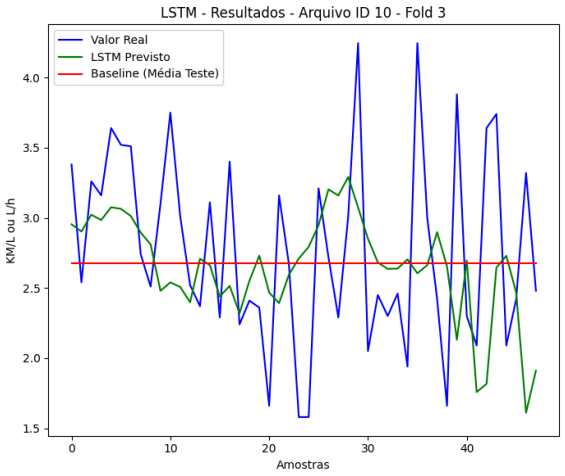


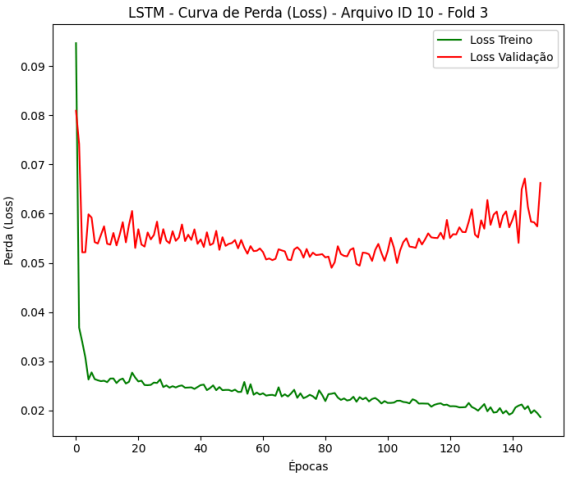
Baseline (Média do Teste) Fold 2 -> RMSE: 0.5540, MAE: 0.4215, MAPE: 18.45%

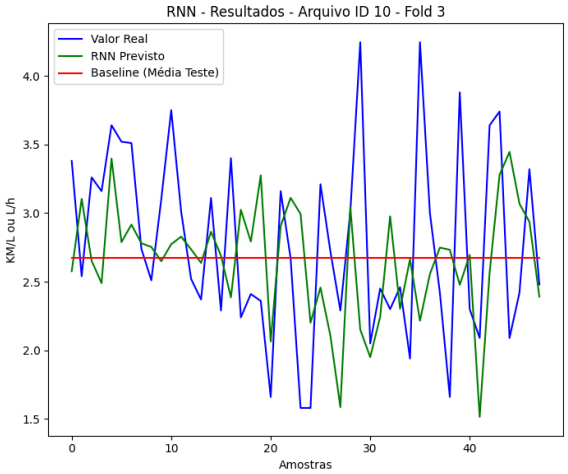
RNN Fold 2 -> RMSE: 0.7578, MAE: 0.6031, MAPE: 25.82%

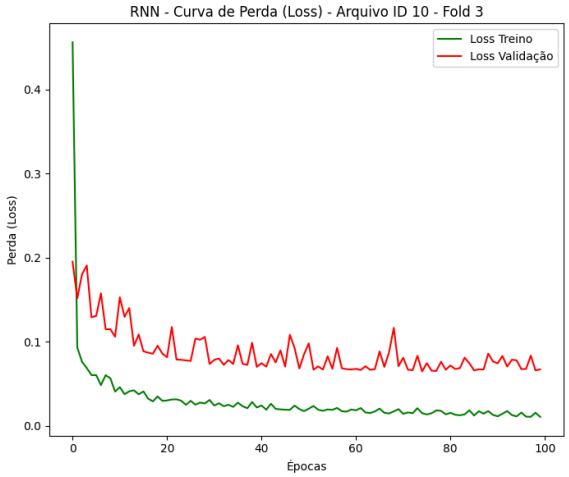
LSTM Fold 2 -> RMSE: 0.6146, MAE: 0.4836, MAPE: 21.22%

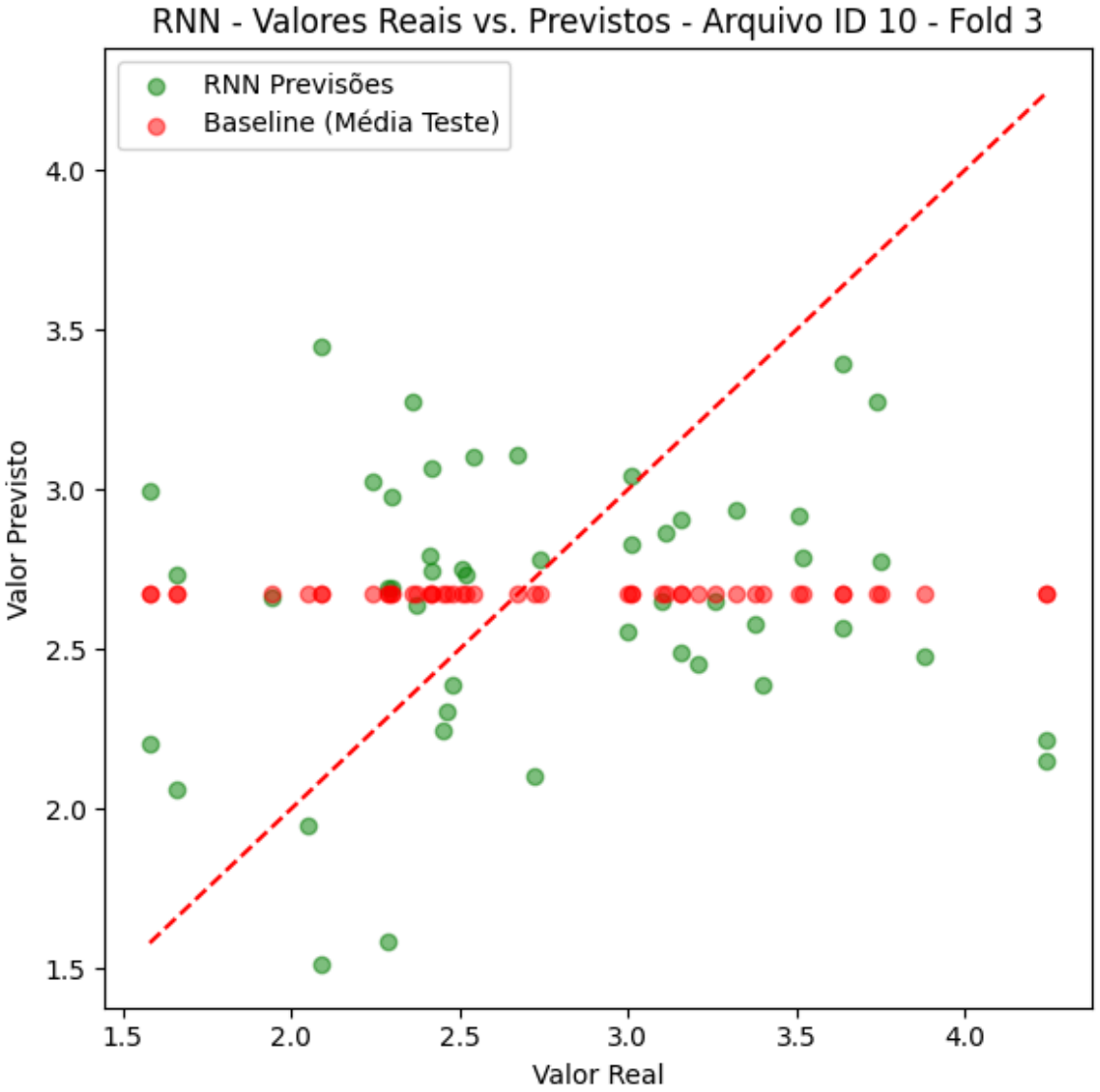
Fold 3

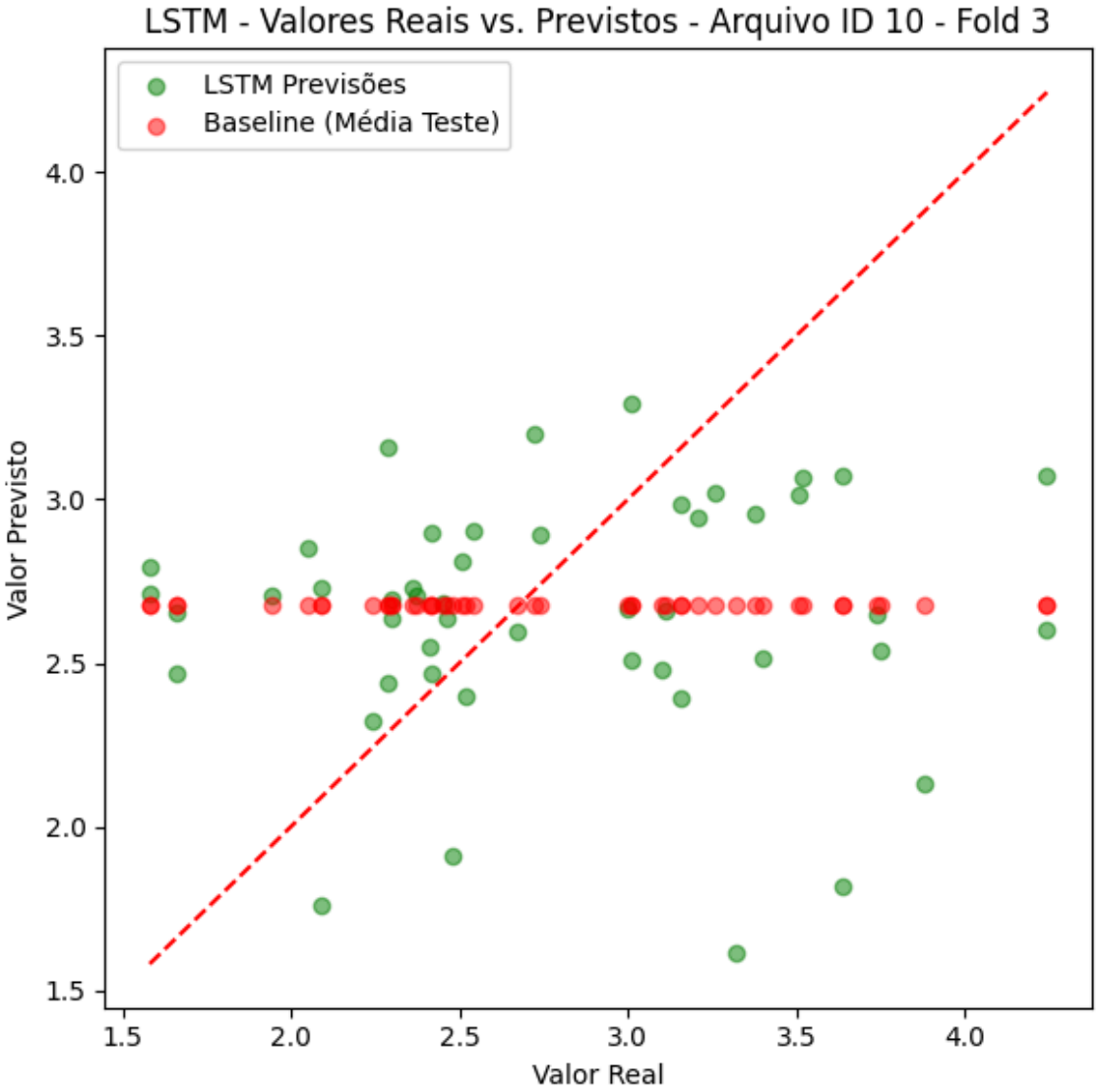

LSTM - Resultados - Arquivo ID 10 - Fold 3
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 10 - Fold 3
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 10 - Fold 3
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 10 - Fold 3
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 10 - Fold 3
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.5
2.0
2.5
3.0
3.5
4.0

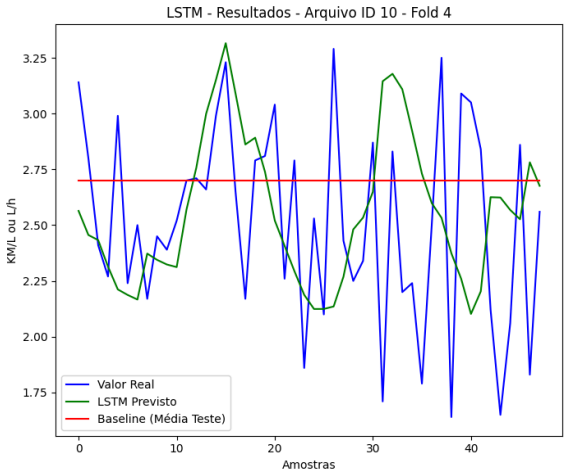


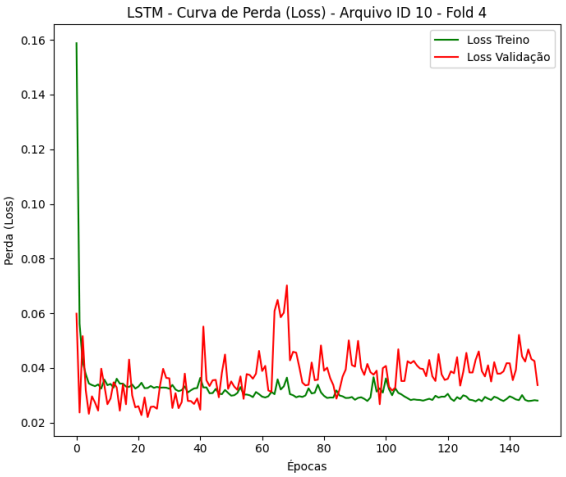
Baseline (Média do Teste) Fold 3 -> RMSE: 0.6865, MAE: 0.5713, MAPE: 21.42%

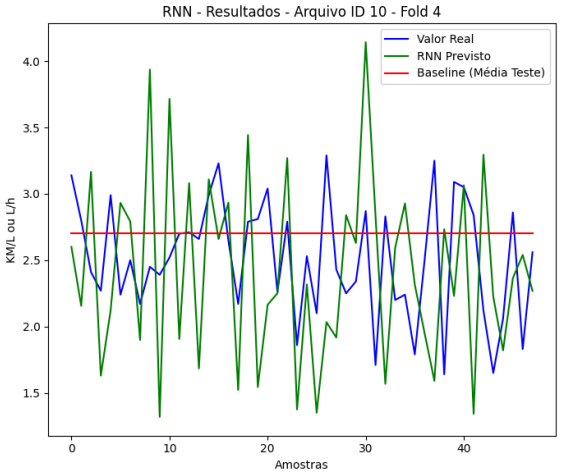
RNN Fold 3 -> RMSE: 0.7698, MAE: 0.6205, MAPE: 23.21%

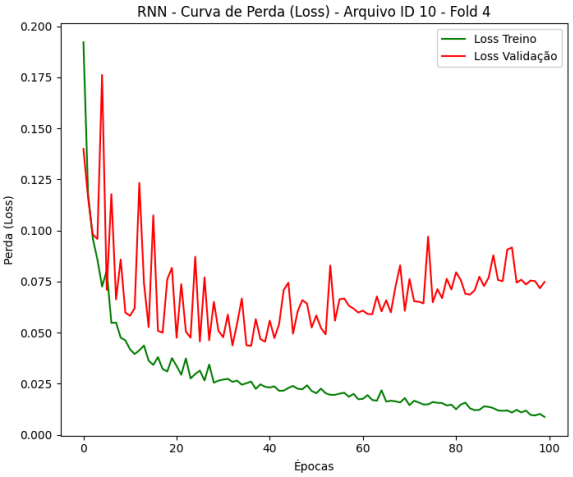
LSTM Fold 3 -> RMSE: 0.7657, MAE: 0.6088, MAPE: 22.68%

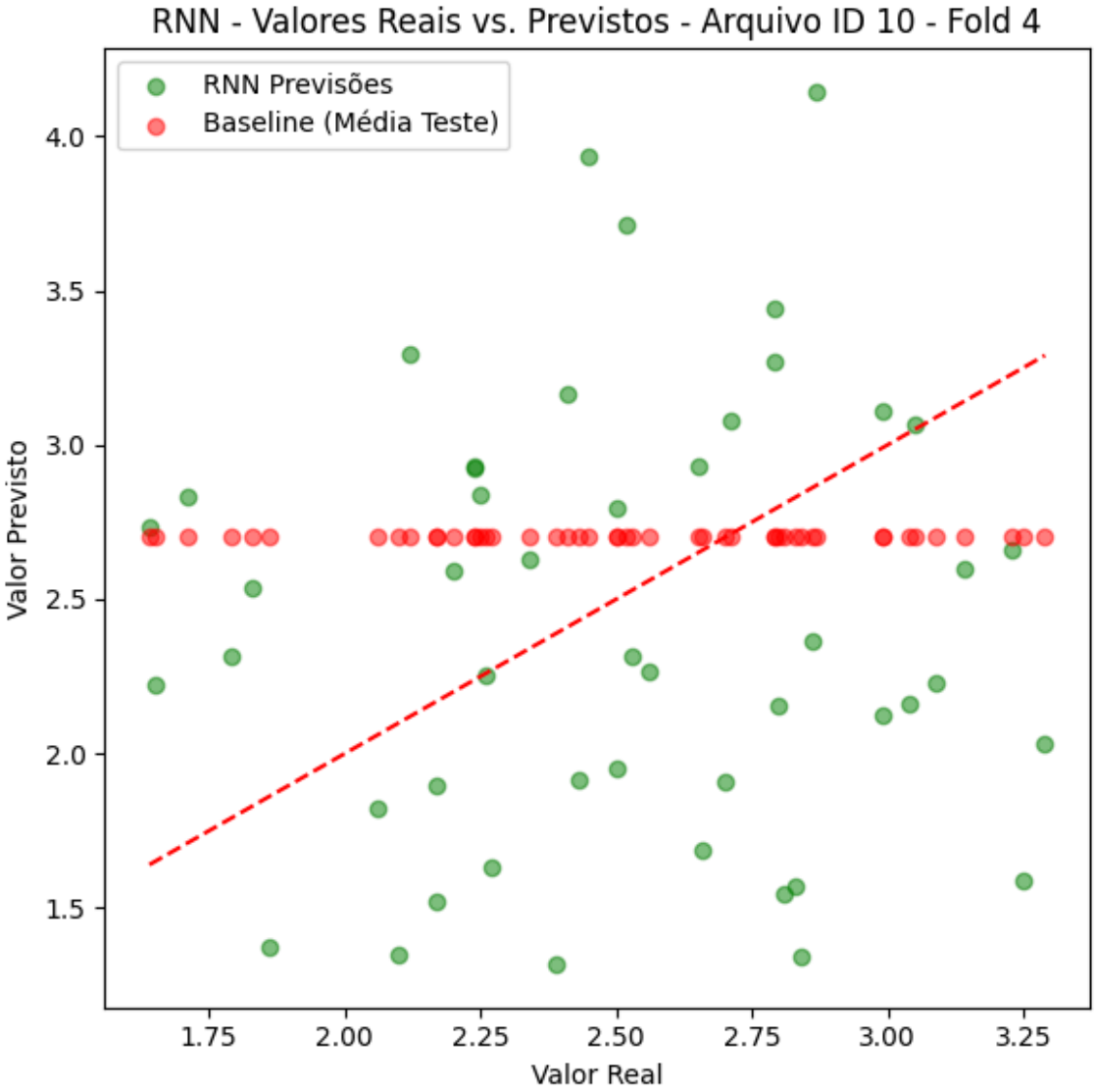
Fold 4

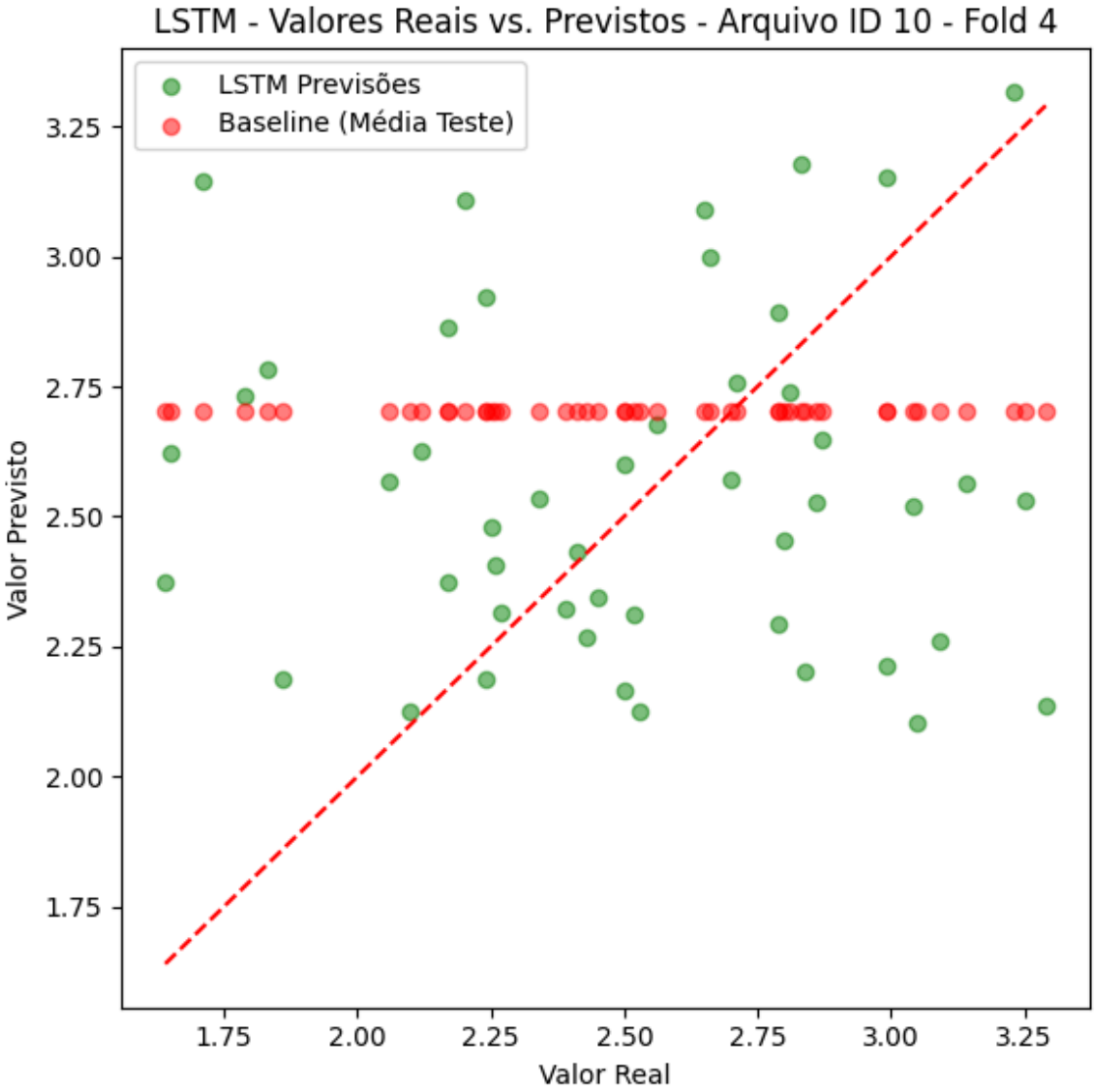

LSTM - Resultados - Arquivo ID 10 - Fold 4
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 10 - Fold 4
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Resultados - Arquivo ID 10 - Fold 4
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 10 - Fold 4
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 10 - Fold 4
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
4.0
3.5
3.0
2.5
2.0
1.5
1.75
2.00
2.25
2.50
2.75
3.00
3.25

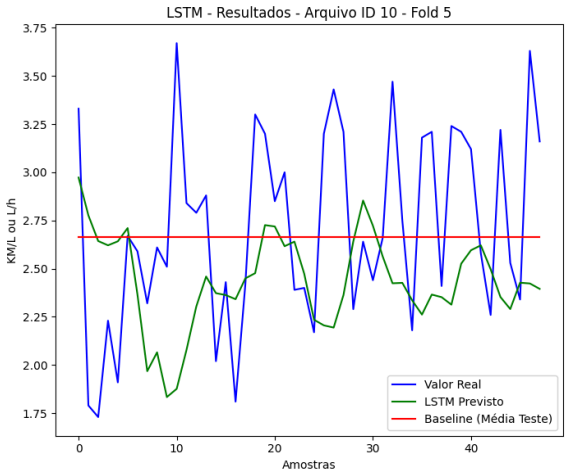


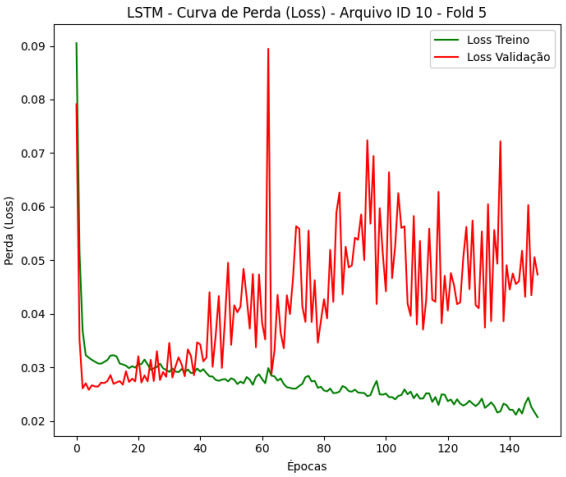
Baseline (Média do Teste) Fold 4 -> RMSE: 0.4757, MAE: 0.3872, MAPE: 17.59%

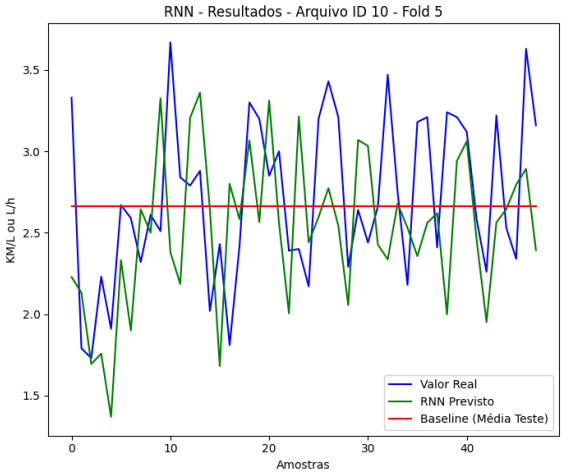
RNN Fold 4 -> RMSE: 0.8137, MAE: 0.7088, MAPE: 28.84%

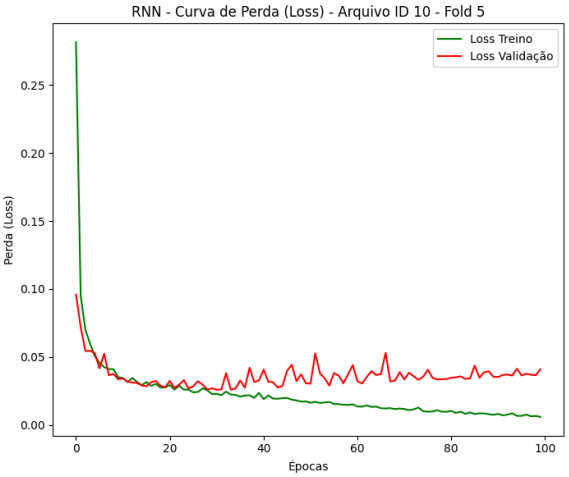
LSTM Fold 4 -> RMSE: 0.5466, MAE: 0.4240, MAPE: 18.20%

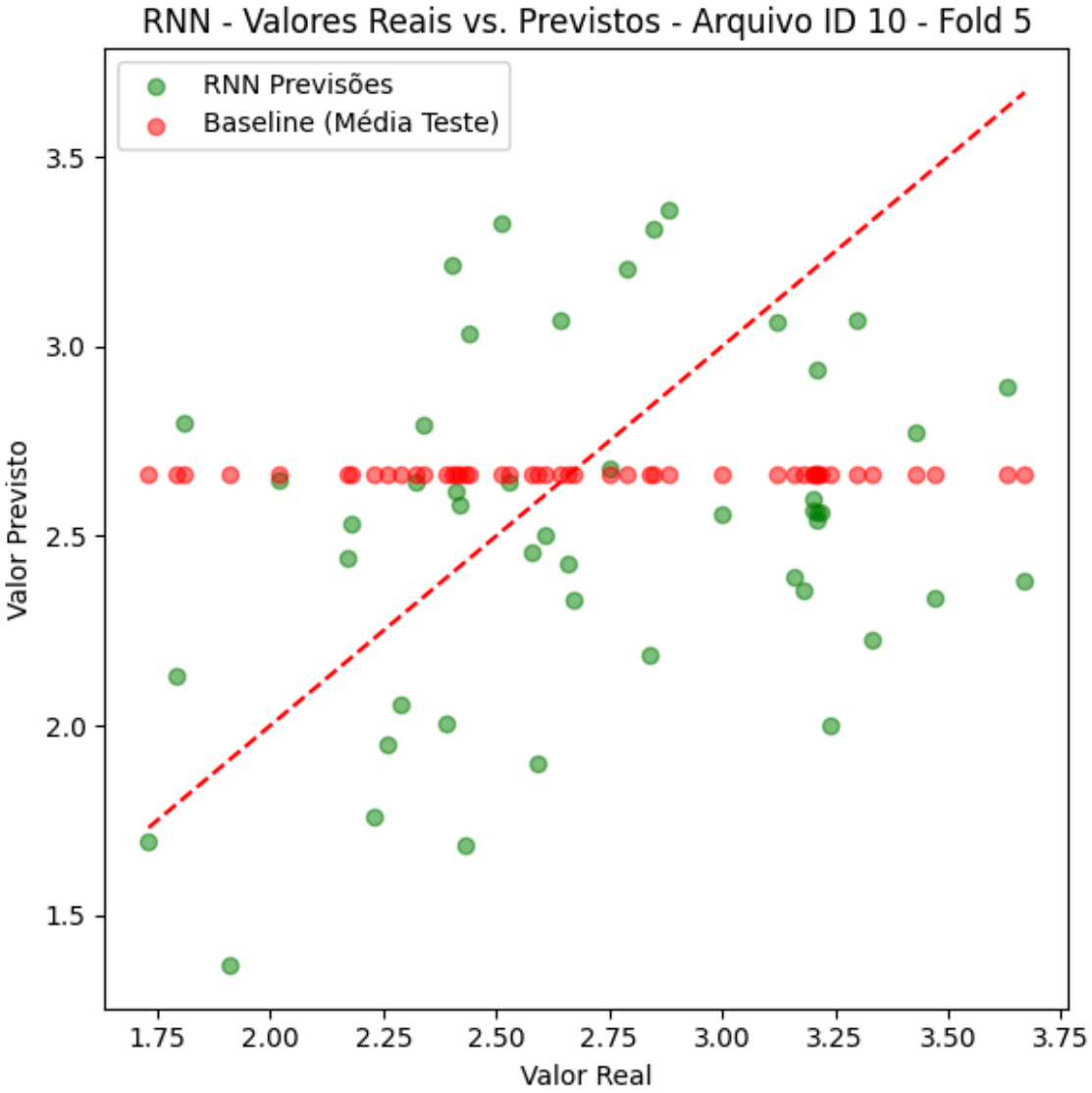
Fold 5

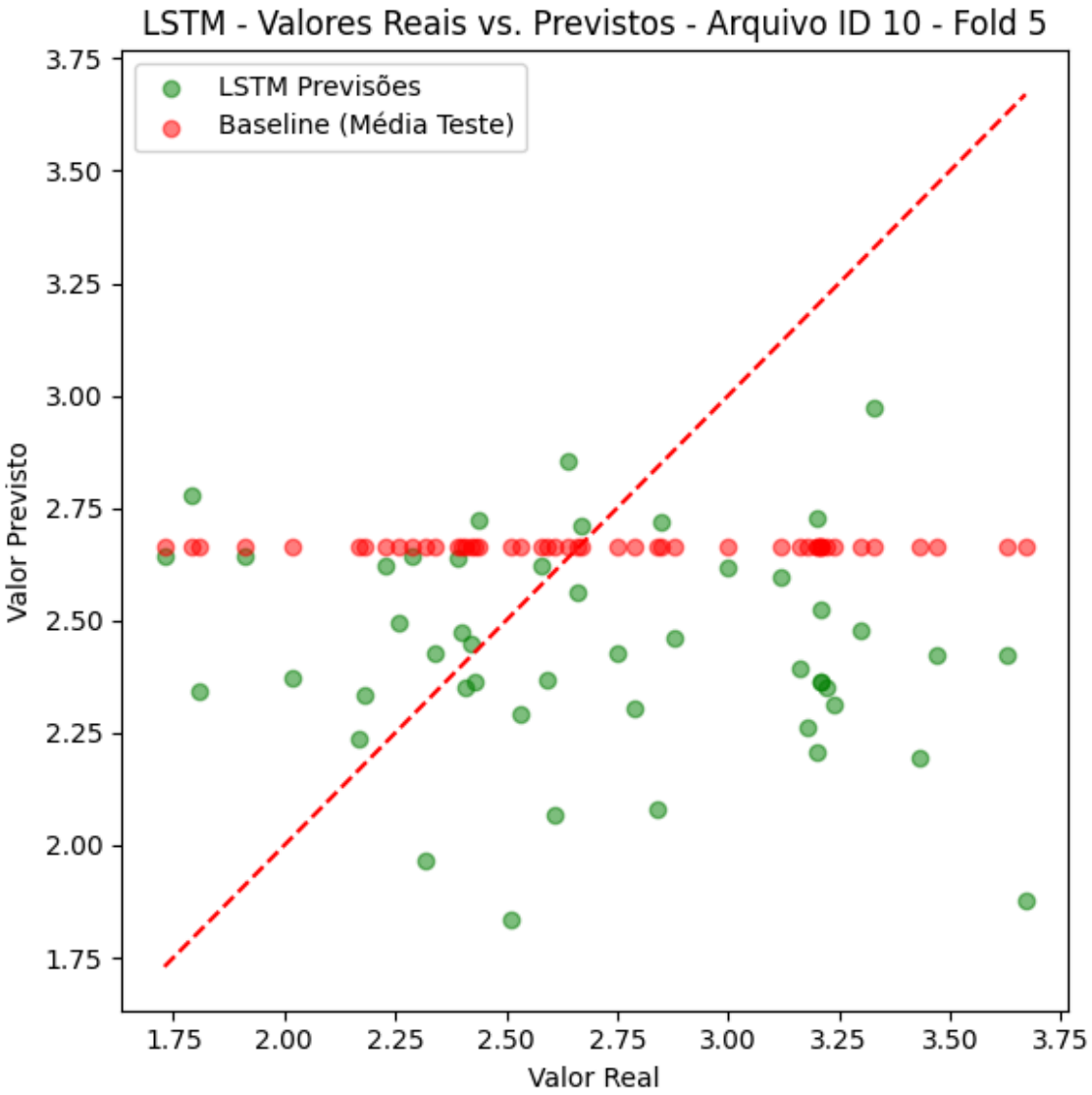

LSTM - Resultados - Arquivo ID 10 - Fold 5
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 10 - Fold 5
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Resultados - Arquivo ID 10 - Fold 5
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 10 - Fold 5
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 10 - Fold 5
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
3.5
3.0
2.5
2.0
1.5
1.75
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75



Baseline (Média do Teste) Fold 5 -> RMSE: 0.5017, MAE: 0.4214, MAPE: 16.15%

RNN Fold 5 -> RMSE: 0.6016, MAE: 0.5159, MAPE: 18.91%

LSTM Fold 5 -> RMSE: 0.6472, MAE: 0.5168, MAPE: 18.64%

Média de todos os folds Arquivo ID 10

Baseline (Média do Teste) Média sobre 5 folds -> RMSE: 0.5422, MAE: 0.4348, MAPE: 17.41%

RNN Média sobre 5 folds -> RMSE: 0.7419, MAE: 0.6126, MAPE: 24.03%

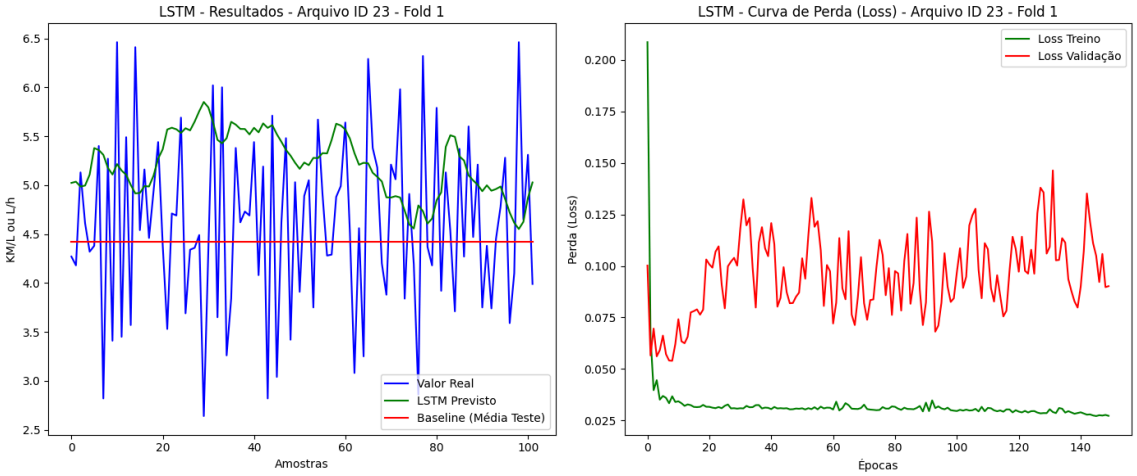
LSTM Média sobre 5 folds -> RMSE: 0.6215, MAE: 0.4860, MAPE: 19.07%

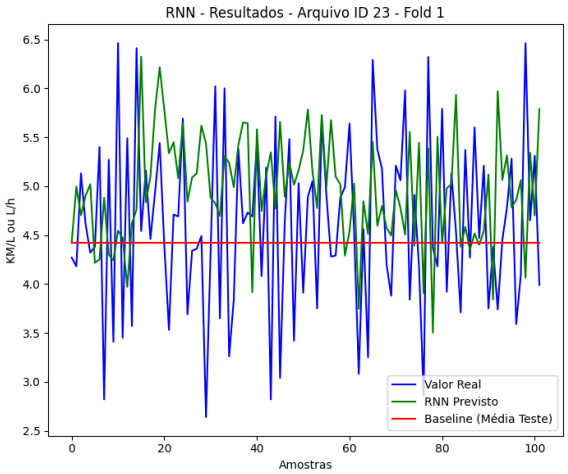
Diferente do “Arquivo ID1” o Arquivo ID 10 teve uma diferença em qual modelo se sobressai, temos que o modelo LSTM apresentou melhor desempenho geral em comparação com a RNN. Sua média de RMSE (0.6215) e MAPE (19.07%) foi significativamente melhor do que a RNN, mostrando maior capacidade de aprendizado.

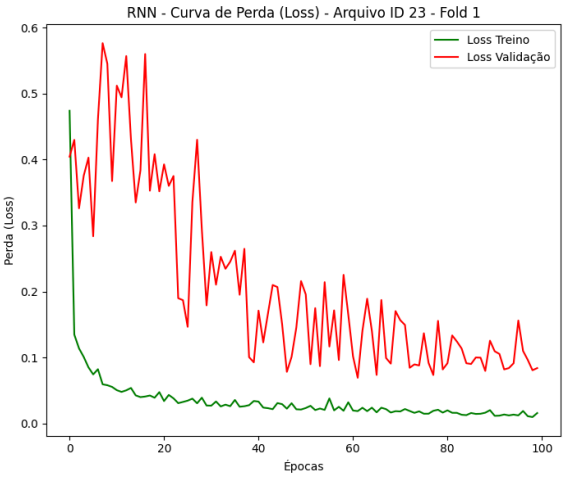
Apesar de superior à RNN, a LSTM ainda não conseguiu superar a simplicidade da Baseline. O modelo RNN apresentou os piores resultados, com MAPE médio de 24.03%.A RNN teve dificuldade em generalizar os dados, com desempenho consistentemente inferior em todos os folds.

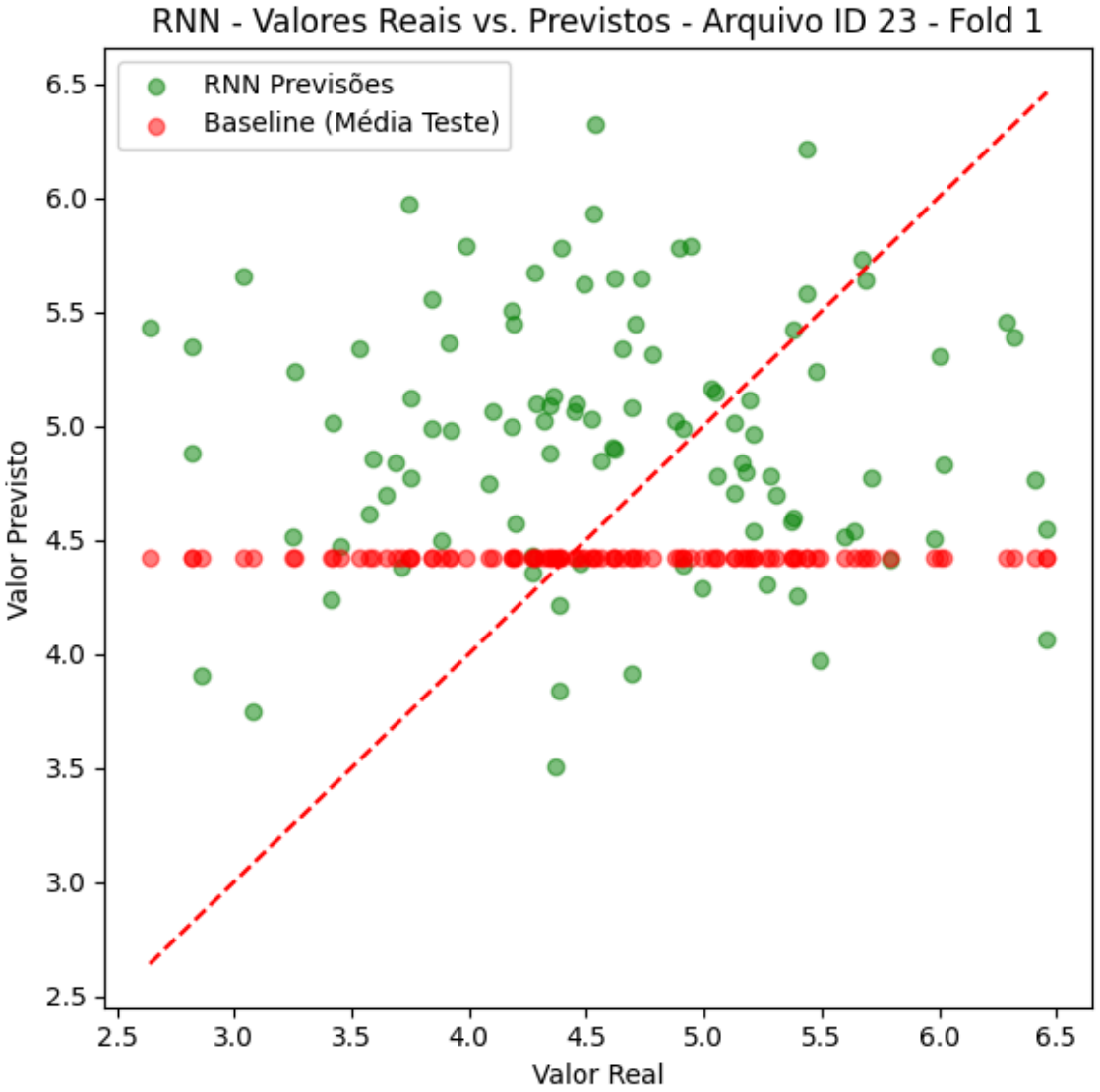
Arquivo ID 23

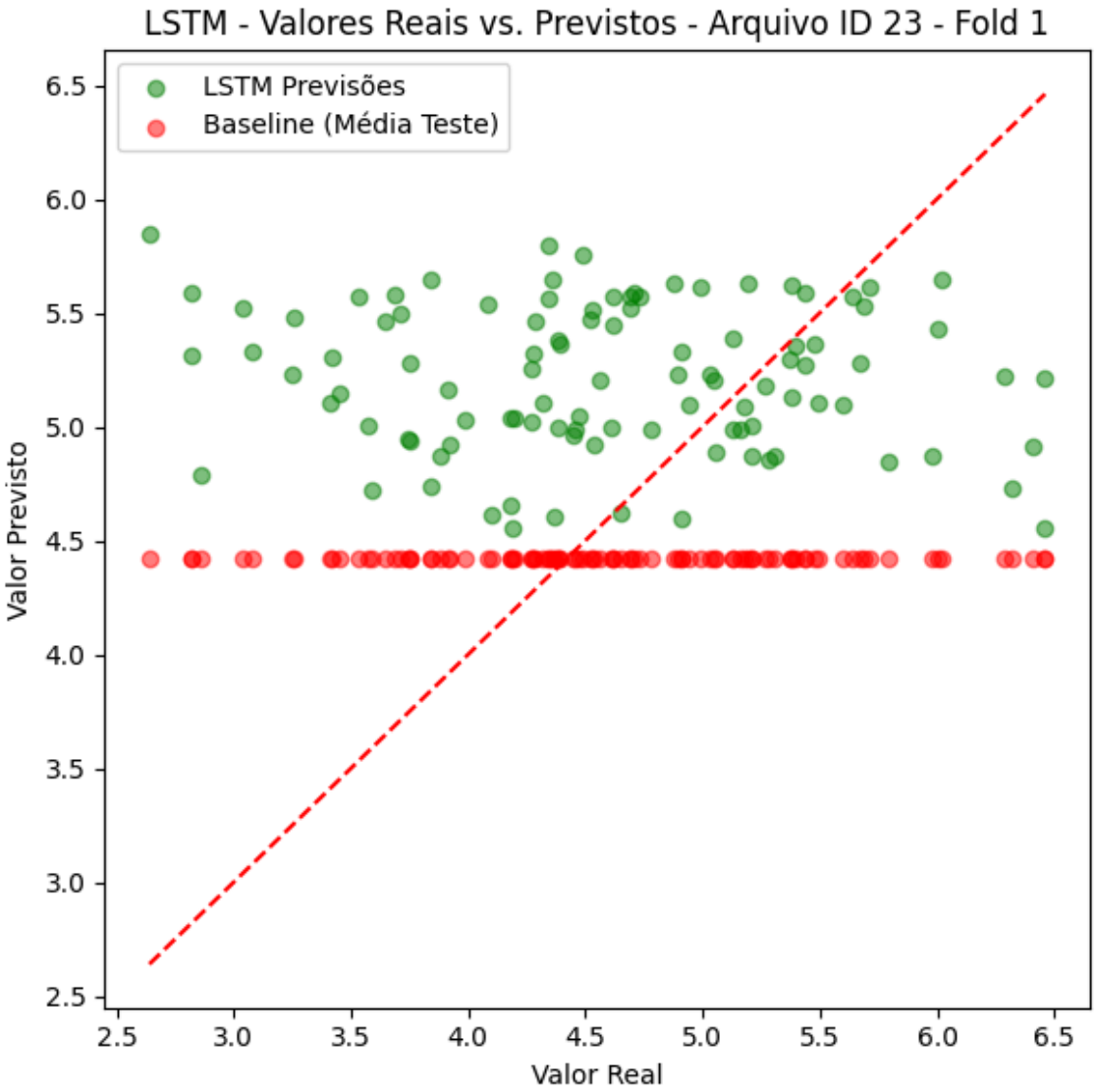
Fold 1








RNN - Valores Reais vs. Previstos - Arquivo ID 23 - Fold 1
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5

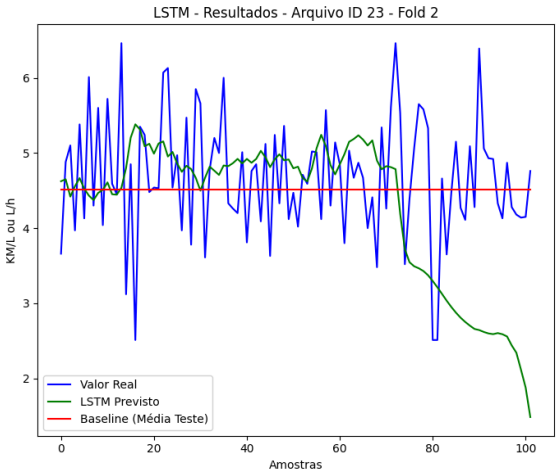


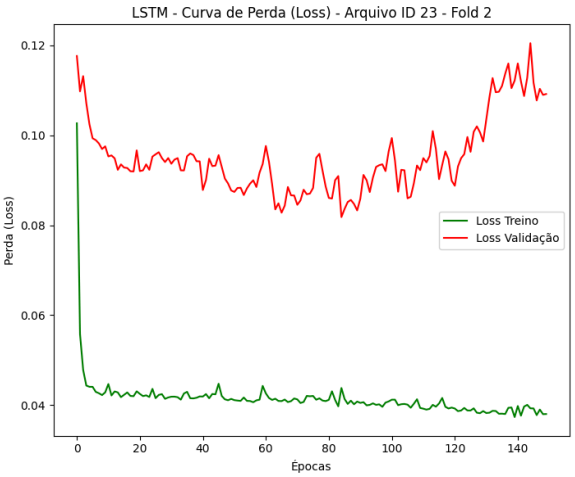
Baseline (Média do Teste) Fold 1 -> RMSE: 0.8939, MAE: 0.7158, MAPE: 16.08%

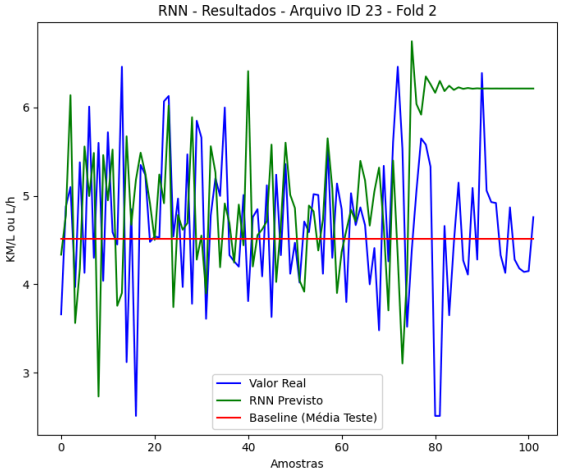
RNN Fold 1 -> RMSE: 1.0962, MAE: 0.9075, MAPE: 21.82%

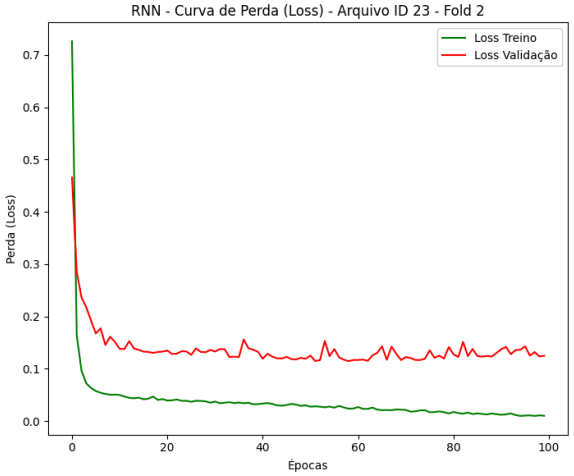
LSTM Fold 1 -> RMSE: 1.1357, MAE: 0.9020, MAPE: 22.86%

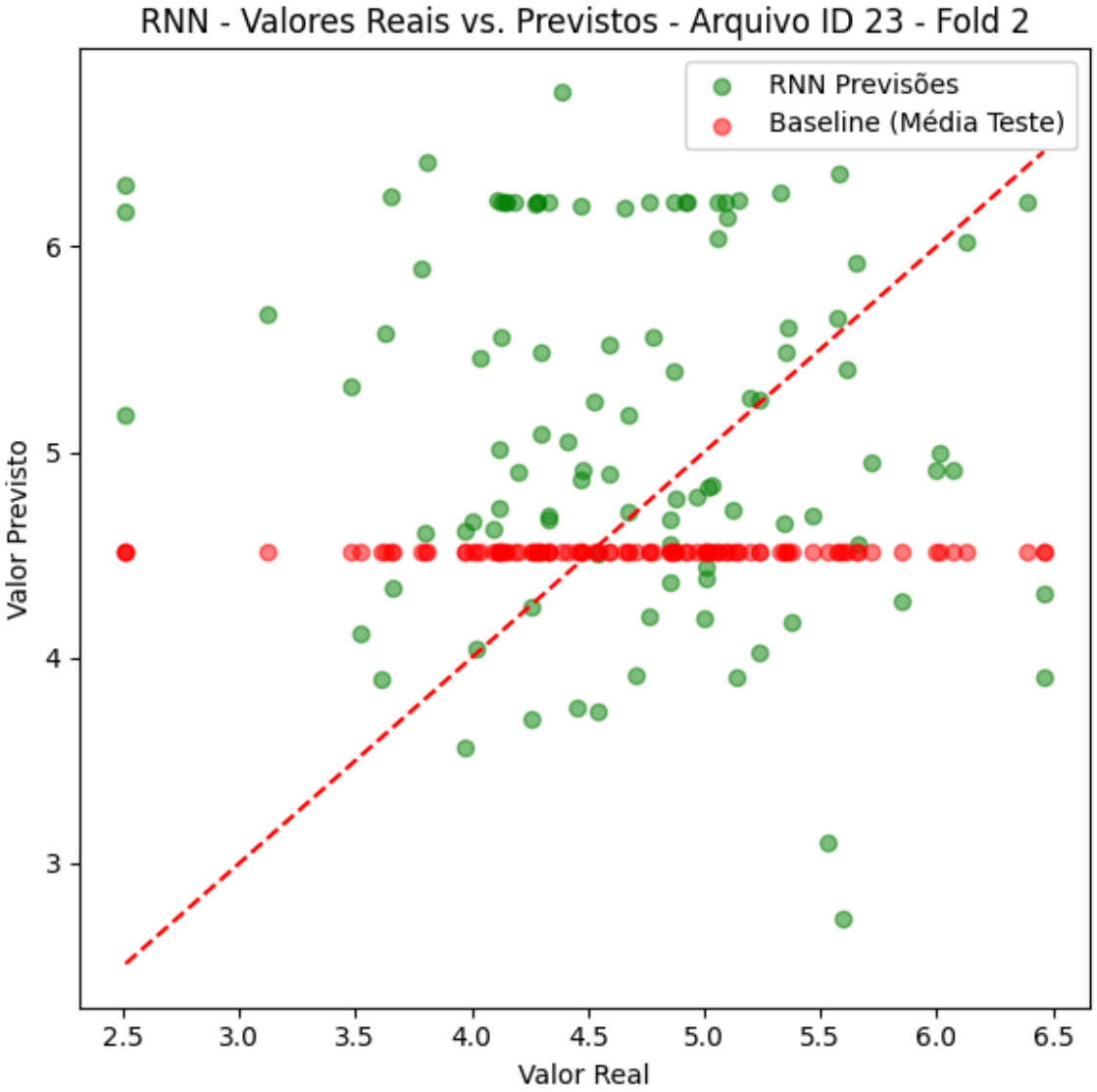
Fold 2

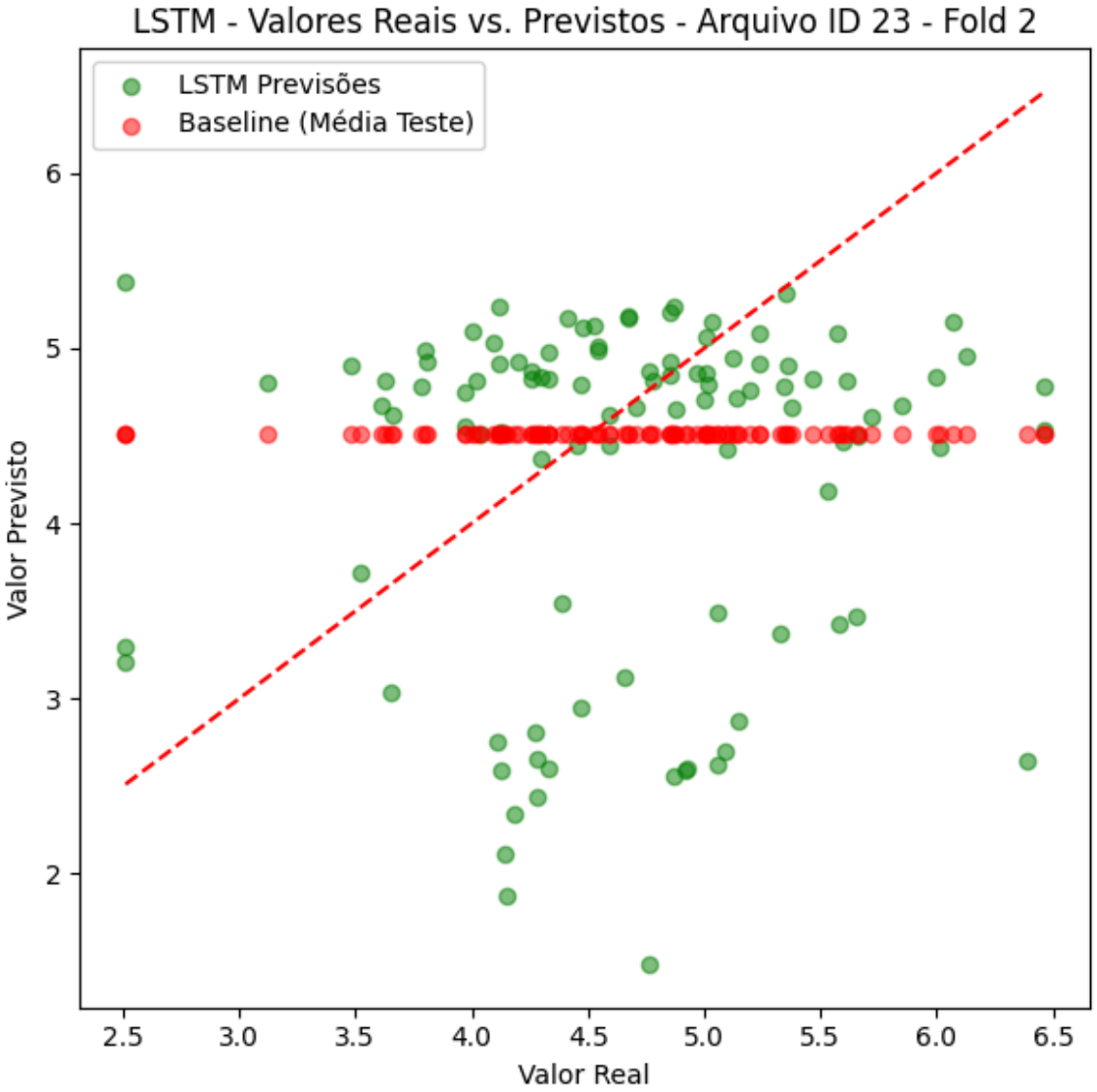

LSTM - Resultados - Arquivo ID 23 - Fold 2
KM/L ou L/h
Amostras
Valor Real
LSTM Previsto
Baseline (Média Teste)


LSTM - Curva de Perda (Loss) - Arquivo ID 23 - Fold 2
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Resultados - Arquivo ID 23 - Fold 2
KM/L ou L/h
Amostras
Valor Real
RNN Previsto
Baseline (Média Teste)


RNN - Curva de Perda (Loss) - Arquivo ID 23 - Fold 2
Perda (Loss)
Épocas
Loss Treino
Loss Validação


RNN - Valores Reais vs. Previstos - Arquivo ID 23 - Fold 2
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
3
4
5
6

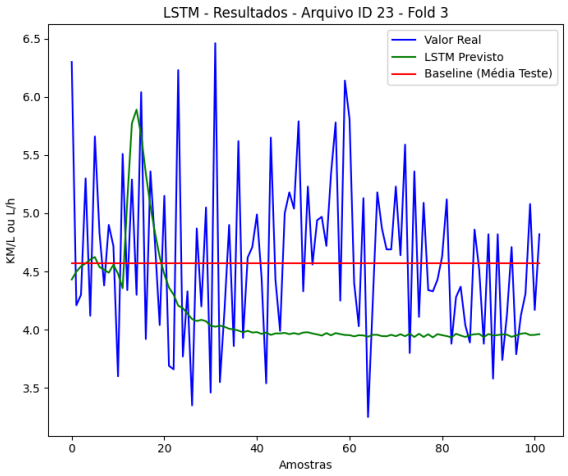


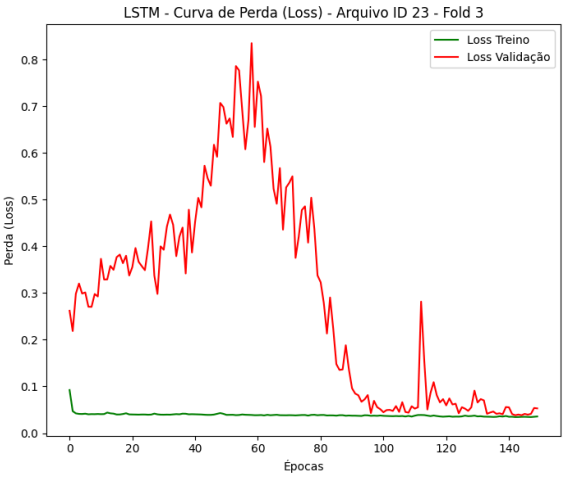
Baseline (Média do Teste) Fold 2 -> RMSE: 0.8120, MAE: 0.6374, MAPE: 14.24%

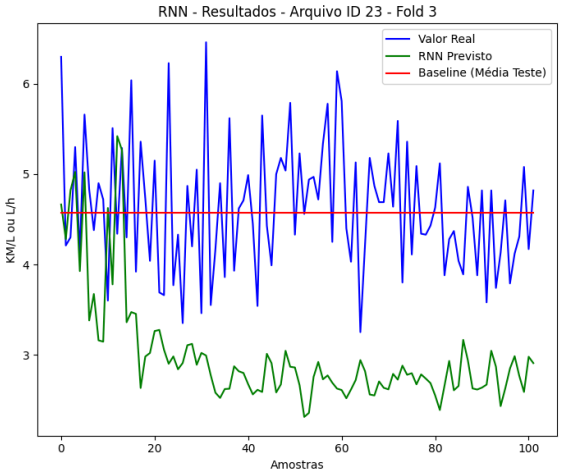
RNN Fold 2 -> RMSE: 1.3495, MAE: 1.0605, MAPE: 25.18%

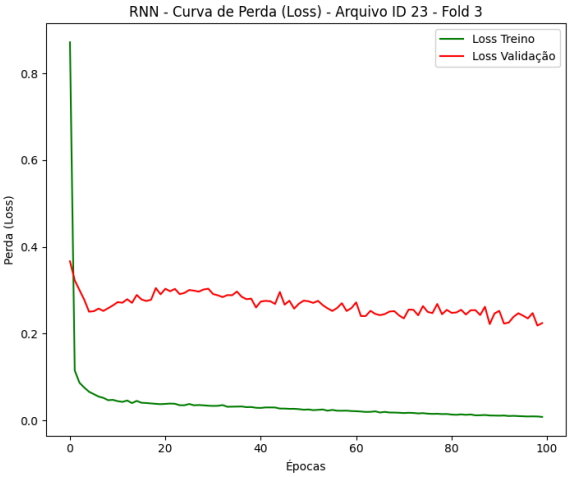
LSTM Fold 2 -> RMSE: 1.2625, MAE: 0.9882, MAPE: 21.85%

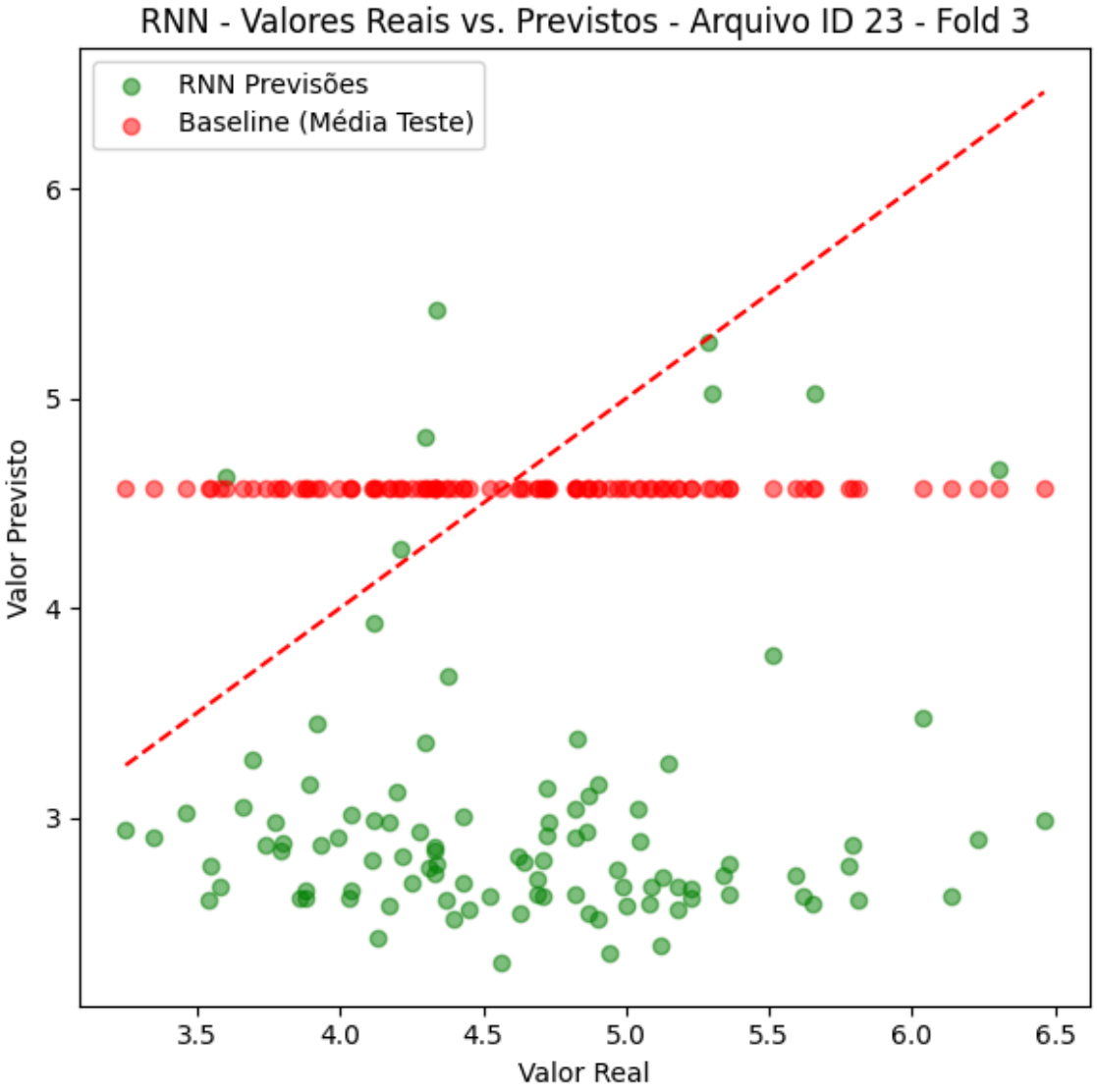
Fold 3

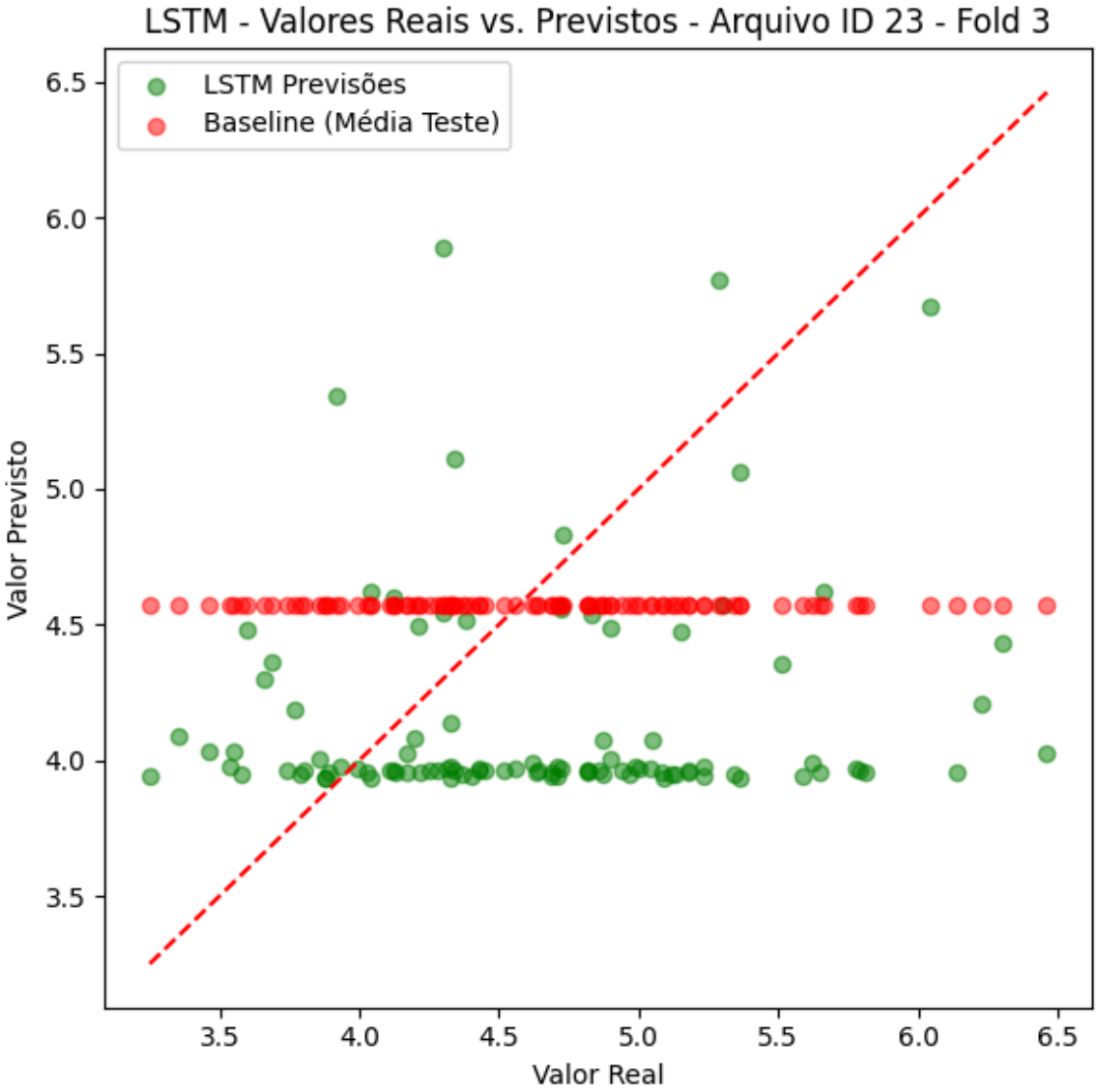

LSTM - Resultados - Arquivo ID 23 - Fold 3
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 23 - Fold 3
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 23 - Fold 3
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 23 - Fold 3
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 23 - Fold 3
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
3
4
5
6
3.5
4.0
4.5
5.0
5.5
6.0
6.5

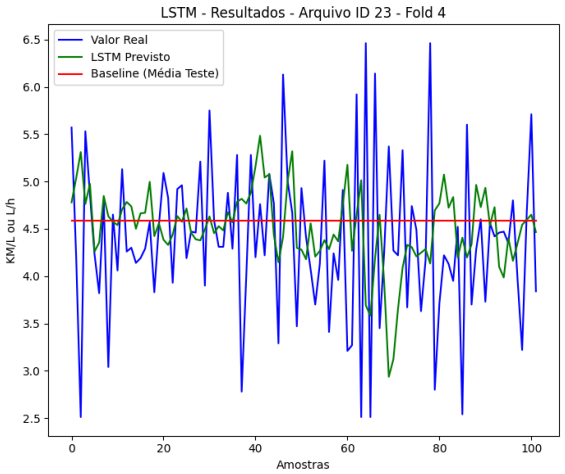


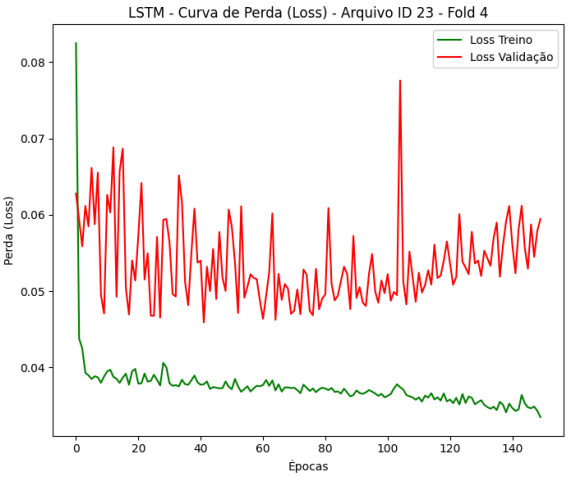
Baseline (Média do Teste) Fold 3 -> RMSE: 0.7053, MAE: 0.5753, MAPE: 12.51%

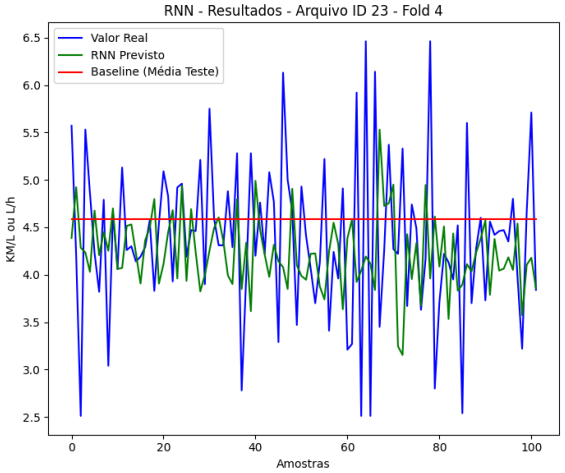
RNN Fold 3 -> RMSE: 1.8709, MAE: 1.6871, MAPE: 35.20%

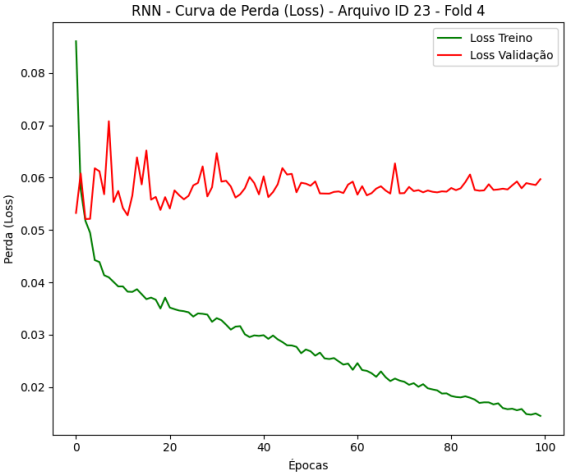
LSTM Fold 3 -> RMSE: 0.9090, MAE: 0.7299, MAPE: 14.92%

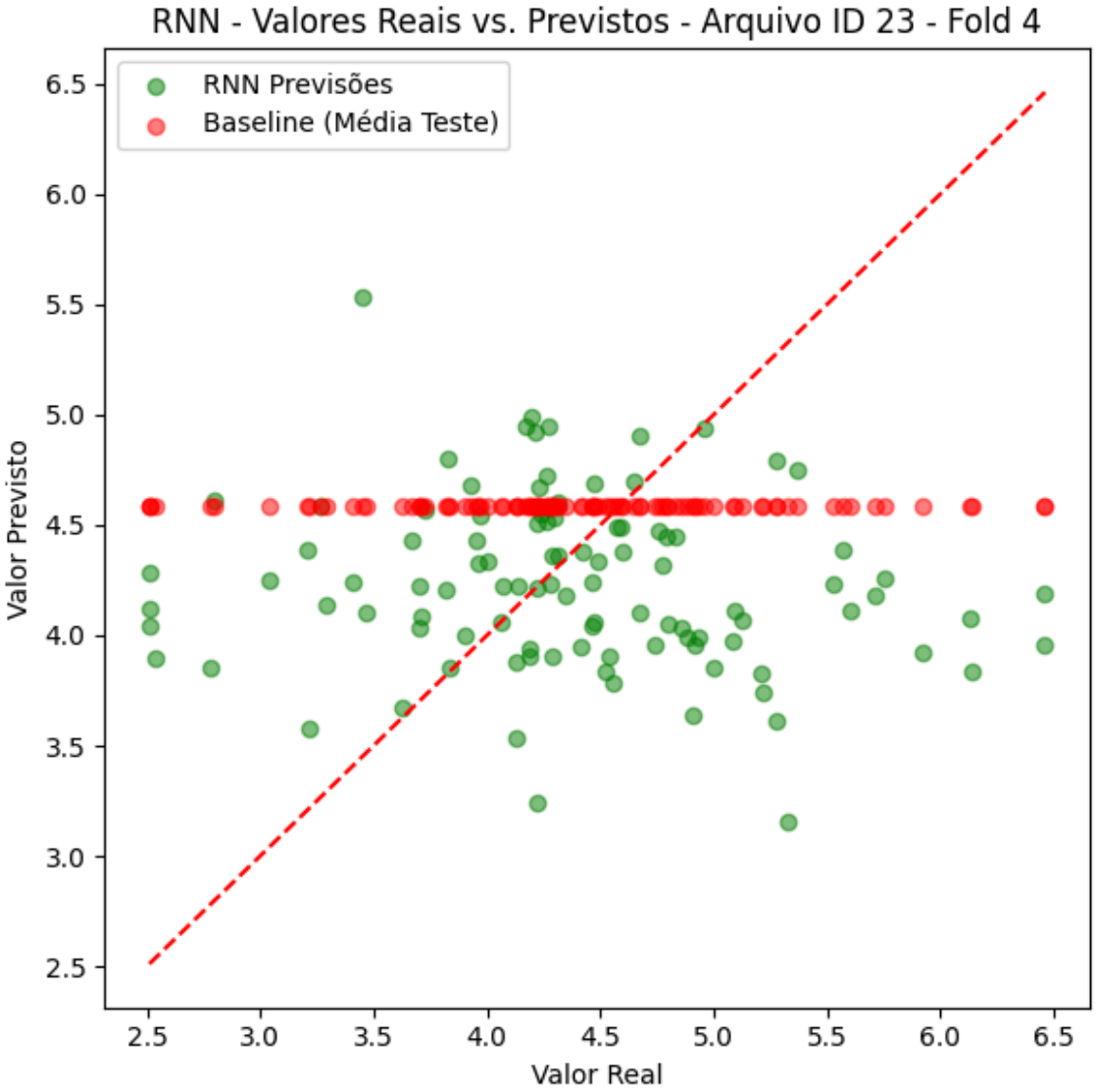
Fold 4

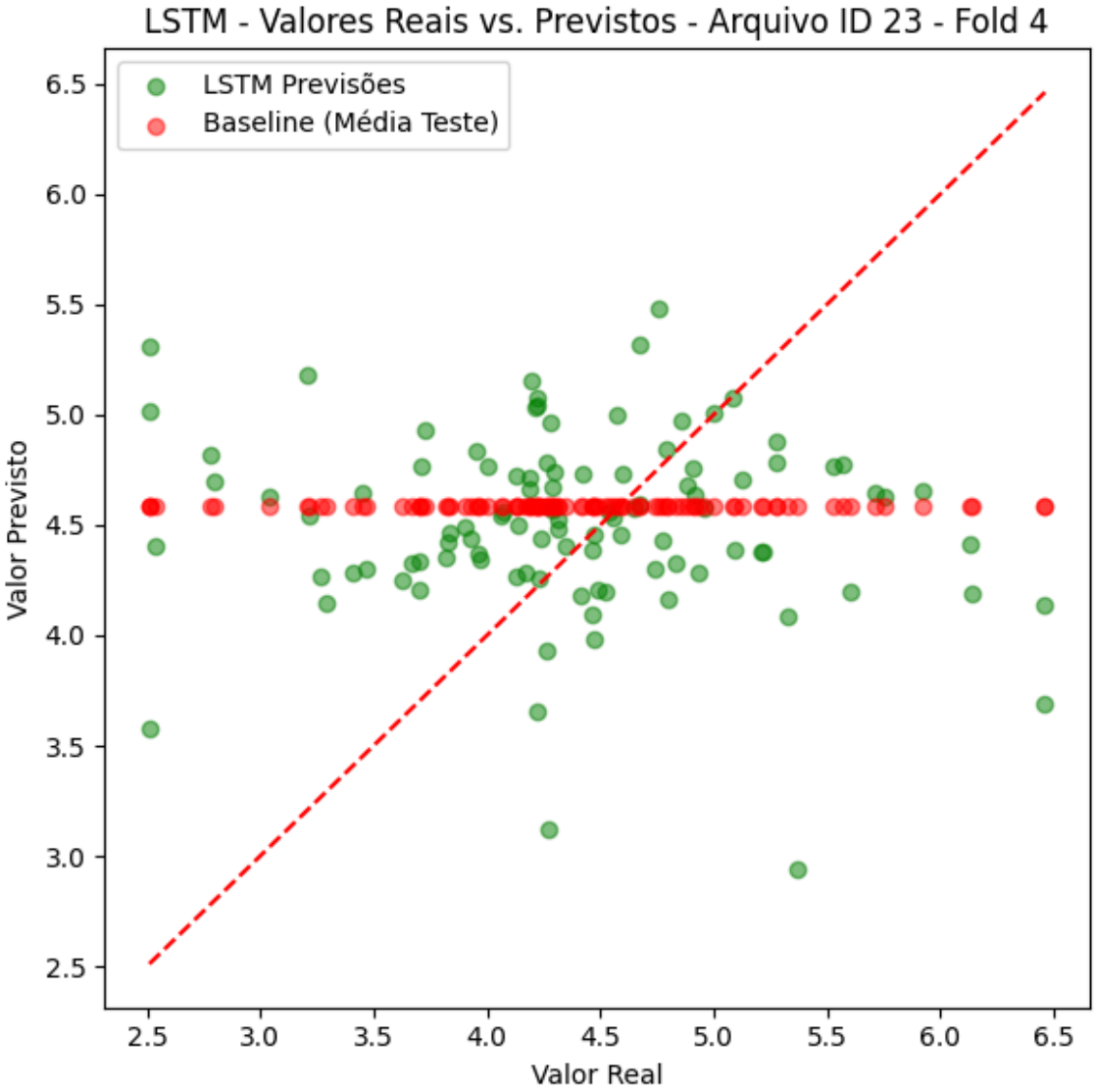

LSTM - Resultados - Arquivo ID 23 - Fold 4
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 23 - Fold 4
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 23 - Fold 4
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 23 - Fold 4
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 23 - Fold 4
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5

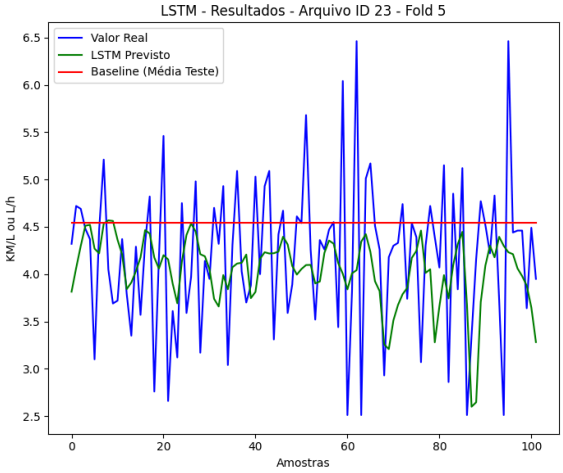


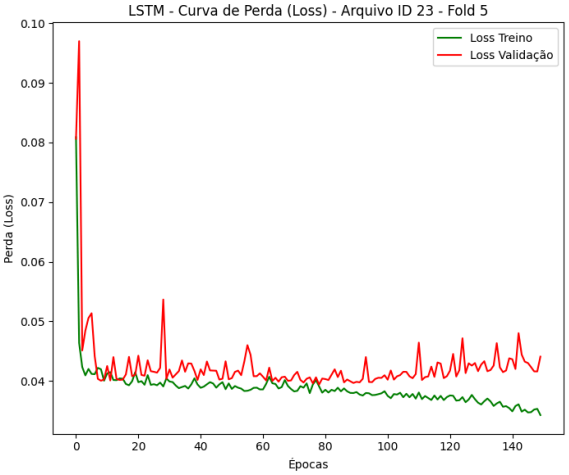
Baseline (Média do Teste) Fold 4 -> RMSE: 0.8485, MAE: 0.6540, MAPE: 16.91%

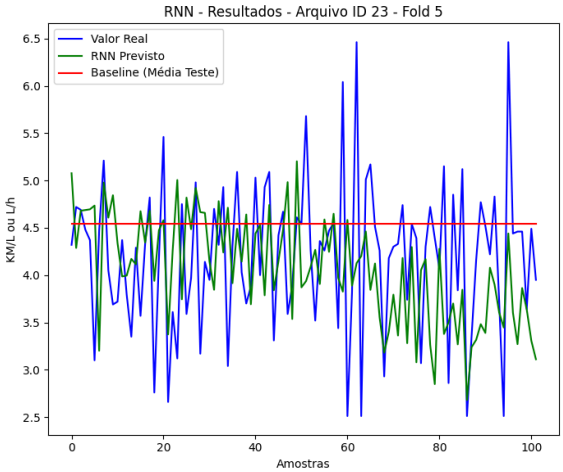
RNN Fold 4 -> RMSE: 0.9651, MAE: 0.7491, MAPE: 17.70%

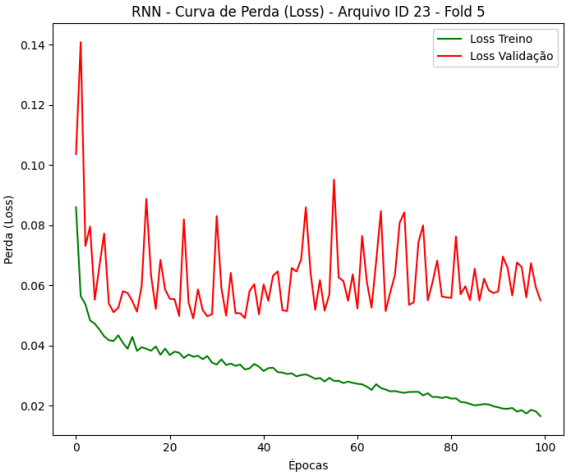
LSTM Fold 4 -> RMSE: 0.9631, MAE: 0.7261, MAPE: 18.29%

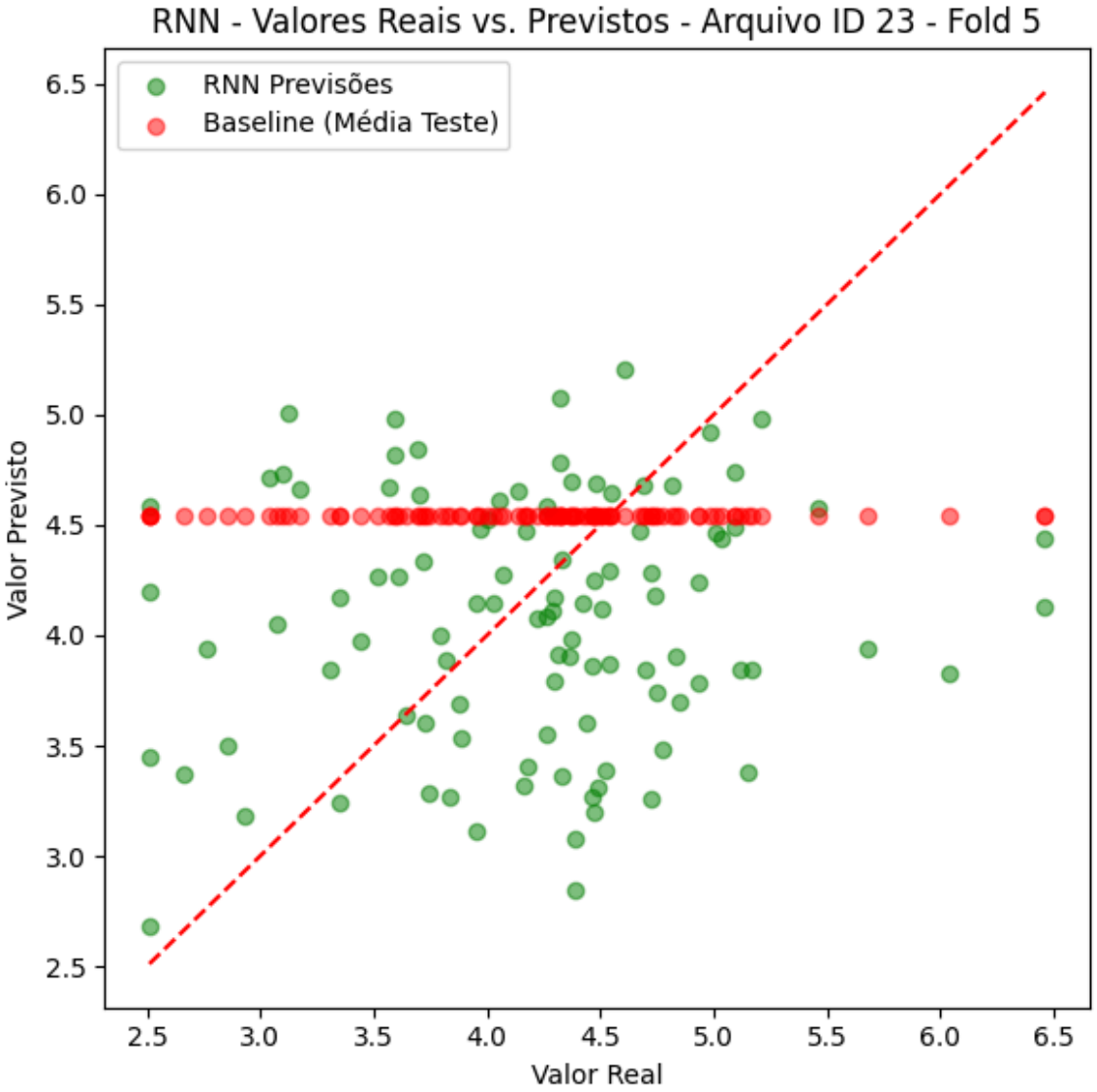
Fold 5

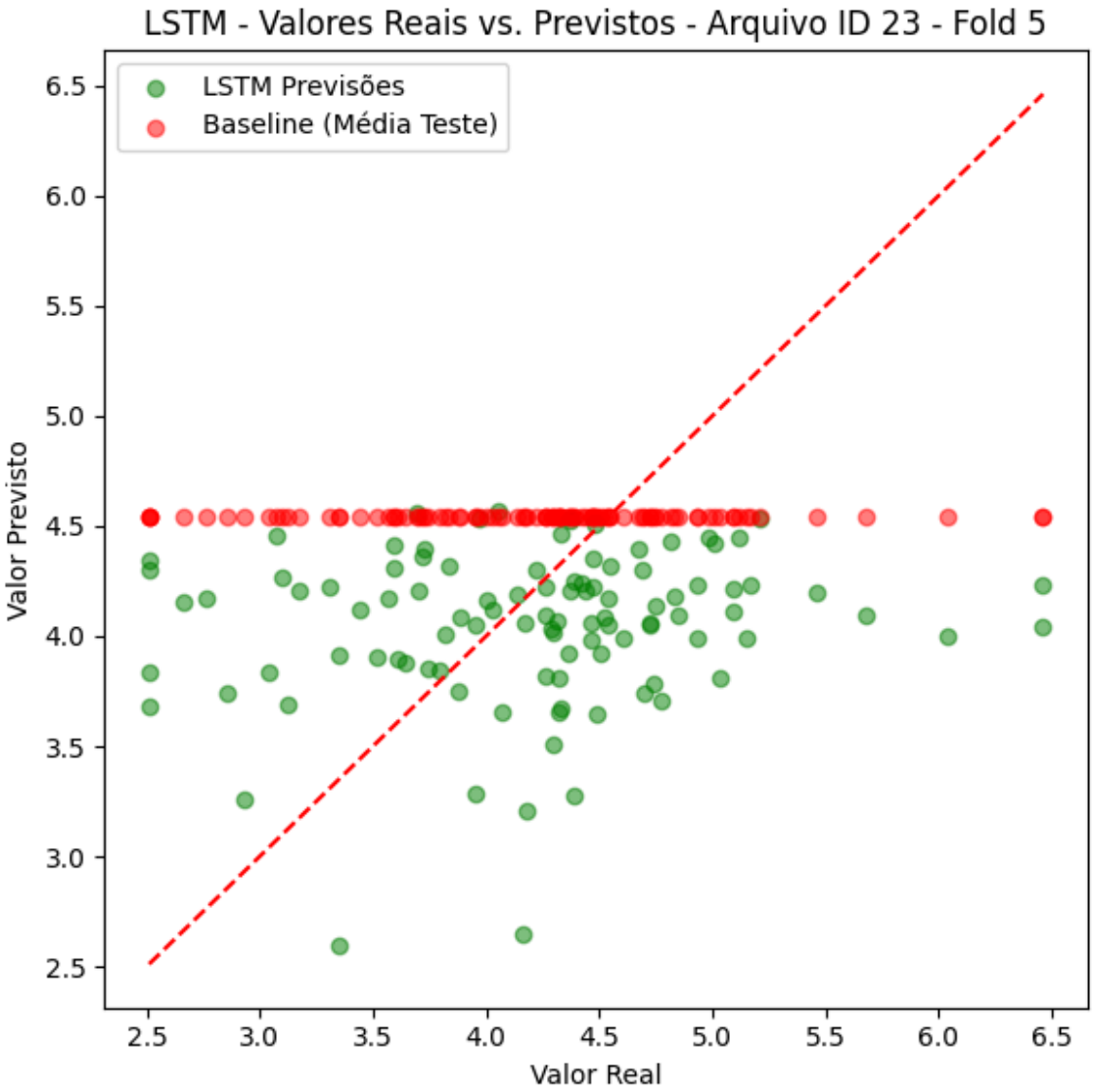

LSTM - Resultados - Arquivo ID 23 - Fold 5
Valor Real
LSTM Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


LSTM - Curva de Perda (Loss) - Arquivo ID 23 - Fold 5
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Resultados - Arquivo ID 23 - Fold 5
Valor Real
RNN Previsto
Baseline (Média Teste)
KM/L ou L/h
Amostras


RNN - Curva de Perda (Loss) - Arquivo ID 23 - Fold 5
Loss Treino
Loss Validação
Perda (Loss)
Épocas


RNN - Valores Reais vs. Previstos - Arquivo ID 23 - Fold 5
RNN Previsões
Baseline (Média Teste)
Valor Previsto
Valor Real
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5



Baseline (Média do Teste) Fold 5 -> RMSE: 0.8579, MAE: 0.6451, MAPE: 17.95%

RNN Fold 5 -> RMSE: 0.9267, MAE: 0.7451, MAPE: 18.54%

LSTM Fold 5 -> RMSE: 0.8294, MAE: 0.6653, MAPE: 16.77%

Média dos teste Arquivo ID 23

Baseline (Média do Teste) Média sobre 5 folds -> RMSE: 0.8235, MAE: 0.6455, MAPE: 15.54%

RNN Média sobre 5 folds -> RMSE: 1.2417, MAE: 1.0298, MAPE: 23.69%

LSTM Média sobre 5 folds -> RMSE: 1.0199, MAE: 0.8023, MAPE: 18.94%

Nesse arquivo tivemos algo parecido com o Arquivo ID10 onde a LSTM teve desempenho melhor que o RNN, com MAPE médio de 18.94%. Apesar disso, a LSTM apresentou instabilidades em alguns folds (como o Fold 1 e Fold 2), sugerindo possível overfitting dos dados. A RNN foi o modelo com o pior desempenho médio,

apresentando MAPE de 23.69%. Em todos os folds, a RNN apresentou dificuldade em capturar padrões dos dados, resultando em erros maiores.

Outro ponto importante a se notar é a análise comparativa com o baseline. Observa-se que os modelos apresentam métricas piores do que a baseline, indicando que eles não estão conseguindo capturar os padrões complexos presentes nos dados. Isso reforça a necessidade de ajustes tanto no modelo quanto na abordagem de tratamento dos dados para alcançar previsões mais precisas e confiáveis.

Contudo, de acordo com os resultados obtidos, observamos que os modelos estão apresentando aprendizado, porém os resultados ainda não são totalmente satisfatórios. Esse desempenho limitado pode estar relacionado a diversos fatores, como a complexidade dos dados, a forma como os dados estão sendo tratados.

Existem pontos de melhoria a serem considerados

Tratamento dos dados: Melhorar o pré-processamento, identificando e removendo outliers, normalizando variáveis e explorando a criação de novas features relevantes.

Ajuste de hiperparâmetros: Realizar uma busca sistemática por melhores valores utilizando funções de otimização de hiperparâmetros, como Grid Search ou Random Search, disponíveis em bibliotecas como Keras.

Redução do overfitting: Implementar métodos já disponíveis, como Dropout, Regularização L2 e Early Stopping, que ajudam a melhorar a capacidade de generalização dos modelos.

Exploração de novos modelos: Considerar arquiteturas alternativas, como GRU ou modelos baseados em Transformers, que têm mostrado eficiência na previsão de séries temporais.

7.Conclusão

A crescente preocupação com a sustentabilidade ambiental e a necessidade de reduzir as emissões de poluentes destacam a relevância do presente estudo, que integra ciência de dados e inteligência artificial como ferramentas para otimizar a eficiência veicular e minimizar os impactos ambientais associados ao setor de transporte. O trabalho buscou desenvolver e aplicar modelos baseados em Redes Neurais Recorrentes (RNNs) e Long Short-Term Memory (LSTM), capazes de prever padrões de consumo de combustível e emissões de poluentes com alta precisão, utilizando séries temporais de dados de veículos. Adicionalmente, a análise multivariada realizada com o algoritmo Random Forest proporcionou uma compreensão aprofundada sobre a influência de variáveis independentes no desempenho automotivo, oferecendo insights valiosos para o desenvolvimento de estratégias mais eficazes.

Os resultados obtidos confirmam o potencial das técnicas avançadas de aprendizado de máquina para lidar com conjuntos de dados heterogêneos e complexos, identificando padrões e relações que escapam às abordagens tradicionais. A aplicação de redes neurais demonstrou ser uma alternativa robusta para modelar a relação dinâmica entre as variáveis de entrada, enquanto o uso de Random Forest e outros algoritmos complementares permitiu avaliar a relevância de cada variável para o problema em questão.

Embora avanços significativos tenham sido alcançados, o estudo também evidencia os desafios ainda presentes na implementação dessas soluções em escala industrial. A disponibilidade e a qualidade dos dados, as limitações de infraestrutura e os custos associados à modernização tecnológica são barreiras que precisam ser superadas para que os benefícios dessas abordagens sejam amplamente adotados. Além disso, o impacto ambiental da cadeia de produção e descarte de tecnologias, como as baterias de veículos elétricos, deve ser considerado no contexto de soluções integradas para a sustentabilidade.

Em conclusão, este trabalho reforça a importância da ciência de dados e da inteligência artificial no enfrentamento de desafios ambientais e econômicos relacionados às emissões veiculares. A combinação de técnicas preditivas e análises multivariadas representa um avanço significativo no entendimento e na mitigação dos impactos do setor automotivo. Ao otimizar processos existentes e propor soluções

inovadoras, esta pesquisa contribui para um desenvolvimento mais sustentável, alinhando-se aos objetivos globais de redução de emissões e melhoria da eficiência energética.

Temos como trabalhos futuros para esse projeto como, estudar mais modelos para melhorar o resultado, testar com outros dados para ter uma noção maior dos resultados, melhorar a forma em que os dados são tratados e expandir os modelos para as variáveis de emissão.

8. Referências Bibliográficas

[1] FERNÁNDEZ, Y.; LÓPEZ, M.; BLANCO, B. Innovation for sustainability: The impact of R&D spending on CO2 emissions. Journal of Cleaner Production, v. 172, p. 3459-3467, 2018. Disponível em: https://www.sciencedirect.com/science/article/pii/S0959652617326513?via%3Dihub. Acesso em: Abril/2024.

[2] GAO, W.; JIANG, Z.; OZBAY, K. Data-driven adaptive optimal control of connected vehicles. IEEE Transactions on Intelligent Transportation Systems, v. 18, p. 1122-1133, 2017. Disponível em: https://ieeexplore.ieee.org/document/7548322. Acesso em: Abril/2024.

[3] SBAYTI, H. et al. Automotive emissions in developing countries: Traffic management and technological control measures. Environmental Engineering Science, v. 18, p. 347-358, 2001. Disponível em: https://www.liebertpub.com/doi/10.1089/109287501753359582. Acesso em: Abril/2024.

[4]LEIBENSPERGER, E. et al. Intercontinental influence of NOx and CO emissions on particulate matter air quality. Atmospheric Environment, v. 45, p. 3318-3324, 2011. Disponível em: https://www.sciencedirect.com/science/article/pii/S1352231011001579?via%3Dihub. Acesso em: Maio/2024.

[5]SINGH, P.; YADAV, D. Link between air pollution and global climate change. 2021. p. 79-108. Disponível em:

https://www.sciencedirect.com/science/article/abs/pii/B9780128229286000095?via%3Dihub. Acesso em: Maio/2024.
[6]GIAMPIERI, A. et al. Moving towards low-carbon manufacturing in the UK automotive industry. Energy Procedia, 2019. Disponível em: https://www.sciencedirect.com/science/article/pii/S1876610219309944?via%3Dihub. Acesso em: Maio/2024.
[7]VEÍCULOS comercializados a partir de 2022 emitirão menos poluentes. Disponível em: https://www.gov.br/ibama/pt-br/assuntos/noticias/2021/veiculos-comercializados-a-partir-de-2022-emitirao-menos-poluentes. Acesso em: Maio/2024.
[8]MOTALLEBIARAGHI, F. et al. High-fidelity modeling of light-duty vehicle emission and fuel economy using deep neural networks. SAE Technical Paper Series, 2021. Disponível em: https://doi.org/10.4271/2021-01-0181. Acesso em: Maio/2024.
[9] Louppe, Gilles. “Understanding Random Forests: From Theory to Practice.” *arXiv: Machine Learning* (2014): n. pag.123
[10] KURSA, M.; RUDNICKI, W. Feature Selection with the Boruta Package. Journal of Statistical Software, v. 36, p. 1-13, 2010. Disponível em: https://www.jstatsoft.org/article/view/v036i11. Acesso em: Jun/2024.
[11]WYNER, Abraham J. et al. Explaining the success of adaboost and random forests as interpolating classifiers. Journal of Machine Learning Research, v. 18, n. 48, p. 1-33, 2017.
[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning, volume 2. Springer, 2009.
[13]INTERNATIONAL ENERGY AGENCY. Global EV Outlook 2019. Disponível em: https://www.iea.org/reports/global-ev-outlook-2019. Acesso em: jun/2024.
[14]DE OLIVEIRA SANTOS, Alessandro et al. Impactos socioambientais decorrentes da nova geração de baterias aplicadas em carros elétricos. Revista Expressão Da Estácio, v. 4, n. 1, p. 42-53, 2020.
[15]PANDOLFI, A.; ADINOLFI, E.; POLVERINO, P.; PIANESE, C. Real-Time Prediction of Fuel Consumption via Recurrent Neural Network (RNN) for Monitoring, Route Planning Optimization and CO2 Reduction of Heavy-Duty Vehicles. SAE Technical Paper 2023-24-0175, 2023. Disponível em: https://doi.org/10.4271/2023-24-0175. Acesso em: jun/2024..
[16]HASSAN, M. A.; SALEM, H.; BAILEK, N.; KISI, O. Random Forest Ensemble-Based Predictions of On-Road Vehicular Emissions and Fuel Consumption in Developing Urban Areas. Sustainability, v. 15, n. 2, p. 1503, 2023. Disponível em:

https://www.mdpi.com/2071-1050/15/2/1503. Acesso em: jun/ 2024.
[17] NELSON, David M. Q.; HOELLER, Arliones; PEREIRA, Allan C. M.; OLIVEIRA, Ricardo A. de. Stock market's price movement prediction with LSTM neural networks. 2017. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7966019. Acesso em: 12 nov. 2024.
[18] XUE, Hao; HUYNH, Du Q.; REYNOLDS, Mark. SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction. 2018. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8354239 . Acesso em: 12 nov. 2024.
[19] NADALETI, Willian Cézar; PRZYBYLA, G. SI engine assessment using biogas, natural gas and syngas with different content of hydrogen for application in Brazilian rice industries: Efficiency and pollutant emissions. 2020. Disponível em: https://www.sciencedirect.com/science/article/pii/S0360319918311984. Acesso em: 12 nov. 2024.
[20] ALFASEEH, Lama; TU, Ran; FAROOQ, Bilal; HATZOPOULOU, Marianne. Greenhouse Gas Emission Prediction on Road Network using Deep Sequence Learning. 2020. Disponível em: https://arxiv.org/abs/2004.08286. Acesso em: 12 nov. 2024.
[21] LI, Xiangqian. A Comparative Study of Statistical and Machine Learning Models on Near-Real-Time Daily Emissions Prediction. 2023. Disponível em: https://arxiv.org/abs/2302.01152. Acesso em: 12 nov. 2024.
[22] SIAMI-NAMINI, Sima; TAVAKOLI, Neda; SIAMI NAMIN, Akbar. A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM. 2019. Disponível em: https://arxiv.org/abs/1911.09512. Acesso em: 12 nov. 2024.
[23] KHALIFA, Nevine; KHALIFA, Ahmed; EL-TAWIL, Khaled; EL-SHENNAWY, Mohamed. Random Forest Ensemble-Based Predictions of On-Road Vehicular Emission Rates under Real-World Driving Conditions. Sustainability, v. 15, n. 2, p. 1503, 2023. Disponível em: https://www.mdpi.com/2071-1050/15/2/1503. Acesso em: 6 dez. 2024.
[24] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning.* Cambridge: MIT Press, 2016. Disponível em: https://www.deeplearningbook.org/. Acesso em: 10 mai. 2024.
[25] GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems, v. 28, n. 10, p. 2222–2232, 2017. Disponível em:

https://doi.org/10.1109/TNNLS.2016.2582924. Acesso em: 10 mai. 2024.
[26] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. Neural Computation, v. 9, n. 8, p. 1735–1780, 1997. Disponível em: https://doi.org/10.1162/neco.1997.9.8.1735. Acesso em: 10 mai. 2024.