

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA**  
**INSTITUTO DE CIÊNCIAS EXATAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**Paulo Vitor Barbosa Ramos**

**Previsão de demanda de energia elétrica residencial utilizando aprendizado  
profundo**

Juiz de Fora

2023

**Paulo Vitor Barbosa Ramos**

**Previsão de demanda de energia elétrica residencial utilizando aprendizado profundo**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Saulo Moraes Villela

Coorientador: Prof. Dr. Bruno Henriques Dias

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Ramos, Paulo Vitor Barbosa.

Previsão de demanda de energia elétrica residencial utilizando aprendizado profundo / Paulo Vitor Barbosa Ramos. – 2023.

135 f. : il.

Orientador: Saulo Moraes Villela

Coorientador: Bruno Henriques Dias

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2023.

1. Previsão de demanda. 2. Energia elétrica. 3. Aprendizado profundo. 4. Seleção de atributos. I. Villela, Saulo Moraes, orient. II. Dias, Bruno Henriques, coorient. III. Título.

**Paulo Vitor Barbosa Ramos**

**Previsão de demanda de energia elétrica residencial utilizando aprendizado profundo**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Ciência da Computação.

Aprovada em 13 de janeiro de 2023.

**BANCA EXAMINADORA**

**Prof. Dr. Saulo Moraes Villela** - Orientador

Universidade Federal de Juiz de Fora

**Prof. Dr. Bruno Henriques Dias** - Coorientador

Universidade Federal de Juiz de Fora

**Prof. Dr. Heder Soares Bernardino**

Universidade Federal de Juiz de Fora

**Prof. Dr. Hugo Morais**

Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento - Lisboa/Portugal

Juiz de Fora, 22/12/2022.



Documento assinado eletronicamente por **Saulo Moraes Villela, Professor(a)**, em 09/03/2023, às 14:35, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Bruno Henriques Dias, Professor(a)**, em 10/03/2023, às 08:39, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Hugo Gabriel Valente Moraes, Usuário Externo**, em 10/03/2023, às 09:16, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Heder Soares Bernardino, Professor(a)**, em 10/03/2023, às 09:20, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Paulo Vitor Barbosa Ramos, Usuário Externo**, em 10/03/2023, às 11:00, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf ([www2.ufjf.br/SEI](http://www2.ufjf.br/SEI)) através do ícone Conferência de Documentos, informando o código verificador **1089542** e o código CRC **6CBB8B86**.

---

À Lá, que sempre me incentivou ao estudo.

## AGRADECIMENTOS

Após ter um pequeno deslumbre da formação deste trabalho, noto diversas pessoas que passaram e permaneceram. Possuo uma imensa gratidão por todas essas pessoas, pois é através delas que é possível ter um vislumbre da evolução de nosso caráter, tanto pelas adversidades impostas quanto pelo auxílio dado. Independente de qual seja o tipo, um aproveitamento pode ser tirado pelo tempo que permanecemos ao seu lado.

Uma pessoa que sempre esteve ao meu lado é minha mãe, Alair (Lá), tanto nas horas de raiva, ansiedade e desespero, quanto naquelas em que o Sol sempre iria brilhar. É essa pessoa que tenho o dever de agradecer, pois sempre me ensinou a importância em criar uma estrutura pelo estudo, formação e conhecimento, deixando o dinheiro, algo passível de se tomado, como consequência da base criada. É à ela que agradeço pela cobrança, encorajamento às audácias, educação e ao questionamento. Sem ela, não haveria este documento.

Não poderia esquecer de agradecer à Lilian. À ela agradeço pelas horas de revisões e comentários feitos nas diversas edições deste documentos.

Tenho muito a agradecer ao Saulo e Bruno, por sempre estarem disponíveis para sanar meus questionamentos, lerem repetidas vezes os trechos escritos e direcionar as etapas do projeto experimental. São os ombros que me aprumo para ver o brilho do Sol e desejar nunca mais voltar para a escuridão.

Agradeço também a todos os professores do PPGCC pelas aulas e dicas durante minha formação, em especial ao Roberto Massi e Luciana Campos. Não poderia deixar de agradecer à Walquiria Silva, Sarah Abreu, Kevyn Swahants, Anrafel Fernandes, Mateus Marim e André Brito. Obrigado a todos vocês pelo suporte, atenção, cuidado e presteza sempre dado ao longo desses anos. À CAPES, pelo apoio financeiro imprescindível para obtenção das ferramentas para execução dos testes experimentais.

Muito obrigado a todos.

“O que fazemos agora ecoa pela eternidade.” (MARCO AURÉLIO).

## RESUMO

Prever a futura demanda de energia elétrica de uma residência possui aplicações em diversas escalas. Partindo da constatação dos indicadores de qualidade do fornecimento de energia, passando pelo gerenciamento do melhor despacho em um sistema interligado, chegando até a fundamentação de investimentos em expansão. Prever a demanda possui um importante papel no desenvolvimento do setor energético. Para tanto, são diversos métodos que auxiliam a inferência desse consumo no mais variado horizonte temporal. Modelos estatísticos, máquinas de vetores suporte e árvores de decisão são algumas maneiras de realizar tal tarefa, porém as redes de aprendizado profundo vêm recebendo um grande destaque pelo baixo erro de inferência obtido, onde múltiplas propriedades vinculadas ao consumo estão presentes. Devido à permissibilidade das redes recorrentes ao problema de previsão de séries temporais, torna-se interessante compará-las à arquitetura *Transformers*, que além de também trazer aspectos de persistência durante o treinamento, está inserida no estado da arte. Não limitado ao comparativo do comportamento entre modelos, é possível levantar estudos de caso de redução de características e granularidade de tempo, a fim de constatar o impacto que as propriedades e o número de amostras possui no desempenho das arquiteturas estudadas. Muito embora seja possível avaliar desempenho, o comparativo torna-se mais completo com a interpretabilidade dos modelos, constatando o impacto dos atributos e, para o conjunto de teste, a importância média de cada característica utilizada no treinamento do modelo. Baseado em um referencial teórico, o comparativo estudado demonstra um ótimo desempenho na utilização dos modelos recorrentes, *Transformers* e baseados em camadas convolucionais. Partindo de um intervalo fixo de tempo, os estudos de caso demonstraram que, à medida que o número de amostras diminui com a resolução de tempo analisado, a redução de atributos faz piorar o erro médio quadrático dos modelos, portanto, a seleção traz bons resultados nas granularidades temporais que trazem um maior número de amostras disponíveis para o treinamento do modelo. Melhoras de até 10,9% são passíveis de serem alcançadas quando o método de fusão tardia por votação, otimizado pela meta-heurística de resfriamento simulado, é utilizado.

Palavras-chave: Previsão de demanda. Energia elétrica. Aprendizado profundo. Seleção de atributos.

## ABSTRACT

Predicting a future electrical energy demand of a residence gets several applications on many scales. Starting with the energy supply quality indicator, going through the best management of dispatch on interconnect energy system, and getting to the bottom of expansion investments. Infer the demand takes a significant role in developing the energy segment. Therefore, certain methods help to infer consumption on various time horizons. Statistical models, support vector machines, and decision trees are some of the ways to solve the task, but deep neural networks come with a great focus on decreasing the error on the prediction, in which multiple features are presented with the consumption reading. Due to the recurrent neural network permissibility of the time-series prediction problem, it becomes interesting to compare these networks with Transformers, recently inserted on state of the art. Not limited by the comparison of model behaviors, it is possible to raise case studies on feature selection and time resolution, intended to see the impact of those features and the number of samples on the studied model's architecture. Even though it is possible to evaluate the performance, the comparison becomes more complete with the interpretability of the models, noticing the feature force and average importance for the whole testing set. The comparison shows a great performance using recurrent models, Transformers, and those with convolutional layers. The case study shows that with the decrease of samples parallel with the resolution, the feature selection makes the accuracy degrade, thus, the selection makes good results where the training set comes with greater time resolution. All the results are possible to become better with the voting ensemble method, optimized by the simulated annealing metaheuristic.

Keywords: Demand prediction. Electrical energy. Deep learning. Feature selection.

## LISTA DE FIGURAS

Figura 1	– Modelo <i>Perceptron</i> de Frank Rosenblatt. . . . .	25
Figura 2	– Demonstrativo da separação (a) linear e (b) não linear de classes. . . . .	26
Figura 3	– Representação gráfica das redes de aprendizado com múltiplas camadas. . . . .	27
Figura 4	– Modelo RNN. . . . .	31
Figura 5	– Diagrama de componentes inseridos nas <i>Vanillas</i> RNNs. . . . .	31
Figura 6	– Representação de uma camada da arquitetura LSTM. . . . .	33
Figura 7	– Representação de uma camada da arquitetura GRU. . . . .	35
Figura 8	– Diferenças principais nas arquiteturas de recorrência. . . . .	36
Figura 9	– Componentes presentes da arquitetura <i>Time-Series Transformers</i> (TST). . . . .	37
Figura 10	– Esquema para construção do método <i>Rolling Window</i> . . . . .	39
Figura 11	– Janelamento de amostras dependente da inferência abordada. . . . .	40
Figura 12	– Demonstração visual dos dados dispostos tridimensionalmente. . . . .	41
Figura 13	– Fluxograma do processo do SA. . . . .	43
Figura 14	– Fluxo da metodologia empregada para análise do comportamento dos modelos. . . . .	49
Figura 15	– <i>Boxplot</i> das características do banco de dados <i>Pecan Street Inc</i> , em resolução de 1 minuto. . . . .	57
Figura 16	– <i>Boxplot</i> das características do banco de dados HUE, em resolução de 1 hora. . . . .	58
Figura 17	– Números de componentes que melhor descreve o <i>dataset</i> . . . . .	59
Figura 18	– <i>Boxplot</i> comparativo dos <i>datasets</i> resultantes da primeira parte da metodologia empregada. . . . .	60
Figura 19	– Análise PDF das diferentes resoluções dos <i>datasets</i> . . . . .	60
Figura 20	– O subgráfico (a) mostra o gráfico QQ utilizando a base da HUE, na resolução de 1 hora. Os subgráficos (b) à (d) mostram o comportamento dos quantiles utilizando os dados da <i>Pecan Street</i> , respectivamente, nas resoluções de 1 hora, 15 minutos e 1 minuto. . . . .	61
Figura 21	– Cálculo da autocorrelação por ACF e PACF nos <i>datasets</i> compilados em 30 <i>lags</i> . . . . .	62
Figura 22	– Previsões das últimas 50 amostras utilizando modelo estatístico SARIMAX. As figuras (a) à (c) detalham as previsões utilizando os dados da <i>Pecan Street</i> , nas resoluções de 1 minuto, 15 minutos e 1 hora, respectivamente. . . . .	63
Figura 23	– Comparativo das curvas de validação para os modelos recorrentes em relação a resolução as três resoluções abordadas pelo <i>dataset Pecan Street</i> . . . . .	66
Figura 24	– Comparativo entre as curvas de validação dos dados da <i>Pecan Street Inc.</i> , utilizando todas as características e aquelas vistas como mais importantes por PCA, em todas as resoluções disponíveis. . . . .	67

Figura 25 – Comportamento da validação utilizando a coleção horária de dados <i>HUE Dataverse</i> , seguindo o comparativo entre os atributos originais e PCA. . . . .	68
Figura 26 – Previsão comparativa utilizando os dados da <i>Pecan Street Inc.</i> entre todas as características e PCA de todos os modelos recorrentes. . . . .	68
Figura 27 – Comportamento da previsão utilizando as coleções de dados da <i>HUE Dataverse</i> , seguindo o comparativo entre todas os atributos e os reduzidos por PCA. . . . .	69
Figura 28 – Comparativo do MSE de avaliação utilizando as resoluções da <i>Pecan Street</i> com todas as características e reduzidas por PCA. . . . .	69
Figura 29 – Comparativo do MSE de avaliação utilizando as resolução horária da HUE com todas as características e reduzidas por PCA. . . . .	69
Figura 30 – Avanço do conjunto de validação para as três resoluções disponíveis na base da <i>Pecan Street</i> utilizando o modelo TST. . . . .	70
Figura 31 – Avanço do treinamento da arquitetura TST utilizando os dados da <i>Pecan Street</i> , em circunstâncias de utilização total dos atributos e reduzidas por PCA. . . . .	71
Figura 32 – Avanço do treinamento da arquitetura TST utilizando a base horária da HUE, em circunstâncias de utilização total dos atributos e reduzidas por PCA. . . . .	71
Figura 33 – Resolução horária. . . . .	71
Figura 34 – Previsão comparativa do modelo TST utilizando os dados da <i>Pecan Street Inc.</i> entre todas as propriedades e as reduzidas por PCA. . . . .	72
Figura 35 – Previsão comparativa do modelo TST utilizando a base horária da HUE entre todas as propriedades e as reduzidas por PCA. . . . .	72
Figura 36 – Arquitetura TST - Resolução de 1 hora. . . . .	72
Figura 37 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA. . . . .	73
Figura 38 – Comparativo dos MSEs alcançados utilizando os modelos recorrentes e TST para a base horária da HUE. . . . .	73
Figura 39 – Comparativo do avanço da validação para os modelos secundários de DL, tendo como base os dados da <i>Pecan Street</i> , utilizando todas as características originalmente pré-processadas. . . . .	75
Figura 40 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando as bases com todas as características e por pré-processamento por PCA. As Figuras (a) demonstra todas os atributos compilados e (b) os reduzidos por PCA, enquanto a (c) a única resolução disponível na HUE. . . . .	76
Figura 41 – Impacto que cada característica possui para a inferência de um único valor. . . . .	78
Figura 42 – Importância média das característica para a inferência dos valores de demanda utilizando os dados da <i>Pecan Street</i> , na resolução de 1 minuto. . . . .	79

Figura 43 – Comparativo do avanço do erro de validação dos modelos utilizando as características relevantes encontradas por SHAP <i>Values</i> , utilizando os dados da <i>Pecan Street</i> . . . . .	80
Figura 44 – Inferências realizadas utilizando características realçadas por SHAP <i>Values</i> , utilizando os dados da <i>Pecan Street</i> para os modelos recorrentes e <i>Transformer</i> . . . . .	81
Figura 45 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando (a) todas as características compiladas; (b) aquelas vistas por PCA; e (c) usando o método de análise por SHAP <i>Values</i> . . . . .	82
Figura 46 – Ponderação de pesos para <i>Ensemble</i> . Os subgráficos (a) à (c) demonstram os dados da <i>Pecan Street</i> em sua originalidade. Os subgráfico (d) à (f) também mostram os dados da <i>Pecan Street</i> , mas utilizando o conjunto resumido por PCA. Gráfico (g) utiliza os dados da HUE originais e (h) os resumidos por PCA. . . . .	83
Figura 47 – Comparativo das inferências realizadas por modelos treinados isoladamente e pela fusão tardia por votação, otimizados por SA. . . . .	84
Figura 48 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando a fusão tardia otimizada por SA. As Figuras (a) e (b) demonstram o comparativo por resolução, enquanto a (c) a única resolução disponível na HUE. . . . .	85
Figura 49 – <i>Boxplot</i> das características do banco de dados <i>Pecan Street</i> , em resolução de 15 minutos. . . . .	100
Figura 50 – <i>Boxplot</i> das características do banco de dados <i>Pecan Street</i> , em resolução de 1 hora. . . . .	100
Figura 51 – Mapa de calor entre característica e número de componentes para o banco de dados <i>Pecan Street</i> , em resolução de 1 minuto. . . . .	101
Figura 52 – Mapa de calor entre característica e número de componentes para o banco de dados <i>Pecan Street</i> , em resolução de 15 minutos. . . . .	102
Figura 53 – Mapa de calor entre característica e número de componentes para o banco de dados <i>Pecan Street</i> , em resolução de 1 hora. . . . .	103
Figura 54 – Mapa de calor entre característica e número de componentes para o banco de dados HUE, em resolução de 1 hora. . . . .	104
Figura 55 – Mapa de calor da correlação existente entre característica para o banco de dados <i>Pecan Street</i> , em resolução de 1 minuto. . . . .	105
Figura 56 – Mapa de calor da correlação existente entre característica para o banco de dados <i>Pecan Street</i> , em resolução de 15 minutos. . . . .	106
Figura 57 – Mapa de calor da correlação existente entre característica para o banco de dados <i>Pecan Street</i> , em resolução de 1 hora. . . . .	107
Figura 58 – Mapa de calor da correlação existente entre característica para o banco de dados HUE, em resolução de 1 hora. . . . .	108

Figura 59 – Diagrama de classes do projeto de treinamento dos modelos. . . . .	109
Figura 60 – Inferência utilizando o modelo SARIMAX e os dados horarios da HUE. . . . .	110
Figura 61 – Os subgráficos (a) à (i) demonstram a melhora no treinamento por meio das curvas de treinamento e validação em relação às épocas, utilizando os dados da <i>Pecan Street</i> . Já os sub-gráficos (j) à (l) há a utilização dos dados da <i>HUE Dataverse</i> . . . . .	111
Figura 62 – Curva de treinamento da arquitetura TST utilizando os dados da <i>Pecan Street</i> . . . . .	112
Figura 63 – Avanço do treinamento do modelo TST utilizando a resolução horária da <i>HUE Dataverse</i> . . . . .	112
Figura 64 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da <i>Pecan Street</i> , na resolução de 1 minuto. . . . .	113
Figura 65 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da <i>Pecan Street</i> , na resolução de 15 minutos. . . . .	113
Figura 66 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da <i>Pecan Street</i> , na resolução de horária. . . . .	114
Figura 67 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da <i>HUE</i> , na resolução horária. . . . .	114
Figura 68 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução de 1 minuto. . . . .	115
Figura 69 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução de 15 minutos. . . . .	115
Figura 70 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução horária. . . . .	116
Figura 71 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da HUE, em resolução horária. . . . .	116
Figura 72 – Inferências de 50 amostras tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução de 1 minuto. . . . .	117
Figura 73 – Inferências de 50 amostras tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução de 15 minutos. . . . .	118
Figura 74 – Inferências de 50 amostras tendo como eixo a base de dados da <i>Pecan Street</i> , em resolução horária. . . . .	119
Figura 75 – Inferências de 50 amostras tendo como eixo a base de dados da HUE, em resolução horária. . . . .	120

Figura 76	– Impacto de característica para a inferência de um único valor utilizando os dados da <i>Pecan Street</i> , na resolução de 1 minuto. . . . .	121
Figura 77	– Impacto de característica para a inferência de um único valor utilizando os dados da <i>Pecan Street</i> , na resolução de 15 minutos, para os modelos principais. . . . .	122
Figura 78	– Impacto de característica para a inferência de um único valor utilizando os dados da <i>Pecan Street</i> , na resolução de 15 minutos, para os modelos complementares. . . . .	123
Figura 79	– Impacto de característica para a inferência de um único valor utilizando os dados da <i>Pecan Street</i> , na resolução horária, para os modelos principais. . . . .	124
Figura 80	– Impacto de característica para a inferência de um único valor utilizando os dados da <i>Pecan Street</i> , na resolução horária, para os modelos complementares. . . . .	125
Figura 81	– Impacto de característica para a inferência de um único valor utilizando os dados da HUE, na resolução horária, para os modelos principais. . . . .	126
Figura 82	– Impacto de característica para a inferência de um único valor utilizando os dados da HUE, na resolução horária, para os modelos complementares. . . . .	127
Figura 83	– Importância média das características para a inferência dos valores de demanda utilizando os dados da <i>Pecan Street</i> , na resolução de 1 minuto, para os demais modelos. . . . .	128
Figura 84	– Importância média das características para a inferência dos valores de demanda utilizando os dados da <i>Pecan Street</i> , na resolução de 15 minutos. . . . .	129
Figura 85	– Importância média das características para a inferência dos valores de demanda utilizando os dados da <i>Pecan Street</i> , na resolução horária. . . . .	130
Figura 86	– Importância média das características para a inferência dos valores de demanda utilizando os dados da HUE, na resolução horária. . . . .	131
Figura 87	– Comparativo dos modelos isolados ao modelo por fusão tardia por votação, utilizando os dados da <i>Pecan Street</i> . . . . .	132
Figura 88	– Comparativo dos modelos isolados ao modelo por fusão tardia por votação, utilizando os dados da HUE. . . . .	133
Figura 89	– Inferências realizadas utilizando características realçadas por <i>SHAP Values</i> , utilizando os dados da <i>Pecan Street</i> para os demais modelos treinados. . . . .	134
Figura 90	– Inferências realizadas utilizando características realçadas por <i>SHAP Values</i> , utilizando os dados da HUE para os demais modelos treinados. . . . .	135

## LISTA DE TABELAS

Tabela 1 – Aspectos descritivos das bases utilizadas em sua versão original. . . . .	55
Tabela 2 – Métricas do modelo SARIMAX para os dois <i>datasets</i> analisados. . . . .	64
Tabela 3 – Tempo de treinamento dos modelos de recorrência. . . . .	66
Tabela 4 – MSE alcançado pelos modelos empregados nas variações de dados. . .	95
Tabela 5 – MAE alcançado pelos modelos empregados nas variações de dados. . .	96
Tabela 6 – MAPE alcançado pelos modelos empregados nas variações de dados. .	97
Tabela 7 – R2 <i>Score</i> alcançado pelos modelos empregados nas variações de dados.	98

## LISTA DE ABREVIATURAS E SIGLAS

ACF	<i>Auto Correlation Function</i>
ACO	<i>Ant Colony Optimization</i>
ARMA	<i>Autogressive Moving Average</i>
ARIMA	<i>Autogressive Integrated Moving Average</i>
ART	<i>Adaptative Resonance Theory</i>
ConvRNN	<i>Convolutional Recorrent Neural Network</i>
CNN	<i>Convolutional Neural Network</i>
DL	<i>Deep Learning</i>
EC	<i>Evolutionary Computation</i>
FCN	<i>Fully Convolutional Network</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphical Processing Unit</i>
GRU	<i>Gated Recorrent Unit</i>
HUE	<i>Houly Usage Energy Database</i>
IA	<i>Inteligência Artificial</i>
KDD	<i>Knowledge Discovery in Database</i>
kNN	<i>K-Nearest Neighbors</i>
LIME	<i>Local Interpretable Model-agnostic Explanations</i>
LSTM	<i>Long-Short Therm Memory</i>
MLP	<i>Multi-Layer Perceptron</i>
MAE	<i>Mean Absolute Error</i>
MAPE	<i>Mean Absolute Percentage Error</i>
MSE	<i>Mean Squared Error</i>
NLP	<i>Neural Language Processing</i>
PACF	<i>Partial Auto Correlation Function</i>
PCA	<i>Principal Component Analysis</i>
PDF	<i>Probability Density Function</i>
QQ	<i>Quantil-Quantil</i>
ResNet	<i>Residual Network</i>
RNA	<i>Redes Neurais Artificiais</i>
RNN	<i>Recurrent Neural Network</i>
SA	<i>Simulated Annealing</i>
SARIMA	<i>Seasonal Autogressive Integrated Moving Average</i>
SARIMAX	<i>Seasonal Autogressive Integrated Moving Average - Exogenous Factors</i>
SHAP	<i>Shapley Additive Explanations</i>
SOTA	<i>State Of The Art</i>
SVD	<i>Singular-Value Decomposition</i>
SVR	<i>Support Vector Regression</i>
TCN	<i>Temporal Convolutional Network</i>
TST	<i>Temporal Time-Series Transformers</i>
XAI	<i>Explainable Artificial Intelligence</i>

## LISTA DE SÍMBOLOS

$y$	Saída.
$y'$	Resultado esperado.
$c_t$	Armazenamento de memória.
$L$	Camada.
$\sigma$	Função sigmoide.
$\alpha^L$	Saída do neurônio da camada $L$ .
$C_0$	Função Custo de Treinamento.
$Z^L$	Estado do neurônio.
$w^L$	Pesos da camada do neurônio analisado.
$h$	Estado escondido .
$f$	Função de ativação.
$x$	Vetor de entrada.
$RNN$	Soma dos quadrados da estimativa dos erros.
$TSS$	Soma total dos quadrados.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>18</b>
1.1	DEFINIÇÃO DO PROBLEMA . . . . .	19
1.2	OBJETIVOS . . . . .	20
1.3	CONTRIBUIÇÕES . . . . .	20
1.4	ORGANIZAÇÃO . . . . .	21
<b>2</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>22</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>	<b>24</b>
3.1	REDES NEURAIS ARTIFICIAIS . . . . .	24
<b>3.1.1</b>	<b><i>Backpropagation</i></b> . . . . .	<b>28</b>
3.2	REDES RECORRENTES . . . . .	30
<b>3.2.1</b>	<b><i>Vanilla RNN</i></b> . . . . .	<b>30</b>
<b>3.2.2</b>	<b>LSTM</b> . . . . .	<b>32</b>
<b>3.2.3</b>	<b>GRU</b> . . . . .	<b>35</b>
<b>3.2.4</b>	<b><i>Transformers</i></b> . . . . .	<b>36</b>
3.3	PREVISÃO DE SÉRIES TEMPORAIS . . . . .	38
3.4	FUSÃO TARDIA . . . . .	40
<b>3.4.1</b>	<b><i>Simulated Annealing</i></b> . . . . .	<b>42</b>
3.5	ANÁLISE DE CARACTERÍSTICAS . . . . .	44
<b>3.5.1</b>	<b>Análise de componentes principais</b> . . . . .	<b>45</b>
<b>3.5.2</b>	<b>Interpretabilidade por SHAP Values</b> . . . . .	<b>46</b>
<b>4</b>	<b>ABORDAGEM PROPOSTA . . . . .</b>	<b>48</b>
4.1	ETAPA DE PRÉ-PROCESSAMENTO . . . . .	48
4.2	ETAPA DE PÓS-TREINAMENTO . . . . .	51
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS . . . . .</b>	<b>53</b>
5.1	BASES DE DADOS . . . . .	53
5.2	IMPLEMENTAÇÃO . . . . .	55
5.3	RESULTADOS . . . . .	56
<b>5.3.1</b>	<b>Análise estatística</b> . . . . .	<b>59</b>
<b>5.3.2</b>	<b>Comportamento dos modelos recorrentes</b> . . . . .	<b>64</b>
<b>5.3.3</b>	<b>Comparativo dos modelos recorrentes e <i>Transformers</i></b> . . . . .	<b>70</b>
<b>5.3.4</b>	<b>Treinamento de outros modelos encontrados no SOTA</b> . . . . .	<b>74</b>
5.4	VARIÁVEIS MAIS IMPACTANTES POR SHAP VALUES . . . . .	76
<b>5.4.1</b>	<b>Treinamento das características analisadas</b> . . . . .	<b>77</b>
5.5	UTILIZAÇÃO DO ENSEMBLE POR SA . . . . .	80
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>86</b>
6.1	TRABALHOS FUTUROS . . . . .	87
	<b>REFERÊNCIAS . . . . .</b>	<b>88</b>

APÊNDICE A – MÉTRICAS COMPLETAS DE PERFORMANCE	94
APÊNDICE B – IMAGENS E RESULTADOS COMPLEMENTARES . . . . .	99

## 1 INTRODUÇÃO

A área de previsão temporal vem ao longo das últimas décadas recebendo sofisticadas ferramentas para tomadas de decisões. Redes neurais, modelos estatísticos e outros métodos de aprendizado de máquina são as principais ferramentas para a previsão mais assertiva de variáveis sensíveis a diversos sistemas (PETROPOULOS et al., 2022).

O trabalho de Petropoulos et al. (2022) fornece uma série de áreas passíveis de serem revolucionadas através da previsão temporal. O ramo energético, principal linha de pesquisa receptora dos métodos de previsão, possui aplicações capazes de obter um sistema eficaz de fornecimento de energia (SANTOS et al., 2020; STRIELKOWSKI, 2020), por exemplo. Prever demanda possui aplicação na detecção de anomalias (UNWIN, 2019; HADJOUT et al., 2022) ou inovações no mercado de energia no modelo par-a-par (*peer-to-peer*) (GOMES et al., 2022).

Possuir um horizonte a ser predito define a aplicação no campo energético. É possível citar: (i) a longo prazo, o qual fornece, por exemplo, o embasamento para futuros investimentos em expansão em transmissão (RICE et al., 2022); (ii) a médio prazo que, por outro lado, ajuda no efetivo controle dos indicadores de qualidade, assegurando os princípios de ininterruptibilidade e segurança do sistema interligado de energia (FERREIRA, 2008); (iii) a curto prazo, oferecendo o padrão de consumo diário, o que ajuda na busca pelo menor custo de utilização de energia, por meio da comutação entre os melhores ofertantes (LIN; WU; BOULET, 2021).

Atrelado ao aspecto temporal de inferência, o domínio de tensão deve ser levado em conta. Haben et al. (2021) expõem que as faixas de tensão aplicadas à transmissão e distribuição são recorrentemente abordadas nos estudos de caso, níveis geralmente aplicados a investimentos e garantia da eficiência do fornecimento do produto energético. Muito embora o trabalho de Haben et al. (2021) ressalte a importância da produção científica nesses domínios, ele traz em pauta a carência em casos de baixa tensão.

Muitos são os métodos capazes de serem aplicados a esses casos. A previsão estatística, tais como a utilização do modelos autorregressivo integrado de médias móveis (*Autoregressive Integrated Moving Average – ARIMA*) ou sazonal (*Seasonal Autoregressive Integrated Moving Average – SARIMA*) estão correntemente inseridos nas pesquisas voltadas à previsão temporal (GHOFRANI et al., 2011), mas somente a título comparativo por haver outras soluções com maior acuracidade de previsão. Mesmo havendo essa utilização voltada à comparação, ainda é possível se deparar com modelos matemáticos direcionados especificamente para a construção da previsão estatística, como em Proedrou (2021). Porém, como revisado por Haben et al. (2021), a gama de tratativas de séries temporais de demanda é grande.

Regressão linear (DING; BÉSANGER; WURTZ, 2015), árvores de decisão (CHEN

et al., 2018) e regressão por vetores suporte (*support vector regression* – SVR) (SOUSA; JORGE; NEVES, 2014) são outras formas de previsão que estão dando lugar às redes neurais artificiais, mais especificamente as de aprendizado profundo. Hong e Wang (2022) demonstram a popularização das redes neurais artificiais (RNA) aplicadas à previsão de demanda ressaltando que, de 1991 até 2021, 1245 pesquisas mencionaram técnicas de inteligência artificial e aprendizado de máquina, sendo 796 descrevendo somente aplicações de RNA em sua forma mais simples. Partindo de 2014, aplicações voltadas ao aprendizado profundo (*deep learning* – DL) ganharam maior destaque pelas possibilidades de customização e a acuracidade obtida com métodos compostos na construção de comitês (*ensembles*).

Muitas são as abordagens direcionadas ao DL. Redes recorrentes (KONG et al., 2019) e redes convolucionais (VOß; BENDER-SAEBELKAMPF; ALBAYRAK, 2018) são algumas das abordagens usadas para serem atreladas a modelos sendo treinados por dados históricos. A operação desta técnica recorrendo aos dados de consumo possibilita a percepção do nível de importância de cada uma, a correlação de outros atributos, tais como as meteorológicas, como visto em Sepulveda et al. (2021), Hong e Wang (2022), e tantos outros autores que modelam a previsão histórica do consumo de energia baseada em características vinculadas à ela.

Como é possível verificar, são muitos aspectos da tratativa do problema de previsão de demanda. Sendo possível utilizar métodos estatísticos, máquinas de vetores suporte, árvores de decisão, regressão linear, redes neurais artificiais e, por fim, modelos de aprendizado profundo. Além da clara visualização da importância de suas abordagens para o problema de previsão, fica evidente a sua relação para a economia, proteção ambiental e avanço da sociedade no correto suprimento de energia para o seu desenvolvimento.

## 1.1 DEFINIÇÃO DO PROBLEMA

Como supracitado, o problema diz respeito à inferência de futuras demandas energéticas em um horizonte temporal de curto prazo, em configurações máximas de até 1 semana. Como é possível destacar, diversos são os fatores influentes na demanda a serem observados, assim sendo, a série histórica a ser visualizada deve comportar diversas características atreladas à leitura de consumo.

Verificando o envolvimento de outros atributos (*features*) atrelados ao consumo, é possível verificar que a série histórica é multivariada, o que comporta com a oportunidade vista da verificação dos diferentes níveis de impacto ao consumo a ser inferido. Além de tudo, o problema de previsão da demanda diz respeito à percepção do comportamento dos diferentes métodos de abordar a série temporal, como a estatística, a regressão por vetores suporte e, principalmente, os modelos de aprendizado profundo.

De forma resumida, o problema abordado pode ser descrito como a inferência

da demanda energética de pequenos consumidores, em uma arquitetura *data-driven*. A influência de outras variáveis ao consumo a ser previsto é presumido, o que inclui a percepção do impacto das diferentes características à variável manipulada. Por fim, como são diferentes métodos passíveis de serem utilizados, o trabalho aqui descrito tem como foco a percepção do comportamento dos diferentes tipos de redes de aprendizado profundo, havendo uma descrição do fundamento de seu funcionamento e um comparativo com os demais métodos.

## 1.2 OBJETIVOS

O objetivo principal do presente trabalho é o estudo do comportamento de redes de aprendizado profundo, já difundidas e daquelas recém inseridas no estado da arte (*State of the Art* – SOTA), para prever a futura demanda energética de consumidores em baixos níveis de tensão. Influenciado por diversas características atreladas à ser inferida, inclui-se nesse objetivo a percepção do impacto que cada uma possui, tanto na etapa de pré-processamento dos dados utilizados quanto para o modelo particularmente analisado.

O horizonte temporal abordado é à curto-prazo. Portanto, o pré-processamento dos atributos atrelados à demanda energética será em resoluções de no máximo 1 hora. Como o trabalho direciona-se a constatação e comparação das métricas avaliativas, elucidando o processo de treinamento seguindo uma etapa de limpeza e engenharia de características, espera-se ajudar a pesquisa em soluções inseridas no mercado de energia e coordenabilidade do sistema energético.

Muitos são os modelos de DL, o que requer uma descrição de seu funcionamento atrelado ao problema visualizado. Além de tudo, outros métodos já foram utilizados. Portanto, torna-se oportuno comparar os diferentes resultados obtidos, sendo possível perceber a acuracidade levantada no seu uso isolado e na fusão tardia, sem perder a possível comparação com os demais métodos.

Ao final, é possível perceber o comportamento dos diferentes modelos durante suas etapas de treinamento e validação, tendo sua métrica de erro devidamente analisada. Além disso, por meios gráficos, é possível descrever a análise de características, tendo como fator determinante o modelo isoladamente treinado. Levantada tal análise, um comparativo pode ser visto tanto pela real previsão quanto pelos indicadores metodológicos de erros aplicados na regressão.

## 1.3 CONTRIBUIÇÕES

O presente documento, por meio do método comparativo, ajudará futuros trabalhos que abordarem o problema de séries temporais utilizando modelos de DL. A oferta da análise de características, respeitando a individualidade do modelo treinado, tendo feito o

devido *benchmarking*, em diferentes aspectos, é a principal contribuição do trabalho.

O conjunto composto pela análise de resultados, a fundamentação, a descrição de sua implementação e as conclusões obtidas para a inferência de futuras demandas energéticas influenciadas por diversas variáveis atreladas à ela, convergem para a conquista do principal objetivo proposto pela previsão de consumo: um completo comparativo do comportamento dos modelos de inferência influenciados por sua resolução e o conjunto de propriedades.

#### 1.4 ORGANIZAÇÃO

Além desse capítulo introdutório, o presente trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta os trabalhos relacionados, demonstrando os principais pontos correlacionados e como é obtida a coleção de documentos a ser analisada. O Capítulo 3 mostra a fundamentação do processo de treinamento dos modelos de inferência utilizados, conceituação das séries temporais, processo de fusão tardia por votação otimizada por meta-heurística e análise e interpretabilidade de atributos. O Capítulo 4 detalha a abordagem proposta para o pré-processamento e análise comparativa do comportamento e dos resultados obtidos. O Capítulo 5 mostra os resultados obtidos utilizando o método estabelecido, estabelecendo detalhes da base de dados, implementação e das inferências dos modelos. Por fim, o Capítulo 6 conclui o trabalho detalhando as percepções obtidas com o comparativo e estabelece uma futura abordagem no tema.

## 2 TRABALHOS RELACIONADOS

Hong e Wang (2022) destacam o aprendizado profundo como principal ferramenta na tarefa previsão do consumo energético, principalmente em aplicações de coordenação, gerenciamento do produto energético e investimentos no setor. Partindo dessas capacidades em prever a demanda energética, torna-se interessante constatar as diferentes abordagens, soluções e métodos para a sua concretização.

Partindo do *review* de Khalil et al. (2022), não só as diferentes aplicabilidades são ressaltadas, mas métodos de aprendizado e modelos são apresentados e fundamentados. Através deste *review*, é possível observar ferramentas e palavras-chaves, colaborando para a construção de uma simples *string* de busca: **“energy consumption AND forecasting AND deep learning”**, aplicada à base *Compendex*.<sup>1</sup> Reduzindo as publicações àquelas ocorridas em periódicos (*journals*) e anais de congresso no intervalo de 2019 à 2022, são 364 documentos passíveis de serem correlacionados ao presente trabalho.

Além de demonstrar aplicações em previsão, os trabalhos mostram novos métodos, técnicas e distintas maneiras em obter uma maior acuracidade na inferência do objeto. Portanto, é interessante constatar similaridades em métodos de pré-processamento, seleção de atributos, tipo de série temporal, descrição na obtenção dos resultados comparativos e tendências para futuras abordagens.

Forootani, Rastegar e Sami (2022) é um dos exemplos de trabalhos que fornecem a similaridade intencionada. O trabalho traz o motivo de abordar a previsão em ambientes residenciais. Fundamentada pela informação de que em 2019, 88% dos 94,8 milhões de medidores inteligentes instalados nos Estados Unidos foram no setor residencial, os autores destacam a importância de evolucionar o campo de pesquisa, principalmente quando aplicações em precisa tarifação de energia, rápida resposta de demanda e apurada programação do despacho de energia são mencionadas.

Muito embora a abordagem esteja restrita à inferência em curto prazo, Forootani, Rastegar e Sami (2022) destacam pontos importantes, como por exemplo a influência de atributos meteorológicos e a atenção em possuir um pré-processamento de limpeza, normalização e seleção de atributos, neste caso pelo método dos *K*-Vizinhos Mais Próximos (*K-Nearest Neighbors* – KNN). É por meio do debate do método de Forootani, Rastegar e Sami (2022) que observa-se uma semelhança com a contribuição aqui proposta, pois, utilizando uma série multivariada previamente pré-processada e reduzindo os atributos da base, os autores definem estudos de casos e comparam os erros de inferência dos modelos separadamente treinados.

Infelizmente, o trabalho de Forootani, Rastegar e Sami (2022) restringe sua compa-

---

<sup>1</sup> Também conhecida como *Engineering Village*, é uma base tecno-científica que reúne pesquisas de 190 disciplinas da engenharia.

ração ao estudar somente o impacto que as características possuem ao objeto de inferência. O único modelo de DL abordado é a rede recorrente *Gated Recurrent Unit* (GRU), com variações em sua camada de regressão. Oliveira, Sousa e Praça (2023) , por outro lado, comparam os modelos *Long-Short Term Memory* (LSTM), *Convolutional Neural Network* (CNN) e *Temporal Convolutional Network* (TCN) pautando a inferência em ambientes industriais, especificamente em uma empresa de manufatura. Todo método comparativo objetiva analisar as métricas performáticas utilizando diferentes granularidades de tempo. Apesar da existência de um cenário comparativo de desempenhos alcançados, é extremamente importante por apresentar a entrada dos modelos como tensores tridimensionais, detalhando a respectiva dimensionalidade do sequenciamento das leituras.

Dubey et al. (2021) não pautam qualquer análise de perfil residencial ou industrial. Seu trabalho é simplesmente a construção dos aspectos comparativos entre o modelo recorrente LSTM e estatísticos autorregressivos de médias móveis (*Autoregressive Moving Average* – ARMA) e ARIMA em sua fundamentação. Além dos conceitos que envolvem essas duas abordagens, o trabalho também torna-se interessante pela construção da análise correlacional existente quando o objeto de inferência é posto junto com outras variáveis, sendo a unidade temporal uma delas. Mesmo que seja um documento caracteristicamente baseado na apresentação do funcionamento dos modelos de previsão, pode ser encontrado o desempenho dos modelos, tão quanto a correlação das diferentes variáveis manipuladas para a inferência da demanda energética..

Em meio ao produto total de trabalhos passíveis de estudo, os documentos supracitados são configurados os mais relacionados por descrever o comportamento dos modelos para a inferência do objeto de pesquisa em diferentes cenários. Além de tudo, fundamenta seu processo de tratamento de erro e incluem a seleção de atributos para constatar a mudança da acurácia da previsão. Pode-se observar estudos de casos direcionados ao perfil de demanda, atributos correlacionados ao objeto de inferência, granularidades temporais e modelos de DL, assim como a dimensão do tensor melhor aceite. Ainda que seja possível destacar que haja um macro-panorama em abordar a inferência série temporal, Petropoulos et al. (2022) ajudam a obter os cuidados a serem tomadas durante a etapa de pré-processamento e instigam à constatar a seleção de atributos por outros métodos diferentes do KNN. Além do mais, traz o cenário de interpretabilidade do comportamento dos modelos, o que levou ao trabalho de Zeng et al. (2021), que também trata séries temporais, mas não focado ao objeto de demanda energética.

### 3 FUNDAMENTAÇÃO TEÓRICA

A pesquisa de trabalhos relacionados trouxe as principais tratativas em previsão de demanda energética, especificamente tendo as redes de aprendizado profundo como principal abordagem. É possível destacar aplicações, métodos, resultados, comparativos e novas propostas de análise. Muito embora os trabalhos realçados tratem macroscopicamente os fundamentos do funcionamento dos modelos de inferência, é por meio deste capítulo que é possível conhecer, de maneira aprofundada, o processo evolutivo do treinamento de um modelo de inteligência artificial (IA), conhecendo arquiteturas, métodos de análise e maneiras de mitigar os erros resultantes.

Partindo do campo evolutivo das redes neurais, o capítulo apresenta: (i) os primeiros avanços na modelagem de um neurônio computacional; (ii) o processo de treinamento de um conjunto destes neurônios, dispostos em camadas, através da propagação de erros; (iii) e a conceituação de persistência encontrada nas redes neurais recorrentes e suas variantes evolutivas. Após apresentar o fundamento das diferentes arquiteturas de DL aplicadas ao problema de regressão, são apresentadas técnicas de janelamento de dados na obtenção de tensores tridimensionais e o uso da meta-heurística de resfriamento simulado para a distribuição de pesos na fusão tardia por votação. Para a análise da influência das características, é vista a seleção de atributos por análise de componentes principais, situação que possibilita o exame do comportamento do modelo com uma reduzida coleção de variáveis atreladas ao consumo de energia. Tendo ressaltado a importância da interpretabilidade dos modelos, o capítulo também apresenta a ferramenta utilizada para observar a força dos atributos para uma única inferência e o grau de importância que cada uma possui no conjunto de dados testados. Após ter sido fundamentado os principais componentes do trabalho, é possível verificar o método de estudo comparativo, tendo como destaque o uso de diferentes bases de dados multivariadas e com diferentes granularidades de tempo.

#### 3.1 REDES NEURAIS ARTIFICIAIS

Adotar máquinas capazes de aprender possibilita a execução de atividades tão complexas que até então, somente seres humanos a partir de sua técnica e experiência, poderiam realizar. Identificar objetos, classificar um estado de ambiente, interpretar uma característica ou prever o comportamento de uma variável contínua no futuro (SILVA; SPATTI; FLAUZINO, 2016) são alguns exemplos.

Embora nem todas essas tarefas sejam atualmente passíveis de serem feitas por máquinas, elas poderão inferir de maneira similar o estado do ambiente a qual está inserido utilizando a abordagem baseada em dados (*data-driven*). Referindo-se a vinculação de um conjunto de amostras a uma conclusão adotada pelo observador especialista, a abordagem ajuda na construção de uma ponderação de pesos e vieses para inferir o objeto, de maneira

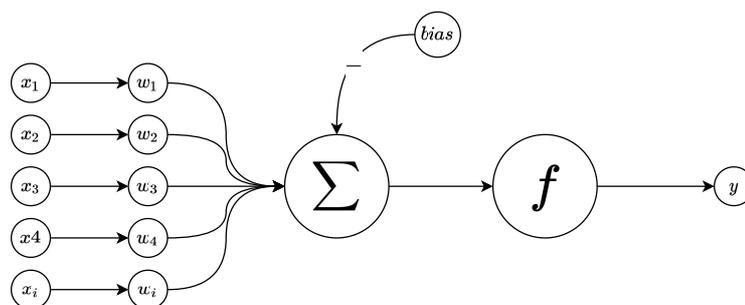
tão próxima quanto o esperado. Todo o processo de ajuste do comportamento da máquina vem da comparação de sua inferência com o previsto, o que traz o conceito de treinamento.

Contudo, o domínio de aprendizado de máquina não surgiu na última década. Foram sucessivas evoluções conceituais, inúmeras revoluções no campo de pesquisa e um breve período de baixa produção científica sobre o tema. Inicialmente com a representação do neurônio e sua formulação, o campo de pesquisa evoluiu a níveis mais avançados já vistos, produzindo soluções para problemas em identificação gráfica ou compressão de dados sem perda.

O primeiro avanço palpável foi em 1943, com o modelo de neurônio de McCulloch-Pitts. Consistindo a primeira modelagem matemática do funcionamento de um neurônio biológico, a reação é definida como uma excitação do estado à partir de suas entradas. Tratando-se de uma representação binária, Rosenblatt (1957) traz o modelo de McCulloch-Pitts como  $n$  entradas ponderadas por uma combinação linear, a qual possui um viés e um limitador, a sua função de ativação. O conceito revisto por Rosenblatt faz nascer o *Perceptron* e sua descrição matemática que, descrita na Equação (3.1), se tornou tão familiar e a Figura 1 amplamente reconhecida no campo da inteligência artificial.

$$y = f \left( \sum_{i=1}^n w_i x_i - \theta \right). \quad (3.1)$$

Figura 1 – Modelo *Perceptron* de Frank Rosenblatt.



Fonte: Elaborada pelo autor (2023).

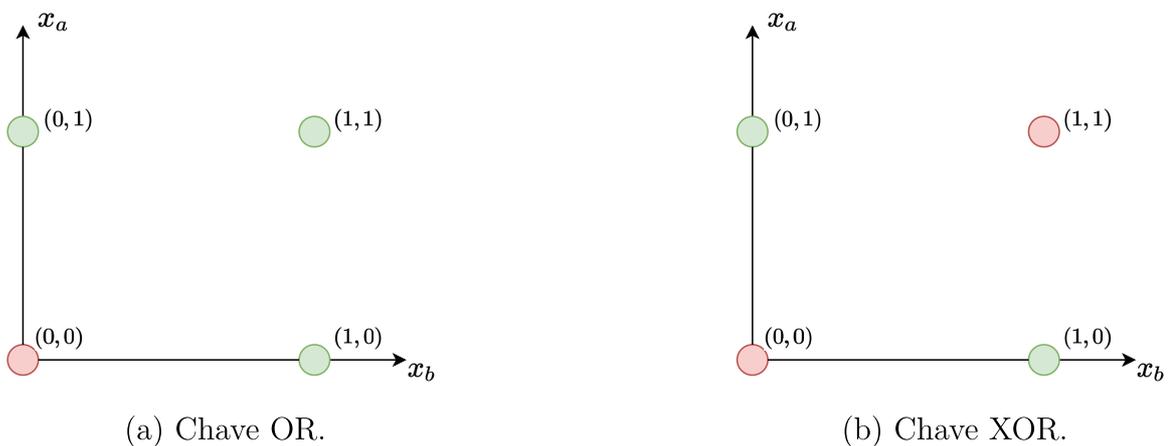
Como são diversas funções de ativação: tipo degrau, logística, exponencial ou sigmoide, todas possuindo a tarefa de definir o intervalo da saída, é possível, por exemplo, utilizar a função de ativação do tipo degrau para descrever o modelo matemático de *Rosenblatt* para obter as seguintes saídas:

$$y = f \left( \sum_{i=1}^n w_i x_i - \theta \right) = \begin{cases} 1, & \text{se } x = \sum_{i=1}^n w_i x_i - \theta \geq 0 \\ 0, & \text{se } x = \sum_{i=1}^n w_i x_i - \theta < 0. \end{cases} \quad (3.2)$$

Até o início da década de 70, o funcionamento do *Perceptron* consistia a melhor solução, porém, com a publicação de “*Perceptrons: An Introduction to Computational*

*Geometry*”, Minsky e Papert (1969) demonstraram que a representação não era suficiente para a separação de dados não-lineares, como uma chave *XOR*, exemplificado pela Figura 2b. Como consequência, a pesquisa no campo de inteligência artificial recebeu um número reduzido de publicações, entrando no período conhecido por “*AI Winter*”.

Figura 2 – Demonstrativo da separação (a) linear e (b) não linear de classes.



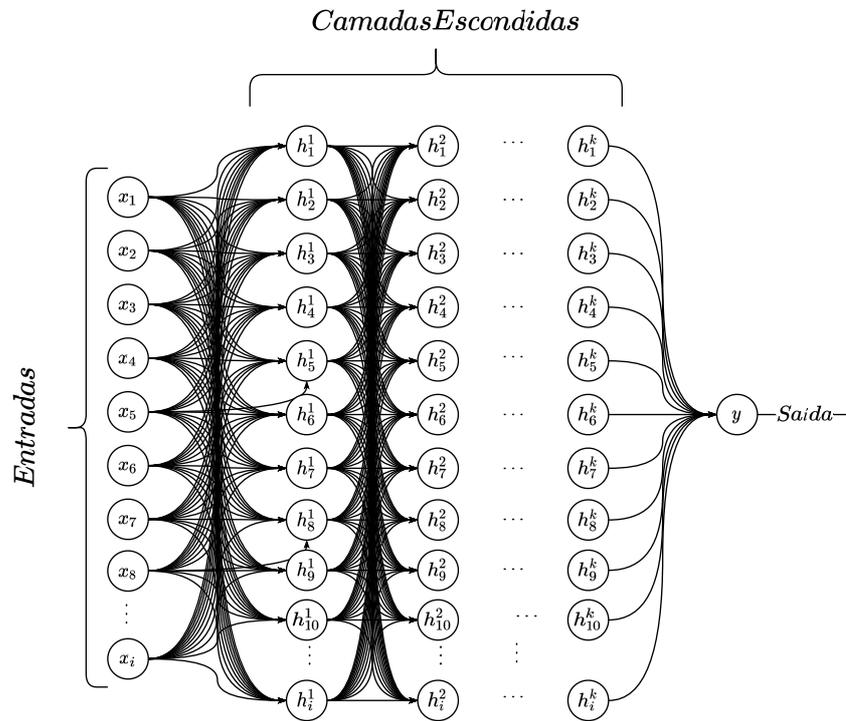
Fonte: Elaborada pelo autor (2023).

Mesmo com a pouca aderência dos pesquisadores à tentativa de solucionar problemas de separações não lineares, é possível destacar publicações que contribuíram para a ocorrência de novos avanços no tema. O algoritmo de *backpropagation* de Werbos e John (1974), a implementação da rede *Adaptative Resonance Theory* (ART) de Grossberg (1976) e os mapas auto-organizáveis de Kohonen (1982) formam algumas das contribuições mais importantes para o acontecimento desse novo entusiasmo no ramo de aprendizado de máquina.

Tão importante quanto as publicações revolucionárias que levaram ao novo período destacado, as melhores condições de poder computacional da década de 1980 e a construção de algoritmos de otimização fizeram com que o estudo de redes neurais voltasse a um novo ciclo de crescimento (SILVA; SPATTI; FLAUZINO, 2016). Contudo, a introdução das redes recorrentes de Hopfield (1982) e a utilização do algoritmo *backpropagation* em redes multicamadas com Rumelhart, Hinton e Williams (1986), são as produções científicas que merecem maior destaque.

O trabalho de Haykin (1998), apresentando o treinamento de redes em multicamadas por ativação não linear, é a essência do treinamento de redes de aprendizado profundo. Grandes avanços nos campos de processamento de linguagens, visão computacional, classificação e processamento de sinais foram concretizados graças a utilização do modelo *feedforward* aliado ao poder computacional das centrais de processamento gráfico (*Graphical Processing Unit* – GPU). A Figura 3 demonstra o conceito das redes de múltiplas camadas de Haykin (1998), base aplicável a qualquer outra derivação de RNA.

Figura 3 – Representação gráfica das redes de aprendizado com múltiplas camadas.



Fonte: Elaborada pelo autor (2023).

É perceptível a existência de uma longa história até chegar ao avanço corrente do SOTA de redes de aprendizado profundo. São décadas de estagnação e outras diversas revisões de conceitos, muitos ainda utilizados, contudo tendo mudanças nos métodos agregados e na sua empregabilidade. O processo evolutivo do domínio apresentado possibilita o entendimento das redes recorrentes. Embora o processo evolutivo destacado ajuda na compreensão do surgimento das redes de aprendizado, ainda é muito superficial o funcionamento de como ocorre o treinamento delas.

Portanto, é necessário obter a base de funcionamento e a compreensão da aplicabilidade das diferentes redes para a previsão de séries temporais sendo possível entender o formato das amostras requeridas e das inferências a serem realizadas. A percepção desse comportamento é vista nas seções subsequentes, porém, para que se possa definir a dimensionalidade requerida durante o treinamento dessas redes, é preciso observar a presença das redes *feedforwards* como base de funcionamento de todas elas. Assim, a Seção 3.1.1 faz compreender o algoritmo *backpropagation* em redes multicamadas (*Multi-Layer Perceptron* – MLP), base conceitual de persistência de neurônios e o processo iterativo de aprendizado.

### 3.1.1 *Backpropagation*

O modelo MLP de Haykin (1998) define que, para que haja um determinado número de saídas  $y_i$ , é necessário alimentar a rede utilizando um vetor representativo do estado de excitação. Como existem  $L$  camadas internas de neurônios, receptáculos dos padrões identificados, entre a saída e a entrada, a importância de cada um deles é ajustada ao longo do treinamento por meio da ponderação da representatividade de suas conexões com os demais neurônios.

O indicador de importância é construído à partir dos *mini-batches*, pequenos subconjuntos amostrais que servem para que a ativação de cada neurônio seja ponderada. A ativação conjunta desses neurônios resulta em uma inferência, a qual é comparada ao seu respectivo gabarito da iteração, um norteador das alterações a serem realizadas nos níveis de ativação de cada neurônio, respeitando a camada a qual pertence. Adotando  $\alpha_i^L$  como a saída do neurônio de uma camada  $L$  e  $y_i$  o resultado esperado, é pela função de custo de treinamento  $C_0$ , dada pela Equação (3.3), que o processo comparativo faz com que o resultado da função diminua, aperfeiçoando o modelo de inferência pelo decrescente erro entre a variável inferida e o resultado esperado.

$$C_0 = \sum_{i=0}^{n_L-1} (\alpha_i^L - y_i)^2. \quad (3.3)$$

Essa iteração não é restrita às camadas mais externas, responsáveis por inferir o valor final de todo o conjunto da rede com o esperado. As camadas mais internas também colaboram, porém em diferentes proporções, o que faz surgir a necessidade de atualizar, na melhor medida, pesos e viés (*bias*) de todas elas. Por conseguinte, uma nova entrada é gerada e o resultado do acionamento dos neurônios de todas as camadas, representativos da saída, é novamente comparado ao real, descrevendo um laço de ajuste do modelo em questão.

Essa visão macroscópica do funcionamento do treinamento permite analisar de maneira aprofundada uma única camada, porém generalizando para as demais. Adotando  $Z_i^L$  como o estado do neurônio (relação entre a saída do neurônio da camada anterior e a relação peso e viés do neurônio em análise), como verificado na Equação (3.4), define-se  $\alpha_i^L$  (a saída) de  $Z_i^L$  utilizando uma função de ativação, como estabelece Rosenblatt (1957). Neste caso, como demonstra a Equação (3.5), é utilizada a ativação sigmoide para definir a saída do neurônio.

$$Z_i^L = w_{ij}\alpha_i^{L-1}b_{ij}, \quad (3.4)$$

$$\alpha_i^L = \sigma(Z_i^L). \quad (3.5)$$

Como supracitado, o significado de treinar uma rede neural é realizar ajustes dos

níveis de ativação e estado dos neurônios. Como a correção é feita em todas as camadas, a regulação dos neurônios mais externas da rede é diferente das internas, ocasionando níveis de proporcionalidade entre os erros das camadas. Portanto é definido a razão entre custo e o peso dos neurônios para que as alterações estejam suficientemente adequadas para que a inferência se aproxime progressivamente do real a cada ajuste. Utilizando, exemplificadamente o que ocorre em somente um neurônio, a razão  $\partial C_0 / \partial w_{ik}^L$  pode ser obtida pelo produto das derivadas dos demais componentes, como demonstrado na Equação (3.6).

$$\frac{\partial C_0}{\partial w_{ik}^L} = \frac{\partial Z_j^L}{\partial w_{ik}^L} \frac{\partial \alpha_j^L}{\partial Z_j^L} \frac{\partial C_0}{\partial \alpha_j^L}. \quad (3.6)$$

Embora haja a descrição dos componentes das relações custo-peso, é só por meio da terceira derivada da Equação (3.6) que é possível destacar que a saída está dependente da força do sinal do neurônio anterior, como demonstrado pela Equação (3.7). Tal fato remete à afirmação supracitada de que o treinamento é uma ponderação da importância de neurônios interligados, traduzindo-se na observação de que neurônios menos importantes tendem a contribuir pouco para saídas mais acuráveis.

$$\frac{\partial Z^L}{\partial w^L} = \alpha^{L-1}. \quad (3.7)$$

As etapas de leitura da entrada, manifestação de uma saída, comparação com o real, ponderação dos pesos dos neurônios de cada camada e, novamente, a entrada de uma nova amostra é realizada repetidas vezes. O objetivo é realizar o ajuste fino dos pesos diminuindo o custo da função, achando a melhor proporção de pesos e *bias* para obter um menor erro comparativo entre saída e rótulo (*label*). O modelo, para avaliar o desempenho dos atributos construídos durante o treinamento e precaver o enviesamento das inferências, é avaliado por outro conjunto, chamado então de validador. Visto que os erros são ínfimos em ambos conjuntos, obtendo bons resultados nas inferências validadoras, é possível dizer que a rede neural está devidamente treinada para obtenção de resultados em meio a amostras nunca vistas.

Esse algoritmo de propagação de erro e correção de ativação fornece a base de funcionamento de qualquer tipo de rede de aprendizado profundo. Sendo chamada de redes *feedforward* ou MLP, sua arquitetura de múltiplas camadas pode ser vista nas redes recorrentes, assunto da Seção 3.2, e componente integrado para obtenção da inferência quando as arquiteturas mais complexas, como as *Transformers* são mencionadas. Mesmo que sua presença esteja implícita no funcionamento das demais, é por meio dos conceitos apresentados pelo algoritmo que é possível entender o processo iterativo de aperfeiçoamento de cada modelo.

## 3.2 REDES RECORRENTES

Embora seja possível levantar o funcionamento do processo de treinamento dos modelos pela propagação de erros, claramente exemplificadas pelas arquiteturas MLP, fica claro que as tarefas de classificação ou regressão são baseadas em conexão unitária entre os *perceptrons*. O fato de dois neurônios em uma mesma camada não estarem interligados remete ao fato de que o neurônio não tem em sua inferência a influência do próprio estado construído em etapas passadas. Analogamente, é possível dizer que o neurônio desse tipo de rede não possui o conceito de memória.

Tarefas de processamento de linguagem natural (*Neural Language Processing – NLP*) e regressão são alguns exemplos nos quais as amostras atuais são influenciadas pelas leituras passadas. Utilizando a primeira tarefa como exemplo, palavras isoladamente analisadas não possuem qualquer significado. É necessário obter um conjunto, guardar o comportamento do neurônio na interpretação de cada palavra dessa amostra e usá-lo para a construção da inferência seguinte. Esse processo é implementado, por exemplo, na análise de sentimento (USAMA et al., 2020), classificação da qualidade textual em textos em inglês (LIU; WANG, 2020) ou a sumarização textual (CHOPRA; AULI; RUSH, 2016). A regressão não é diferente, pois leituras passadas são importantes por indicarem sazonalidades e um padrão a ser seguido no horizonte temporal em destaque, como é visto em Mahesh, Afrouz e Areeckal (2022) ou em Nadeem e Arshad (2021).

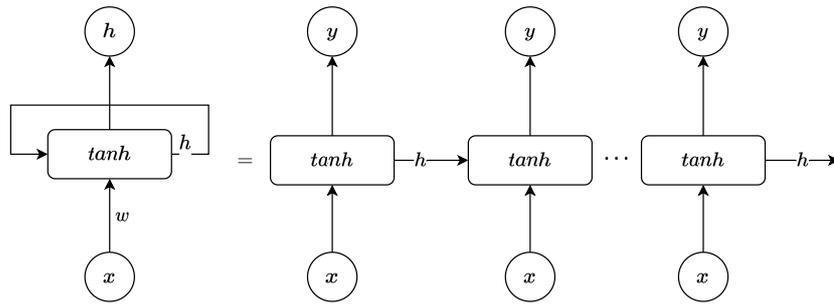
Em meio as possibilidades supracitadas, devido à implementação da persistência, como representado na Figura 4, as redes neurais recorrentes (*Recurrent Neural Network – RNN*) são amplamente utilizadas quando a intenção é abordar problemas que necessitam de conhecimentos passados para ajudar a inferir o resultado presente. Para tanto, planejando levantar o funcionamento base dos componentes que fazem parte do modelo, as próximas Subseções são direcionadas a explicação do avanço do diferentes tipos de arquiteturas voltados a recorrência. Partindo das *Vanillas RNN*, é possível verificar o problema de *Vanishing Gradient* e a solução trazida pela rede LSTM ou GRU. Por fim, é descrita a transformação de sequências em *tokens* pela rede *Transformers*, a qual também é amplamente utilizada em tarefas de NLP ou regressão.

### 3.2.1 *Vanilla RNN*

Para que possam serem descritos os componentes de uma rede recorrente básica (*Vanilla RNN*), adota-se  $h_t$  como o estado escondido para o período,  $f$  a função de ativação,  $x_t$  o vetor de entrada analisado e  $y_t$  a inferência feita pela rede. Visto graficamente cada uma dessas propriedades, a Figura 5 demonstra a interação de cada uma delas por meio da topologia de uma única camada.

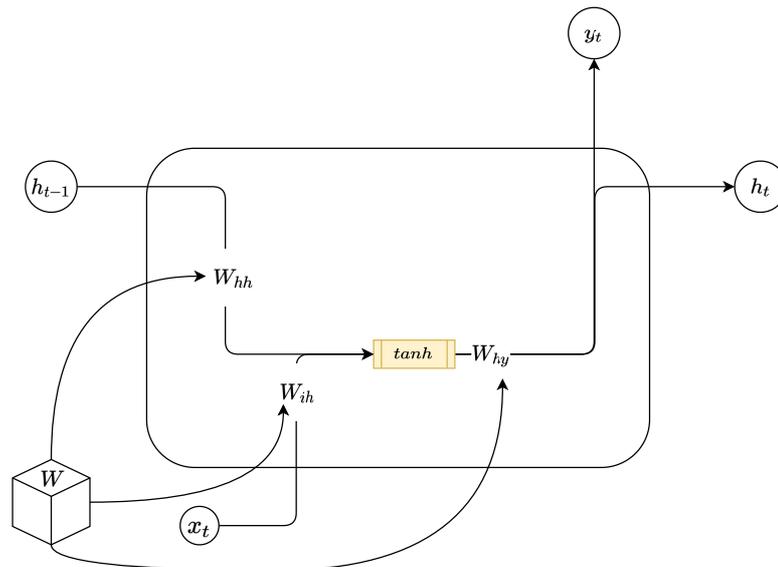
Detalhando um pouco mais a Figura 5, é possível verificar a existência de pesos  $W$  específicas para cada componente da topologia, mas havendo uma matriz global a qual são

Figura 4 – Modelo RNN.



Fonte: Adaptado de Olah (2015).

derivadas. Os *bias*  $b$ , implícitos aos pesos, são adicionados ao seu produto com a entrada,  $h_t$  e  $x_t$ . Para tanto, é perceptível uma complexidade no processo de *backpropagation* neste tipo de rede, pois a propagação dos níveis de ativação não está restrita aos seus componentes.

Figura 5 – Diagrama de componentes inseridos nas *Vanillas* RNNs.

Fonte: Adaptado de Olah (2015).

Tendo como foco o *hidden-state*, é perceptível pela Equação (3.8) que ela é fruto do estado anterior  $h_{t-1}$  e  $x_t$ , além de seus respectivos pesos e vieses. Neste ponto, é claro a constatação do atributo de memória declarado anteriormente. Aliás, as propriedades que geram esse componente é ativado por uma função não-linear, o que garante a separação dos dados deste tipo.

$$h_t = \tanh(h_{t-1}W_{hh} + x_tW_{ih} + b_{ih} + b_{hh}). \quad (3.8)$$

Produzida a propriedade de memória, ainda não foi definido a inferência  $y$  para ser

comparada com o real  $y'$ . Observando a topologia descrita pela Figura 5, a saída nada mais é que uma cópia do estado escondido  $h_t$  com o seus respectivos pesos. A Equação (3.9) demonstra como é obtido  $y_t$ .

$$y_t = W_{hy}h_t. \quad (3.9)$$

A partir de então, todo o processo de *backpropagation* é realizado, porém respeitando a particularidade da arquitetura de haver uma matriz global a ser herdada para os demais componentes. Entretanto, todo o desenvolvimento do treinamento continua o mesmo, ainda existindo a tarefa de diminuir o erro entre real e predito usando alterações de níveis de ativação.

Mesmo que a topologia adotada possa ser simples, até mesmo trivial, havendo semelhanças com as MLP, este tipo de arquitetura recorrente possui um alto nível de complexidade. Além do fato de haver uma matriz de pesos a ser herdada para os componentes que fazem parte da arquitetura, existe o fato de que a quantidade de camada é definida pelo tamanho das sequências a serem utilizadas como entrada. Existindo portanto essa equivalência, quanto maior for uma sequência, maior será a quantidade de camadas e o número de derivações de pesos a ser feita. Para tanto, o processo de propagação dos ajustes dos níveis de ativação do *backpropagation* será mais complexo.

Surge então o problema de *Vanishing Gradient*, visto por Bengio, Frasconi e Simard (1993). Os autores definem que, em meio à sequência muito longa, a rede fica tão grande que a proporção de ajuste dos *hidden-states* nas últimas camadas faz com as ativações das primeiras tornem as inferências maiores do que o real em etapas posteriores. O ato faz surgir uma reação em cadeia de erros em proporções, o que desencadeia a piora do treinamento e das validações.

Como solução, Bengio, Frasconi e Simard (1993) propuseram alterar a função de otimização utilizada. Além dessa mudança, ainda existiu a possibilidade de separar a disposição da camada de *hidden-state*, como feito por Elman (1990). Mesmo que haja o aumento da complexidade de tempo e espaço utilizando esta implementação e uma ínfima melhora dos resultados pela substituição da função otimizadora, a solução definitiva viria com a proposição da LSTM de Hochreiter e Schmidhuber (1997), a qual alterou todos componentes internos da camada, onde é melhor detalhada na Seção 3.2.2.

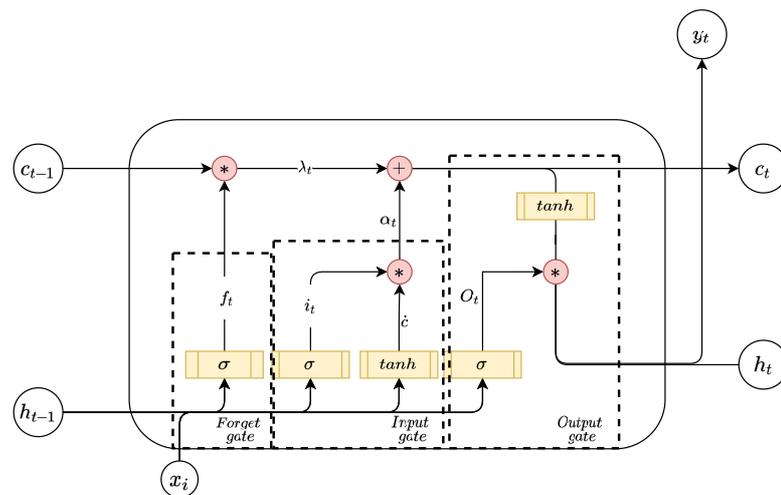
### 3.2.2 LSTM

Mesmo que a *Vanilla* RNN possa gerar o desencadeamento do treinamento em meio a longas sequências, bons resultados ainda podem surgir delas, ademais quando a implementação de Elman (1990) é utilizada com a otimização de Kingma e Ba (2014). Realçando ainda seu funcionamento, a arquitetura proposta de Hochreiter e Schmidhuber

(1997) direciona-se primariamente a solução do *Vanishing Gradient* em meio a problemas que requeiram longas entradas para a construção de suas inferências.

Hochreiter e Schmidhuber (1997), propondo uma arquitetura totalmente renovada, inserem somente dois componentes: (i) uma unidade de memória  $c_t$ , que serve como armazenamento de memória; e (ii) um processo decisório que determinará o que será ou não incorporado no primeiro item. A Figura 6 exemplifica uma camada da arquitetura da rede LSTM.

Figura 6 – Representação de uma camada da arquitetura LSTM.



Fonte: Adaptado de Olah (2015).

Empregando como suporte a Figura 6, todo o processo de construção das saída, a representação desse armazenamento e o estado  $h_t$  vêm das seguintes etapas decisórias:

1. Percepção dos importantes fragmentos inseridos no estado de armazenamento de memória. O responsável por tal tarefa é o elemento chamado *forget gate*. Para elaborar o resultado deste item, a sequência de entrada e o estado escondido anterior  $h_{t-1}$  geram um estado temporário  $f_t$ , como demonstrado na Equação (3.10). Este estado temporário  $f_t$  é agregado ao estado de armazenamento de memória anterior  $c_{t-1}$ , gerando  $\lambda_t$ , como destacado na Equação (3.11).  $\lambda_t$  representará os fragmentos do estado de memória anterior, porém com componentes representativos da entrada.

$$f_t = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + b_f), \quad (3.10)$$

$$\lambda_t = c_{t-1} \odot f_t. \quad (3.11)$$

2. Percepção de quais partes da nova entrada possuem atributos importantes para serem gravadas no novo estado de armazenamento ou, simplesmente, novo candidato  $c_t$ . Para tanto, os elementos *input gate*  $i_t$  e *candidate memory gate*  $\tilde{c}$  também receberam a nova sequência de entrada e o *hidden-state* anterior, para então serem agrupados,

formando  $\alpha_t$ . Por meio da Equações (3.12) e (3.13), respectivamente, figurativas dos itens *input gate* e *candidate memory gate*, é possível observar que os dois, separadamente, identificam componentes lineares e não-lineares da relação entrada-estado. A Equação (3.14), que demonstra o agrupamento  $\alpha_t$  desses dois itens, é uma matriz a ser inserida ao resultado do *forget gate*  $\lambda_t$ , o que gera o armazenamento de memória  $c_t$  a ser passado para a camada seguinte, colaborando para que haja a preservação tanto dos componentes das entradas quanto dos fragmentos já vistos como mais importantes. A Equação (3.15) demonstra que este estado  $c_t$  é simplesmente uma soma de matrizes.

$$i_t = \sigma(x_t W_{xt} + h_{t-1} W_{hi} + b_i), \quad (3.12)$$

$$\dot{c} = \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c), \quad (3.13)$$

$$\alpha_t = i_t \odot \dot{c}, \quad (3.14)$$

$$c_t = \lambda_t + \alpha_t. \quad (3.15)$$

3. Usando o mesmo método de confecção do *input gate*, a entrada  $x_t$  e o estado escondido anterior  $h_{t-1}$  são novamente utilizados para dar origem ao *output gate*. Esse elemento, junto com uma cópia da matriz do novo candidato  $c_t$  faz com que um novo *hidden-state* seja gerado e passe para as próximas camadas a relação existente entre os fragmentos destacados e a nova entrada analisada. As Equações (3.16) e (3.17) demonstram, respectivamente, a construção do *output gate* e do estado  $h_t$ .

$$O_t = \sigma(x_t W_{xh} + h_{t-1} W_{ho} + b_o), \quad (3.16)$$

$$h_t = \tanh(c_t) \odot O_t. \quad (3.17)$$

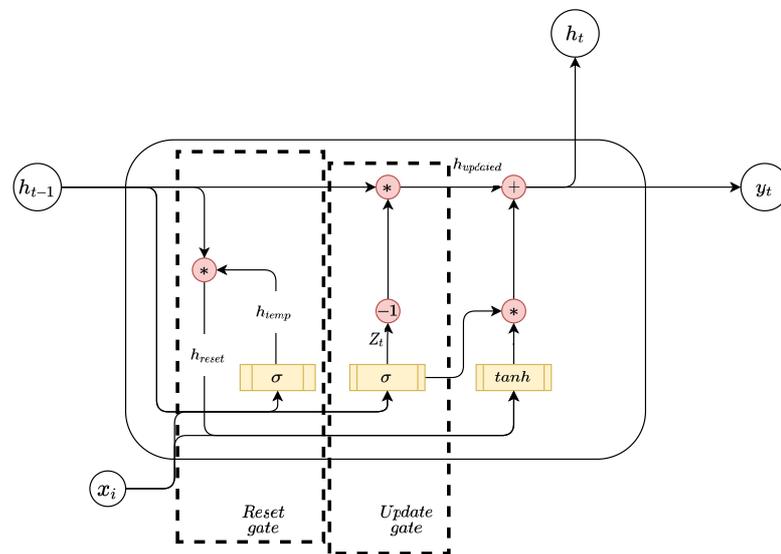
É possível perceber que, ao contrário do que acontece nas *Vanilla RNN*, a arquitetura ainda herda fragmentos de entradas e estados passadas. Após haver a preservação de itens de camadas anteriores, a saída  $y_t$  segue o mesmo roteiro de copiar o estado escondido  $h_t$  gerado e manipulá-lo com uma matriz de pesos  $W_{yh}$ . Portanto toda rotina de propagação de erros é novamente executada, sendo realizada a comparação entre inferência e resultado esperado, construção de novos indicativos de ativação de neurônio e inserção de uma nova amostra. O que garante a não ocorrência do desencadeamento do erro durante o treinamento é a ponderação existente do que será ou não passado adiante, conservando os atributos anteriores e definindo a propriedade de “memória de curto e longo prazo.”

A arquitetura de Hochreiter e Schmidhuber (1997) já é uma solução para o *Vanishing Gradient* de Bengio, Frasconi e Simard (1993), porém a complexidade de implementação da arquitetura é um pouco grande. Visando construir um modelo enxuto que tenha resultados tão satisfatórios quanto a LSTM, a GRU é a próxima etapa a ser estudada. Para tanto, os atributos que fazem parte desse modelo são descritos na Seção **3.2.3**.

### 3.2.3 GRU

Visando simplificar a solução de Hochreiter e Schmidhuber (1997), a arquitetura GRU também utiliza as mesmas propriedades de preservar os estados externos e internos das camadas das redes, correspondendo à analogia de memórias de curto e longo prazo. Ao invés de haver dois estados separados de armazenamento das memórias, a arquitetura GRU engloba-os no *hidden-state* passado entre as camadas. Por consequência, menos *gates* são necessários para que ocorra o filtro do que será passado adiante, sendo somente necessário o estado oculto  $h_t$ . Para auxiliar na explicação dos componentes inseridos na arquitetura, a Figura 7 demonstra o diagrama da topologia de uma camada da GRU.

Figura 7 – Representação de uma camada da arquitetura GRU.



Fonte: Adaptado de Olah (2015).

Partindo do procedimento do *reset gate*, o item define o quanto do estado anterior  $h_{t-1}$  será incorporado ao estado a ser construído. Sendo uma espécie de filtro, esse *gate* engloba o funcionamento do *input e forget gate* da LSTM. Perceba que, aqui há somente o descarte dos fragmentos inúteis do estado oculto anterior. A sua manipulação para que haja itens representativos da entrada vêm por meio do *update gate*.

Tendo como pauta *reset gate*, como demonstrado pela Equação (3.18), uma função sigmoide  $\sigma$  a entrada  $x_t$  será mesclada ao estado anterior  $h_{t-1}$  para gerar um estado totalmente novo, porém temporário  $h_{temp}$ . Este será mesclado ao estado anterior  $h_{t-1}$ , formando  $h_{reset}$ .

$$h_{reset} = \tanh[\sigma(x_t W_{th} + h_{t-1} W_{hh} + b_h) \odot h_{t-1}] + (x_t W_{th} + b_x). \quad (3.18)$$

Por enquanto, existe tão somente um novo *hidden-state* com atributos da entrada combinado ao estado anterior. Ainda é necessário atualizar  $h_{t-1}$  com a nova observação,

a fim de que seus atributos sejam incluídos a este antigo estado. O real *hidden-state* anterior será contrabalanceado por  $Z_t$ , produto da nova observação incorporada ao mesmo *hidden-state*, como demonstra a Equação (3.19), havendo, portanto, sua atualização  $h_{updated}$ .

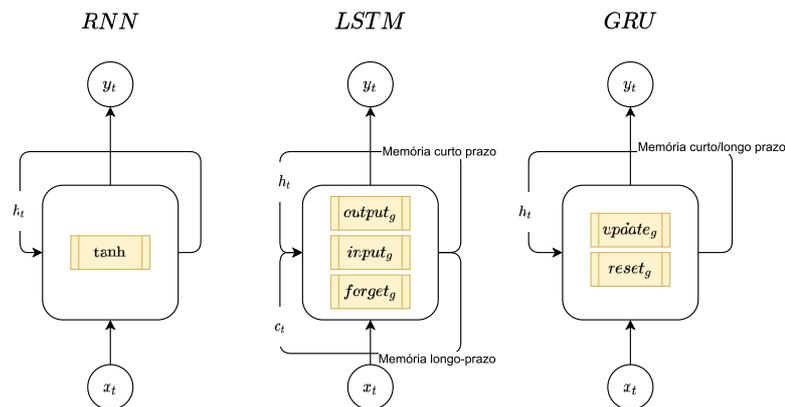
$$Z_t = \sigma(x_t W_{th} + h_{t-1} W_{hh} + bz). \quad (3.19)$$

O *hidden-state* candidato  $h_{reset}$ , um atualizado com as novas observações  $h_{updated}$  e um totalmente novo  $Z_t$ , representam o que da nova iteração pode ser utilizado de antigo, ser alterado e ser inserido como memória, respectivamente. O produto de tudo isso constrói o *hidden-state* a ser repassado para a próxima camada, como define a Equação (3.20).

$$h_t = (h_{t-1} \odot Z_t) + ([1 - Z_t] \odot h_{reset}). \quad (3.20)$$

Além de a saída, assim como as demais arquiteturas supracitadas, ser a próprio *hidden-state* ponderado por uma matriz de pesos própria, como já visto na Equação (3.9), fica perceptível que o funcionamento do modelo GRU possui os mesmos aspectos da rede LSTM, havendo o intuito de preservar fragmentos da entrada observadas em épocas passadas e repassar um estado totalmente novo. Ao contrário do que acontece na LSTM, a rede aqui descrita repassa o componente através de uma única entrada. A Figura 8 demonstra as principais diferenças existentes entre as arquiteturas e a função de cada.

Figura 8 – Diferenças principais nas arquiteturas de recorrência.



Fonte: Adaptado de Olah (2015).

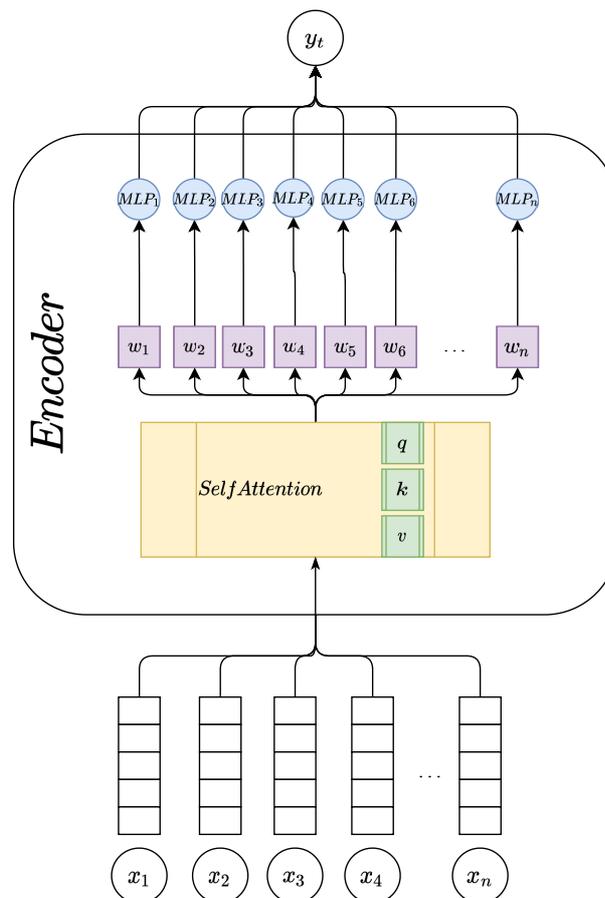
### 3.2.4 Transformers

Como visto, as redes recorrentes possuem uma tradicionalidade em soluções de problemas de séries temporais e NLP, devido a sua característica de utilizar estado de inferências passadas como parâmetro para as novas. Por meio dos funcionamentos descritos na Seção 3.2, o primeiro avanço foi por uma arquitetura que tinha seu treinamento

desencadeado por haver longas seqüências. Em seguida, a rede LSTM é apresentada como solução desse problema pela inclusão de *gates* que funcionassem como filtro e preservadores dos atributos constatados nas camadas anteriores, o estado escondido. Por fim, a GRU é publicada como uma enxuta arquitetura, reduzindo os *gates*, porém mantendo o desempenho da LSTM. Por meio desse processo evolutivo das redes recorrentes tradicionais, é interessante compará-las com as estruturas *Transformers* de Vaswani et al. (2017), verificando o funcionamento base e as similaridades que ela possui com as demais já descritas.

A proposta original de Vaswani et al. (2017) define um *encoder* composto por uma camada de *Self-Attention* ligada à uma arquitetura comum *feedforward*, isto é, uma rede MLP simples. A entrada é definida como múltiplos vetores transformados em seqüências representativas do quão fortemente estão interligados aos demais já lidos, formando um sistema chamado *Multihead Self-Attention*. Perceba que sua base de funcionamento assemelha-se às redes recorrentes pelo fato de que as camadas são criadas à partir dos vetores de entradas.

Figura 9 – Componentes presentes da arquitetura *Time-Series Transformers* (TST).



Fonte: Adaptado de Zerveas et al. (2021).

Percebendo que na proposta apresentada por Vaswani et al. (2017) havia a restrição

na aplicação em problemas de NLP (RAFFEL et al., 2020), Zerveas et al. (2021) apresentam sua aplicação para séries temporais multivariadas com uma arquitetura derivada de Vaswani et al. (2017). Todo o *kernel* teria como base de funcionamento o *Transformers* de Vaswani et al. (2017), porém retirando o *decoder* para transformar uma matriz em vetor, que tinha, por exemplo, a função de transformar a probabilidade de ocorrência em uma classe qualquer. As entradas funcionariam da mesma forma, diferentes sequências que seriam tratadas para construir a representatividade por *Multihead Self-Attention*. À partir de então, o funcionamento é o mesmo do *Transformers* original de Vaswani et al. (2017), tendo o vetor de entrada transformado em abstrações representativos da entrada, chamados de vetores  $q$ ,  $k$  e  $v$ . A Figura 9 exemplifica o *encoder* proposto.

Composto por três vetores, *query*  $q$ , *key*  $k$  e *value*  $v$ , abstrativos da entrada, o *Multihead Self-Attention* origina um vetor de pesos  $w$  representativo das sequências originais, ponderando e delineando a importância de cada entrada. Perceba que o *hidden-state* é interna à própria arquitetura e não aos que são passados adiante. A saída  $y$  é obtida através do mesmo cálculo visto na Seção 3.2.1, porém a matriz de pesos  $W_o$  é referente à saída da camada *feedforward* da rede, como demonstrado na Equação (3.21).

$$y = W_o \bar{Z}. \quad (3.21)$$

O funcionamento das arquiteturas *Transformers* de Zerveas et al. (2021) e Vaswani et al. (2017) lembra muito as redes recorrentes tradicionais. Pode-se verificar os mesmos atributos de construção de representatividade das sequências de entrada, uma forma similar de ponderar a saída da rede e, não menos importante, o mesmo intuito de persistir os fragmentos das entradas passadas.

Muito embora haja a percepção da forma de tratar as sequências e a constatação do treinamento dessas redes por amostras de tensores, ainda é muito importante verificar como a tarefa de previsão temporal se encaixa em ferramentas que foram originalmente elaboradas para tratar a classificação ou previsão de uma palavra. Portanto, a Seção 3.3 trata da manipulação dos dados tabulados das séries temporais, processando um conjunto tabulado para se tornarem tensores tridimensionais, representativos da quantidade de exemplos, das amostras e de suas propriedades vinculadas.

### 3.3 PREVISÃO DE SÉRIES TEMPORAIS

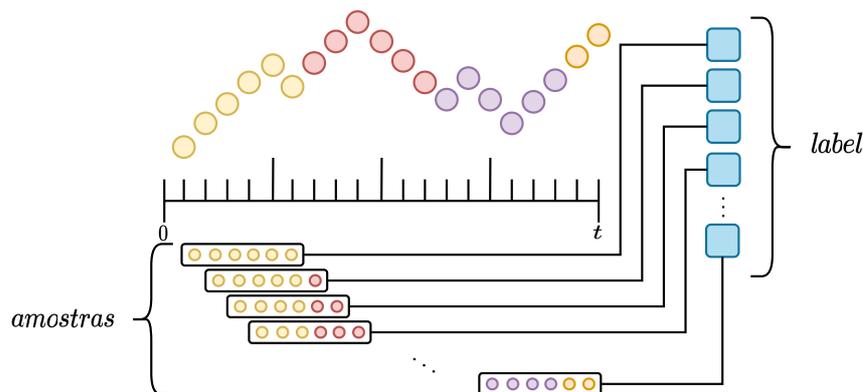
Conceituar uma série temporal é dizer que existe uma coleção de dados, de uma ou múltiplas propriedades, ordenadamente dispostas em relação a sua estampa de tempo (IENCO; INTERDONATO, 2023). Sem essa ordem bem definida, o conjunto de amostras não possui qualquer significado, não sendo possível identificar padrões, correlações ou conclusões acerca dos dados. Se estes são descritos temporalmente em uma única propriedade,

define-se que a série é univariada. Caso contrário, quando existem múltiplos atributos ligados ao objeto, as séries são multivariadas.

Tendências, estacionariedades, distribuições e outros componentes que definem estatisticamente essa coleção de dados podem advir da análise prévia dessas séries temporais. Porém, tratando-se de um modelo de aprendizado profundo, a relação que um grupo de amostras possui com as anteriores precisa estar explícita, pois o modelo inferirá sua próxima leitura baseada nas anteriores.

A relação que um pequeno conjunto de amostras possui com um objeto a ser inferido pode ser construída por meio do método de janelamento temporal, ou, simplesmente, *rolling window* (CHANG; CHEN; HUANG, 2022). Definindo um horizonte amostral e outro para o resultado, é possível subdividir todo o conjunto de dados em pequenas relações de exemplos, cada uma relacionada a um objeto de comparação a ser feito durante a inferência do modelo. A Figura 10 demonstra graficamente como o método funciona. No Apêndice se encontra o Algoritmo 10 implementado para a construção dessas janelas temporais.

Figura 10 – Esquema para construção do método *Rolling Window*.

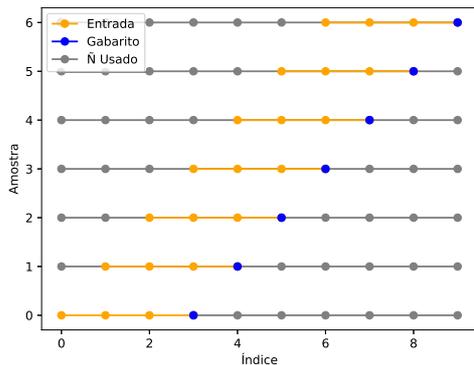


Fonte: Elaborada pelo autor (2023).

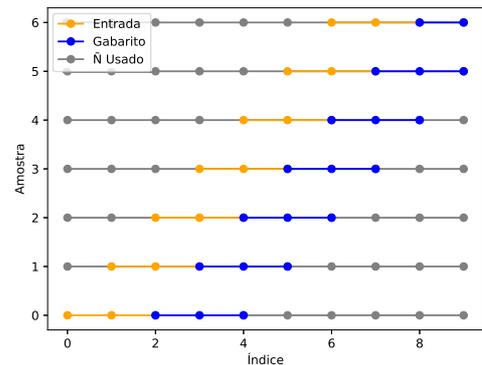
Como acentuado anteriormente, o método flexibiliza a construção tanto do horizonte de amostras quanto de inferência. Horizontes a serem comparados vão subdividir a tarefa de previsão em *single-step* e *multi-step*, respectivamente, para uma e múltiplas amostras à frente. A título de implementação, a diferença é simplesmente a presença de um valor escalar como saída no primeiro e um vetor para o segundo.

Como visto em An e Anh (2015), a inferência do modelo utilizando previsão *multi-step* gera erros maiores à medida que o horizonte a ser predito aumenta, pois o modelo infere múltiplos resultados utilizando a mesma sequência de amostras. Ademais, é possível utilizar o *single-step*, a qual possui uma maior acuracidade, para inferir múltiplas etapas usando o próprio método de *rolling window*. A Figura 11 exemplifica graficamente a principal diferença entre os passos de inferência.

Figura 11 – Janelamento de amostras dependente da inferência abordada.



(a) Previsão em passo único.



(b) Previsão em múltiplos passos.

Fonte: Elaborada pelo autor (2023).

Como o processo de *backpropagation* necessita de um conjunto de amostras e um resultado comparativo, a técnica de janelamento fornece, assim como um “valor resposta”, a possibilidade da construção de um horizonte amostral variável. O formato dos *tensores* em questão é bidimensional: sendo o tamanho da sequência destacada  $s$  como a primeira; e a quantidade de atributos vinculados ao consumo  $f$  como a segunda. Portanto, cada entrada do modelo possui um conjunto de dados bidimensionais vinculado a um *label*. A terceira dimensão desses *tensores*, é o *batch-size*  $b$ , a qual reparte essa cadeia de amostras bidimensionais em exemplos a serem consideradas como entradas pelo modelo:  $[b \cdot s \cdot f]$ . A Figura 12 ajuda na compreensão da transformação dos dados em sequência com múltiplas propriedades, a qual são acompanhadas por um *label* e, por fim, faz parte de um conjunto de treinamento.

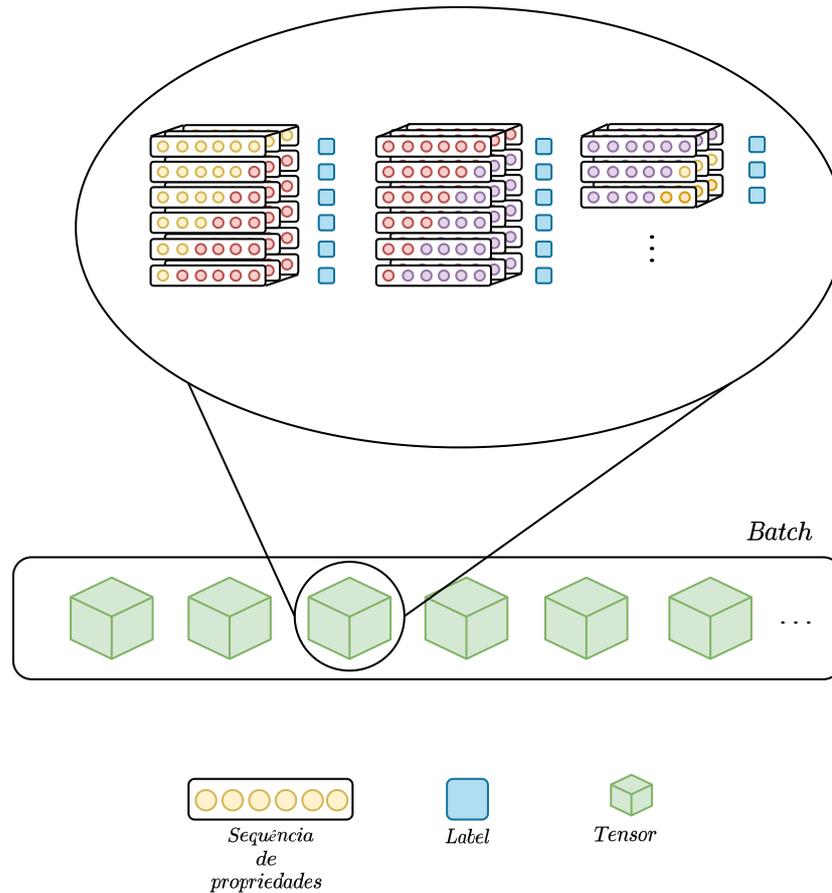
Esta técnica possibilita o uso dos modelos recorrentes supracitados, assim como outros que, muito embora não possuam um maior destaque, figuram como métodos para a literatura. A existência de múltiplas arquiteturas passíveis de uso, as quais possuem suas particularidades performáticas, motiva o uso da fusão tardia. Portanto, a Seção 3.4 é direcionada a construção do entendimento de como acontece a fusão dos modelos de inferência.

### 3.4 FUSÃO TARDIA

Todos os modelos fundamentados, assim como outros, têm seu característico funcionamento na inferência da série temporal, atingindo medidas distintas de erros de inferência. Visando agregar esses perfis de previsão, a fusão tardia utiliza cada modelo separadamente treinado para a construção de uma nova inferência, que englobe todos os atributos separadamente.

Diferentes métodos podem ser obtidos no SOTA para este fim. Partindo da

Figura 12 – Demonstração visual dos dados dispostos tridimensionalmente.



Fonte: Elaborada pelo autor (2023).

utilização de média simples, por exemplo, cada modelo tem sua colaboração igualitária. Muito embora existam os métodos *bagging* (BREIMAN, 1996a), *boosting* (SCHAPIRE, 1990), *stacking* (BREIMAN, 1996b) ou *gradient boosting* distribuído (XGBoost) (WANG et al., 2021), é possível aproveitar os modelos já treinados e usar o seu erro como referência de votação. Adotando, portanto, uma fusão tardia por votação, os modelos com indicadores maiores de erro colaboram menos para a inferência a ser gerada pelo *ensemble*.

Objetivando o baixo custo computacional, a distribuição de pesos para esta média ponderada precisa ser otimizada e possuir uma solução ótima. Usar a busca em formato *grid search*, assim como feito por Tian et al. (2020), não é interessante pois a complexidade aumenta de acordo com a quantidade de modelos a serem combinados. Assim como Taghavi e Sajedi (2014) e Brito et al. (2021), os quais também tiveram essa preocupação, o algoritmo de resfriamento simulado (*Simulated Annealing* – SA) é utilizado para a tarefa de encontrar o melhor conjunto de distribuição de pesos de votação. A Seção 3.4.1 fundamenta o funcionamento do algoritmo.

### 3.4.1 *Simulated Annealing*

Como mencionado, a intenção da fusão tardia é a construção de uma solução ótima que possua os melhores atributos de cada modelo separado. Como foi escolhido o método de fusão por votação, é necessário distribuir os melhores pesos para que esta “votação” pondere melhor aqueles com melhor desempenho, porém sendo a distribuição feita durante a validação de cada modelo. Meta-heurísticas são poderosas ferramentas para a construção dessa combinação otimizada.

Meta-heurística, termo introduzido por Glover (1977), consiste em uma classe de algoritmos heurísticos, normalmente aplicados a problemas de otimização combinatória, que visa a busca de soluções genéricas, tendo como propriedade a obtenção de uma solução próxima à ótima global (BLUM; ROLI, 2001). *Ant Colony Optimization* (ACO) é um exemplo dessa classe além do algoritmo de Resfriamento Simulado (*Simulated Annealing* – SA).

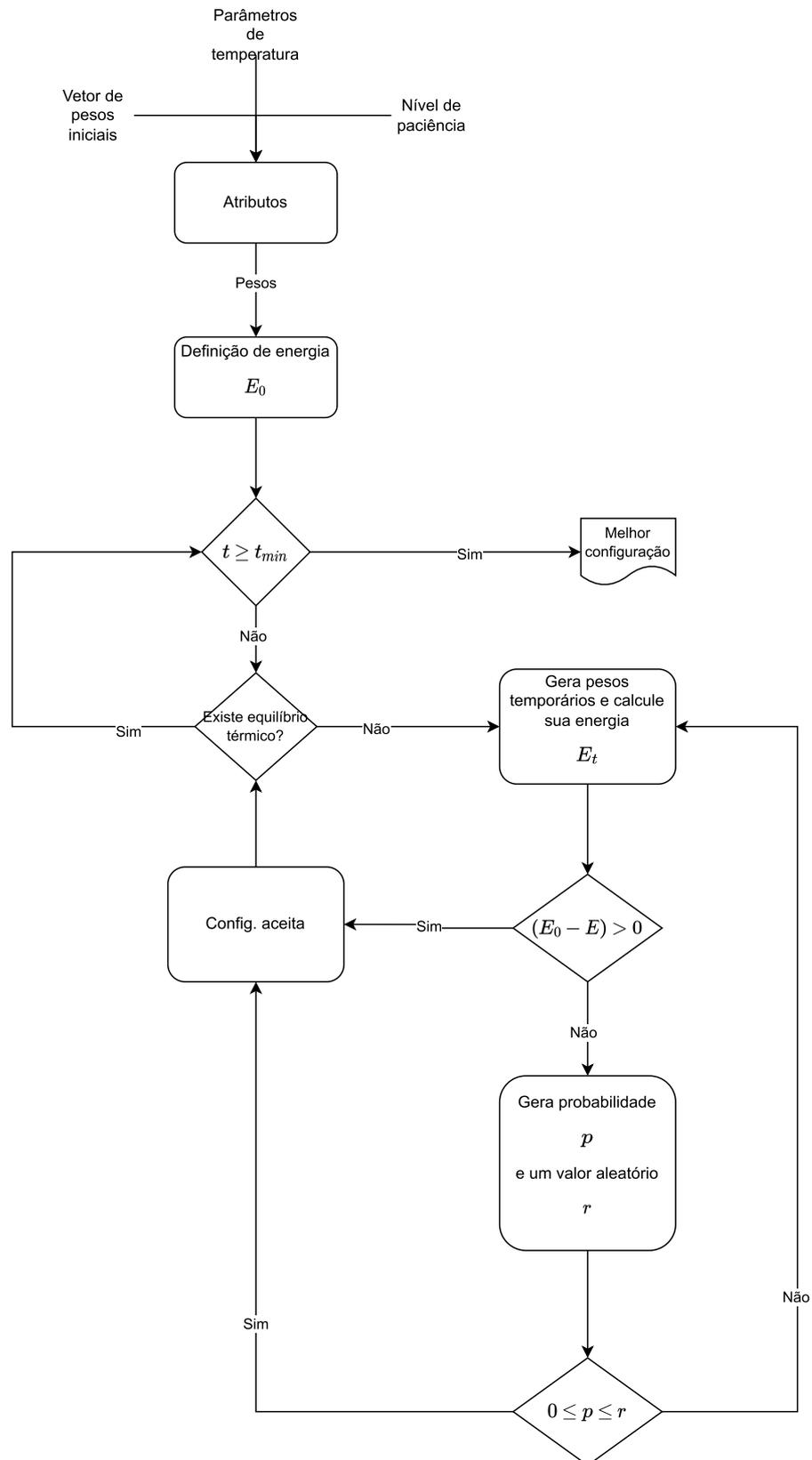
O SA, o algoritmo escolhido para fazer parte do método de escolha dos pesos da fusão, é um dos mais antigos algoritmos meta-heurísticos usados para o problema de combinação otimizada. O funcionamento é análogo à termodinâmica, pois a técnica busca soluções probabilísticas verificando o quão ótima ela é por meio de uma medida de energia.

Partindo de uma solução inicial, necessariamente composta por um vetor de pesos igualmente distribuídos entre as partes, e uma temperatura inicial  $T_0$ , a técnica consiste em realizar perturbações aleatórias no sistema e verificar, através de uma função objetivo, a energia resultante desse. O objetivo principal da técnica é encontrar a cristalização do sistema, ou seja, por meio de vários conjuntos de pesos, diminuir a função objetivo até que não haja mais alterações significativas. Em outras palavras, por meio de um conjunto inicial de pesos, igual à quantidade de modelos utilizados, o SA perturba a distribuição desses pesos até que a função objetiva indique o menor indicador de erro possível. A Figura 13 apresenta o fluxograma do processo de criação das configurações da votação.

Tendo como referência o trabalho de Villela e Carvalho (2006), os pesos iniciais, a taxa de decréscimo da temperatura e a função objetivo, são os parâmetros utilizados no processo de construção da votação, consistente em 5 passos:

1. Gerar uma perturbação no sistema pela escolha aleatória de duas posições e um valor flutuante aleatoriamente escolhido;
2. Elaborar um novo conjunto de pesos à partir dessa perturbação;
3. Verificar a energia gerada à partir do novo conjunto de pesos:
  - Se a diferença entre a energia atual e a anterior for maior que 0, então o novo conjunto de pesos será considerado como atual e a energia gerada por este será

Figura 13 – Fluxograma do processo do SA.



Fonte: Elaborada pelo autor (2023).

acoplada a verificação da entropia.

- Se a diferença for negativa, então será calculada a probabilidade  $p = e^{\frac{-\Delta E}{T*B}}$ , onde  $B$  é a constante de Boltzman. O valor gerado a partir desta probabilidade será comparado a um valor flutuante aleatoriamente gerado que, caso esteja no intervalo  $[0, p]$ , a nova configuração de pesos será aceita. Caso contrário, um novo processo se inicia.
4. Calcular a diferença entre a entropia atual e há  $i$  iterações passadas. Caso a diferença seja menor que o limite adotado por  $\varepsilon$ , considera-se então um equilíbrio térmico e a temperatura é diminuída pelo índice  $\alpha$ , iniciando então uma nova iteração com uma nova temperatura. Caso contrário, um novo peso é gerado utilizando a mesma temperatura.
  5. O processo somente finaliza quando a temperatura chegar próxima a zero, momento dito em que houve a “cristalização” do sistema. A configuração de pesos gerado nesse momento é que será adotada, pois são os valores que geraram o índice mais baixo da função objetivo.

Agora, direcionando o enfoque para a coleção de características inseridas nos conjuntos utilizados, torna-se interessante analisá-las e compreender a importância que cada uma possui para a inferência e construção de uma coleção de treinamento mais enxuto. Deste modo, a Seção 3.5 traz alguns métodos para a análise das características, antes e depois de sua utilização nos modelos de previsão.

### 3.5 ANÁLISE DE CARACTERÍSTICAS

Realizar a previsão de uma série multivariada requer a utilização de um certo número de propriedades correlacionadas ao objeto de inferência. Porém, a construção exacerbada dessas variáveis durante a etapa de engenharia de características pode fazer com que o modelo piore ao invés de melhorar (BELLMAN, 2015).

Portanto, como as séries possuem múltiplas características, torna-se interessante interpretá-las e verificar seu nível de importância, o que abre oportunidade de comparar se houve a melhora da acuracidade do modelo com a redução das variáveis, principalmente, analisando em que circunstâncias ocorreu tal fato. Duas podem ser as abordagens dessa análise: antes do treinamento, com a utilização da análise de componentes principais (*Principal Component Analysis* – PCA), e depois dela, usando a ferramenta chamada *Shapley Additive Explanations* (SHAP), ou mesmo, *SHAP Values*. Esta segunda etapa é muito interessante, pois, além de ser um método novo para a descrição do comportamento do modelo, define a força das características para uma única inferência e o nível de suas importâncias para todo o conjunto de teste. Como os modelos são instanciados na época

com o menor erro médio quadrático, a utilização deste conjunto é essencial para não ocorrer enviesamento das previsões. Assim, as próximas seções tratam de forma detalhada essas duas ferramentas.

### 3.5.1 Análise de componentes principais

Muitos trabalhos trazem a previsão de série temporais utilizando as técnicas de PCA. Elattar (2013), por exemplo, defende a construção de redes neurais compostas pela redução de características, mas encapsuladas em seu *kernel*. Zhang et al. (2020), por outro lado, direcionam a abordagem ao pré-processamento da base de dados, filtrando aquelas mais importantes à descrição do consumo de energia. Fato esse importante, pois é uma forma construtiva de seleção de características (*feature selection*), onde o modelo, independentemente de qual a arquitetura adotada, recebe os mesmos dados vistos como mais importantes para a descrição do *dataset* e da variável objeto.

Sabendo que o PCA é utilizado como uma etapa de seleção de características estranhas ao processo de engenharia de características, é possível verificar a forma como são obtido esses componentes principais. Toda a base de cálculo é derivada do *Singular-Value Decomposition* (SVD), extraindo autovetores e autovalores, ou os então melhores referenciados, *eigenvectors* e *eigenvalues* do *dataset*, agora representado por uma matriz de valores normalizados. Assim sendo, dada uma matriz  $A_{ij}$ , é possível obter uma matriz descritiva de sua covariância, demonstrando a pontuação do quão cada característica é predita em relação às demais. Valores positivos entre duas variáveis indicam uma dependência unidirecional entre elas, ou seja, se uma delas crescer e as outras relacionadas à ela também irão. O oposto acontece com valores negativos, ou seja, a dependência é inversa, o acréscimo de uma remete ao decréscimo das demais. Já nulas remetem à falta de relação previsível entre elas.

Essa matriz de covariância é necessária para a construção do *eigenvector*, vetor que descreve a direção descritiva da matriz  $A_{ij}$  depois de linearmente transformada, e *eigenvalue*, medida escalar desse vetor. Dessa forma, sabendo que  $Z_{ij}$  é a matriz de pontuações gerada pela covariação da matriz  $A_{ij}$ , *eigenvalue* é tido através de Equação (3.22), sabendo que  $\lambda$  é o *eigenvalue* e  $I$  a matriz identidade.

$$\det(Z_{ij} - \lambda I) = 0. \quad (3.22)$$

Através do *eigenvalue* é possível definir os *eigenvectors*, sabendo que é um sistema de equações lineares, o que possibilita a solução pela eliminação de Gauss. Sabendo que esses *eigenvectors* são vetores que apontam a direção dos dados em diferentes novas dimensões sem perder sua descritividade, é necessário escolher um conjunto de direção dependente do número de componentes representativos. Uma forma de tomar essa decisão é ranquear esses *eigenvectors* pelos seus respectivos *eigenvalues*, que melhor representam

o *dataset* nesse novo sub-espço apontado. Reduzindo então, por exemplo, um espaço tridimensional em bidimensional, é combinado os *eigenvectors* dos dois maiores valores de *eigenvalues*, gerando uma nova matriz  $W_{23}$ . Essa matriz  $W_{ij}$ , por fim é combinada à matriz  $A_{ij}$  original, exemplificada pela Equação (3.23), para obtenção dos reais valores que compõem os melhores componentes do banco de dados.

$$y = A_{ij}W_{ij}^{-1}. \quad (3.23)$$

### 3.5.2 Interpretabilidade por SHAP Values

Embora haja a explicação do funcionamento das redes neurais, como visto na Seção 3.2, ainda é custoso saber qual a real influência de cada característica para uma determinada previsão. Tais modelos são muitas vezes referenciados como “caixas-pretas”, os quais recebem um conjunto de entrada para inferir um determinado valor de saída. A avaliação feita entre a relação saída e entrada do sistema somente diz ao analista o erro e a aderência do modelo aos dados utilizados para o seu treinamento, porém não existe de maneira explícita de saber qual o real comportamento dos atributos envolvidos para que ocorrência da determinada inferência.

Medidas de erros e a *feature selection* por PCA ajudam no entendimento e redução das *features*, mas ainda não é possível analisar o comportamento isolado dessas variáveis. Por meio da interpretação por SHAP Values, arquiteturas novas são criadas, previsões são desmistificadas e influências dos atributos são destacados, o que abre para a análise para o estudo do comportamento de um modelo isolado para o conjunto de dados colocado em pauta.

É muito comum haver uma confusão entre os métodos de interpretação e avaliação dos modelos. Baptista, Goebel e Henriques (2022) definem que interpretar diz respeito a mapear localmente o comportamento de cada característica influente no modelo para uma inferência isolada. Diferentemente de avaliar, que somente ajuda a entender a condução do treinamento, verificando *overfitting* e outros tipos de desencadeamentos de erros.

A interpretação do funcionamento de modelos está no ramo de pesquisa de *Explainable Artificial Intelligence* (XAI), e são muitos métodos disponíveis para a execução dessa tarefa (BAPTISTA; GOEBEL; HENRIQUES, 2022). *Local Interpretable Model-agnostic Explanations* (LIME) (RIBEIRO; SINGH; GUESTRIN, 2016) e *Captum* (KOKHLIKYAN et al., 2020) são algumas das ferramentas para conseguir criar uma metodologia de interpretação das previsões feitas em domínios de classificação e regressão.

Muitos exemplos de utilização desses métodos estão disponíveis na literatura. Zeng et al. (2021), por exemplo, utilizam técnicas de interpretabilidade de modelos de aprendizado profundo para melhorar e explicar a previsão de complicações pós-cirúrgicas por meio de séries temporais de pressão sanguínea e outras características relacionadas à

saúde do paciente.

Baptista, Goebel e Henriques (2022) e Zeng et al. (2021) utilizam o SHAP *Values* (LUNDBERG; LEE, 2017), métrica que ajuda na obtenção da importância de cada característica para o momento da inferência. Desenvolvida especificamente para os complexos modelos de aprendizado profundo, SHAP *Values* integra tanto os métodos de Ribeiro, Singh e Guestrin (2016) quanto os algoritmos vistos na biblioteca *Captum* (KOKHLIKYAN et al., 2020), em uma só ferramenta que disponibiliza por meio de sua métrica a atribuição do peso de cada *feature* para as inferências isoladas, respeitando a especificidade de cada modelo analisado.

## 4 ABORDAGEM PROPOSTA

A etapa de fundamentação traz importantes aspectos, bases de funcionamento e outras importantes ferramentas para serem utilizadas. Essa contextualização ajuda a observar empiricamente o comportamento de cada modelo na previsão de séries temporais. Intencionando a comparação das características de treinamento e inferência, é necessário o desenvolvimento de um método agregativo de todo o embasamento teórico feito até então.

Portanto, para a construção de um comparativo, o método de análise necessita descrever: (i) o pré-processamento das bases escolhidas; (ii) o processo de escolha dos atributos mais importantes por PCA; (iii) as observações e conclusões tidas durante a etapa de treinamento, validação e previsão dos modelos usando o conjunto total e os apartados na etapa passada; (iv) a construção do *ensemble* e comparação de suas inferências com os resultados já obtidos; (v) a força que cada variável possui para o modelo durante sua previsão; (vi) e, por fim, (vii) o treinamento, validação e previsão dos modelos utilizando aquelas características interpretadas como as mais influentes por SHAP *Values*. Para melhor entender o método comparativo do comportamento dos modelos, a Figura 14 demonstra graficamente os passos tomados.

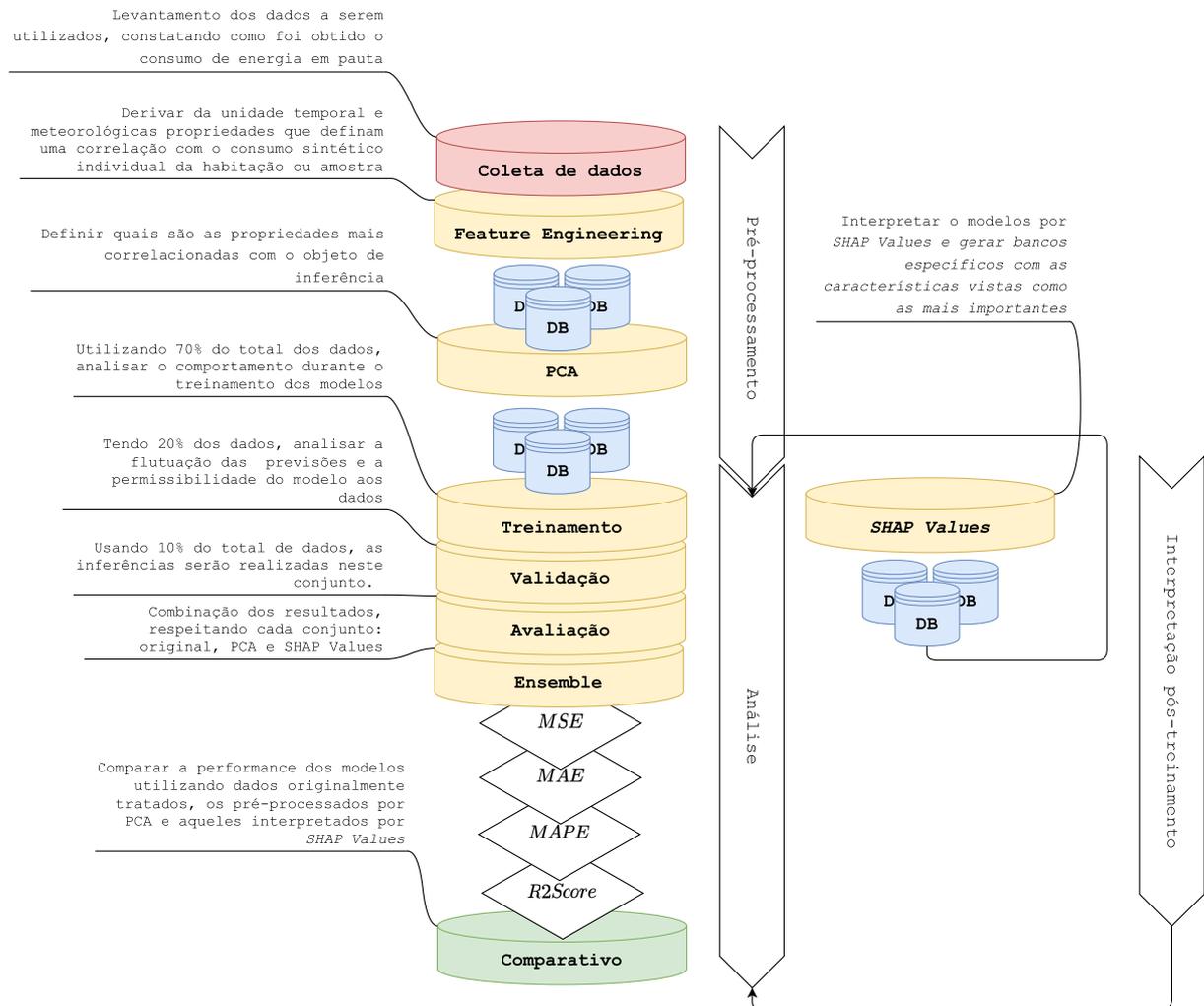
O método empregado pelo presente documento é bem similar ao proposto por Sepulveda et al. (2021), tendo o diferencial de possibilitar a interpretação dos modelos e a observação empírica do comportamento durante o treinamento e inferência destes. A fim de detalhar melhor cada um desses passos, o método pode ser melhor descrito nas Seções 4.1 e 4.2.

### 4.1 ETAPA DE PRÉ-PROCESSAMENTO

Antes que cada modelo possa ser devidamente treinado, é de extrema importância realizar uma análise superficial na base de dados a ser utilizada. Esta primeira etapa permite constatar sua correta ordenação e verificar a propriedade de uma série multivariada. Todos esses fatores possibilitarão construir indicadores descritivos da distribuição amostral dos dados.

Sazonalidades e *outliers* são passíveis de serem encontrados por meio dessa etapa. É importante destacar que os *datasets* compostos por amostras resumidas a equipamentos, a disponibilização desse conjunto de dados pode ser comum a diversas situações amostrais, portanto, existe a possibilidade da ocorrência de medições não inseridas a um ou mais conjuntos. Em outras palavras, bases que descrevam consumo por equipamento, sua generalização para outras fontes de leituras pode gerar a falta do valor objeto. Por consequência, deve-se atentar ao fato de que podem existir leituras nulas. Tais valores nulos ou discrepantes impactam tanto no treinamento para a ocorrência da futura inferência quanto na representação dos componentes por PCA, gerando erros na percepção dos pesos

Figura 14 – Fluxo da metodologia empregada para análise do comportamento dos modelos.



Fonte: Elaborada pelo autor (2023).

das características.

Como cada consumidor possui um perfil de consumo, é preciso agregar todos os seus componentes descritivos dessa demanda, desassociando-a dos demais itens presentes na base. Em outras palavras, leituras correspondentes a uma mesma medida são somadas como, por exemplo, potência demanda e tensão.

Embora haja a diminuição do número de atributos presentes na coleção de dados, essas características apartadas não são significativas pela falta de valor descritivo, o que poderia ocasionar a falha do treinamento de qualquer modelo a ser utilizado. A etapa de *feature engineering* se estende a expansão dos atributos de estampa de tempo, gerando: valores unitários de data (dia, mês, anos, *Unix epoch*, contagem da semana e outras derivadas) e componentes senoidais e cossenoidais das estampas do tempo, como demonstrado por Strielkowski (2020). Porém, não resumido aos atributos de unidade

temporal, caso a demanda de energia processada figure a potência eficaz, suas derivações devem ser inclusas à esta etapa.

A engenharia de característica supracitada fornece uma quantidade significativa de características para o consumo e o sequenciamento das leituras, o qual possui sua integridade. Porém, é preciso obter dados meteorológicos da mesma região das leituras de consumo, pois, de acordo com Hong e Wang (2022), a meteorologia impacta na previsão de demanda visto que, em períodos demasiadamente quentes ou frios, o padrão pré-estabelecido do consumo de energia tende a mudar, fato também explicado por Dryar (1944).

Dessa maneira, a etapa de *feature engineering* prolonga-se à inclusão de características meteorológicas, tais como: a cobertura das nuvens, precipitação, hora do crepúsculo entre outras. Importante destacar que, caso não seja disponibilizado pela fonte original leituras desses atributos, dados externos devem ser obtidos na mesma resolução, respeitando a localidade da medição do objeto de previsão.

Fim do processo de *feature engineering*, o conjunto dos dados descreve: (i) fatores de unidade temporal, (ii) o estado meteorológico e (iii) os conjuntos relacionados às medidas elétricas demandadas do circuito analisado. Cada conjunto respeita uma resolução, uma localidade e um perfil de demanda energética. Portanto, são amostras agrupadas por um determinado atributo, como por exemplo, região, tipo de residência ou identificador de coleta, que fornecem sequencialmente o objeto de inferência com diversos outros atributos relacionados.

A base de dados originalmente compilado apresenta um número amplo de características. Objetivando comparar e constatar a existência da maldição da dimensionalidade ocasionada pelo aumento substancial das *features*, gerando um possível *overfitting* (BELLMAN, 2015), o PCA é aplicado para construir outras bases com os atributos mais importantes, sendo mantida a individualidade das leituras e o objeto de inferência.

Esta análise de componentes principais traz hierarquicamente as *features* com maior descritividade do *dataset* (HE et al., 2020). Usando deste procedimento, tanto a dimensionalidade é comparada quanto o tempo e a melhora no processo de treinamento usando todos os dados compilados e os apartados.

Independente do conjunto abordado, os dados são normalizados, conforme dita a técnica de descoberta de conhecimento da base (*Knowledge Discovery in Database – KDD*), e divididos em subconjuntos de treinamento, validação e avaliação, estando nas respectivas proporções: 70%, 20% e 10% do total de dados. Cada um deles é transformado em sequências de janela temporal utilizando o *rolling window*. Abordando uma previsão *single-step*, a janela de amostras escolhida foi de 60 leituras para uma inferência a frente. A escolha dessa janela foi subjetiva a fim de manter um comparativo, portanto, um estudo de caso em que o número de amostras inseridos na entrada não foi incluso no estudo.

A etapa de pré-processamento da metodologia é finalizada com a obtenção dos

subconjuntos em formato de sequência temporal normalizados pelo valor máximo, estado no intervalo de  $[-1, 1]$ . A próxima subseção descreve o procedimento a ser adotado no pós-treinamento do modelo, referindo-se a análise das métricas, fusão tardia e interpretação das características.

## 4.2 ETAPA DE PÓS-TREINAMENTO

Com as características originalmente processadas e apartadas por PCA, subdivididas nos subconjuntos de manipulação, todos os modelos poderão ser treinados e analisados em estudos de caso. Direcionando-se a princípio ao comportamento do treinamento e validação dos dados, a adequação de cada arquitetura é constatada e um indicativo de seu aprendizado é visto. Portanto, em um primeiro momento, todos os modelos são treinado utilizando uma sequência de dados que possua todas as características originalmente pré-processadas. Com isso, a melhora por época da inferência do modelo poderá ser vista comparando os erros médio quadrático de treinamento e validação de cada.

Ao utilizar essa análise em diferentes resoluções, mas com um intervalo de amostra bem definido, é possível verificar o impacto que a quantidade de amostra possui no processo de melhora do modelo. Tendo sido concluído o treinamento dos modelos e percebido o comportamento durante a validação de sua inferência, é possível realizar a previsão do conjunto amostral de 10% direcionado à avaliação. Sendo, portanto, a avaliação do treinamento dos modelos sem amostras usadas para as duas etapas anteriores, é destacável o comportamento dos modelos em identificar mudanças abruptas de demanda, percebendo seu nível de inércia para a inferência do objeto.

A totalidade do subconjunto de avaliação é avaliada pelas métricas de *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE) e *R2 Score*, respectivamente descritas pelas Equações (4.1), (4.2), (4.3) e (4.4). Essas métricas, principais referências avaliativas na tarefa de regressão, auxiliam na percepção da acurácia dos modelos, visto em diferentes resoluções e conjuntos de dados.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (4.1)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (4.2)$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (4.3)$$

$$R^2 = 1 - \frac{RNN}{TSS}. \quad (4.4)$$

Paralelo à ocorrência dos treinamentos e avaliações dos modelos, é possível interpretá-los por meio de SHAP *Values*. A ferramenta traz o impacto das características para a

inferência do modelo treinado em duas situações: uma única inferência e o impacto médio referente à análise de todo o conjunto de teste. Esse segundo produto permite realizar uma nova separação dos dados e treiná-los novamente para a constatação de um melhor desempenho. É importante ressaltar que o produto é tido em cima do conjunto de dados originalmente compilados, visto que o comparativo é o desempenho dos modelos treinados usando o conjunto total de propriedades, as resumidas por PCA e, por fim, as simplificadas por SHAP *Values*. Portanto, este último conjunto não é gerado à partir do PCA.

Novamente, o comportamento do treinamento, validação, inferência de uma pequena amostra e a avaliação total dos modelos está em pauta. Porém, não só o impacto de quantidade de amostras por resolução está descrito, mas a quantidade e a importância dos atributos, sendo perceptível a melhora ou a piora da acurácia nesses dois aspectos.

O último passo da etapa de pós-treinamento é a tentativa da melhora da acurácia das inferências por um *ensemble*, feito por votação e utilizando os modelos com os pesos das melhores épocas de treinamento. Para tanto, a escolha da ponderação do comitê foi otimizado pelo algoritmo SA no conjunto de validação. Para que seja observado o comportamento do novo componente comparativo, as etapas de análise de inferência e das métricas comparativas são levantadas, percebendo tanto os modelos mais impactantes quanto a sua possível melhora.

O processo de *ensemble* finaliza o método e o resultado é um comparativo de: (i) todas as características previamente compiladas; (ii) variáveis apartadas por PCA; (iii) propriedades importantes para os modelos isolados, utilizando SHAP *Values*; (iv) o impacto da resolução em um intervalo definido de coleta de amostras; (v) e a possível melhora da inserção do *ensemble*. Usando deste método, pretende-se obter uma análise ampla dos modelos e dos dados utilizados, verificando o pressuposto do impacto do número de atributos e amostras. O Capítulo 5 traz detalhadamente a descrição da origem dos dados brutos, a implementação e os resultados da aplicação do método proposto.

## 5 EXPERIMENTOS E RESULTADOS

### 5.1 BASES DE DADOS

O comparativo do comportamento dos modelos de inferência está baseada no pré-processamento e análise de duas bases de dados: *Houly Usage Energy Database* (HUE) (MAKONIN, 2018) e *Pecan Street* (STREET, 2022). A manipulação do método proposto utilizando essas bases traz a possibilidade de analisar o comportamento de cada modelo em diferentes resoluções de tempo e montante de amostras. Além de tudo, é possível constatar o conjunto que melhor representa toda a coleção de dados utilizando o PCA, sendo perceptível a particularidade de cada modelo na diminuição das propriedades para aquelas mais representativas de toda a coleção. Porém, antes que seja possível destacar os resultados detalhadamente, é necessário obter um panorama da origem dos dados envolvidos.

A base de dados da HUE, publicada pela *Harvard Dataverse*, possui leituras de 28 residências da Colúmbia Britânica, estado do Canadá. Existem 9 diferentes classes de residências, que podem ser listadas como: (i) residências características da década de 1940, (ii) do tipo bangalô, (iii) construções características das décadas 1965 à 1989, (iv) de aspecto moderno, (v) duplex, (vi) triplex, (vii) *townhouses* (duas casas que compartilham uma única parede), (viii) apartamentos e (ix) *laneway* (pequenas casas construídas nos fundos da casa principal). Cada uma dessas classes de residência possui sua demanda característica, sendo fatores definidores o tipo de construção, finalidade e os equipamentos inseridos para satisfazer as necessidades diárias de seus habitantes. Tendo sido disponibilizado um dicionário com seus identificadores e suas respectivas classes, é possível constatar a completude dos dados e o início da amostragem, que podem variar entre 2012 à 2018, ou seja, nesta base há diferentes intervalos de coleta de amostras. Embora haja a desvantagem de estar somente disponível na resolução horária, essas informações sugerem a construção de um grande conjunto de dados a ser manipulado, para então ser parametrizado ao modelo de inferência.

Ainda tendo como descritivo a base da HUE, os dados de demanda originalmente estarão inseridos no domínio univariado, pois somente há o dado de demanda vinculado a medida de tempo. Mesmo que não haja qualquer outra característica que possa ser correlacionada ao consumo, com exceção da própria leitura temporal, felizmente a publicadora disponibiliza datas festivas e a medição meteorológica da região canadense, o que retira a necessidade de inserção de dados externos à editora. Os dados em questão abrem a possibilidade de inseri-los à coleção original e expandir a unidade temporal, o que converte o domínio da série temporal para multivariado.

Mesmo que existam fatores que contribuam à tratativa da demanda em meio a um conjunto multivariado, ainda é necessário expandir a estampa de tempo. A construção

dessas novas características baseadas na unidade temporal faz surgir uma melhor correlação entre o objeto de inferência e o tempo propriamente dito. Porém, a base de dados somente disponibiliza coleta de amostras em granularidade horária.

Intencionando obter um estudo de caso de acurácia voltado às resoluções de tempo, o *dataset* disponibilizado pela *Pecan Street* foi a segunda fonte de dados escolhida para a condução dos experimentos. A base consiste em uma versão estudantil e outra comercial, sendo a quantidade de observações e a variabilidade das propriedades relacionadas o diferencial. A construção da bancada de teste tem como suporte a versão estudantil, onde há o detalhamento do consumo de energia de 25 residências das regiões americanas de Austin, capital do Texas, Nova Iorque e Califórnia, todas disponibilizadas somente no intervalo de 1 ano (janeiro de 2018 à janeiro de 2019). A própria editora descreve a completude dos dados de cada uma das regiões disponibilizadas, sendo a coleção de Austin o mais atrativo em relação às demais regiões por haver 99% dos dados completos em todas as resoluções (1 segundo, 1 minuto e 15 minutos). Além das características de leitura, o conjunto de dados na versão utilizada está baseada no tipo de residência de família única. Por outro lado, sua versão comercial possui outros 4 tipos: (i) apartamentos, (ii) casas urbanas, (iii) casas à venda e (iv) *motorhomes*.

Embora não haja uma precisa propriedade descritiva da leitura de demanda temporal na base da *Pecan Street*, existem características que, isoladamente, oferecem a carga de equipamentos e cômodos. Assim sendo, a medida de carga para aquela hora ou minuto está implícita em inúmeros medidores, o que sugere uma série originalmente multivariada.

*A priori*, a informação de ambos os *datasets* que são manipulados remete à necessidade de verificar qual a classe de residência da HUE melhor se aproxima do tipo de carga da *Pecan Street*, visto que somente há uma disponível nessa. A classe de residência bangalô na HUE foi escolhida, por ser um tipo de construção de um pavimento e remeter à vivência de uma família inteira, o que oferece uma similaridade a demanda da classe oferecida pela *Pecan Street*. Referindo-se a construção das resoluções, como a base de dados da HUE somente disponibiliza os dados em resolução horária e os da *Pecan Street* estão dispostas em até 15 minutos, o método proposto poderá ser experimentado nas resoluções horárias em ambas, uma vez que é possível escalonar as granularidades disponíveis na *Pecan Street*. O inverso seria passível de erro.

Tendo ressaltado as características macroscópicas dos *datasets*, a Tabela 1 resume a dimensão de ambas as coleções de dados, porém em sua forma bruta. Como o objetivo dos modelos é a inferência da propriedade sintetizada do consumo de energia, em suas diversas resoluções de tempo, a etapa de pré-processamento para ambos serve como etapa de lapidação, o que retira inconsistências e faz existir demanda energética relacionada ao tempo e outras diversas propriedades. Ao final, a compilação possibilita a obtenção de uma correlação entre a leitura de demanda e as propriedades vinculadas.

Tabela 1 – Aspectos descritivos das bases utilizadas em sua versão original.

Dataset	Participant   Tipo de residência	Formato original do dataset (obs. x características)	Percentual de completude	Descrição absoluta da demanda energética ?
Pecan Street Inc.	661 - Residência Familiar	1 minuto: (525.530, 79); 15 minutos: (35.032, 79); 1 hora: (8.759, 79);	99%	Não
HUE	1 - Bangalô	1 hora: (29.202, 3)	100%	Sim

Fonte: Elaborada pelo autor (2023).

A utilização da etapa de pré-processamento, primeira etapa do método empregado, garante a existência da propriedade sintética de demanda, respeitando a resolução de tempo estudada. Através da Seção 5.3, direcionada a demonstração de seus resultados, é possível perceber a conseqüente mudança da disposição dos atributos, a construção da correlação entre a demanda total e as variáveis temporais e meteorológicas e, finalmente, as distribuições estatísticas dessa resultante e a construção da correta dimensionalidade aplicável aos modelos de aprendizado profundo.

## 5.2 IMPLEMENTAÇÃO

A implementação dos *scripts* de execução dos estudos de caso foram totalmente feitos em *Python* 3.9.2 para a construção das arquiteturas de aprendizado profundo <sup>1</sup>. Idealizando a facilitação do estudo para futuros alunos que utilizem o presente documento como referência, a implementação conta com a utilização do paradigma de orientação a objetos para transformar todos os componentes de construção do modelo, laço de treinamento, validação e arquivamento de *logs* das métricas em objetos herdáveis. O intuito de seguir esse formato é possibilitar o fácil entendimento das difíceis estruturas de DL, organizar os pontos vitais dos objetos de treinamento e diminuir a curva de aprendizado. A Figura 59, encontrada no Apêndice B demonstra o diagrama de classes dos objetos envolvidos no projeto, o qual está publicamente disponível<sup>2</sup>.

Referentes à descrição de *hardware*, os modelos foram treinados utilizando a *NVIDIA GTX 1660 Super* de 6Gb GDDR6, gerenciada pelo pacote de desenvolvimento *CUDA* 11.3 e a biblioteca *cuDNN* 8.7. Os *logs* de treinamento e validação, assim como o uso do *hardware* durante a utilização dos modelos, estão publicamente disponíveis no *Weights & Biases*<sup>3</sup>.

Destacando-se os hiperparâmetros utilizados, o *batch-size*, valor de exemplos a serem carregados em memória para o modelo, foram ponderados respeitando dois outros parâmetros: (i) o número de núcleos da central de processamento (*Central Processing Unit* – CPU) disponíveis, chamado de *workers* e (ii) a seqüência a ser estabelecida na janela

<sup>1</sup> *Pytorch* 1.11.0 com a biblioteca *Pytorch Lightning* 1.6.3

<sup>2</sup> <https://github.com/pv08/consumption-forecasting>

<sup>3</sup> <https://wandb.ai/pvbr08/EnergyConsumption>

temporal. Como a bancada de teste possui uma *CPU Intel Core I5 9400F @ 2.9GHz*, com 6 núcleos e 12 *threads*, mas com uma quantidade de 16Gb de memória DDR4 de 2400MHz, foi escolhido globalmente um *batch-size* de 32 e uma sequência de janela temporal de 60 amostras para um *label*, seguindo o padrão de previsão em passo único. Tais parâmetros foram escolhidos para que não houvesse qualquer exceção de memória e a configuração de *hardware* tivesse o melhor desempenho. Porém a implementação conta com o pacote Optuna (AKIBA et al., 2019), a fim de escolher os melhores hiper-parâmetros da rede, com exceção do horizonte amostral.

Com esses hiper-parâmetros globais, cada modelo utilizado no comparativo foi treinado em 200 épocas, sem qualquer parada precoce ou nível de paciência, a fim de perceber as flutuações do treinamento e validação. Mesmo que haja o treinamento exaustivo, os pesos da melhor época são destacados no arquivo de *checkpoint*. Tendo sido descrito os componentes de implementação, a Seção 5.3 descreve o avanço dos treinamentos e as inferências dos modelos, utilizando ou não a etapa de componentes principais.

### 5.3 RESULTADOS

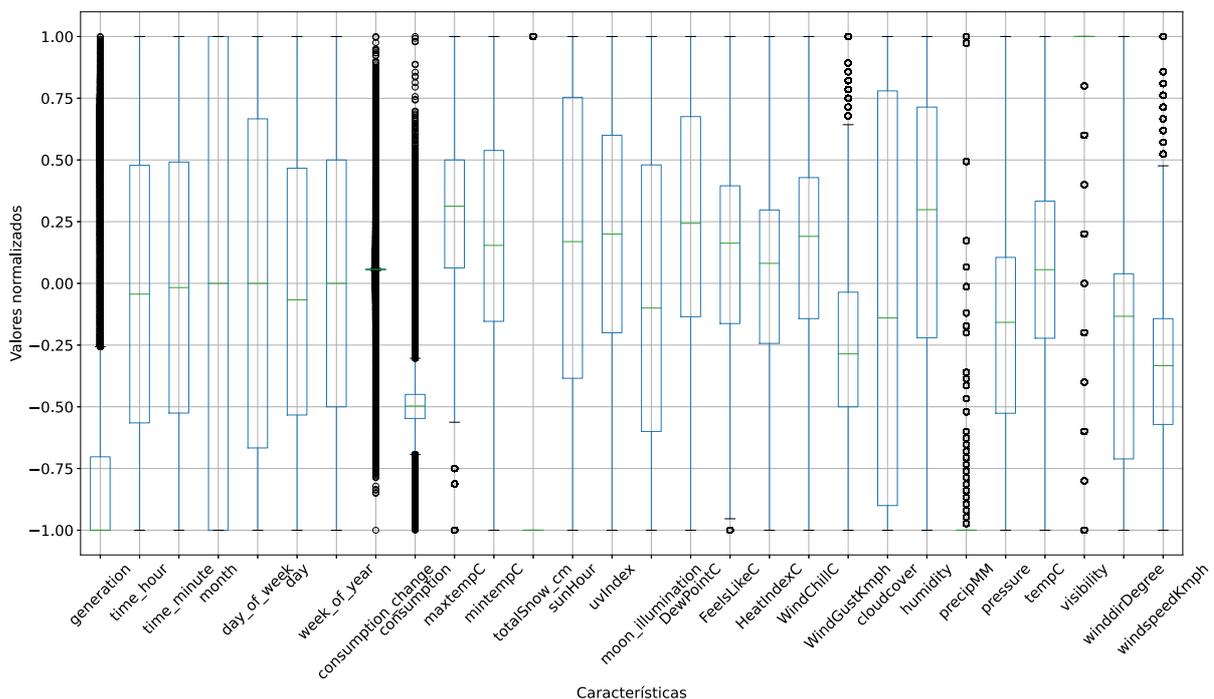
Como sugere a etapa de treinamento estabelecido pelo método proposto, dois são os tipos de dados a serem utilizados como entrada para as arquiteturas de aprendizado profundo. O primeiro tipo refere-se ao produto final da lapidação dos *datasets* originais, em que somente o emprego das técnicas de *feature engineering*. O segundo conjunto adiciona o destaque das principais características ao emprego das técnicas do primeiro conjunto, havendo uma consequente redução de propriedades para aquelas mais importantes. Para ambos, há uma característica implícita de consumo energético, que é sintetizada respeitando a resolução em destaque.

Inicialmente referindo-se a base da *Pecan Street Inc.*, a amostra selecionada possui 98% de completude, porém sua demanda total está dissolvida em 79 diferentes medidores, tendo somente 75 referenciados a construção de carga demandada, 2 a sua geração fotovoltaica e outras 2 a tensão de circuitos. Para a residência descrita na Tabela 1, somente existem 12 equipamentos de consumo disponíveis, ou seja, os outros 63 equipamentos não estão presentes na residência 661 da *Pecan Street*. A falta destes equipamentos não é um empecilho, visto que a primeira etapa do pré-processamento visa somar diferentes propriedades de consumo e geração para a estampa de tempo. Portanto, o consumo disperso por equipamento será substituído por apenas um atributo exemplificativo da demanda da residência.

Embora seja palpável a drástica redução de características por essa etapa, não existia qualquer propriedade que definisse o consumo pontual da residência. Contudo, estando a amostragem inclusa no domínio univariado, é necessário expandir os atributos temporais e incluir os dados meteorológicos da região de Austin, respeitando a granularidade em

destaque. O resultado final da aplicação da etapa de pré-processamento traz o sintético consumo de energia, medidos em resoluções de 1 minuto, 15 minutos e 1 hora, no intervalo de 1 ano, tendo outras 27 propriedades passíveis de serem correlacionadas ao objeto de inferência pelo modelo a ser treinado. Exemplificadamente, a Figure 15 ilustra os desvios, as médias e os quartis de todas as características envolvidas no banco de dados da *Pecan Street Inc.*, de minuto. É possível verificar que as Figuras 49 e 50, disponíveis no Apêndice B, demonstram as mesmas características, porém em diferentes distribuições e resoluções.

Figura 15 – *Boxplot* das características do banco de dados *Pecan Street Inc.*, em resolução de 1 minuto.

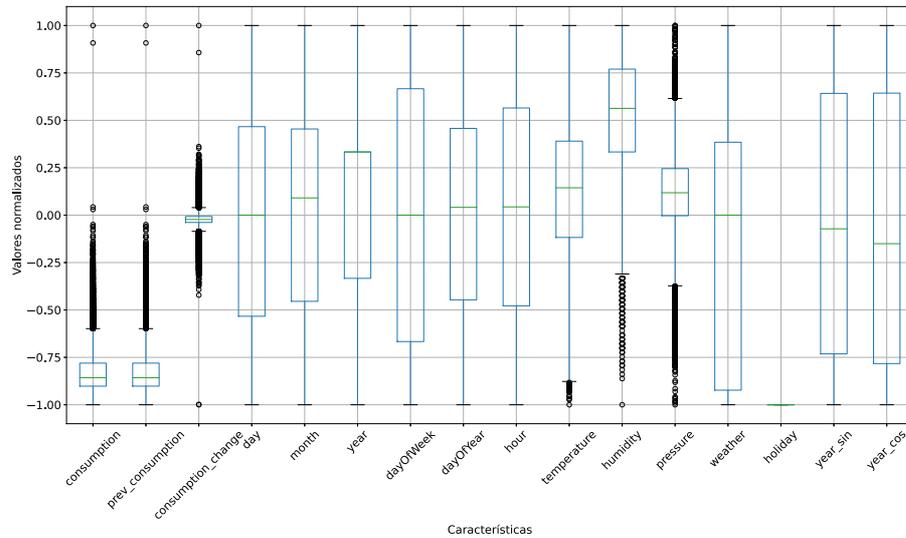


Fonte: Elaborada pelo autor (2023).

Ao contrário do que acontece no *dataset* da *Pecan Street*, a base da HUE já possui uma propriedade absoluta de consumo pontual, ou seja, já é possível perceber a flutuação do consumo total na resolução horária. Assim sendo, a etapa de pré-processamento é reduzida à tarefa de expansão da unidade temporal e a inclusão das medidas meteorológicas, já disponibilizadas pela editora dos dados, passo que gera uma quantidade menor de propriedades relacionadas ao tempo da Colúmbia Britânica. A Figura 16 ajuda na percepção dos 16 componentes que fazem parte do banco de dados horário da HUE.

As duas bases resultantes são produtos de um pré-processamento para obter uma completude de dados, assim como uma representatividade do consumo de isoladas construções, respeitando a resolução de tempo e do período a ser analisado. Mas, como é parte da metodologia de análise, é necessário obter quais são as propriedades representativas para ambos *datasets*. Sendo ainda parte da etapa de pré-processamento, o PCA ajudará a

Figura 16 – *Boxplot* das características do banco de dados HUE, em resolução de 1 hora.



Fonte: Elaborada pelo autor (2023).

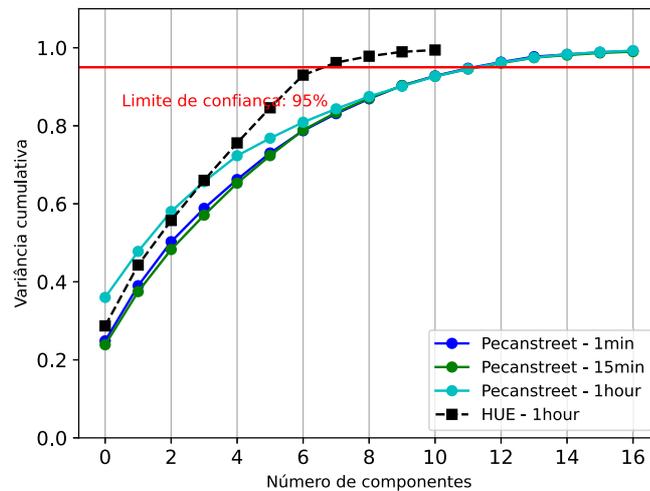
encontrar quais as características globalmente representativas do banco de dados.

A redução das propriedades é baseada no número de componentes representativo de cada *dataset*. Para que não fique subjetivo sua escolha, é possível obter, por meio da soma cumulativa da variância, a correta dimensão que represente no mínimo, 95% da amostra. Por meio da Figura 17 é possível perceber, para cada base de dados, a melhor quantidade de propriedades representativa, em sua respectiva resolução. Vale lembrar que, a propriedade de consumo necessariamente tem que estar listada nas mais importantes, já que a sequência de janelas temporais é em relação a ela. As Figuras 51, 52, 53 e 54, no Apêndice B, demonstram o mapa de calor entre o número de componentes e as características de cada banco de dados.

Com a redução da coleção dos atributos para as mais importantes, é possível verificar que a primeira comparação do comportamento dos modelos possui 2 grupos: o primeiro, composto por todas as propriedades compiladas da manutenção do *dataset* original; o segundo, conjunto reduzido por PCA que descreve as propriedades que melhor representam todo o *dataset* previamente compilado. Cada grupo, tem 4 conjuntos de dados, sendo 3 da *Pecan Street Inc.*, nas resoluções de 1 minuto, 15 minutos e 1 hora, e 1 para o *dataset* da HUE, na resolução horária. Cada um desses grupos possui seu subconjunto de treinamento, validação e avaliação, nas respectivas percentagens em relação ao total de observações, como levantado pelo método proposto: 70%, 20% e 10%

Fim da compilação dos *datasets*, é possível obter componentes estatísticos de cada resolução. A Seção 5.3.1 fornece o estudo do tipo de distribuição e os testes realizados para a definição do tipo de série temporal lidada. Além da possível análise, é perceptível a construção de parâmetros para a construção do modelo estatístico *Seasonal Autogressive*

Figura 17 – Números de componentes que melhor descreve o *dataset*.



Fonte: Elaborada pelo autor (2023).

*Integrated Moving Average - Exogenous Factors* (SARIMAX), constantemente representado na literatura como linha de base para os modelos de DL (GARCIA; MENDOZA; NAVARRO, 2020), (PATSAKOS; VROCHIDOU; PAPAKOSTAS, 2022).

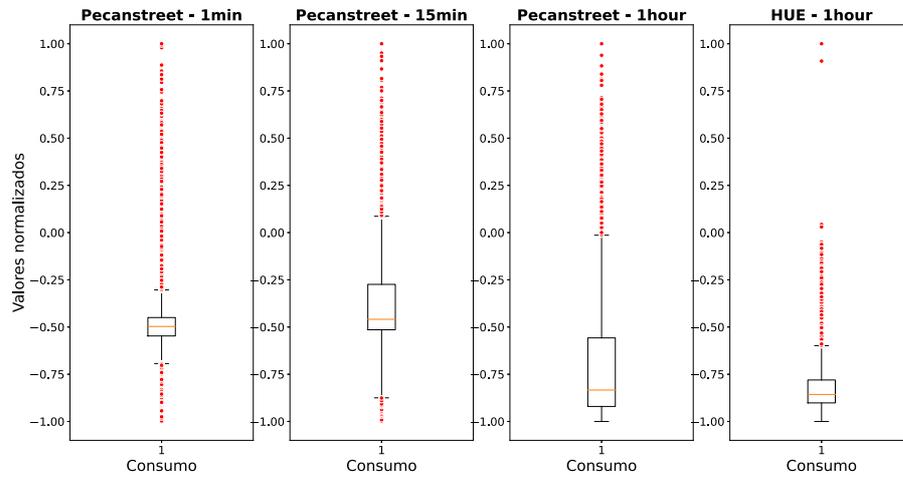
### 5.3.1 Análise estatística

O pré-processamento das bases resulta em leituras de consumo energético em 4 diferentes resoluções, dispostos em dois estudos de caso voltados aos atributos. Antes que possam ser demonstrados os resultados dos principais modelos de inferência, é importante delinear uma base comparativa. Sendo o modelo estatístico SARIMAX este *baseline*, ele é parametrizado por meio das leituras de consumo sintético para obter a métrica de sua acuracidade. É destacável que, embora exista diferentes coleções de dados a serem analisados, somente as amostras de granularidade são vistas, pois o modelo em questão foi implementado para inferir o objeto de pesquisa utilizando somente um atributo, a própria demanda. Portanto, testes de estacionariedade, distribuição estatística e métricas de dispersão estarão baseadas na leitura de demanda em diferentes granuralidades de tempo. Todos os demais modelos, posteriormente descritos, terão o modelo SARIMAX como base comparativa.

Antes que possam ser estudados os erros de inferência do conjunto de avaliação, o processamento dos estudos de caso possibilita a constatação dos componentes estatísticos das amostras de demanda. A correlação existente entre os atributos poderão ser analisadas pelas Figuras 55, 56, 57 e 58, presentes no Apêndice B.

Desta forma, tendo 4 diferentes séries históricas de consumo, é possível comparar as métricas bases de dispersão. A Figura 18 demonstra lado-a-lado as localizações das médias, quartis e os *outliers* existentes em cada uma das resoluções.

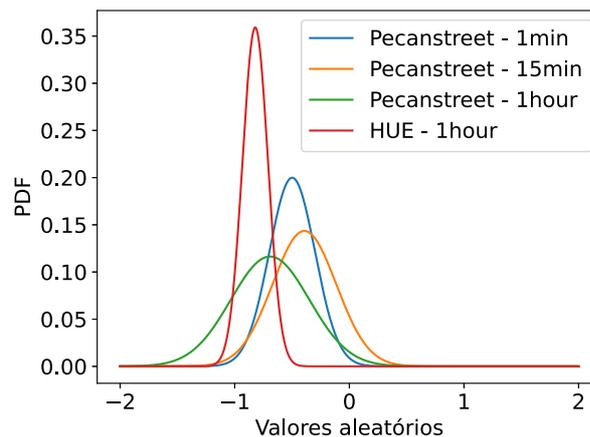
Figura 18 – *Boxplot* comparativo dos *datasets* resultantes da primeira parte da metodologia empregada.



Fonte: Elaborada pelo autor (2023).

Tendo as métricas de dispersão que fornecem apenas uma percepção macroscópica do nível das leituras de consumo, é possível utilizar a média e o desvio padrão para a análise da função densidade de probabilidade (*Probability Density Function* – PDF). A Figura 19 descreve a probabilidade da ocorrência dos diferentes níveis de consumo. Pela distribuição, é possível interpretar uma maior probabilidade de ocorrências de valores negativos, lembrando que estão normalizados.

Figura 19 – Análise PDF das diferentes resoluções dos *datasets*.

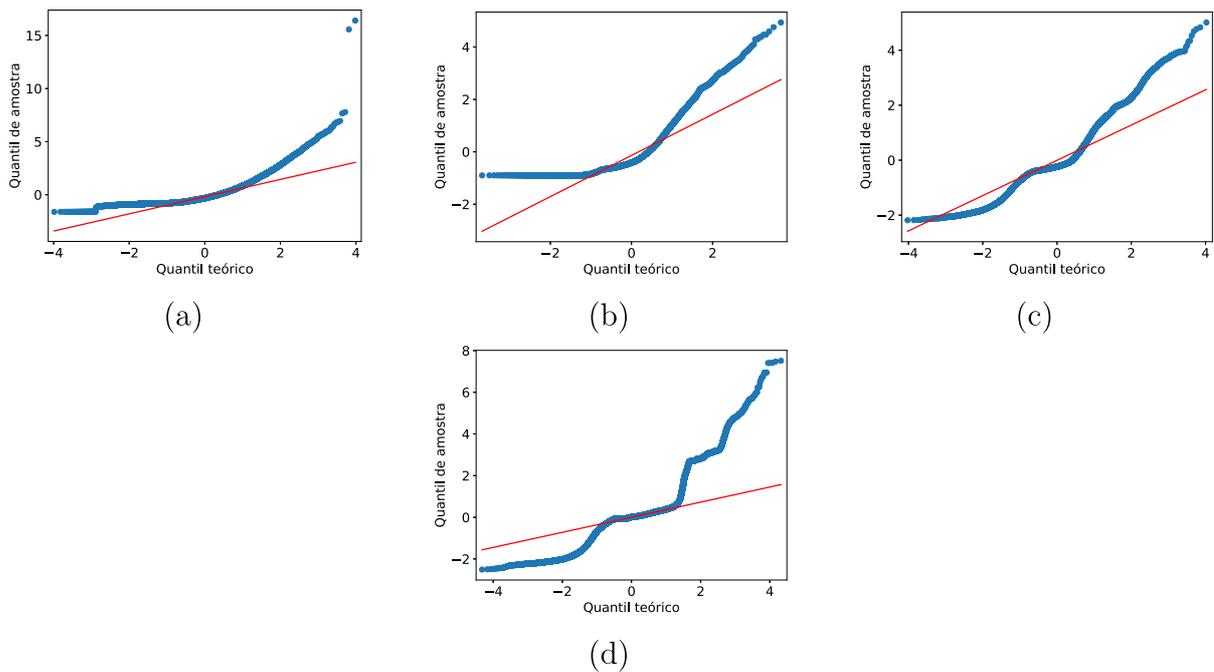


Fonte: Elaborada pelo autor (2023).

A utilização do PDF, junto com o desvio padrão e as médias, somente indicam uma série que possui uma distribuição normal. É necessário utilizar os testes de *Kolmogorov-Smirnov*, *Dickey-Fuller* e os gráficos Quantil-Quantil (QQ) para a percepção das tendências de estacionariedade e confirmar a distribuição normal, previamente destacada. Primeiramente utilizando o teste de *Kolmogorov-Smirnov*, a recusa da hipótese-nula, por meio

do  $p$  – *value*, demonstrou a estacionariedade da série, fato confirmado pelo teste de *Dickey-Fuller*. A Figura 20 mostra o gráfico QQ em uma tendência de crescimento linear, seguindo o quantil teórico, o que confirma a distribuição normal.

Figura 20 – O subgráfico (a) mostra o gráfico QQ utilizando a base da HUE, na resolução de 1 hora. Os subgráficos (b) à (d) mostram o comportamento dos quantiles utilizando os dados da *Pecan Street*, respectivamente, nas resoluções de 1 hora, 15 minutos e 1 minuto.



Fonte: Elaborada pelo autor (2023).

Os testes de *Kolmogorov-Smirnov* e *Dickey-Fuller* indicaram uma estacionariedade dos dados, um facilitador para a constatação das inferências por modelos estatísticos, não sendo necessário realizar novas manipulações na séries histórica do consumo energético. Essa percepção, antes que possa ser descrito a parametrização do modelo SARIMAX, possibilita verificar indicadores de autocorrelações.

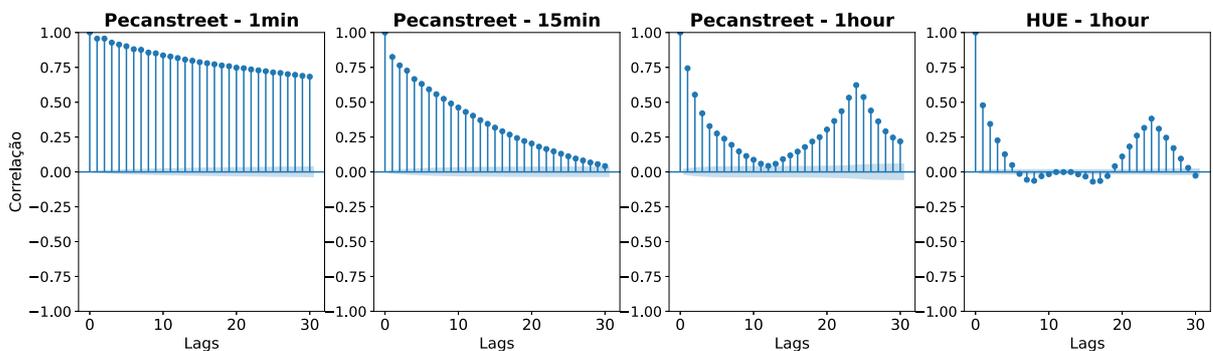
Antes que se possa demonstrar os resultados das funções de autocorrelação, é necessário explicar seus conceitos e suas particularidades, visto que são complementares. A autocorrelação é similar à correlação, ambas estudam o quão uma leitura atual de consumo é impactada pela leitura de  $n$  período passados, ou seja, o quão a série é similar a si mesma. A função de autocorrelação (*Auto Correlation Function* – ACF) e a de autocorrelação parcial (*Partial Auto Correlation Function* – PACF) são as responsáveis por dizer os níveis dessa correlação por meio dos *lags*, deslocamentos dos períodos da própria série.

Partindo do ACF, a função diz, para cada nível de *lag*, o quanto a leitura atual  $\tau$  está linearmente relacionada às passadas  $\tau - lag_n$ , onde  $lag_n$  são os períodos passados. Essa autocorrelação é dada no intervalo  $[-1, 1]$ , indicando o quão a relação linear é positiva ou negativamente ligadas às passadas. O PACF funciona de forma similar, tendo somente

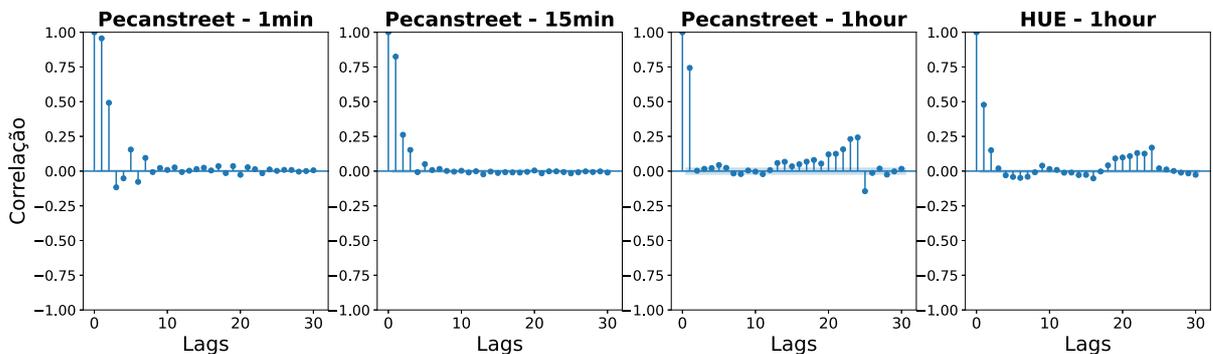
a diferença de demonstrar a autocorrelação isolada entre a leitura atual e o valor lido  $n$  lags atrás, não existindo autocorrelação acumulada, somente entre a atual e a primeira. Por meio da Figura 21, a qual descreve o comportamento dos *datasets* em 30 lags, a Figura 21a permite verificar uma autocorrelação positiva na maioria dos lags, sendo aos poucos diminuída.

Importante destacar que o primeiro lag é ignorado, pois a autocorrelação é em relação à ela própria, o que justifica uma correlação +1. Tendo a Figura 21b como descritivo do PACF, a qual demonstra a relação existente entre o lag e a primeira leitura, é possível perceber que não é necessário construir lags muitos grandes para obter uma correlação forte. As 5 primeiras leituras já possuem uma forte autocorrelação.

Figura 21 – Cálculo da autocorrelação por ACF e PACF nos *datasets* compilados em 30 lags.



(a) ACF utilizando 30 lags em todos os *datasets*.



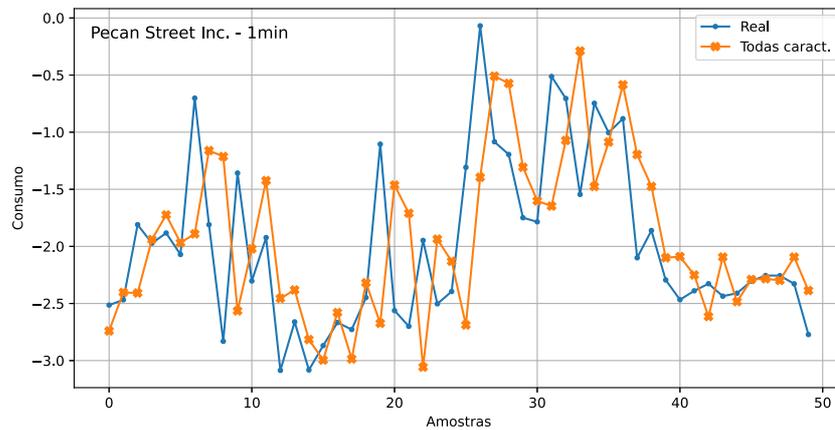
(b) PACF utilizando 30 lags em todos os *datasets*.

Fonte: Elaborada pelo autor (2023).

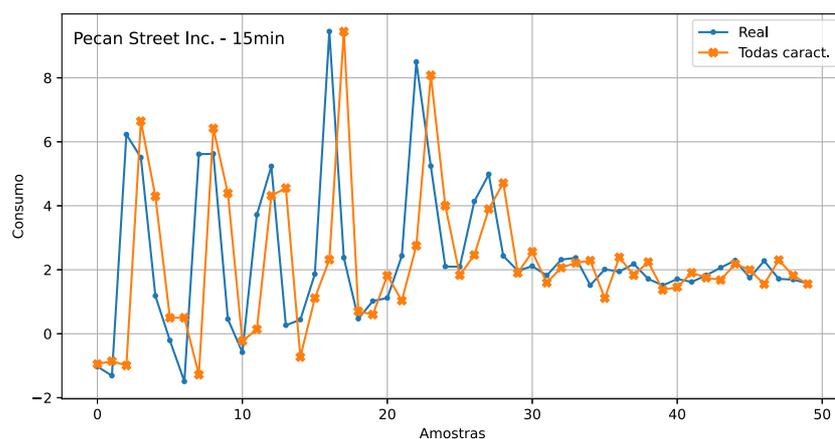
A constatação das autocorrelações por meio das funções ACF e PACF oferecem dados importantes para a correta parametrização do modelo regressivo estatístico. Como a série temporal possui propriedade sazonal, é utilizado o modelo SARIMAX para a realização das inferências, a qual agrega componentes sazonais, autorregressivos e de média móvel. Utilizando uma ordem de 4 propriedades autorregressivas ( $p$ ) e 3 de média móvel ( $q$ ), tendo seus respectivos componentes sazonais da ordem de 2 unidades ( $P$ ) e 12

( $Q$ ), a Figura 22 exemplifica a previsão utilizando os dados da *Pecan Street* em todas as resoluções.

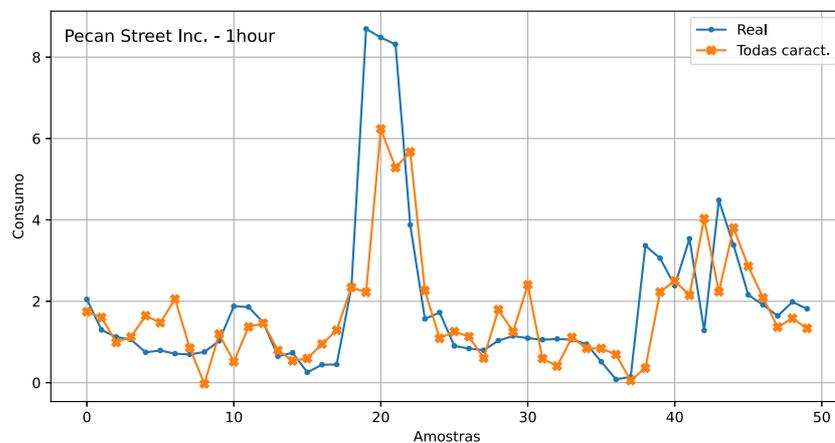
Figura 22 – Previsões das últimas 50 amostras utilizando modelo estatístico SARIMAX. As figuras (a) à (c) detalham as previsões utilizando os dados da *Pecan Street*, nas resoluções de 1 minuto, 15 minutos e 1 hora, respectivamente.



(a) 1 minuto.



(b) 15 minutos.



(c) 1 hora.

Fonte: Elaborada pelo autor (2023).

Por meio da inferência destacada, é possível obter um panorama reduzido do comportamento do modelo de regressão, por haver a análise das últimas 50 amostras de cada resolução. Assim sendo, as inferências demonstradas pela Figura 22 descrevem esse escopo. A Figura 60, nos Apêndices, detalha as previsões utilizando a base horária da HUE.

Mesmo que haja a percepção das inferências, é preciso, por meio das métricas de MSE, MAE e MAPE, avaliar todos os 10% do conjunto de dados, representando o subconjunto de avaliação previamente destacado. Dessa forma, por meio da Tabela 2 são destacados os índices obtidos na manipulação de ambos os *datasets*, em suas respectivas resoluções.

Tabela 2 – Métricas do modelo SARIMAX para os dois *datasets* analisados.

Dataset	Atributos	Resolução	MSE	MAE	MAPE	R2 Score
Pecan Street Inc.	Todos	1min	0.003380	0.022216	0.151010	0.913472
		15min	0.010599	0.055534	0.926418	0.815273
		1hour	0.037332	0.109684	3.398128	0.528440
HUE	Todos	1hour	0.205413	0.263126	2.607050	0.446193

Fonte: Elaborada pelo autor (2023).

Como especificado no início da Seção, o modelo estatístico é apenas uma base a ser comparada com o real objeto de estudo do presente documento, as redes de aprendizado profundo. Dessa forma, os resultados apresentados na Seção 5.3.2 são comparativamente analisados às métricas obtidas pelo modelo SARIMAX, em ambas ocasiões de conjunto de dados, a íntegra e a reduzida por PCA.

### 5.3.2 Comportamento dos modelos recorrentes

Como supracitado, as métricas trazidas pela Seção 5.3.1 na análise do comportamento do modelo estatístico SARIMAX oferece somente uma base comparativa para os modelos RNN, LSTM e GRU, a serem apresentados na presente na Seção. Importante ressaltar novamente que as séries tratadas no modelo estatístico são univariadas, não havendo qualquer construção da correlação e influência de outras variáveis à demanda energética. Como o próprio modelo utiliza medidas passadas para a construção da sua atual inferência, não é possível a construção do nível de importância de cada característica, muito menos a utilização da base criada por PCA.

Ao contrário dos modelos estatísticos, os de DL oferecem a possibilidade de customizar o conjunto de dados a ser utilizado para o treinamento das arquiteturas, tanto por meio do janelamento quanto pela quantidade de características das amostras. Por meio do janelamento, cada um dos subconjuntos de treinamento, validação e avaliação tem uma quantidade de amostras vinculada a um *label*, tendo a matriz no formato  $[b \cdot s \cdot f]$ , sabendo que  $b$  é o *batch-size*,  $s$  o tamanho da sequência e  $f$  a quantidade de propriedades.

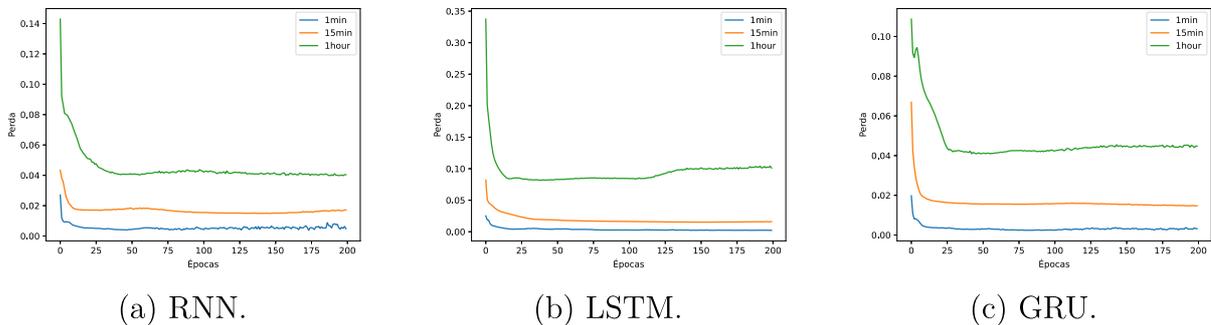
Por exemplo, utilizando os hiper-parâmetros previamente estabelecidos e a resolução de minuto do *dataset* da *Pecan Street*, a matriz de treinamento estaria no formato  $[32 \cdot 60 \cdot 28]$ , correspondentes aos 32 exemplos das 60 amostras, cada uma com 28 atributos. Cada exemplo possui uma matriz *label* bidimensional,  $[32 \cdot 1]$ , ou seja, 32 exemplos de somente 1 propriedade, o consumo esperado.

Como os dados da *Pecan Street* e da HUE possuem propriedades singulares, tanto para o conjunto total quanto para o PCA, cada um desses *datasets* possui sua matriz particular. A partir de então, essas matrizes, *tensores*, são utilizados como parâmetro de entrada para cada modelo. A função dessas entradas é, por meio da oferta de exemplos e um resultado esperado, adequar gradualmente os pesos dos neurônios das camadas mais internas da rede até que o erro entre o valor inferido pela rede e real seja mínimo. Como o processo subdivide-se nas etapas de treinamento e validação, as duas curvas representativas de seus erros devem decrescer ao longo das épocas, demonstrando que existe um erro progressivamente menor nas duas etapas. Tendo como análise primária as redes recorrentes, a Figura 61, disponível no Apêndice B, demonstra exatamente o esperado, um erro de treinamento e validação progressivamente menor até sua estabilização em níveis baixíssimos.

Mesmo que seja importante a análise do progresso durante a etapa de treinamento, a de validação é ainda mais importante, pois é quando os pesos atribuídos são testados. É possível observar a piora, a estabilização ou a melhora da acuracidade das inferências. Como o número de amostras varia de acordo com a resolução, é preciso analisar o comportamento durante a validação. Utilizando os dados da *Pecan Street*, por meio Figura 23 é possível notar as curvas de validação alcançadas pelos modelos recorrentes, nas 3 resoluções disponíveis. É perceptível que o erro da resolução de 1 minuto foi menor que a de 15 minutos, estando a resolução horária na pior posição de avanço durante o treinamento dos modelos. Como os pesos das camadas internas da arquitetura são ajustados com base nos exemplos, essa piora provavelmente é dada pela reduzida quantidade de amostras presentes nessas duas últimas granularidades de tempo, que reduz os subconjuntos usados no treinamento das redes de DL.

Como supracitado, os dados até então utilizados são do conjunto de características compiladas pelo método de pré-processamento. O próximo passo é estudar o comportamento de cada um desses modelos utilizando as bases com atributos reduzidos por PCA. Como todos os hiper-parâmetros foram mantidos, é possível observar o comportamento dos modelos somente pela diminuição das propriedades que, *a priori*, são as representativas de todo o conjunto de dados original. Assim, por meio Figura 24 e 25 que, respectivamente, descrevem o comportamento em meio aos dados da *Pecan Street* e HUE, é comparável o avanço da validação utilizando atributos reduzidos por PCA e os previamente processados. Ao comparar o avanço do treinamento para os modelos em destaque, não é vista alteração significativa no comportamento de validação, seguindo o mesmo padrão de avanço. Porém,

Figura 23 – Comparativo das curvas de validação para os modelos recorrentes em relação a resolução as três resoluções abordadas pelo *dataset Pecan Street*.



Fonte: Elaborada pelo autor (2023).

existem outros fatores a serem analisados.

O tempo de treinamento pode ser um fator destacável. A redução das características pode ocasionar a facilidade do modelo em inferir. A Tabela 3, a qual expõe dados extraídos do *Weights & Biases*, demonstra reduções mínimas no tempo de treinamento entre casos pautados. A bancada de teste não foi, em sua maior parte, utilizada especificamente para o treinamento dos modelos, portanto, é possível que o uso diário possa ter afetado no desempenho temporal do treinamento.

Tabela 3 – Tempo de treinamento dos modelos de recorrência.

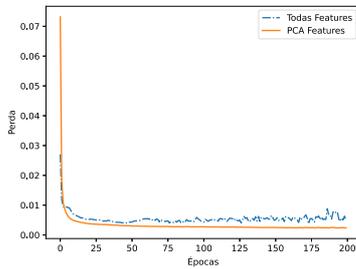
Dataset	Atributos	Resolução	Tempo de treinamento		
			RNN	LSTM	GRU
Pecan Street Inc.	Todos	1min	10h:2min	7h:32min	8:31min
		15min	3h:54min	3h:14min	3h:11min
		1hour	1:38min	1h:35min	2h:24min
	PCA	1min	8h:26min	8h:10min	7h:40min
		15min	2h:6min	2h:58min	2h:55min
		1hour	33min:55s	46min:13s	45min:5s
HUE	Todos	1hour	1h:30min	2h:55min	2h:49min
	PCA	1hour	2h:50min	3h:21min	3h:13min

Fonte: Elaborada pelo autor (2023).

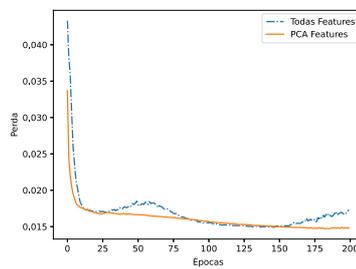
Antes que seja possível verificar os erros obtidos durante a avaliação dos modelos. As Figuras 26 e 27, demonstrando, respectivamente, a inferência das 50 amostras da *Pecan Street* e HUE, mostram uma boa identificação de *spikes* e mudanças bruscas de padrões.

A análise gráfica das inferências traz informações importantes sobre o comportamento em meio às amostras, principalmente quando comparadas às curvas de treinamento e validação. Porém, como a análise esta restrita a uma pequena amostra, é preciso utilizar

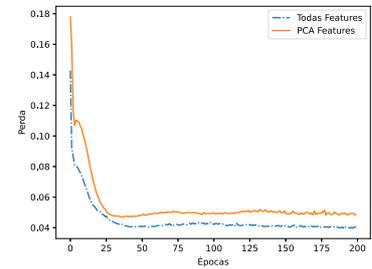
Figura 24 – Comparativo entre as curvas de validação dos dados da *Pecan Street Inc.*, utilizando todas as características e aquelas vistas como mais importantes por PCA, em todas as resoluções disponíveis.



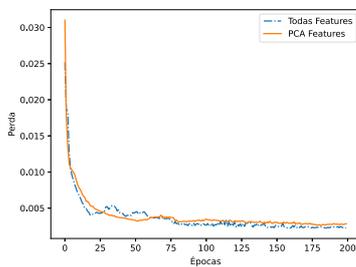
(a) RNN - Resolução de 1 minuto.



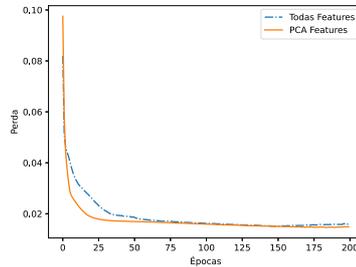
(b) RNN - Resolução de 15 minutos.



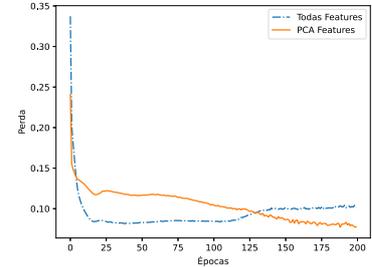
(c) RNN - Resolução de 1 hora.



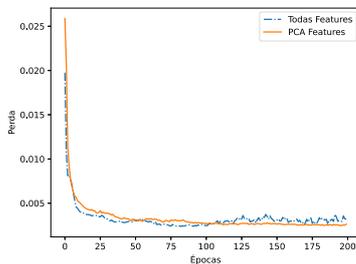
(d) LSTM - Resolução de 1 minuto.



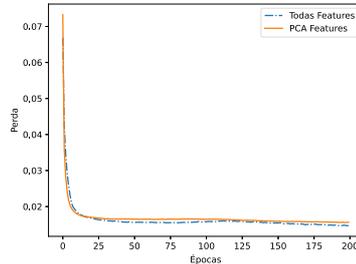
(e) LSTM - Resolução de 15 minutos.



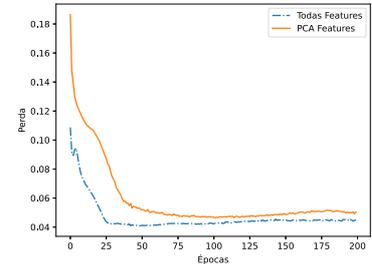
(f) LSTM - Resolução de 1 hora.



(g) GRU - Resolução de 1 minuto.



(h) GRU - Resolução de 15 minutos.



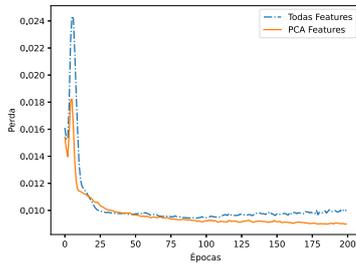
(i) GRU - Resolução de 1 hora.

Fonte: Elaborada pelo autor (2023).

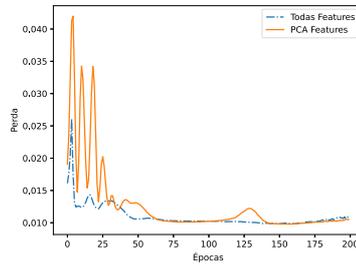
as métricas e, enfim, compará-las àquelas obtidas pela inferência do modelo SARIMAX. As Figuras 28 e 29, respectivamente, demonstram os índices de MSE alcançados pela base da *Pecan Street* e HUE para a situação analisada. Comparando-as com as inferências das Figuras 26 e 27 é perceptível que as previsões realizadas utilizando as características do PCA foram mais próximas do real. As métricas do PCA das Figuras 28 e 29 fortalecem essa percepção, visto que o erro foi menor, tendo pouca variação somente no uso da resolução de 15 minutos da *Pecan Street*.

As Tabelas 4, 5, 6 e 7, presentes no Apêndice A, demonstram as métricas performáticas alcançadas pelo modelos recorrentes analisados. Ao ser observada a Tabela 4, tendo como pauta a amostra da *Pecan Street* de 1 minuto, o modelo RNN obteve um acréscimo

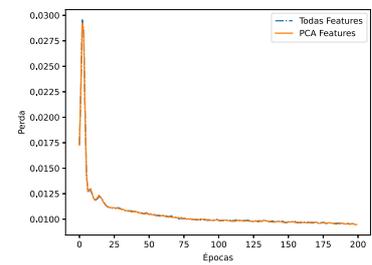
Figura 25 – Comportamento da validação utilizando a coleção horária de dados *HUE Dataverse*, seguindo o comparativo entre os atributos originais e PCA.



(a) RNN - Resolução de 1 hora.



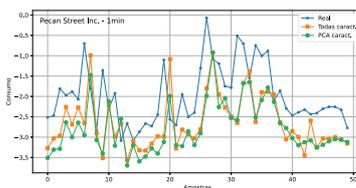
(b) LSTM - Resolução de 1 hora.



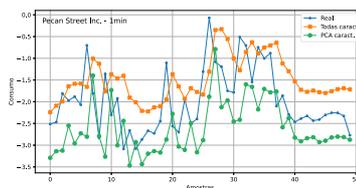
(c) GRU - Resolução de 1 hora.

Fonte: Elaborada pelo autor (2023).

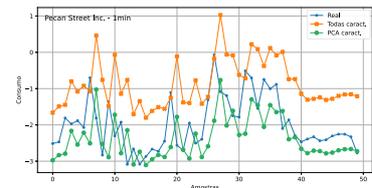
Figura 26 – Previsão comparativa utilizando os dados da *Pecan Street Inc.* entre todas as características e PCA de todos os modelos recorrentes.



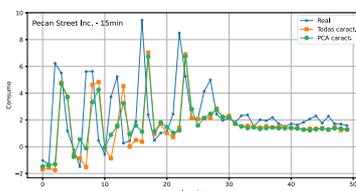
(a) Arquitetura RNN - Resolução de 1 minuto.



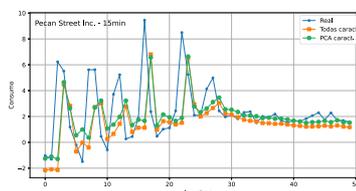
(b) Arquitetura LSTM - Resolução de 1 minuto.



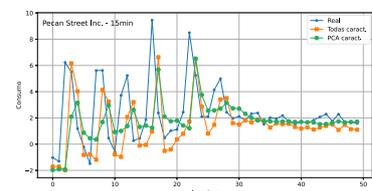
(c) Arquitetura GRU - Resolução de 1 minuto.



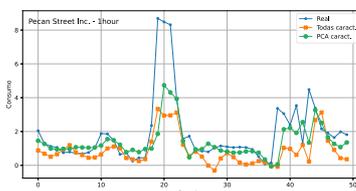
(d) Arquitetura RNN - Resolução de 15 minutos.



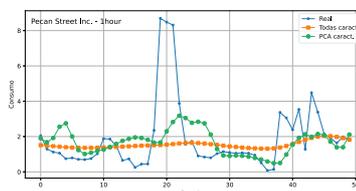
(e) Arquitetura LSTM - Resolução de 15 minutos.



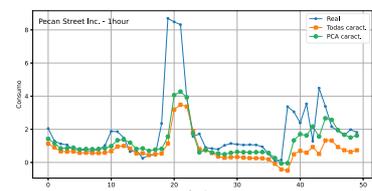
(f) Arquitetura GRU - Resolução de 15 minutos.



(g) Arquitetura RNN - Resolução de 1 hora.



(h) Arquitetura LSTM - Resolução de 1 hora.



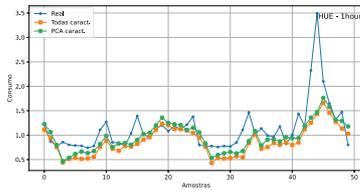
(i) Arquitetura GRU - Resolução de 1 hora.

Fonte: Elaborada pelo autor (2023).

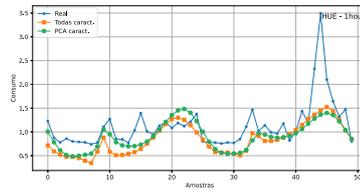
de 17.21% na acuracidade se comparado ao modelo SARIMAX. Porém, ao observar as demais resoluções de tempo, somente existe melhora quando a base reduzida por PCA é utilizada para a granularidade de 15 minutos da *Pecan Street* e a horária da *HUE*.

A base horária da *Pecan Street* obteve bons resultados somente no modelo estatístico.

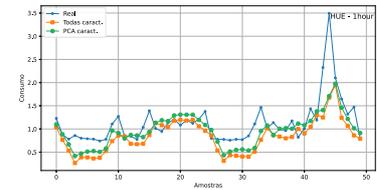
Figura 27 – Comportamento da previsão utilizando as coleções de dados da *HUE Dataverse*, seguindo o comparativo entre todas os atributos e os reduzidos por PCA.



(a) Arquitetura RNN - Resolução de 1 hora.



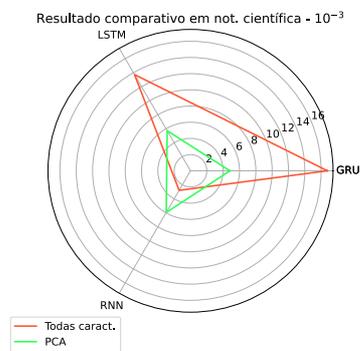
(b) Arquitetura LSTM - Resolução de 1 hora.



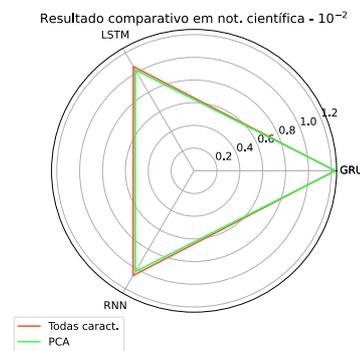
(c) Arquitetura GRU - Resolução de 1 hora.

Fonte: Elaborada pelo autor (2023).

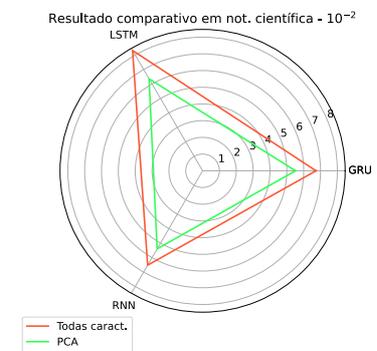
Figura 28 – Comparativo do MSE de avaliação utilizando as resoluções da *Pecan Street* com todas as características e reduzidas por PCA.



(a) 1 minuto.



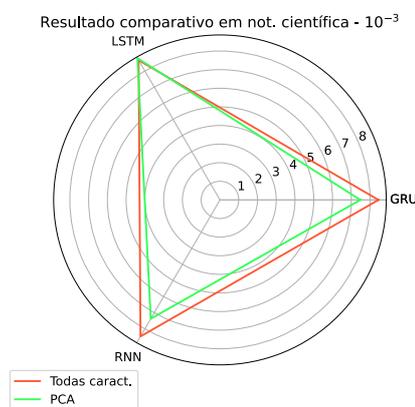
(b) 15 minutos.



(c) 1 hora.

Fonte: Elaborada pelo autor (2023).

Figura 29 – Comparativo do MSE de avaliação utilizando as resolução horária da HUE com todas as características e reduzidas por PCA.



Fonte: Elaborada pelo autor (2023).

Supondo que a quantidade de amostras disponíveis para o treinamento influencie na acuracidade, é possível supor que modelos mais simples performem melhor na presença de

poucos dados, já que a base em questão é a que possui o menor número de amostras.

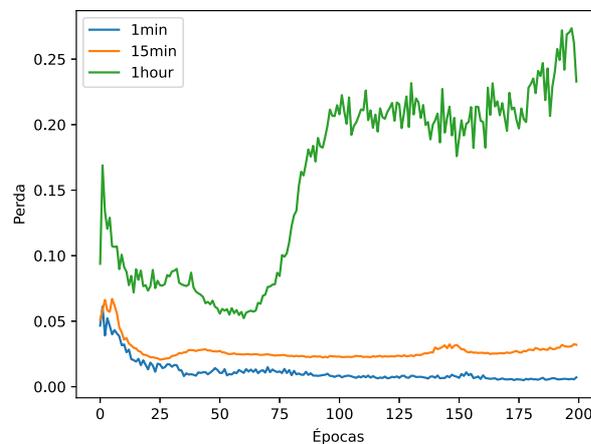
Contudo, o quantitativo de arquiteturas é muito pouco para obter essa análise. Portanto, é destacável a sua comparação com novas arquiteturas inseridas no SOTA e aqueles pouco referenciados na demonstração dos resultados. A Seção 5.3.3 descreve o comportamento dos modelos *Transformers*.

### 5.3.3 Comparativo dos modelos recorrentes e *Transformers*

Embora exista a tradicionalidade do uso dos modelos recorrentes, é muito interessante verificar o resultados de outras arquiteturas, pois particularidades podem ser encontradas, ainda mais quando há o cenário de múltiplas granularidades e atributos envolvidos.

A implementação de Zerveas et al. (2021), chamada de *Temporal Time-Series Transformers* (TST), é ótima para ser estudada e comparada com os modelos recorrentes. Mantendo as mesmas condições de hiper-parâmetros previamente estabelecidas, a Figura 30 demonstra comparativamente o MSE de validação alcançado em todas as resoluções disponíveis na *Pecan Street*, no cenário onde todas os atributos pré-processados estão presentes, sendo possível constatar a ocorrência de um *overfitting* na resolução horária. Muito embora a resolução em questão ofereça a percepção da ocasional piora do modelo, possivelmente por enviesamento das inferências ocasionada por falta de amostras suficientes, os níveis de erro alcançados nas demais granularidades de tempo é comparável aos recorrentes. As Figuras 62 e 63, presentes no Apêndice B, detalham melhor o avanço do treinamento do modelo TST para a coleção de dados da *Pecan Street* e HUE, respectivamente.

Figura 30 – Avanço do conjunto de validação para as três resoluções disponíveis na base da *Pecan Street* utilizando o modelo TST.

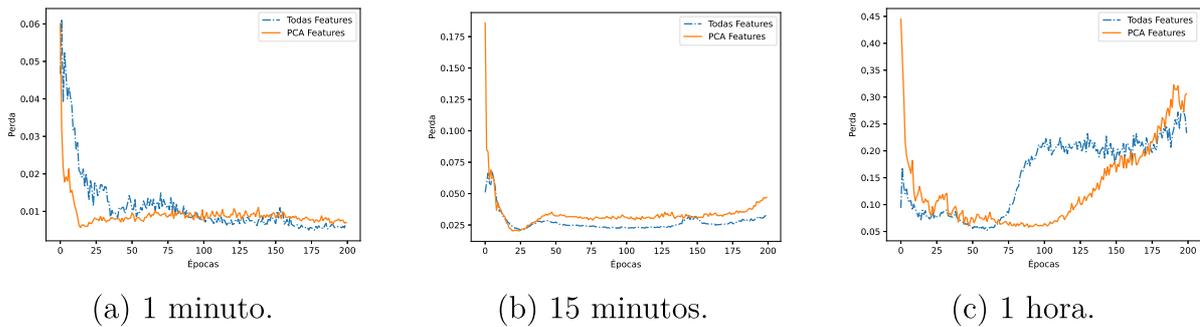


Fonte: Elaborada pelo autor (2023).

Somente tendo sido analisado a circunstância da utilização da base com todas os atributos, é necessário analisar o comportamento dos modelos também pautando as

propriedades reduzidas por PCA, constatando o comportamento das curvas de treinamento em meio a simplificação do conjunto de características. As Figuras 31 e 33 detalham o avanço do treinamento nas duas circunstâncias, sendo melhor detalhado o *overfitting* gerado na base horária da *Pecan Street*.

Figura 31 – Avanço do treinamento da arquitetura TST utilizando os dados da *Pecan Street*, em circunstâncias de utilização total dos atributos e reduzidas por PCA.



Fonte: Elaborada pelo autor (2023).

Figura 32 – Avanço do treinamento da arquitetura TST utilizando a base horária da HUE, em circunstâncias de utilização total dos atributos e reduzidas por PCA.

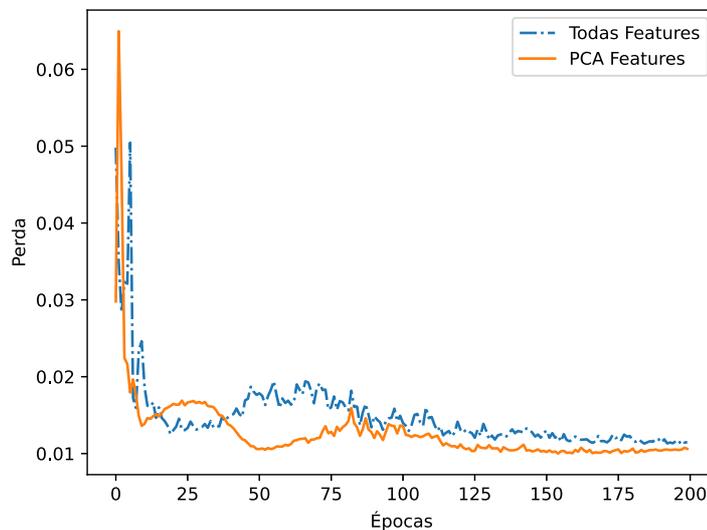
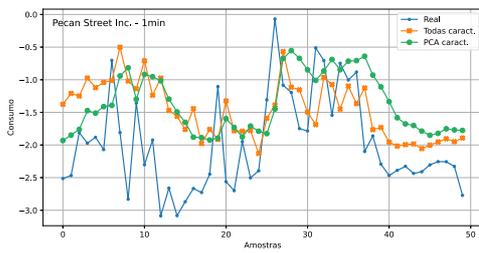


Figura 33 – Resolução horária.

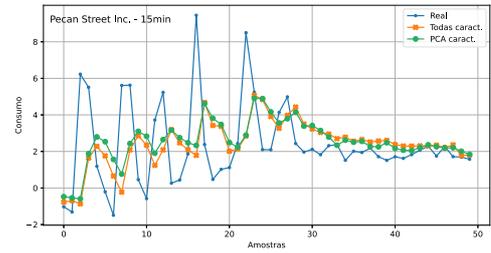
Fonte: Elaborada pelo autor (2023).

Como visto para os modelos recorrentes, as Figuras 34 e 36 mostram as inferências do modelo TST nas duas situações pautadas, também tendo como base as últimas 50 amostras do conjunto de avaliação. É possível destacar um grau acentuado de similaridade de previsão entre o modelo *Transformer* com as redes recorrentes, porém é destacável a facilidade de o modelo TST identificar melhor os picos de leituras e as mudanças bruscas de comportamento, melhor perceptível quando os *spikes* são analisados.

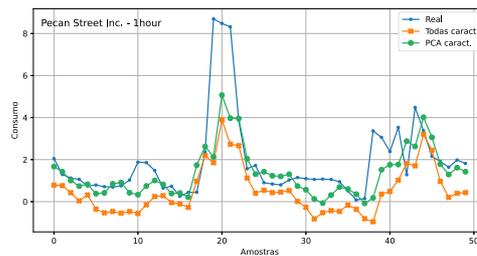
Figura 34 – Previsão comparativa do modelo TST utilizando os dados da *Pecan Street Inc.* entre todas as propriedades e as reduzidas por PCA.



(a) 1 minuto.



(b) 15 minutos.



(c) 1 hora.

Fonte: Elaborada pelo autor (2023).

Figura 35 – Previsão comparativa do modelo TST utilizando a base horária da HUE entre todas as propriedades e as reduzidas por PCA.

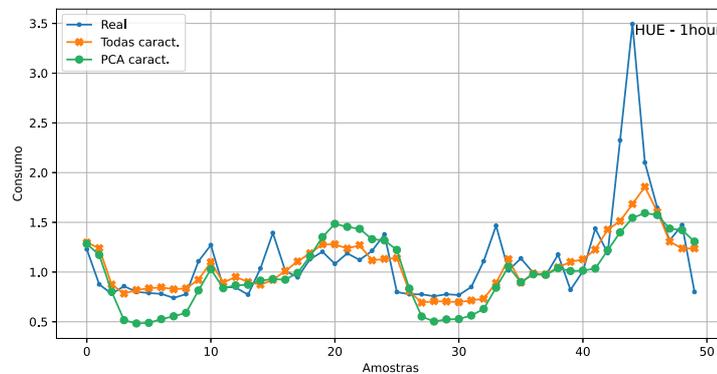


Figura 36 – Arquitetura TST - Resolução de 1 hora.

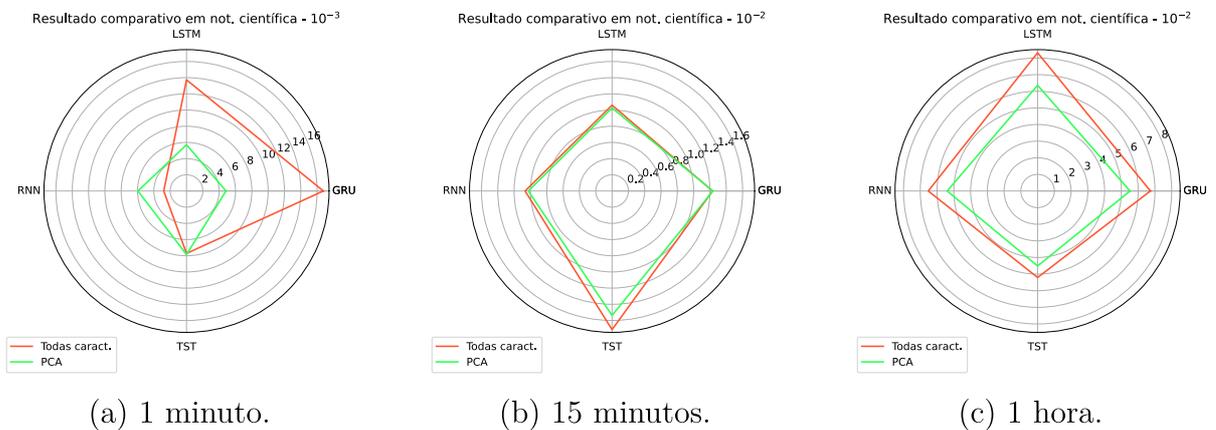
Fonte: Elaborada pelo autor (2023).

Importante destacar que, apesar de existir um *overfitting* na resolução horária da *Pecan Street*, por meio da Figura 31c é constatável que o *checkpoint* salvo é antes do desencadeamento do treinamento. Portanto, é possível verificar que o patamar da perda durante a validação nessa resolução foi próxima das demais. Essa visualização ressalta a afirmação de que o conjunto de teste dos dados foi avaliado utilizando os melhores pesos dos modelos.

Sendo o último passo da análise, as Figuras 37 e 38 comparam as métricas MSE

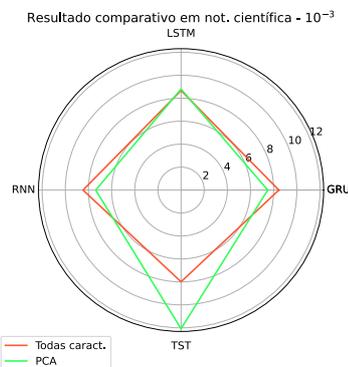
obtidas quando todo o subconjunto de teste é avaliado pelas melhores épocas salvas. Seguindo os resultados presentes na Tabela 4, ressaltados no Apêndice A, baseado na resolução de 1 e 15 minutos da *Pecan Street*, o modelo não apresentou qualquer melhora quando comparado aos modelos recorrentes e estatístico SARIMAX, em ambas ocasiões de coleção de atributos. A arquitetura TST, em relação à RNN, se destacou na avaliação da base horária da *Pecan Street*, possuindo uma melhora de 20,2% quando todos os atributos são utilizados e 16,36% quando o domínio do PCA é analisado. Comparado-o ao SARIMAX, não houve qualquer melhora, em ambos os casos. A mesma situação é vista ao observar a base horária da HUE: melhora ao compará-la aos modelos recorrentes e piora ao estatístico.

Figura 37 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA.



Fonte: Elaborada pelo autor (2023).

Figura 38 – Comparativo dos MSEs alcançados utilizando os modelos recorrentes e TST para a base horária da HUE.



Fonte: Elaborada pelo autor (2023).

Os modelos *Transformers* trazem resultados comparáveis aos recorrentes, tradicionalmente abordados em pesquisas de séries temporais. Tendo sido analisados os resultados

em diferentes granularidades de tempo, é perceptível seu desempenho em meio às amostras resumidas, muito embora haja a ocorrência de um *overfitting*. O desempenho visto ressalta o destaque da arquitetura no SOTA.

Contudo, existem outras arquiteturas que, muito embora tenham um maior destaque em outras domínios, são ocasionalmente referenciados como tratativas para a previsão da demanda energética, portanto, interessantes de serem inclusas no método comparativo. Desta forma, a Seção 5.3.3 trata da descrição do comportamento dos demais modelos encontrados para a previsão de séries temporais.

#### 5.3.4 Treinamento de outros modelos encontrados no SOTA

Observar o comportamento da arquitetura TST de Zerveas et al. (2021) permitiu constatar a diminuição do erro nas resoluções horárias de ambos *datasets* analisados. Apesar de as resoluções horárias terem sido o conjunto de dados com menor indicador de erro, foi o caso de ocorrência de *overfitting*, provavelmente causado pela quantidade insuficiente de amostras para treinamento e validação.

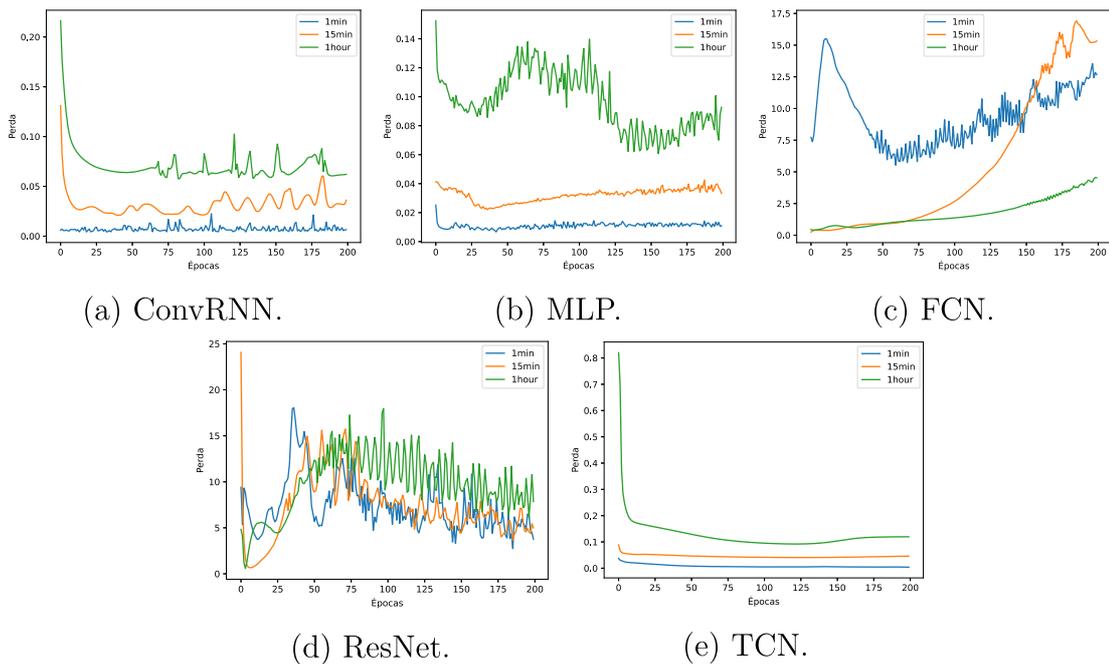
Como os algoritmos de árvore de decisão, máquinas de suporte, redes convolucionais, MLP, entre outras, também estão presentes no SOTA para tarefas de regressão, todos eles são incluídos no comparativo. Portanto, as seguintes redes foram implementadas para a obtenção das curvas de treinamento e validação: (i) MLP, uma implementação primariamente voltada a classificação, porém customizada à inferência de séries temporais; (ii) *Fully Convolutional Network* (FCN), uma implementação de Wang, Yan e Oates (2016) utilizando camadas convolucionais que, embora seja usada para classificação, foi customizada para a regressão; (iii) *Residual Network* (ResNet), arquitetura mais adotada no domínio de visão computacional, mas também implementada por Wang, Yan e Oates (2016) para ser comparada à FCN; (iv) *Convolutional Recurrent Neural Network* (ConvRNN), um modelo que adota camadas recorrentes e convolucionais; (v) TCN, a qual também segue a diretiva de inserção de aspectos convolucionais (BAI; KOLTER; KOLTUN, 2018); (vi) *XGBoost*, modelo inserido na categoria de árvore de decisão; (vi) SVR, proposta também muito utilizada, porém para resoluções com menores granularidades.

Cada um desses modelos seguiu o mesmo método previamente proposto, também mantendo os hiper-parâmetros já testados. As Figura 64, 65 e 66, disponibilizadas nos Apêndices, demonstram as curvas de treinamento e validação das resoluções disponíveis na base da *Pecan Street*. As curvas, resultantes da utilização da base horária da HUE poderá ser analisada através da Figura 67, também disponível no Apêndice B.

Para comparar o comportamento da validação nas diferentes resoluções, a Figura 39 demonstra a curva comparativa de validação dos modelos citados em todas as resoluções disponíveis para os dados da *Pecan Street*. Por meio dessa Figura, é perceptível que o índice de erro desses modelos é maior em todas as resoluções, tendo os modelos ConvRNN

e TCN se sobressaído no grupo em pauta. Através da grande oscilação presente durante a validação, é possível constatar que os modelos MLP, FCN e ResNet não obtiveram sucesso satisfatório com o conjunto validação. O contraste do desempenho entre esses modelos e os demais mais performáticos, pode ser melhor visto pela comparação da constância do erro de validação alcançados.

Figura 39 – Comparativo do avanço da validação para os modelos secundários de DL, tendo como base os dados da *Pecan Street*, utilizando todas as características originalmente pré-processadas.



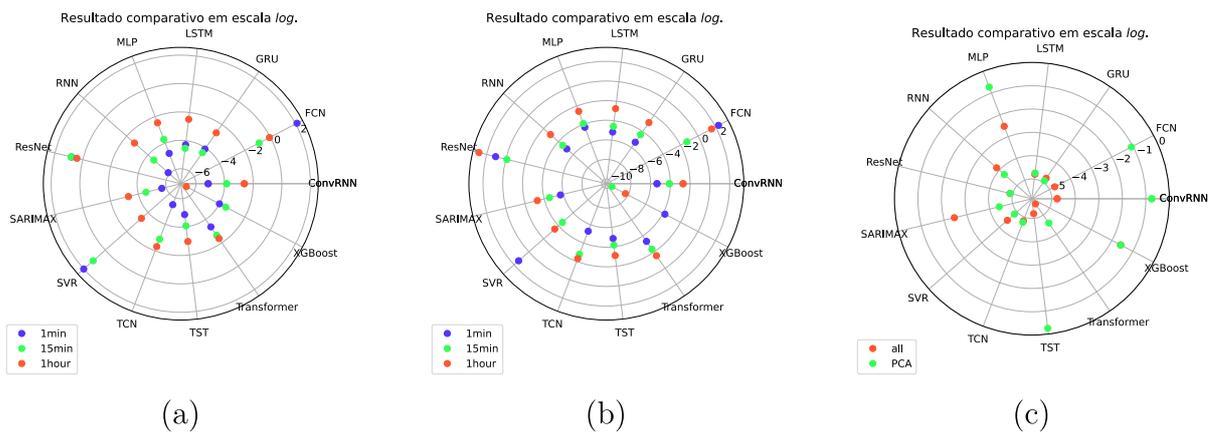
Fonte: Elaborada pelo autor (2023).

Os modelos também foram avaliados com características reduzidas por PCA. O avanço dos modelos utilizando as duas categorias de dados, poderá ser melhor analisada no Apêndice B, por meio das Figuras 68 a 70, quando os dados da *Pecan Street* são utilizados, e pela Figura 71, que demonstra o comparativo dos dados da HUE. As Figuras 72 a 74 demonstram a manipulação dos dados da *Pecan Street* com a Figura 75 para os dados da HUE para as inferências das 50 últimas amostras. Em todas as figuras destacadas é possível verificar o o desempenho ressaltado dos modelos ConvRNN e TCN.

Como a intenção é levantar um comparativo de erros alcançados pelos modelos, a Figura 40 demonstra a acuracidade de todos eles, para ambas bases tratadas. Através dessa constatação, por meio da Figura 40a, percebe-se que os modelos mais simples se comportaram melhor na manipulação de estruturas com poucos dados, sendo alcançado erros menores. É possível perceber que o modelo *XGBoost* ofereceu bons resultados em relação aos demais modelos, porém somente em resoluções horárias, também as que tinham pouco volume de dados. Quando a Figura 40b é comparada à 40a, fica perceptível que

grande parte dos modelos se comportou melhor com a utilização de todas as características, pois seus indicadores de erro estão mais ao centro do gráfico. O mesmo pode ser visto na Figura 40c com os dados horários da HUE. O comparativo poderá ser melhor construído utilizando a Tabela 4, presente no Apêndice A.

Figura 40 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando as bases com todas as características e por pré-processamento por PCA. As Figuras (a) demonstra todas os atributos compilados e (b) os reduzidos por PCA, enquanto a (c) a única resolução disponível na HUE.



Fonte: Elaborada pelo autor (2023).

#### 5.4 VARIÁVEIS MAIS IMPACTANTES POR SHAP VALUES

Utilizar o PCA para apartar as principais *features* teve resultados satisfatórios quando a resolução horária da *Pecan Street* é vista, granulação que possui o menor número de amostras disponíveis. Um dos objetivos de usar essa extração de atributos é simplificar a coleção dos dados, sem que haja a perda da representabilidade de toda a coleção original de propriedades.

Os resultados apresentados na Seção 5.3 demonstram tão somente o comportamento de cada modelo utilizando essa coleção representativa. Mesmo assim, ainda há o questionamento do comportamento de cada modelo, quando o ponto de análise é o atributo. A ferramenta SHAP Values ajuda na análise do impacto desses atributos para uma previsão isolada.

Para tanto, assim como feito na avaliação do conjunto de teste, o melhor *checkpoint* é instanciado e todos os hiper-parâmetros foram mantidos. Importante destacar que a interpretação é feita sobre o conjunto de dados originalmente processado, ou seja, aquela que possui todas as características. Esse conjunto dá origem a um novo *dataset* com os atributos vistos como importantes para a ferramenta, sendo os modelos novamente treinados e avaliados, incluindo as métricas como comparativos para os resultados já conquistados.

O SHAP *Values* fornece a força e a importância de cada atributo isolado, respeitando individualmente o modelo. A resolução de 1 minuto da *Pecan Street* como bancada de análise, por sua complexidade, e os modelos recorrentes e TST como testes, a Figura 41 pode ser traduzida como a força que cada *feature* possui para a inferência de um único valor normalizado. O lado esquerdo (cor vermelha) das subfiguras correspondem às características que forcem a inferência do modelo para valores positivos, enquanto o lado direito (cor azul) fazem o contrário. O ponto de convergência entre ambos faz surgir o valor inferido. Cada força atuante nesse valor possui seu grau de impacto, definida pelo tamanho da barra. Assim, por exemplo, a Figura 41a demonstra a propriedade *cloudcover* como a mais impactante para que a inferência tenha valores menores. Do lado contrário, a variável *generation* oferece a maior força para que a inferência adote valores mais elevados. A força das duas em conjunto com as demais, fazem com que o valor inferido seja  $-0,46$ .

O impacto das características para os demais modelos pode ser melhor analisado pelas Figuras 76 à 80, que trazem as resoluções da *Pecan Street*, e pela Figuras 81 e 82 que abordam o banco de dados da HUE. Para detalhar a importância real de cada atributo para todo o conjunto de avaliação, a Figura 42 mostra a média de pontuação de cada característica utilizando os dados de 1 minuto da *Pecan Street*.

Por estar direcionado a somente uma única previsão, o gráfico de força não diz muito sobre as características e a inferência da demanda. O gráfico de importância, por outro lado, é perceptível alguns atributos que o modelo direcionou uma maior importância que, de fato, colaboram para a mudança do perfil de consumo ao longo do ano. Propriedades da unidade do mês, semana do ano, mínimo de temperatura, cobertura das nuvens e tantas outras relacionadas a unidade temporal e meteorológica, listadas nesse gráfico de importância, corroboram a afirmativa de que tais são influentes no alto e baixo consumo de energia ao longo do ano. Através dessa visualização, é perceptível que a prática desenvolvida durante o pré-processamento se alinha com a teoria fundamentada.

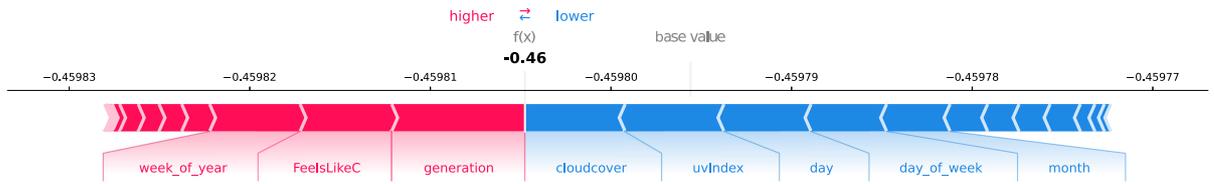
As Figuras 83 à 85, presentes no Apêndice B, mostram a importância utilizando os dados da *Pecan Street*. A Figura 86, também no Apêndice B, utiliza a base de dados da HUE como base de teste.

#### 5.4.1 Treinamento das características analisadas

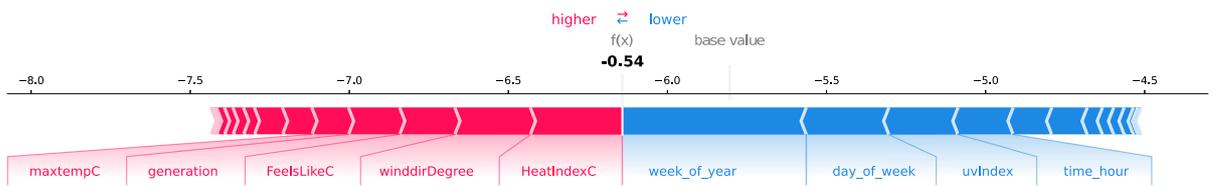
A ferramenta SHAP *Values* fornece uma coleção de novas propriedades vistas como mais importantes. Mesmo que seja tão direcionada a compreensão do comportamento dos modelos, até então “caixas-pretas”, esse novo conjunto pode ser utilizado novamente para verificar a diminuição do erro entre o real e a inferência.

Seguindo o método proposto de análise e comparação, a coleção de propriedades vistas como mais importantes por SHAP *Values* foi utilizada como entrada para os mesmos modelos, em todas as resoluções disponíveis. A Figura 43 demonstra a comparação

Figura 41 – Impacto que cada característica possui para a inferência de um único valor.

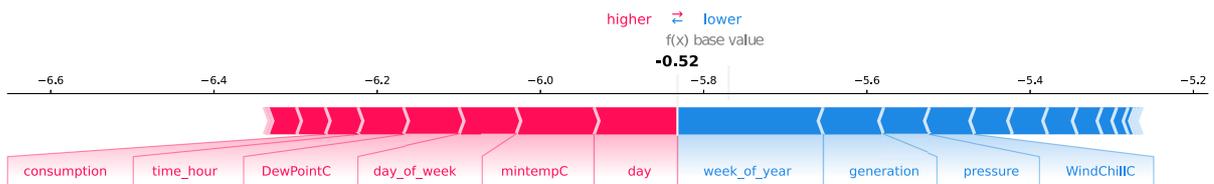


(a) RNN.



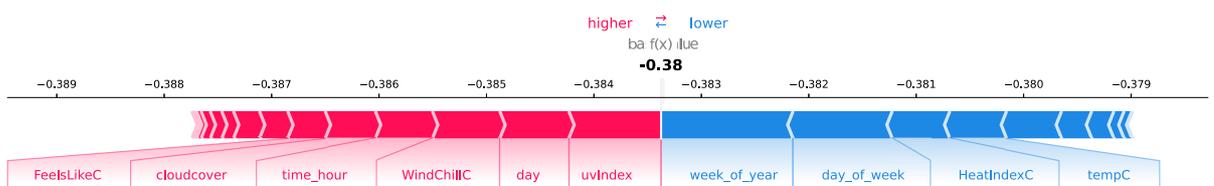
1e-7-5.38498e-1

(b) LSTM.



1e-5-5.165e-1

(c) GRU.



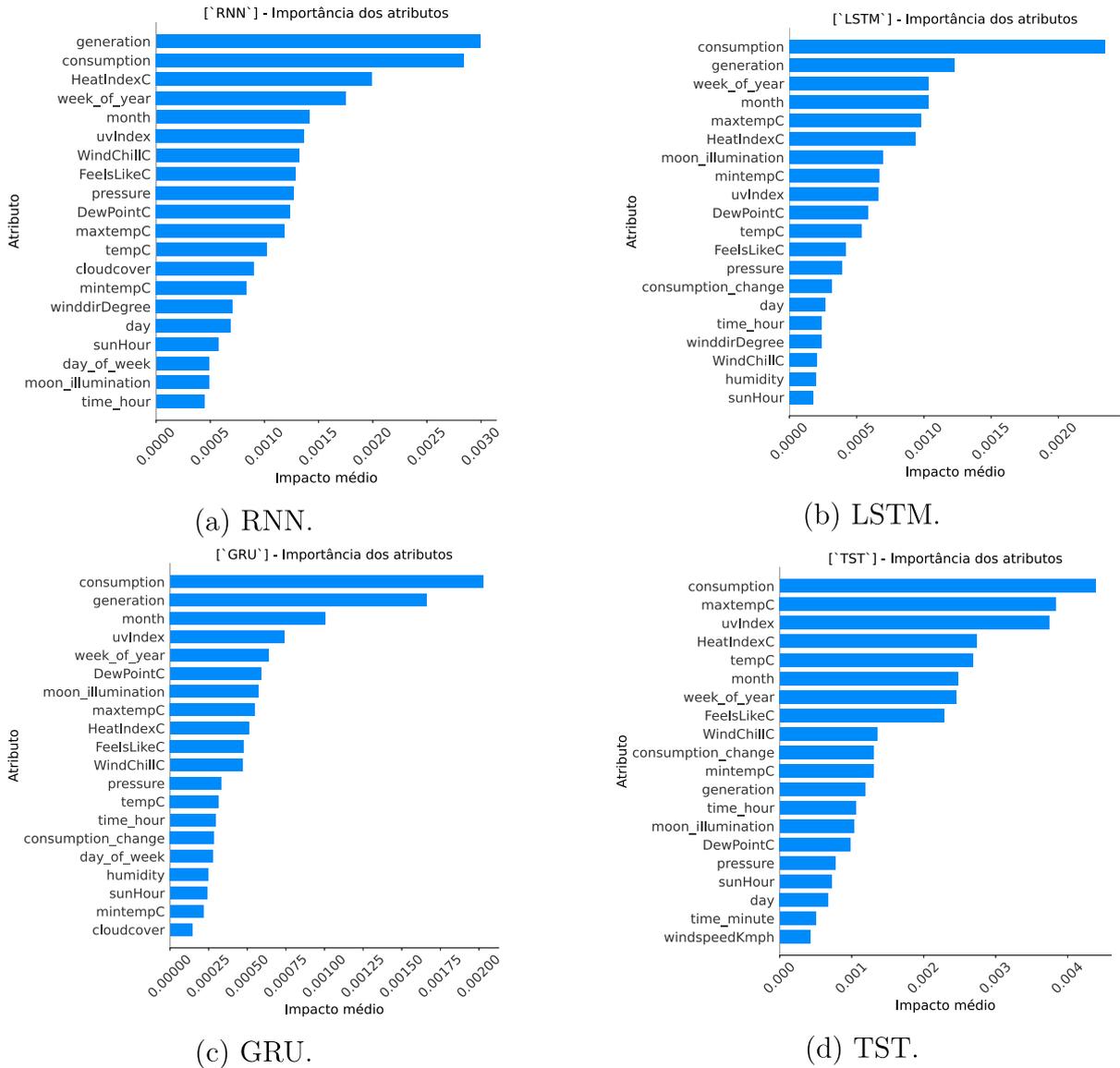
(d) TST.

Fonte: Elaborada pelo autor (2023).

do avanço da validação dos modelos tendo esse conjunto como destaque. É possível verificar algumas melhoras, principalmente, referindo-se ao modelo ResNet, pois, como demonstrado pela Figura 43h, a curva de erro em todas as resoluções possui menos flutuações se comparada com a Figura 39d, mesmo embora não tenha erros que definam a estabilização do modelo.

Antes que seja possível verificar o comportamento dos modelos treinados por essas

Figura 42 – Importância média das característica para a inferência dos valores de demanda utilizando os dados da *Pecan Street*, na resolução de 1 minuto.

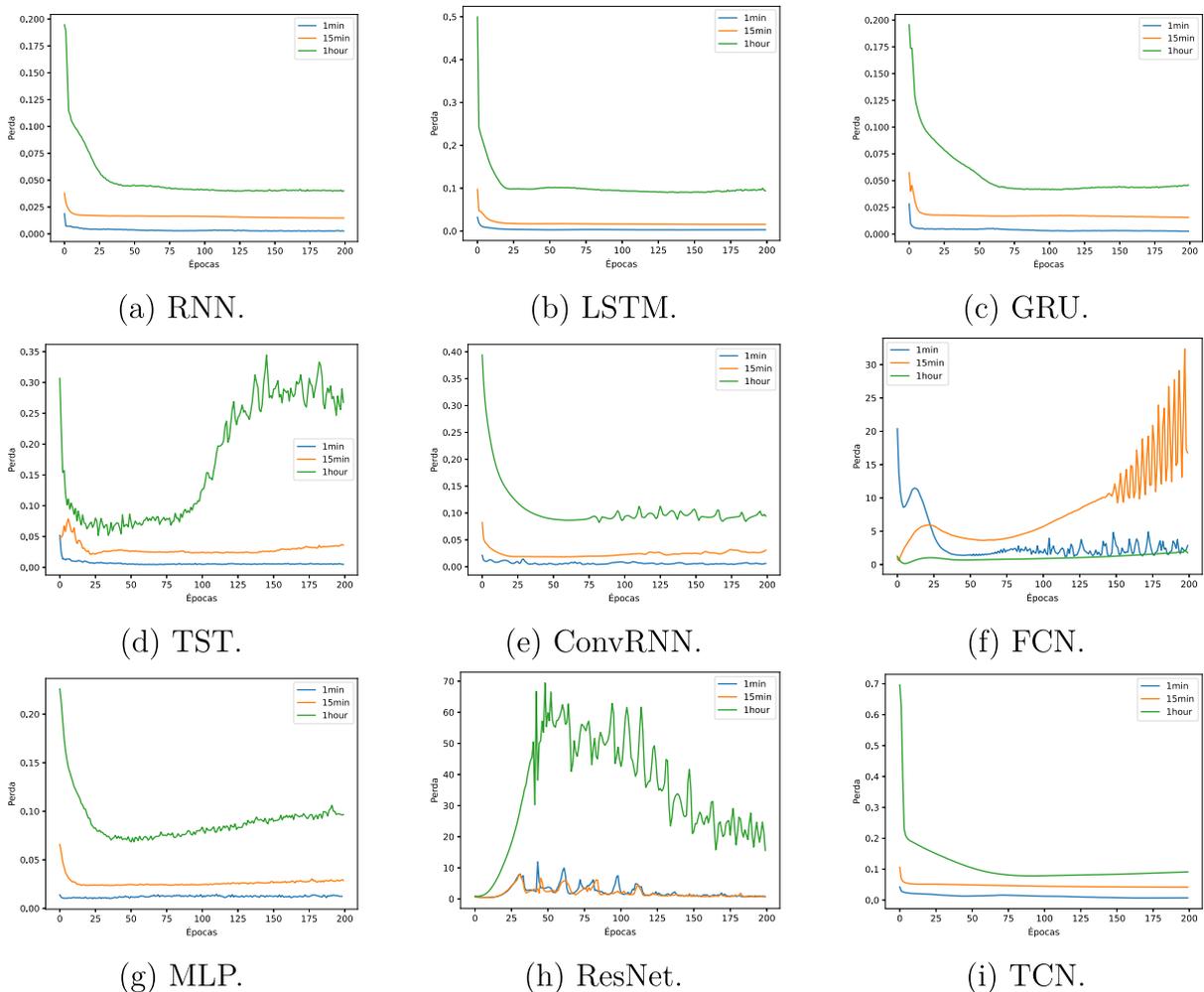


Fonte: Elaborada pelo autor (2023).

características analisadas por SHAP *Values*, é necessário verificar as inferências das mesmas amostras comparadas com os demais métodos. Assim, utilizando os dados da *Pecan Street* com os modelos recorrentes e TST, a Figura 44 insere as previsões por SHAP *Values*, a comparação do PCA e de todas as características previamente compiladas.

Todos os modelos que tiveram resultado satisfatórios antes, continuam tendo com as variáveis vistas por SHAP *Values*. Porém, sendo a etapa de validação que somente declara que o treinamento possui a robustez necessária, é preciso comparar as métricas atuais com as já fornecidas. Esse comportamento pode ser melhor visto entre o comparativo da utilização de todas as propriedades compiladas, as previamente reduzidas por PCA e, agora, reduzidas por SHAP *Values*, como demonstrado nas Figuras 45. As Tabelas 4, 5, 6

Figura 43 – Comparativo do avanço do erro de validação dos modelos utilizando as características relevantes encontradas por SHAP *Values*, utilizando os dados da *Pecan Street*.



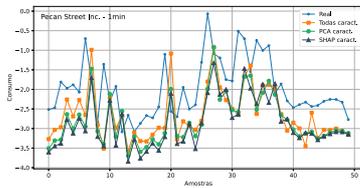
Fonte: Elaborada pelo autor (2023).

e 7, presentes no Apêndice A, demonstram melhor desempenho alcançada pelos modelos treinados.

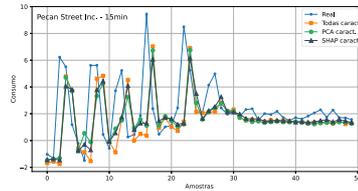
## 5.5 UTILIZAÇÃO DO ENSEMBLE POR SA

As métricas até agora abordadas trazem a acuracidade isolada dos modelos. A Figura 40 possibilitou a comparação entre elas, em uma escala global. Porém, como demonstrado no decorrer da apresentação dos resultados, cada um desses modelos possui sua particularidade de inferência, tendo propriedades de identificação de *spikes* e inércia na mudança de padrões de pico para estacionariedade, ou vice-versa. Outros modelos, que não sejam de DL, também ofertaram bons resultados, que também possuindo suas propriedades singulares.

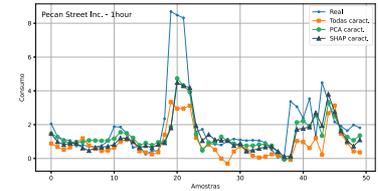
Figura 44 – Inferências realizadas utilizando características realçadas por SHAP *Values*, utilizando os dados da *Pecan Street* para os modelos recorrentes e *Transformer*.



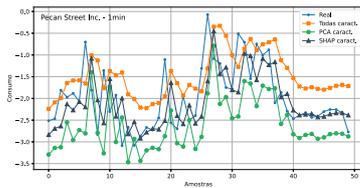
(a) Arquitetura RNN - Resolução de 1 minuto.



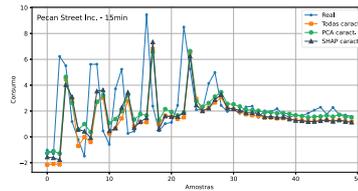
(b) Arquitetura RNN - Resolução de 15 minutos.



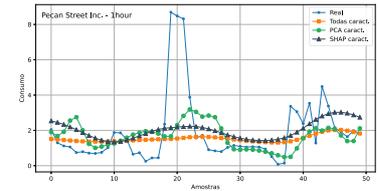
(c) Arquitetura RNN - Resolução horária.



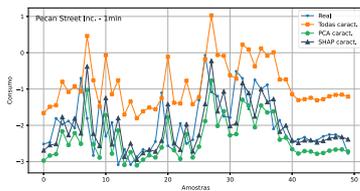
(d) Arquitetura LSTM - Resolução de 1 minuto.



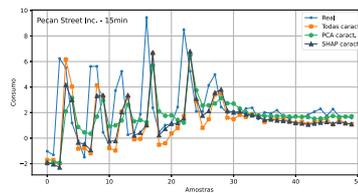
(e) Arquitetura LSTM - Resolução de 15 minutos.



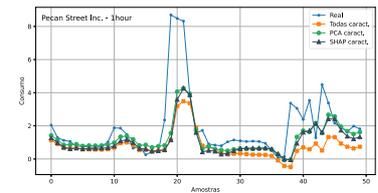
(f) Arquitetura LSTM - Resolução horária.



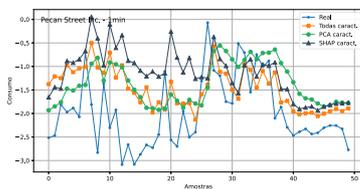
(g) Arquitetura GRU - Resolução de 1 minuto.



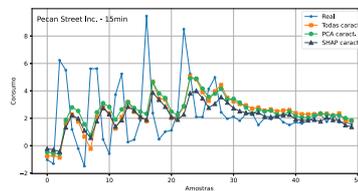
(h) Arquitetura GRU - Resolução de 15 minutos.



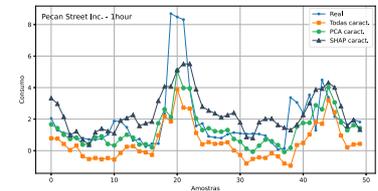
(i) Arquitetura GRU - Resolução de 1 hora.



(j) Arquitetura TST - Resolução de 1 minuto.



(k) Arquitetura TST - Resolução de 15 minutos.



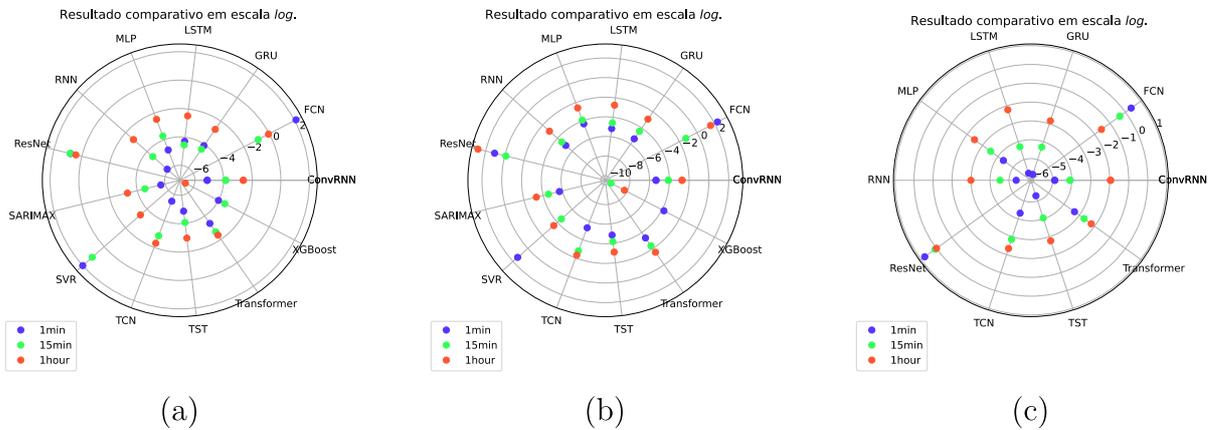
(l) Arquitetura TST - Resolução horária.

Fonte: Elaborada pelo autor (2023).

É possível, por meio do *ensemble*, agregar a inferência de todos esses modelos em uma única, fazendo com que ela comporte-se como a média das demais. Por meio da ponderação de pesos em que cada modelo influencia para a nova previsão, é possível fazer com que os piores modelos sejam os menos impactantes.

Tratando-se de um *ensemble* ponderado, feito após o teste e a inferência de cada um separadamente, é necessário distribuir os pesos de acordo com seu nível de erro. Realizar essa distribuição por meio de *grid search* provoca um enorme custo computacional por dois fatores: (i) são 9 modelos a serem combinados e (ii) é necessário uma alta precisão para a distribuição dos pesos.

Figura 45 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando (a) todas as características compiladas; (b) aquelas vistas por PCA; e (c) usando o método de análise por SHAP Values.



Fonte: Elaborada pelo autor (2023).

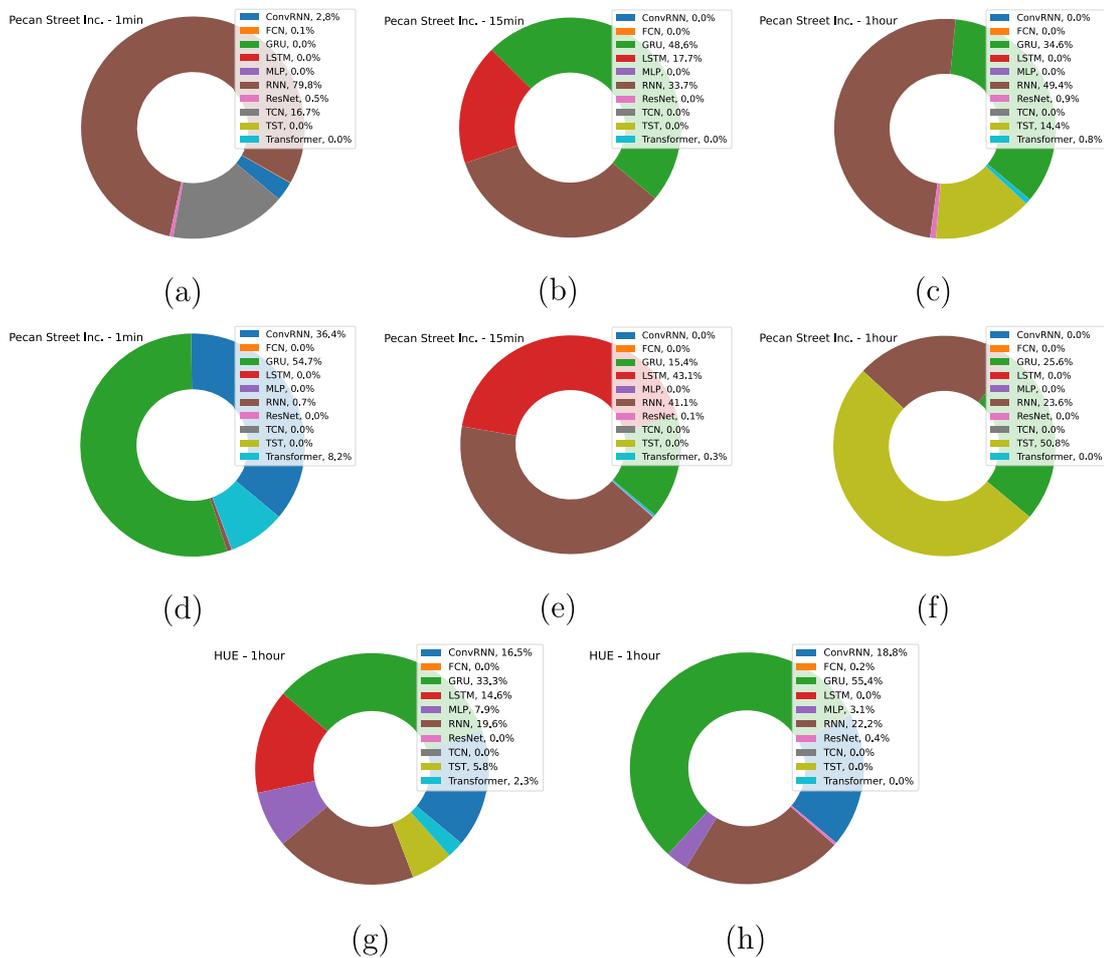
Para contornar esse problema e otimizar a busca por essa ponderação, o algoritmo *Simulated Annealing*. As inferências realizadas durante a validação guiarão a ponderação dos pesos que cada modelo possui para a inferência a ser realizada no conjunto de teste. Assim, a função MSE é usado como guia dos testes para a transferência dos pesos entre os modelos. Vale lembrar que, como trata-se de uma fusão tardia, o melhor *checkpoint* de cada modelo é escolhido, portanto a inferência para a tarefa conta com melhor desempenho de cada.

Tendo como base os dois tipos de características processadas, todas e por PCA, a Figura 46 demonstra os pesos finais ponderados durante a etapa de validação para todos os modelos de DL. É perceptível a participação de modelos recorrentes e de *Transformers* na maior parte dos tipos de dados utilizados, o que ressalta suas importâncias e as visualizações levantadas nas Seções de análise.

Através de cada ponderação de pesos, é possível realizar a inferência da pequena amostra para a verificação do quão próximo o *ensemble* está do real. Exemplificadamente, a Figura 47 mostra as inferências dos modelos recorrentes e baseado em *Transformers*, todos treinados isoladamente, em comparação com o *ensemble* por votação realizado, utilizando os dados da *Pecan Street*. Fica claro a melhor aproximação do real das inferências feitas por fusão tardia, pois o melhor de cada modelo está presente em um só modelo. A Figura 87 e Figura 88, ambas apresentadas no Apêndice B, demonstram os demais comparativos com os dados da *Pecan Street* e HUE.

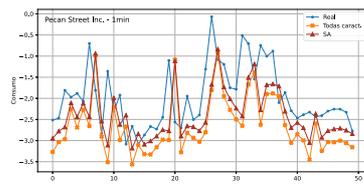
É possível, portanto, inserir a fusão tardia ao comparativo das métricas dos erros. Assim sendo, a Figura 48 mostra o gráfico de radar previamente apresentado, agora inserindo como parte comparativa os resultados por fusão tardia. As Tabelas 4, 5, 6 e 7, presentes no Apêndice A, demonstram o desempenho alcançado pelo *ensemble*.

Figura 46 – Ponderação de pesos para *Ensemble*. Os subgráficos (a) à (c) demonstram os dados da *Pecan Street* em sua originalidade. Os subgráfico (d) à (f) também mostram os dados da *Pecan Street*, mas utilizando o conjunto resumido por PCA. Gráfico (g) utiliza os dados da HUE originais e (h) os resumidos por PCA.

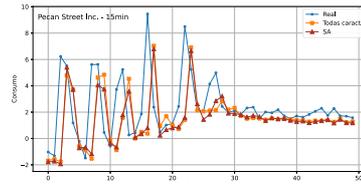


Fonte: Elaborada pelo autor (2023).

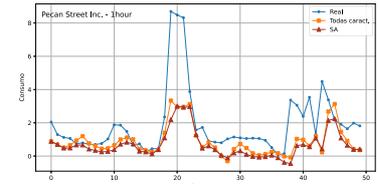
Figura 47 – Comparativo das inferências realizadas por modelos treinados isoladamente e pela fusão tardia por votação, otimizados por SA.



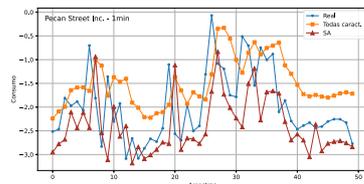
(a) Arquitetura RNN - Resolução de 1 minuto.



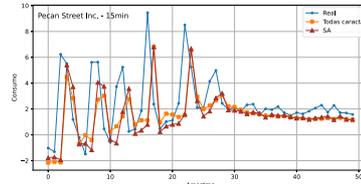
(b) Arquitetura RNN - Resolução de 15 minutos.



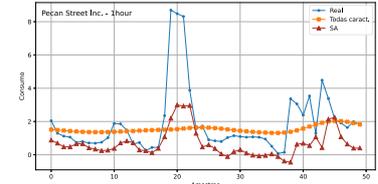
(c) Arquitetura RNN - Resolução horária.



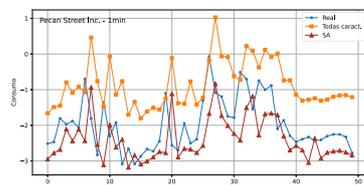
(d) Arquitetura LSTM - Resolução de 1 minuto.



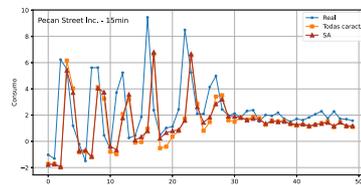
(e) Arquitetura LSTM - Resolução de 15 minutos.



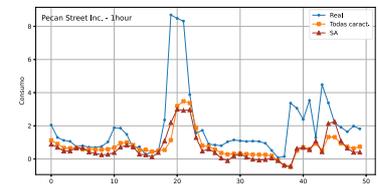
(f) Arquitetura LSTM - Resolução horária.



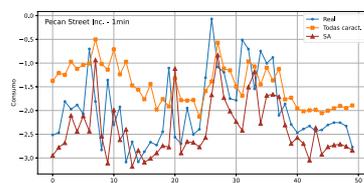
(g) Arquitetura GRU - Resolução de 1 minuto.



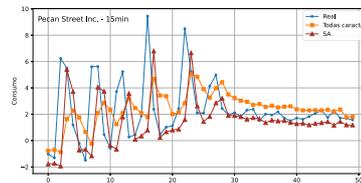
(h) Arquitetura GRU - Resolução de 15 minutos.



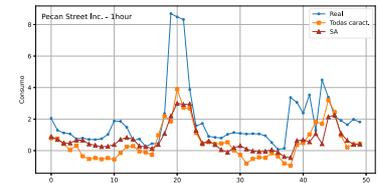
(i) Arquitetura GRU - Resolução de 1 hora.



(j) Arquitetura TST - Resolução de 1 minuto.



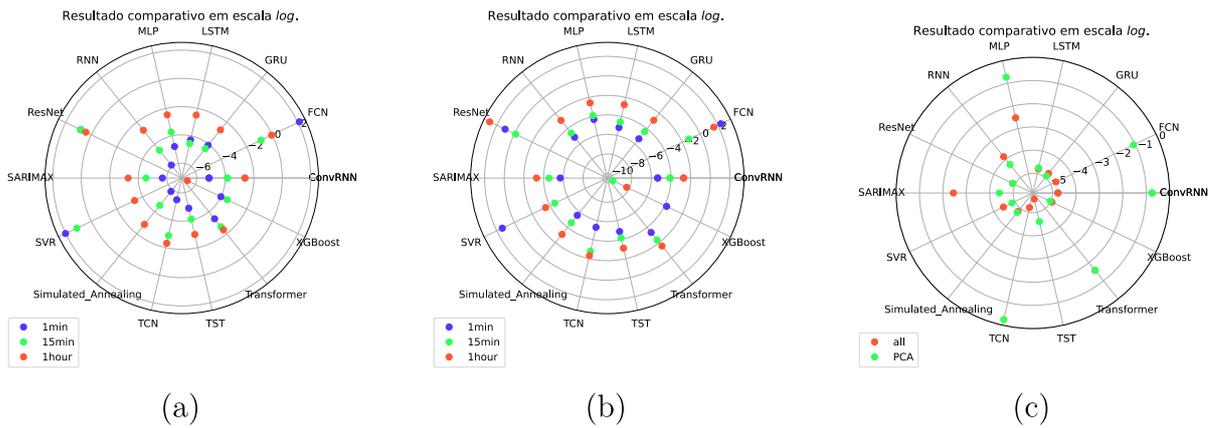
(k) Arquitetura TST - Resolução de 15 minutos.



(l) Arquitetura TST - Resolução horária.

Fonte: Elaborada pelo autor (2023).

Figura 48 – Comparativo das métricas, em escala logarítmica, de avaliação utilizando a fusão tardia otimizada por SA. As Figuras (a) e (b) demonstram o comparativo por resolução, enquanto a (c) a única resolução disponível na HUE.



Fonte: Elaborada pelo autor (2023).

## 6 CONCLUSÃO

O estudo comparativo levanta pontos de análise do comportamento, por treinamento e inferência, de modelos de aprendizado de máquina levando em consideração a coleção de atributos e a granularidade temporal. Aplicado a previsão multivariada de demanda energética à curto-prazo, é possível interpretar a real força que cada atributo possui para um modelo isolado. Estando no horizonte de curto-prazo e tratando o objeto de inferência influenciado por múltiplos atributos correlacionados, a previsão pode ser aplicada à temas de mercado de energia ou coordenabilidade do despacho de energia.

O método comparativo apresentado oferece suporte para o tratamento generalizado das bases que apresentem séries de demanda energética, pois conta com as principais ferramentas e formas de análise para o pré-processamento das amostras. Por meio do janelamento temporal, é possível utilizar os dados processados para inferir a demanda no horizonte de escolha, além de preparar as amostras na correta dimensionalidade dos *tensores* de entrada dos modelos apresentados, estando toda a implementação está publicamente disponível.

Estando o modelo estatístico no papel de *baseline* para a constatação do desempenho comparativo dos modelos recorrentes e TST, principais arquiteturas de DL estudadas, os resultados mostram bons resultados, ainda mais quando diversos atributos estão presentes para a inferência do objeto principal, a demanda energética. Possuindo o comparativo principal, o comportamento de métodos tradicionais com os em ascensão no SOTA, é necessário incluir outros modelos que, muito embora não possuam um grande destaque na pesquisa.

Além dos resultados obtidos pelos modelos primordiais, os modelos ConvRNN e TCN recebem o título de arquiteturas de ótimo desempenho. Possuindo a proposta de inclusão de camadas convolucionais, amplamente presentes em problemas de visão computacional, essas duas arquiteturas tiveram erros comparáveis ao recorrentes e baseados em *Transformers*.

Agora, analisando os estudos de caso de conjunto de atributos e granularidade, é possível observar, por meio das Tabelas 4 e 7, que em resoluções menores, como em 1 minuto da *Pecan Street*, reduzir os atributos por SHAP *Values* oferece um menor erro à inferência, resultado ocasionado provavelmente pelo respeito à particularidade do modelo e a presença de um conjunto muito grande de amostras. Analisando o aumento da granularidade temporal nesta mesma base de dados, percebe-se que os melhores modelos foram treinados com o conjunto separado por PCA. O conjunto de todas as características somente possuiu o menor erro comparativo quando a base horária da HUE é utilizada, provavelmente porque a quantidade de atributos é menor e essa redução provocou a piora do modelo na definição de padrões entre características.

Resultados significativo foram obtidos pela utilização da técnica de *ensemble* por votação. Sendo uma maneira simples e prática, por não haver a necessidade de treinar um novo modelo, a técnica agrega os melhores componentes de cada modelo. A técnica com o algoritmo meta-heurístico SA possibilitou uma ótima solução para melhor distribuir a importância de cada modelo.

## 6.1 TRABALHOS FUTUROS

Diversas são as abordagens possíveis para o problema de previsão de consumo de energia. A *string* de busca vista no Capítulo 2 possibilitou verificar trabalhos tratando de modelos DL em camadas convolucionais e recorrentes. Portanto, bons comparativos poderão ser obtidos utilizando modelos compostos, tais como CNN-LSTM.

Muito embora essas abordagens tragam ótimos resultados, o aprendizado por reforço pode ser um grande destaque, possivelmente uma ótima ferramenta para as transações *peer-to-peer* nos mercados descentralizados de energia. Referindo-se ao presente trabalho, uma ótima proposta é construir o *ensemble* abordado como um modelo a ser treinado e implementar o SA como uma função otimizadora. Essa proposta traria uma arquitetura totalmente nova e teria resultados compatíveis ao método de fusão tardia.

## REFERÊNCIAS

- AKIBA, Takuya; SANO, Shotaro; YANASE, Toshihiko; OHTA, Takeru; KOYAMA, Masanori. Optuna: a next-generation hyperparameter optimization framework. In: **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. New York, NY, USA: Association for Computing Machinery, 2019. p. 2623–2631.
- AN, Nguyen Hoang; ANH, Duong Tuan. Comparison of strategies for multi-step-ahead prediction of time series using neural network. In: **2015 International Conference on Advanced Computing and Applications (ACOMP)**. [S.l.: s.n.], 2015. p. 142–149.
- BAI, Shaojie; KOLTER, J. Zico; KOLTUN, Vladlen. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. **CoRR**, 2018.
- BAPTISTA, Marcia L.; GOEBEL, Kai; HENRIQUES, Elsa M.P. Relation between prognostics predictor evaluation metrics and local interpretability shap values. **Artificial Intelligence**, 2022.
- BELLMAN, Richard E. **Adaptive Control Processes: a guided tour**. [S.l.]: Princeton University Press, 2015.
- BENGIO, Yoshua; FRASCONI, Paolo; SIMARD, Patrice Y. The problem of learning long-term dependencies in recurrent networks. **IEEE International Conference on Neural Networks**, v. 3, p. 1183–1188, 1993.
- BLUM, Christian; ROLI, Andrea. Metaheuristics in combinatorial optimization: overview and conceptual comparison. **ACM Comput. Surv.**, v. 35, p. 268–308, 01 2001.
- BREIMAN, Leo. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.
- BREIMAN, Leo. Stacked regressions. **Machine learning**, Springer, v. 24, n. 1, p. 49–64, 1996.
- BRITO, André. S; BERNARDES, Marcelo. V; VILLELA, Saulo. M; TACON, Hemerson; CHAVES, Hugo. L; MAIA, Helena. A; CONCHA, Darwin. T; PEDRINI, Helio. Weighted voting of multi-stream convolutional neural networks for video-based action recognition using optical flow rhythms. **Journal of Visual Communication and Image Representation**, v. 77, p. 103–112, 2021.
- CHANG, Chih-Hao; CHEN, Zih-Bing; HUANG, Shih-Feng. Forecasting of high-resolution electricity consumption with stochastic climatic covariates via a functional time series approach. **Applied Energy**, v. 309, 2022.
- CHEN, Tairen; LEHR, Jane M.; LAVROVA, Olga; MARTÍNEZ-RAMÓN, Manel. Distribution feeder-level day-ahead peak load forecasting methods and comparative study. **IET Generation, Transmission & Distribution**, v. 12, n. 13, p. 3270–3278, 2018.
- CHOPRA, Sumit; AULI, Michael; RUSH, Alexander M. Abstractive sentence summarization with attentive recurrent neural networks. In: **Proceedings of the 2016 Conference of the North American Chapter of the Association for**

**Computational Linguistics: Human Language Technologies.** [S.l.]: Association for Computational Linguistics, 2016. p. 93–98.

DING, Ni; BÉSANGER, Yvon; WURTZ, Frédéric. Next-day MV/LV substation load forecaster using time series method. **Electric Power Systems Research**, v. 119, p. 345–354, 2015.

DRYAR, Henry A. The effect of weather on the system load. **Electrical Engineering**, v. 63, n. 12, p. 1006–1013, 1944.

DUBEY, Ashutosh Kumar; KUMAR, Abhishek; GARCÍA-DÍAZ, Vicente; SHARMA, Arpit Kumar; KANHAIYA, Kishan. Study and analysis of SARIMA and LSTM in forecasting time series data. **Sustainable Energy Technologies and Assessments**, v. 47, p. 101–474, 2021.

ELATTAR, Ehab Elsayed. Day-ahead price forecasting of electricity markets based on local informative vector machine. **IET Generation, Transmission & Distribution**, v. 7, n. 10, p. 1063–1071, 2013.

ELMAN, Jeffrey L. Finding structure in time. **Cognitive Science**, v. 14, n. 2, p. 179–211, 1990.

FERREIRA, Vitor. H. **Desenvolvimento de modelos neurais autônomos para previsão de carga elétrica.** Tese (Doutorado) — COPPE/UFRJ, 2008.

FOROOTANI, Ali; RASTEGAR, Mohammad; SAMI, Ashkan. Short-term individual residential load forecasting using an enhanced machine learning-based approach based on a feature engineering framework: a comparative study with deep learning methods. **Electric Power Systems Research**, v. 210, 2022.

GARCIA, Tania B. Lopez; MENDOZA, Alberto Coronado; NAVARRO, José A. Domínguez. Artificial neural networks in microgrids: a review. **Engineering Applications of Artificial Intelligence**, v. 95, 2020.

GHOFRANI, M.; HASSANZADEH, M.; ETEZADI-AMOLI, M.; FADALI, M. S. Smart meter based short-term load forecasting for residential customers. In: **2011 North American Power Symposium.** [S.l.: s.n.], 2011. p. 1–5.

GLOVER, Fred. Heuristics for integer programming using surrogate constraints. **Decision Sciences**, v. 8, n. 1, p. 156–166, 1977.

GOMES, Luis; MORAIS, Hugo; GONÇALVES, Calvin; GOMES, Eduardo; PEREIRA, Lucas; VALE, Zita. Impact of forecasting models errors in a peer-to-peer energy sharing market. **Energies**, v. 15, p. 3543, 05 2022.

GROSSBERG, Stephen. Adaptive pattern classification and universal recoding: II. feedback, expectation, olfaction, illusions. **Biol. Cybern.**, Springer-Verlag, v. 23, n. 4, p. 187—202, 1976.

HABEN, Stephen; ARORA, Siddharth; GIASEMIDIS, Georgios; VOSS, Marcus; Vukadinović Greetham, Danica. Review of low voltage load forecasting: methods, applications, and recommendations. **Applied Energy**, v. 304, 2021.

HADJOUT, D.; TORRES, J.F.; TRONCOSO, A.; SEBAA, A.; MARTÍNEZ-ÁLVAREZ, F. Electricity consumption forecasting based on ensemble deep learning with application to the algerian market. **Energy**, v. 243, p. 1–10, 2022.

HAYKIN, Simon. **Neural Networks: a comprehensive foundation**. 2nd. ed. [S.l.]: Prentice Hall PTR, 1998.

HE, Yan; WU, Pengcheng; LI, Yufeng; WANG, Yulin; TAO, Fei; WANG, Yan. A generic energy prediction model of machine tools using deep learning algorithms. **Applied Energy**, v. 275, p. 115402, 2020.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.

HONG, Tao; WANG, Pu. Artificial intelligence for load forecasting: history, illusions, and opportunities. **IEEE Power and Energy Magazine**, v. 20, n. 3, p. 14–23, 2022.

HOPFIELD, John J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982.

IENCO, Dino; INTERDONATO, Roberto. Deep semi-supervised clustering for multi-variate time-series. **Neurocomputing**, v. 516, p. 36–47, 2023.

KHALIL, Mohamad; MCGOUGH, A. Stephen; POURMIRZA, Zoya; PAZHOOHESH, Mehdi; WALKER, Sara. Machine learning, deep learning and statistical analysis for forecasting building energy consumption: a systematic review. **Engineering Applications of Artificial Intelligence**, v. 115, p. 105–287, 2022.

KINGMA, Diederik; BA, Jimmy. Adam: a method for stochastic optimization. **International Conference on Learning Representations**, 2014.

KOHONEN, Teuvo. Self-organized formation of topologically correct feature maps. **Biological cybernetics**, Springer, v. 43, n. 1, p. 59–69, 1982.

KOKHLIKYAN, Narine; MIGLANI, Vivek; MARTIN, Miguel; WANG, Edward; ALSALLAKH, Bilal; REYNOLDS, Jonathan; MELNIKOV, Alexander; KLIUSHKINA, Natalia; ARAYA, Carlos; YAN, Siqi; REBLITZ-RICHARDSON, Orion. Captum: a unified and generic model interpretability library for PyTorch. 2020.

KONG, Weicong; DONG, Zhao Yang; JIA, Youwei; HILL, David J.; XU, Yan; ZHANG, Yuan. Short-term residential load forecasting based on LSTM recurrent neural network. **IEEE Transactions on Smart Grid**, v. 10, n. 1, p. 841–851, 2019.

LIN, Weixuan; WU, Di; BOULET, Benoit. Spatial-temporal residential short-term load forecasting via graph neural networks. **IEEE Transactions on Smart Grid**, v. 12, n. 6, p. 5373–5384, 2021.

LIU, Cheng; WANG, Xiaofang. Quality-related english text classification based on recurrent neural network. **Journal of Visual Communication and Image Representation**, v. 71, 2020.

LUNDBERG, Scott M; LEE, Su-In. A unified approach to interpreting model predictions. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 30**. [S.l.]: Curran Associates, Inc., 2017. p. 4765–4774.

MAHESH, Konduri Praveen; AFROUZ, Shaik Ashar; AREECKAL, Anu Shaju. Detection of fraudulent credit card transactions: a comparative analysis of data sampling and classification techniques. **Journal of Physics: Conference Series**, IOP Publishing Ltd., v. 2161, n. 1, 2022.

MAKONIN, Stephen. **HUE: the hourly usage of energy dataset for buildings in british columbia**. [S.l.]: Harvard Dataverse, 2018. Acesso em: 2022-10-01.

MINSKY, Marvin; PAPER, Seymour. **Perceptrons: an introduction to computational geometry**. Cambridge, MA, USA: MIT Press, 1969.

NADEEM, Ahmad; ARSHAD, Naveed. A data-driven approach to reduce electricity theft in developing countries. **Utilities Policy**, v. 73, p. 101304, 2021.

OLAH, Christopher. **Understanding LSTM Networks**. 2015. Acesso: 2022-12-04. Disponível em: <<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>>.

OLIVEIRA, Nuno; SOUSA, Norberto; PRAÇA, Isabel. Deep learning for short-term instant energy consumption forecasting in the manufacturing sector. In: OMATU, Sigeru; MEHMOOD, Rashid; SITEK, Pawel; CICERONE, Serafino; RODRÍGUEZ, Sara (Ed.). **19th International Conference, Distributed Computing and Artificial Intelligence**. [S.l.]: Springer International Publishing, 2023. p. 165–175.

PATSAKOS, Ioannis; VROCHIDOU, Eleni; PAPAKOSTAS, George A. A survey on deep learning for building load forecasting. **Mathematical Problems in Engineering**, v. 2022, 2022.

PETROPOULOS, Fotios; APILETTI, Daniele; ASSIMAKOPOULOS, Vassilios; BABAI, Mohamed Zied. Forecasting: theory and practice. **International Journal of Forecasting**, v. 38, n. 3, p. 705–871, 2022.

PROEDROU, Elisavet. A comprehensive review of residential electricity load profile models. **IEEE Access**, v. 9, p. 12114–12133, 2021. ISSN 2169-3536.

RAFFEL, Colin; SHAZEER, Noam; ROBERTS, Adam; LEE, Katherine; NARANG, Sharan; MATENA, Michael; ZHOU, Yanqi; LI, Wei; LIU, Peter J. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, v. 21, n. 1, 2020.

RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos. Why should i trust you?: explaining the predictions of any classifier. In: . New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1135–1144.

RICE, Rowan; NORTH, Kristina; HANSEN, Geoffrey; PEARSON, Drew; SCHAER, Oliver; SHERMAN, Thomas; VASSALLO, Daniel. Time-series forecasting energy loads: a case study in texas. In: **2022 Systems and Information Engineering Design Symposium (SIEDS)**. [S.l.: s.n.], 2022. p. 196–201.

- ROSENBLATT, F. **The perceptron, a perceiving and recognizing Automaton Project Para.** [S.l.]: Cornell Aeronautical Laboratory, 1957. (Report: Cornell Aeronautical Laboratory).
- RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning internal representations by error propagation. In: RUMELHART, David E.; MCCLELLAND, James L. (Ed.). **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.** [S.l.]: MIT Press, 1986. p. 318–362.
- SANTOS, José A. F. A.; JONG, Pieter de; C., Caiuby A.; TORRES, Ednildo A. Combining wind and solar energy sources: Potential for hybrid power generation in Brazil. **Utilities Policy**, v. 67, 2020.
- SCHAPIRE, Robert E. The strength of weak learnability. **Machine learning**, Springer, v. 5, n. 2, p. 197–227, 1990.
- SEPULVEDA, Luis F.M.; DINIZ, Petterson S.; DINIZ, João O.B.; NETTO, Stelmo M.B.; CIPRIANO, Carolina L.S.; ARAÚJO, Alexandre C.; LEMOS, Victor H.B.; PESSOA, Alexandre C.P.; QUINTANILHA, Darlan B.P.; ALMEIDA, João D.S.; SILVA, Aristófanés C.; PAIVA, Anselmo C.; BRAZ, Geraldo; SILVA, Márcia I.A.; MONTEIRO, Eliana M.G.; SILVA, Italo F.S.; FERNANDES, Eduardo C. Forecasting of individual electricity consumption using optimized gradient boosting regression with modified particle swarm optimization. **Engineering Applications of Artificial Intelligence**, v. 105, 2021.
- SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas: fundamentos teóricos e aspectos práticos.** 2. ed. [S.l.]: ArtLiber, 2016. v. 1.
- SOUSA, João C.; JORGE, Humberto M.; NEVES, Luís P. Short-term load forecasting based on support vector regression and load profiling. **International Journal of Energy Research**, v. 38, n. 3, p. 350–362, 2014.
- STREET, Pecan. **Pecan Street Dataport.** 2022. Acesso em: 2022-10-01. Disponível em: <<https://dataport.pecanstreet.org>>.
- STRIELKOWSKI, Wadim. Chapter 6 - consumers, prosumers, and the smart grids. In: STRIELKOWSKI, Wadim (Ed.). **Social Impacts of Smart Grids.** [S.l.]: Elsevier, 2020. p. 191–238.
- TAGHAVI, Zahra Sadat; SAJEDI, Hedieh. Ensemble selection using simulated annealing walking. In: **2014 International Conference on Advances In Computing, Electronics and Electrical Technology (CEET).** [S.l.: s.n.], 2014.
- TIAN, Zhenya; XIAO, Jialiang; FENG, Haonan; WEI, Yutian. Credit risk assessment based on gradient boosting decision tree. **Procedia Computer Science**, v. 174, p. 150–160, 2020.
- UNWIN, Antony. Multivariate outliers and the o3 plot. **Journal of Computational and Graphical Statistics**, Taylor & Francis, v. 28, n. 3, p. 635–643, 2019.

USAMA, Mohd; AHMAD, Belal; SONG, Enmin; HOSSAIN, M. Shamim; ALRASHOUD, Mubarak; MUHAMMAD, Ghulam. Attention-based sentiment analysis using convolutional and recurrent neural network. **Future Generation Computer Systems**, v. 113, p. 571–578, 2020.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N; KAISER, Ł ukasz; POLOSUKHIN, Illia. Attention is all you need. In: GUYON, I.; LUXBURG, U. Von; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2017. v. 30.

VILLELA, Saulo Moraes; CARVALHO, Luís Alfredo Vidal. **O Resfriamento Simulado no Projeto Ótimo de Autômatos Celulares para a Geração de Chaves em Criptografia de Fluxo**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro (UFRJ), 2006.

VOß, Marcus; BENDER-SAEBELKAMPF, Christian; ALBAYRAK, Sahin. Residential short-term load forecasting using convolutional neural networks. In: **2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)**. [S.l.: s.n.], 2018. p. 1–6.

WANG, Yuanyuan; SUN, Shanfeng; CHEN, Xiaoqiao; ZENG, Xiangjun; KONG, Yang; CHEN, Jun; GUO, Yongsheng; WANG, Tingyuan. Short-term load forecasting of industrial customers based on SVM and XGBoost. **International Journal of Electrical Power & Energy Systems**, v. 129, 2021.

WANG, Zhiguang; YAN, Weizhong; OATES, Tim. Time series classification from scratch with deep neural networks: a strong baseline. **2017 International Joint Conference on Neural Networks (IJCNN)**, p. 1578–1585, 2016.

WERBOS, Paul; JOHN, Paul. Beyond regression : new tools for prediction and analysis in the behavioral sciences. In: . [S.l.: s.n.], 1974.

ZENG, Xian; HU, Yaoqin; SHU, Liqi; LI, Jianhua; DUAN, Huilong; SHU, Qiang; LI, Haomin. Explainable machine-learning predictions for complications after pediatric congenital heart surgery. **Scientific Reports**, v. 11, 2021.

ZERVEAS, George; JAYARAMAN, Srideepika; PATEL, Dhaval; BHAMIDIPATY, Anuradha; EICKHOFF, Carsten. A transformer-based framework for multivariate time series representation learning. In: **Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery; Data Mining**. New York, NY, USA: Association for Computing Machinery, 2021. (KDD '21), p. 2114–2124.

ZHANG, Yang; DENG, Caibo; ZHAO, Ran; LETO, Sebastian. A novel integrated price and load forecasting method in smart grid environment based on multi-level structure. **Engineering Applications of Artificial Intelligence**, v. 95, p. 103852, 2020.

## APÊNDICE A – MÉTRICAS COMPLETAS DE PERFORMANCE

As Tabelas 4, 5, 6 e 7, presentes neste Apêndice, demonstram as métricas performáticas de todos os modelos, respectivamente correspondentes ao MSE, MAE, MAPE e R2 *Score*. As granularidades e as fontes de dados estão explícitas, estando em negrito os melhores valores alcançados em cada métrica. Fica perceptível, principalmente pela Tabela 4, que o conjunto de dados apartados por PCA tiveram um bom desempenho em resoluções de tempo em que não havia muitas amostras. O conjunto separado por SHAP *Values* teve destaque na resolução de 1 minuto da *Pecan Street*. Importante ressaltar que os valores obtidos na avaliação dos modelos foram baseados na melhor época de treinamento, mesmo que tenha ocorrido *overfitting*, como no modelo TST e a resolução horária da *Pecan Street*.

Tabela 4 – MSE alcançado pelos modelos empregados nas variações de dados.

Métrica	MSE											
Dataset	Pecan Street									HUE		
Resolução	1min			15min			1hour			1hour		
Atributos	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP
RNN	<b>0.00279</b>	0.00595	0.00461	0.01065	<b>0.01025</b>	0.01080	0.06524	0.05395	<b>0.05022</b>	0.00847	<b>0.00737</b>	0.00802
LSTM	0.01372	0.00571	<b>0.00315</b>	0.01060	<b>0.01025</b>	0.01388	0.08315	<b>0.06360</b>	0.10878	<b>0.00866</b>	0.00879	0.00876
GRU	0.01676	0.00483	<b>0.00295</b>	<b>0.01231</b>	0.01235	0.01358	0.06731	<b>0.05515</b>	0.05887	0.00844	<b>0.00748</b>	0.00760
TST	0.00771	0.00785	<b>0.00505</b>	0.01712	<b>0.01537</b>	0.01744	0.05204	<b>0.04512</b>	0.06251	<b>0.00798</b>	0.01207	0.00964
ConvRNN	0.00584	<b>0.00474</b>	0.00747	0.02124	<b>0.01657</b>	0.01671	0.07268	<b>0.06644</b>	0.14099	<b>0.00755</b>	0.00776	0.01276
MLP	<b>0.00847</b>	0.01333	0.01274	0.02432	<b>0.02016</b>	0.02914	0.08480	<b>0.07605</b>	0.08350	<b>0.00833</b>	0.00851	0.00952
FCN	8.25760	10.3296	<b>1.45320</b>	0.41447	<b>0.27995</b>	0.69230	0.94316	4.68446	<b>0.21022</b>	<b>0.08018</b>	0.48447	0.15399
ResNet	2.23833	2.89366	<b>2.13189</b>	2.31748	<b>0.89168</b>	1.08180	1.53051	16.41117	<b>0.98816</b>	<b>0.08627</b>	0.77796	0.27713
TCN	<b>0.00417</b>	0.00501	0.01332	<b>0.05626</b>	0.06188	0.05751	0.09842	0.09936	<b>0.09582</b>	0.01166	<b>0.01027</b>	0.01641
<i>Ensemble</i>	<b>0.00289</b>	0.00376		0.01024	<b>0.00999</b>		0.05753	<b>0.04566</b>		0.00714	<b>0.00656</b>	
SVR	7.42879	<b>3.96012</b>	-	3.08206	<b>0.01072</b>	-	0.03403	<b>0.03004</b>	-	<b>0.00551</b>	0.47276	-
XGBoost	<b>0.01823</b>	0.02271	-	<b>0.02981</b>	<b>0.05983</b>	-	0.00133	<b>0.00025</b>	-	<b>0.00377</b>	0.34326	-
SARIMAX	<b>0.00337</b>	-	-	<b>0.01059</b>	-	-	<b>0.03733</b>	-	-	<b>0.20541</b>	-	-

Fonte: Elaborada pelo autor (2023).

Tabela 5 – MAE alcançado pelos modelos empregados nas variações de dados.

Métrica	MAE											
Dataset	Pecan Street									HUE		
Resolução	1min			15min			1hour			1hour		
Atributos	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP
RNN	<b>0.02650</b>	0.06192	0.05041	0.05967	<b>0.05339</b>	0.05454	0.16139	<b>0.12345</b>	0.13206	0.05835	<b>0.05107</b>	0.05552
LSTM	0.10407	0.05833	<b>0.02353</b>	0.05701	<b>0.05430</b>	0.06807	0.19378	<b>0.16430</b>	0.25806	0.05781	<b>0.05603</b>	0.05982
GRU	0.12142	0.05074	<b>0.02573</b>	0.06728	<b>0.06519</b>	0.07016	0.13235	<b>0.12082</b>	0.12571	0.06372	<b>0.05528</b>	0.05679
TST	0.06021	0.06627	<b>0.04091</b>	0.09608	<b>0.08182</b>	0.08792	0.14368	<b>0.13829</b>	0.18738	<b>0.05451</b>	0.07641	0.06423
ConvRNN	0.05607	<b>0.04361</b>	0.06179	0.09559	<b>0.08232</b>	0.08900	0.17565	<b>0.17111</b>	0.28312	<b>0.05987</b>	0.06305	0.08548
MLP	0.06581	0.07008	<b>0.06483</b>	0.10038	<b>0.09819</b>	0.10875	0.20491	<b>0.17724</b>	0.19723	0.05359	<b>0.05277</b>	0.05961
FCN	2.46375	2.72781	<b>0.93816</b>	0.51989	<b>0.43285</b>	0.68219	0.76949	1.70751	<b>0.38178</b>	<b>0.22906</b>	0.60603	0.35249
ResNet	<b>1.09798</b>	1.34136	1.18267	1.11981	<b>0.75079</b>	0.84797	0.97476	3.70548	<b>0.79325</b>	<b>0.22062</b>	0.63251	0.41432
TCN	<b>0.03321</b>	0.03708	0.09722	0.16833	0.20496	<b>0.16348</b>	0.20806	0.23536	<b>0.18053</b>	0.06936	<b>0.06448</b>	0.09441
<i>Ensemble</i>	<b>0.02652</b>	0.03592		0.05358	<b>0.05258</b>		0.13363	<b>0.12349</b>		0.05145	<b>0.04874</b>	
SVR	2.36098	<b>1.53873</b>	-	1.43284	<b>0.09158</b>	-	<b>0.17547</b>	0.16271	-	0.04510	<b>0.45296</b>	-
XGBoost	<b>0.06970</b>	0.07683	-	0.11077	<b>0.00467</b>	-	0.02385	<b>0.01094</b>	-	0.04146	<b>0.35482</b>	-
SARIMAX	<b>0.02221</b>	-	-	<b>0.05553</b>	-	-	<b>0.10968</b>	-	-	<b>0.26312</b>	-	-

Fonte: Elaborada pelo autor (2023).

Tabela 6 – MAPE alcançado pelos modelos empregados nas variações de dados.

Métrica	MAPE											
Dataset	Pecan Street									HUE		
Resolução	1min			15min			1hour			1hour		
Atributos	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP
RNN	<b>0.14220</b>	0.30428	0.21210	0.92183	<b>0.81904</b>	0.90441	4.88546	<b>3.79263</b>	4.11293	0.09153	<b>0.08186</b>	0.08780
LSTM	0.48304	0.28010	<b>0.17468</b>	1.00041	<b>0.81856</b>	0.99095	4.33073	4.20825	<b>3.74409</b>	0.09147	<b>0.08986</b>	0.09359
GRU	0.53133	0.23991	<b>0.16368</b>	<b>0.91839</b>	1.06218	1.06875	4.35518	4.21207	<b>4.20590</b>	0.09654	<b>0.08631</b>	0.08784
TST	0.34994	0.29644	<b>0.27696</b>	1.00007	<b>0.87233</b>	0.96062	4.56583	4.59006	<b>4.25748</b>	<b>0.08580</b>	0.08631	0.09964
ConvRNN	<b>0.22020</b>	0.25833	0.28317	1.37777	1.02070	<b>1.00774</b>	5.13323	<b>4.70171</b>	6.05600	<b>0.08850</b>	0.09114	0.53038
MLP	<b>0.26471</b>	0.44704	0.47655	<b>1.24312</b>	1.14281	1.43518	5.38558	5.71011	<b>5.36075</b>	0.08622	<b>0.08589</b>	0.09459
FCN	<b>8.73249</b>	8.90999	3.97944	3.23451	<b>1.50761</b>	4.25863	11.02001	17.97226	<b>6.95738</b>	<b>0.29699</b>	0.76546	0.43055
ResNet	<b>3.36829</b>	4.12096	5.85828	6.37581	<b>3.80967</b>	6.07990	4.32818	42.33065	<b>3.73976</b>	<b>0.28268</b>	0.77889	0.53038
TCN	<b>0.21167</b>	0.27051	0.53340	1.85153	1.26917	<b>2.110639</b>	4.56645	<b>4.16572</b>	4.88300	0.10587	<b>0.10071</b>	0.14056
<i>Ensemble</i>	<b>0.15628</b>	0.23502		0.89930	<b>0.81167</b>		4.61107	<b>4.29652</b>		0.08183	<b>0.07766</b>	
SVR	9.44577	<b>5.78149</b>	-	6.31611	<b>0.36384</b>	-	<b>0.79564</b>	0.79641	-	0.76676	<b>0.37610</b>	-
XGBoost	<b>0.95270</b>	1.08521	-	1.19227	<b>0.03143</b>	-	0.40294	<b>0.09572</b>	-	<b>0.10496</b>	0.46293	-
SARIMAX	<b>0.15100</b>	-	-	<b>0.92641</b>	-	-	<b>3.39812</b>	-	-	<b>0.26070</b>	-	-

Fonte: Elaborada pelo autor (2023).

Tabela 7 – R2 Score alcançado pelos modelos empregados nas variações de dados.

Métrica	R2 Score											
Dataset	Pecan Street									HUE		
Resolução	1min			15min			1hour			1hour		
Atributos	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP	Todas	PCA	SHAP
RNN	<b>0.91983</b>	0.84811	0.88221	0.81609	<b>0.82299</b>	0.81351	0.20192	0.33999	<b>0.38558</b>	0.13196	<b>0.24420</b>	0.17742
LSTM	0.68827	0.85417	<b>0.91949</b>	0.81702	<b>0.82298</b>	0.76050	-0.01724	<b>0.22197</b>	-0.33072	<b>0.11176</b>	0.09883	0.10221
GRU	0.59062	0.87661	<b>0.92451</b>	<b>0.78781</b>	0.78684	0.76568	0.17656	<b>0.32530</b>	0.27976	0.13425	<b>0.23313</b>	0.22091
TST	0.83398	0.79954	<b>0.87104</b>	0.70449	<b>0.73470</b>	0.69895	0.36334	<b>0.44801</b>	0.23541	<b>0.18152</b>	-0.23784	-0.30817
ConvRNN	0.81516	<b>0.87894</b>	0.80934	0.63349	<b>0.71427</b>	0.71152	0.11085	<b>0.18772</b>	-0.72470	<b>0.22580</b>	0.20413	-0.30817
MLP	<b>0.77968</b>	0.45713	0.67474	0.58034	<b>0.65212</b>	0.49707	-0.03074	<b>0.06963</b>	-0.02154	<b>0.14578</b>	0.12751	0.02395
FCN	-251.65753	-262.63471	<b>-36.08856</b>	-6.15132	<b>-3.83041</b>	-10.94495	-10.53742	-56.30353	<b>-1.57157</b>	<b>-7.21670</b>	-48.64789	-14.78137
ResNet	<b>-77.99939</b>	-72.85197	-53.40999	-38.98575	<b>-14.38502</b>	-17.66547	-17.72228	-199.75251	<b>-11.08786</b>	<b>-7.74121</b>	-78.72427	-27.39951
TCN	<b>0.88122</b>	0.87207	0.65979	<b>0.02914</b>	-0.06783	0.00780	-0.20402	<b>-0.12550</b>	-0.17217	-0.19586	<b>0.05263</b>	-0.68257
<i>Ensemble</i>	<b>0.92601</b>	0.90394		0.82324	<b>0.82756</b>		0.29615	<b>0.44145</b>		0.26783	<b>0.32702</b>	
SVR	-189.19805	<b>-100.39026</b>	-	-52.69748	<b>0.81307</b>	-	0.57062	<b>0.62091</b>	-	<b>0.98515</b>	-0.27461	-
XGBoost	<b>0.53312</b>	0.41844	-	0.48049	<b>0.99907</b>	-	0.98317	<b>0.99683</b>	-	<b>0.98983</b>	0.07454	-
SARIMAX	<b>0.91347</b>	-	-	<b>0.81527</b>	-	-	<b>0.52844</b>	-	-	<b>0.44619</b>	-	-

Fonte: Elaborada pelo autor (2023).

## APÊNDICE B – IMAGENS E RESULTADOS COMPLEMENTARES

Além de o Algoritmo 10 demonstrar o processo de construção da janela temporal, o Apêndice demonstra os gráficos *boxplots* dos atributos, tendo as Figuras 49 e 50 correspondentes ao banco de dados da *Pecan Street*. As Figuras 51, 52 e 53 mostram o mapa de calor da correlação entre número de componentes do PCA com os atributos quando a base de dados da *Pecan Street* é utilizada. Para a base da HUE, o mesmo pode ser visto pela Figura 54. A correlação entre atributos pode ser analisada pelo mapa de calor demonstrado pela Figura 55, 56 e 58 quando os dados da *Pecan Street* é usado. A Figura 58 demonstra a correlação dos atributos presentes na base da HUE.

Tendo como base os modelos treinados para a inferência do consumo de energia, a Figura 60 detalha a previsão de 50 amostras do conjunto de teste do *dataset* da HUE. O treinamento dos modelos recorrentes utilizando a base da *Pecan Street* e HUE pode ser analisado pela Figure 61. Como o comparativo se estende ao modelo TST e outros referenciados no SOTA, as Figuras 62, 63, 64, 65, 66, 67, 68, 69, 70, 71 demonstram o avanço da rede para as duas bases treinadas. As inferências de 50 amostras do conjunto de teste para esses modelos pode ser vista pelas Figuras 72, 73, 74 e 75.

Tratando da análise por SHAP *Value*, as Figuras 76, 77, 78, 79, 80, 81 e 82 mostram o impacto das características para cada modelo treinado. Já as Figuras 83, 84, 85, 86 mostram a importância desses atributos quando a ferramenta de interpretabilidade é utilizada.

As inferências das 50 amostras do conjunto de teste utilizando a fusão tardia por votação otimizada por SA podem ser analisadas pelas Figuras 87 e 88. Como as variáveis apartadas por SHAP *Values* também foram utilizadas para o treinamento dos modelos, as Figuras 89 e 90 mostram as previsões conquistadas.

---

**Algoritmo 1:** Construção de sequências *sliding windows*.

---

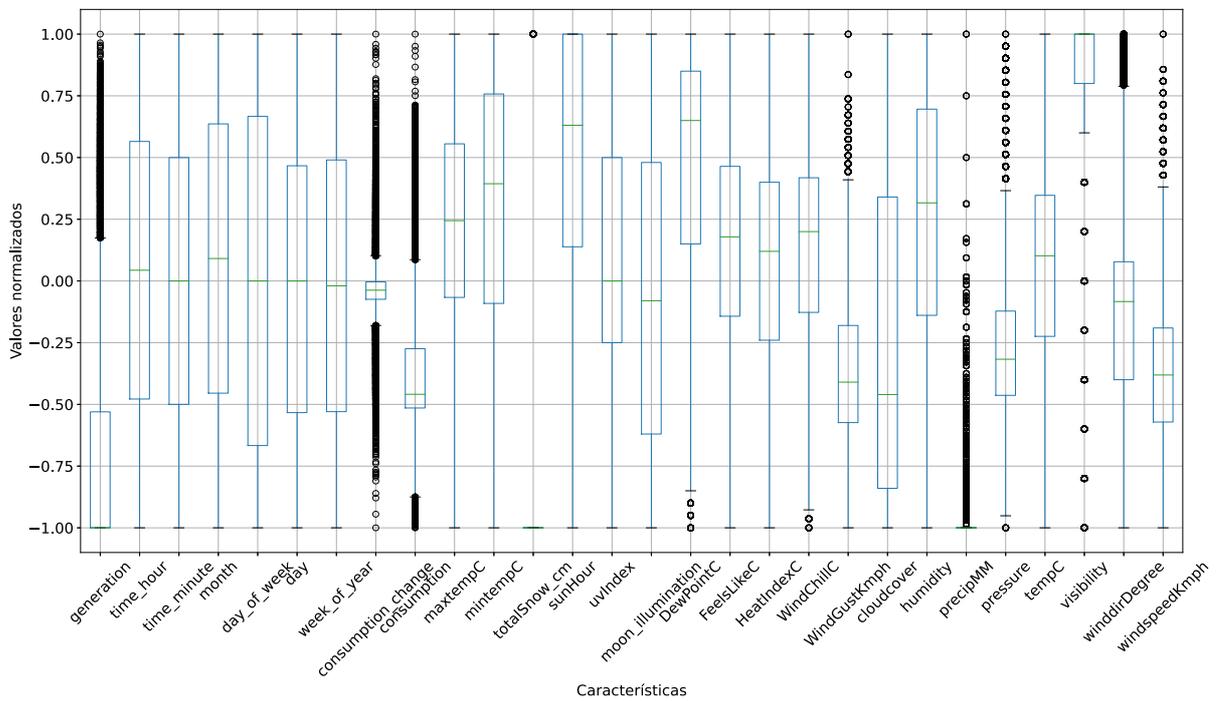
**Entrada:** Vetor de dados,  $(\alpha)$ ;  
 Horizonte temporal de entrada,  $(j)$ ;  
 Horizonte temporal de saída,  $(k)$ ;  
**Saída:** Vetor de dados,  $(\beta)$

```

1 início
2    $\beta = []$ ;
3   para  $i$  no intervalo de tamanho $(\alpha) - j$  faça
4      $label_p = i + j$ ;
5      $g = \alpha[i, label_p]$ ;
6      $h = \alpha[label_p, k]$ ;
7      $\beta \leftarrow \beta + (g, h)$ ;
8   fim para
9   Retorna  $\beta$ ;
10 fim
```

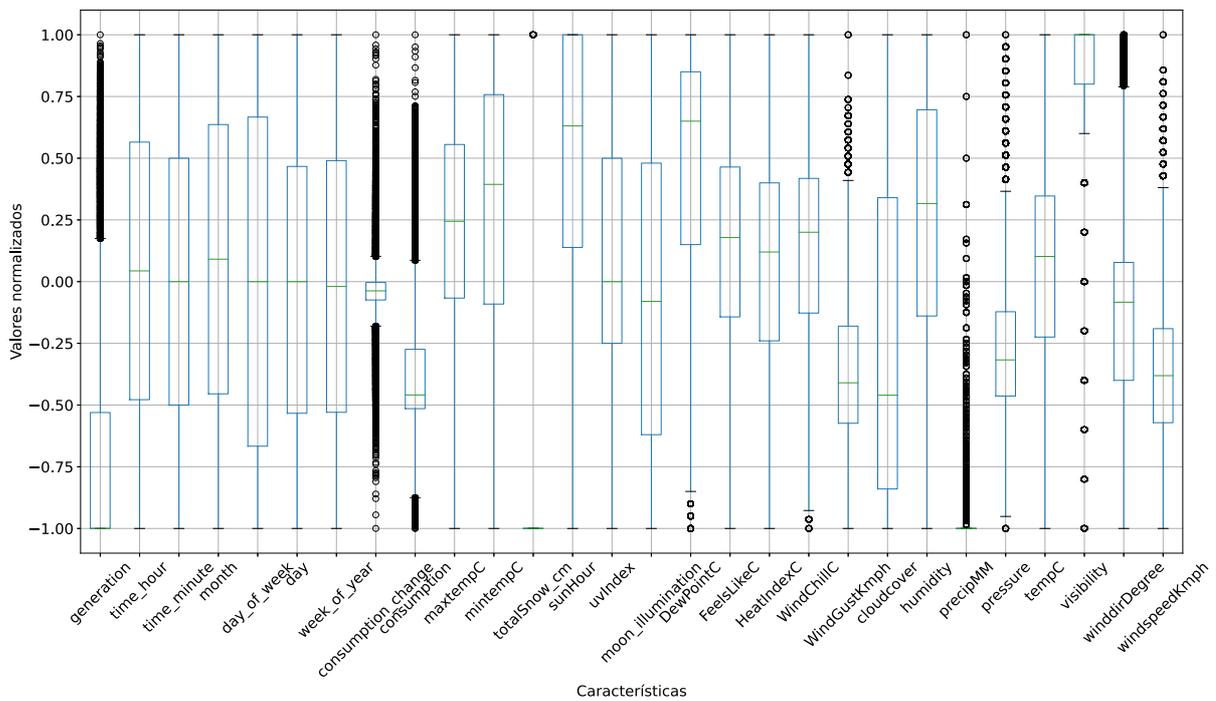
---

Figura 49 – *Boxplot* das características do banco de dados *Pecan Street*, em resolução de 15 minutos.



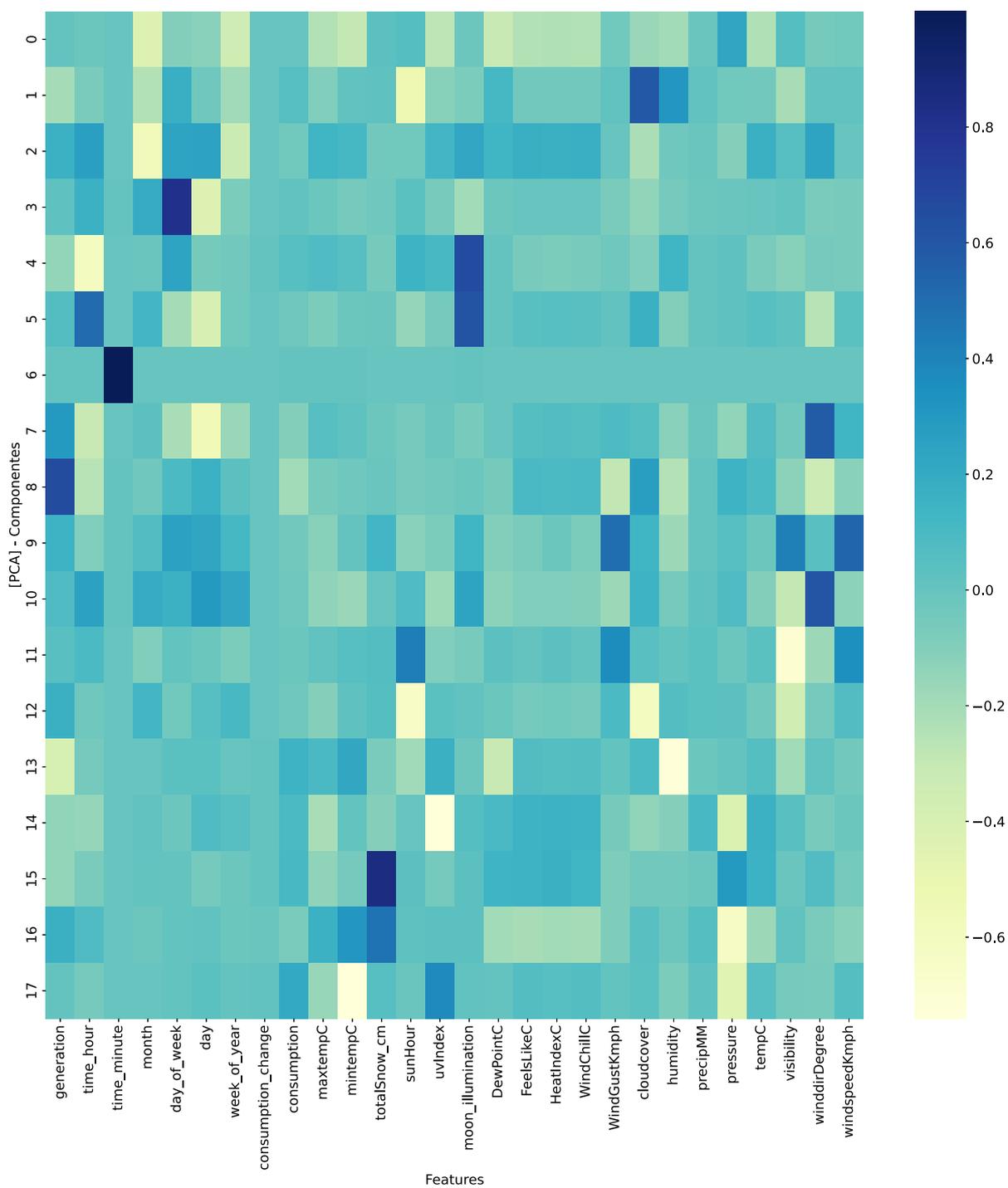
Fonte: Elaborada pelo autor (2023).

Figura 50 – *Boxplot* das características do banco de dados *Pecan Street*, em resolução de 1 hora.



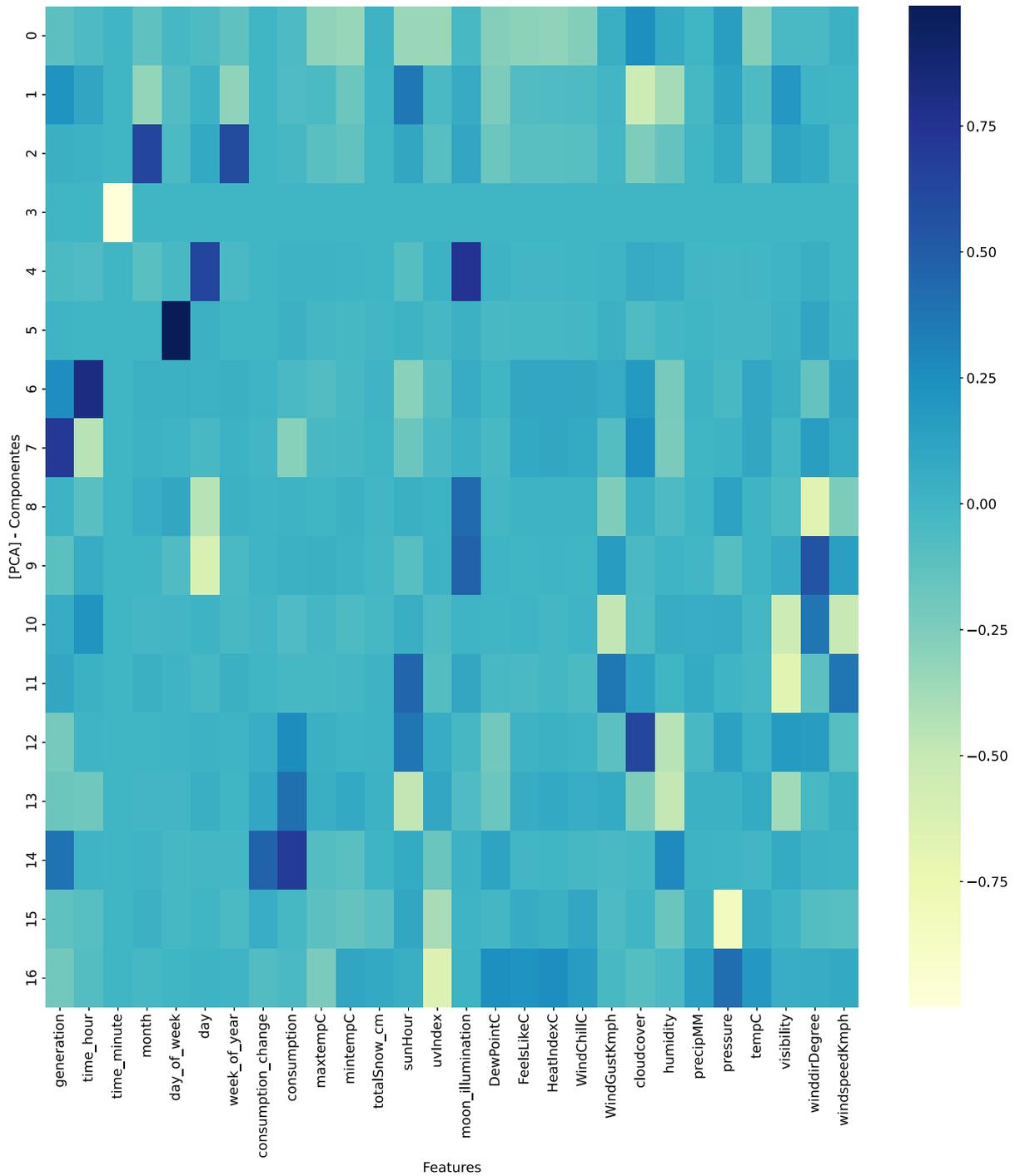
Fonte: Elaborada pelo autor (2023).

Figura 51 – Mapa de calor entre característica e número de componentes para o banco de dados *Pecan Street*, em resolução de 1 minuto.



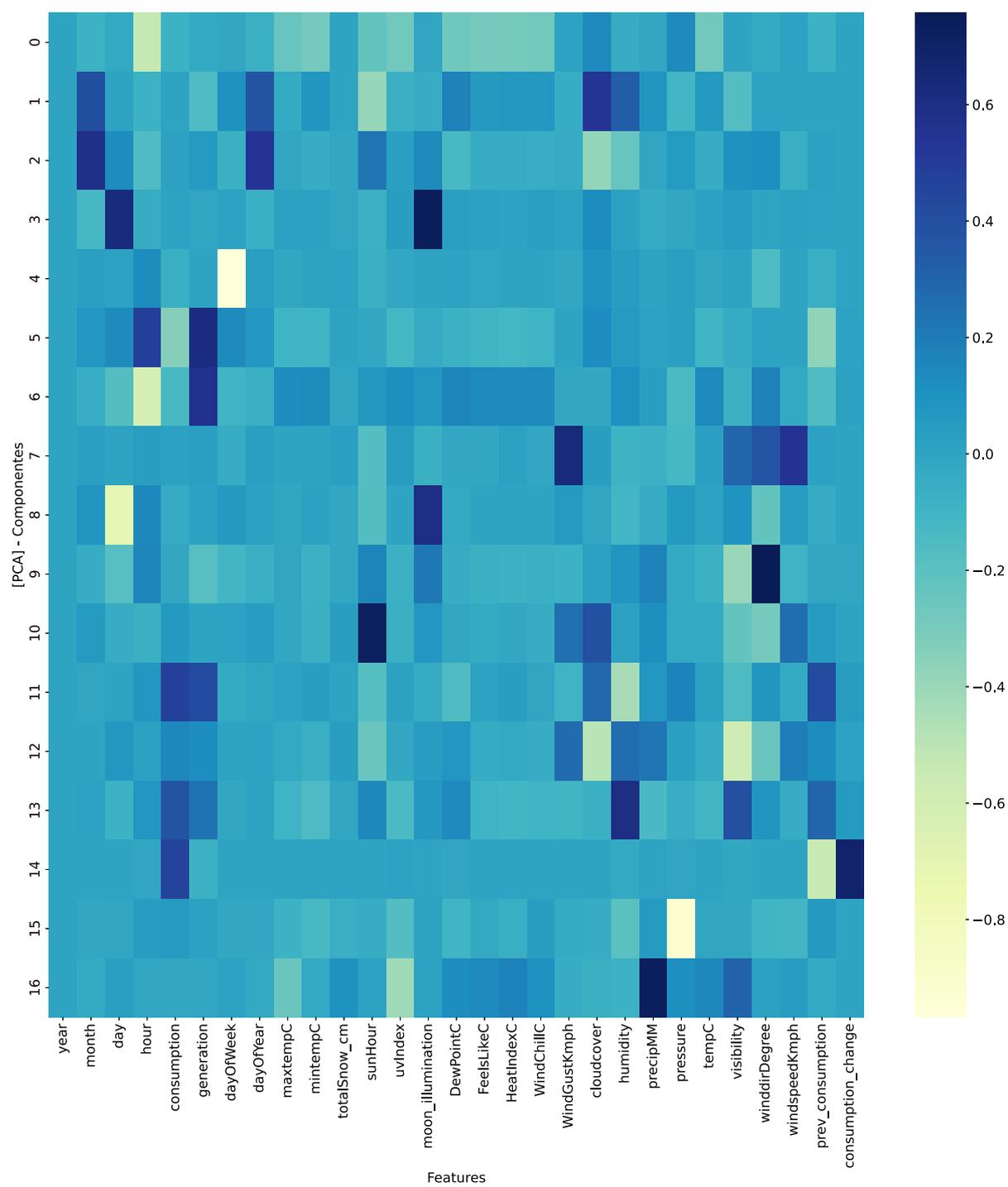
Fonte: Elaborada pelo autor (2023).

Figura 52 – Mapa de calor entre característica e número de componentes para o banco de dados *Pecan Street*, em resolução de 15 minutos.



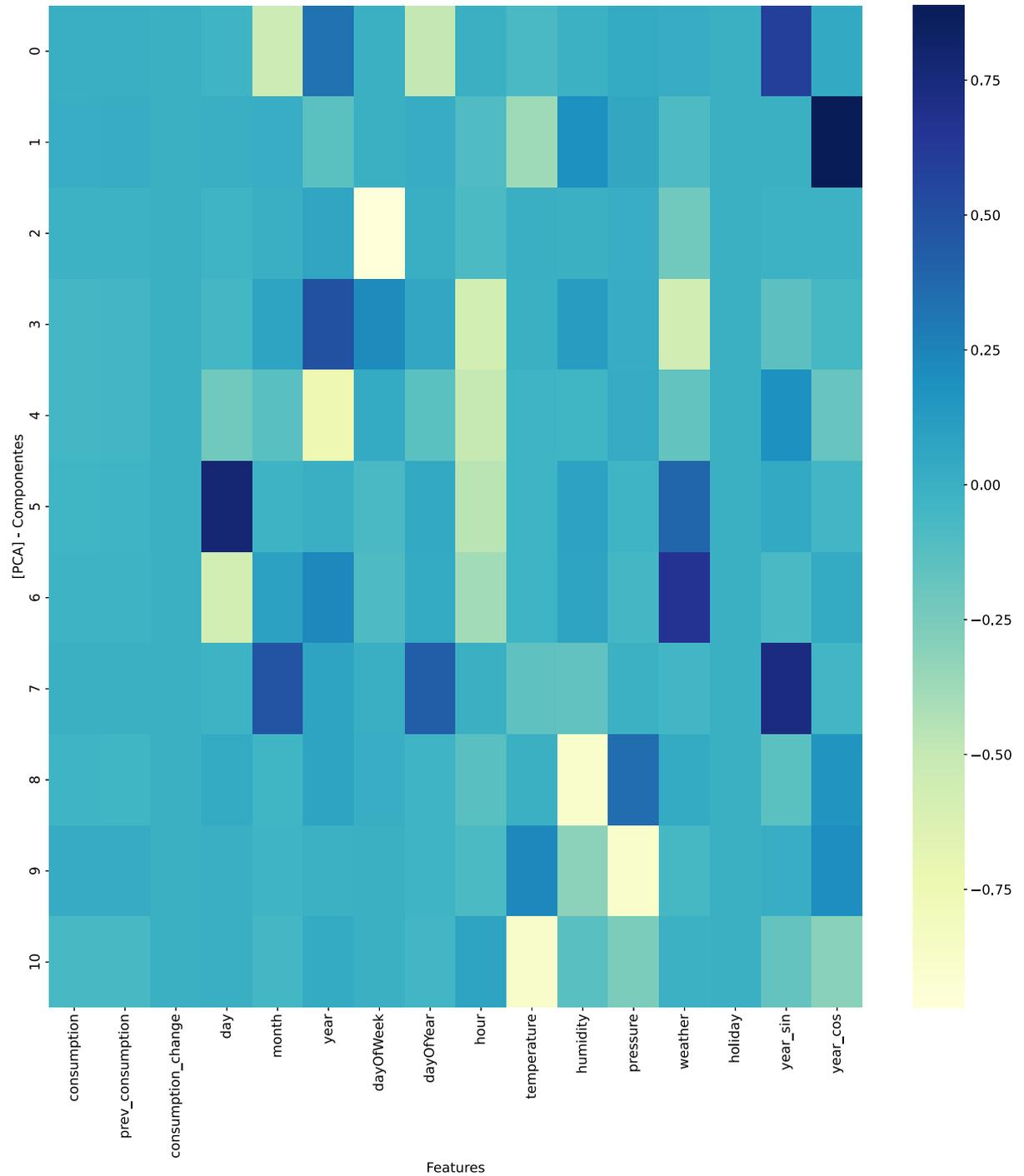
Fonte: Elaborada pelo autor (2023).

Figura 53 – Mapa de calor entre característica e número de componentes para o banco de dados *Pecan Street*, em resolução de 1 hora.



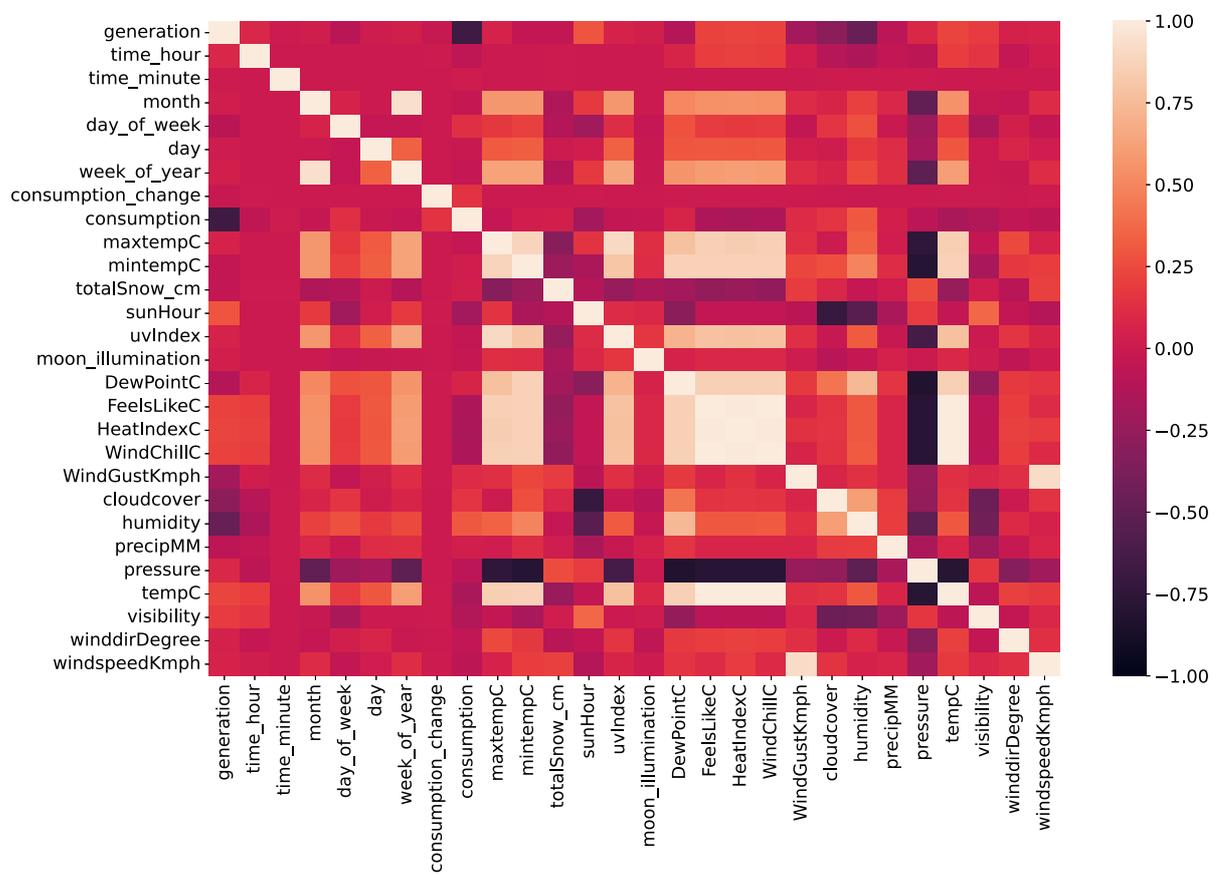
Fonte: Elaborada pelo autor (2023).

Figura 54 – Mapa de calor entre característica e número de componentes para o banco de dados HUE, em resolução de 1 hora.



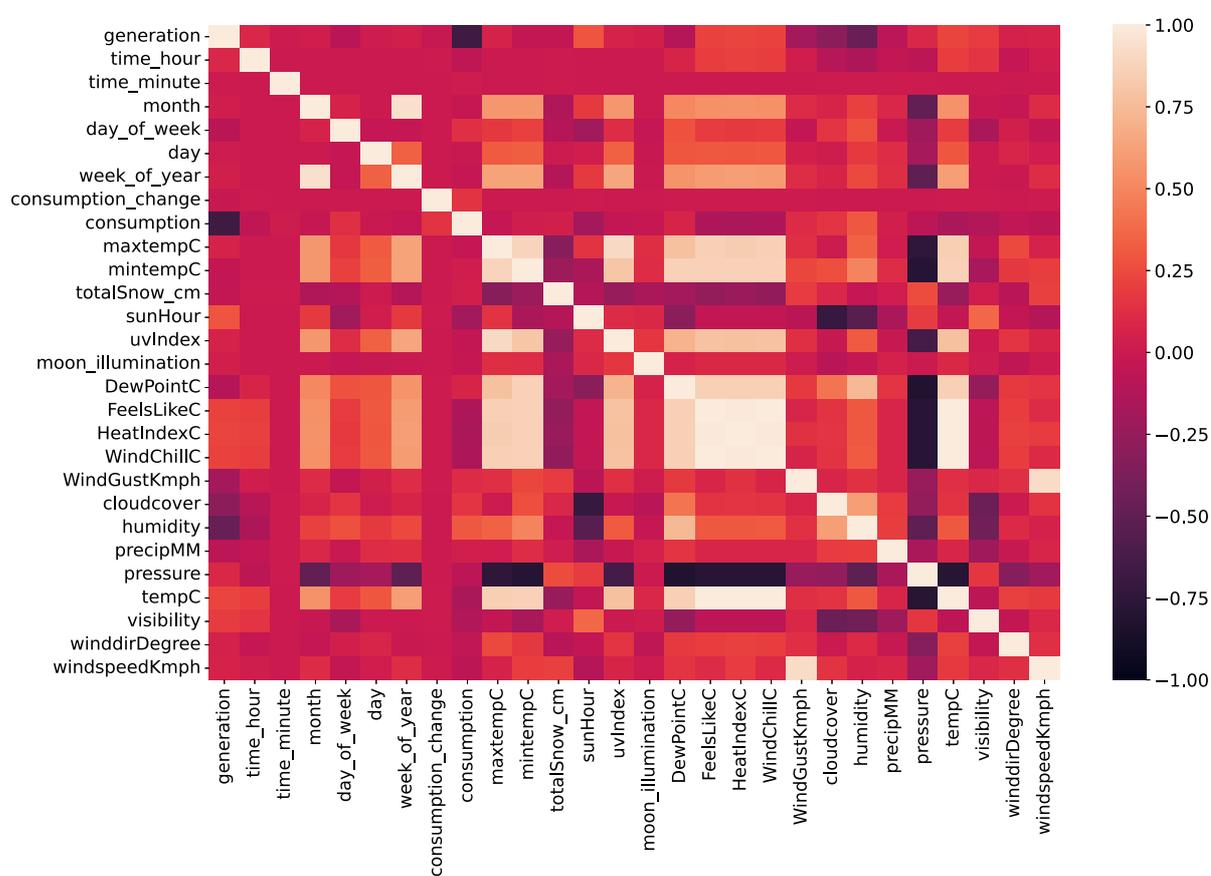
Fonte: Elaborada pelo autor (2023).

Figura 55 – Mapa de calor da correlação existente entre característica para o banco de dados Pecan Street, em resolução de 1 minuto.



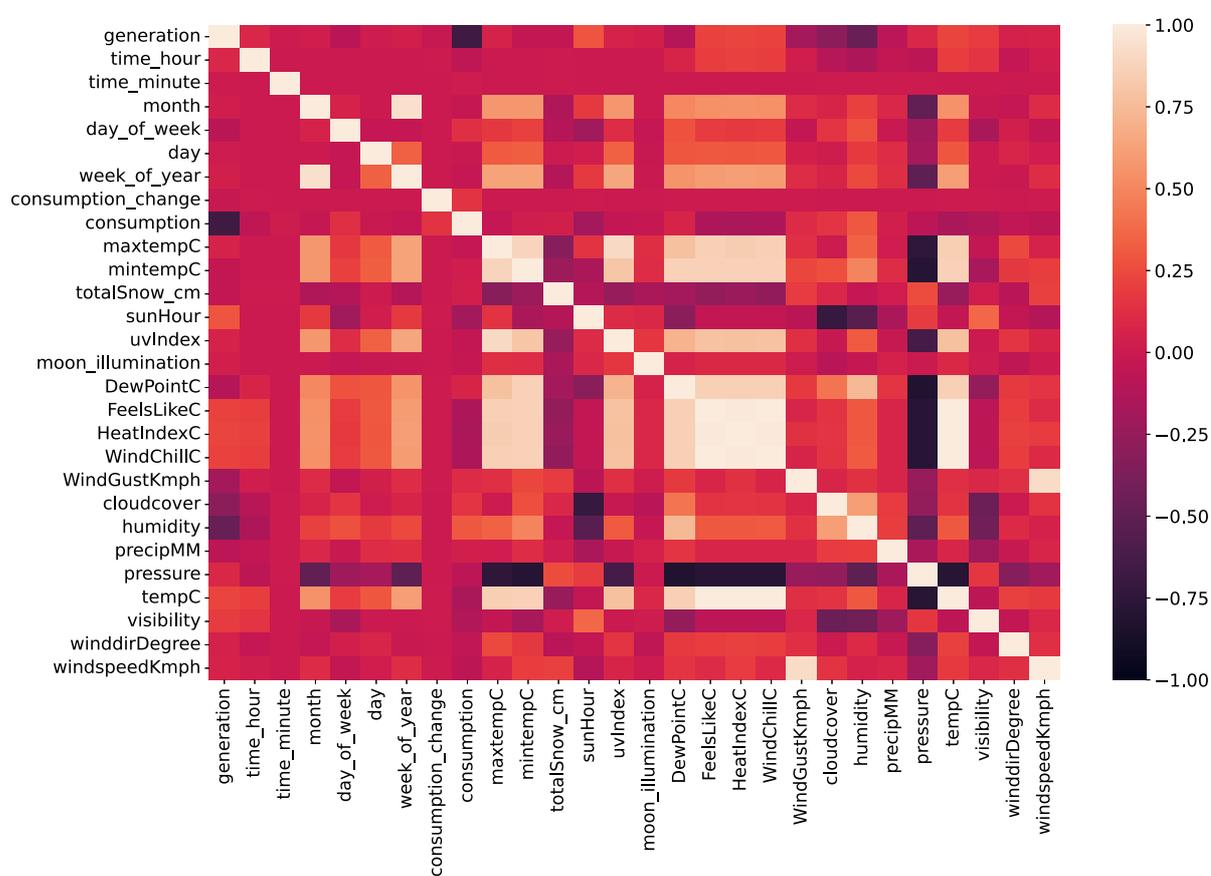
Fonte: Elaborado pelo autor (2023).

Figura 56 – Mapa de calor da correlação existente entre característica para o banco de dados Pecan Street, em resolução de 15 minutos.



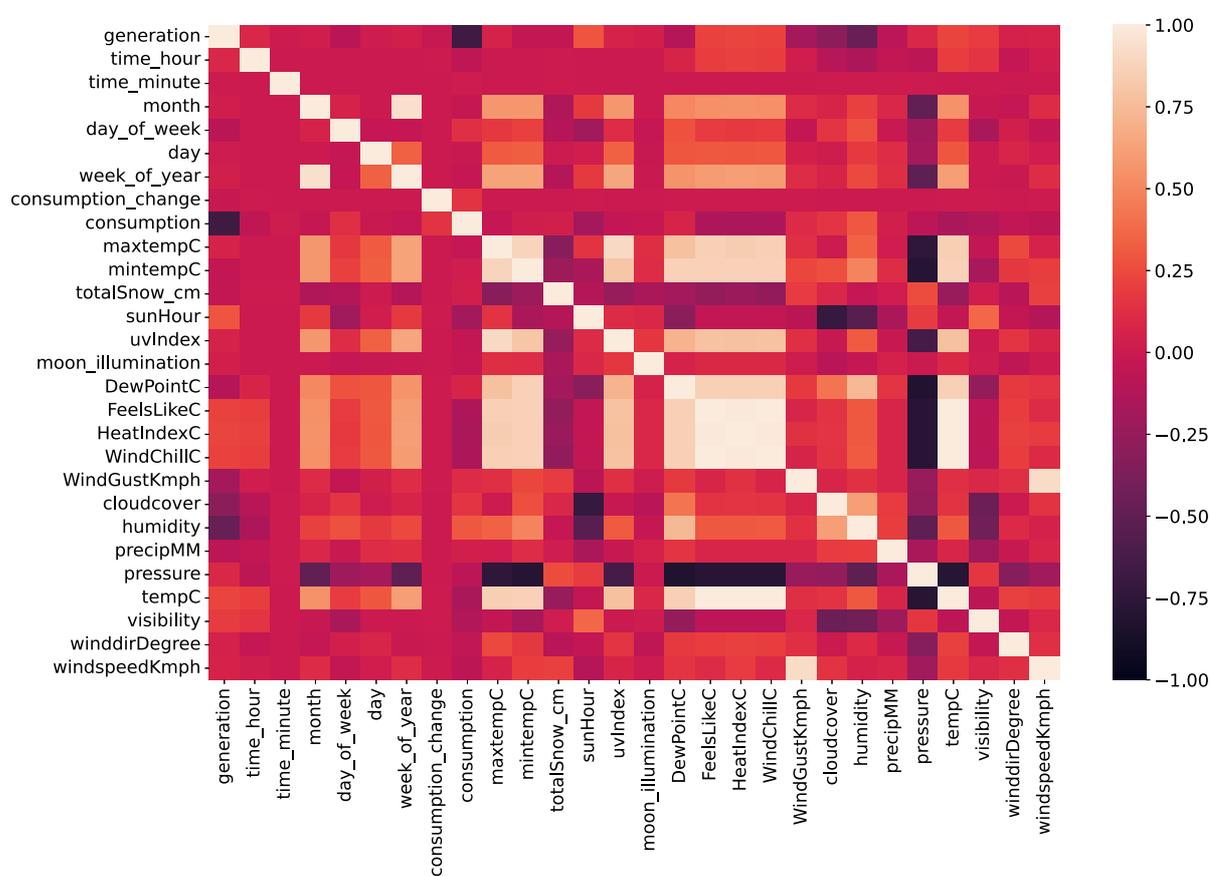
Fonte: Elaborada pelo autor (2023).

Figura 57 – Mapa de calor da correlação existente entre característica para o banco de dados Pecan Street, em resolução de 1 hora.



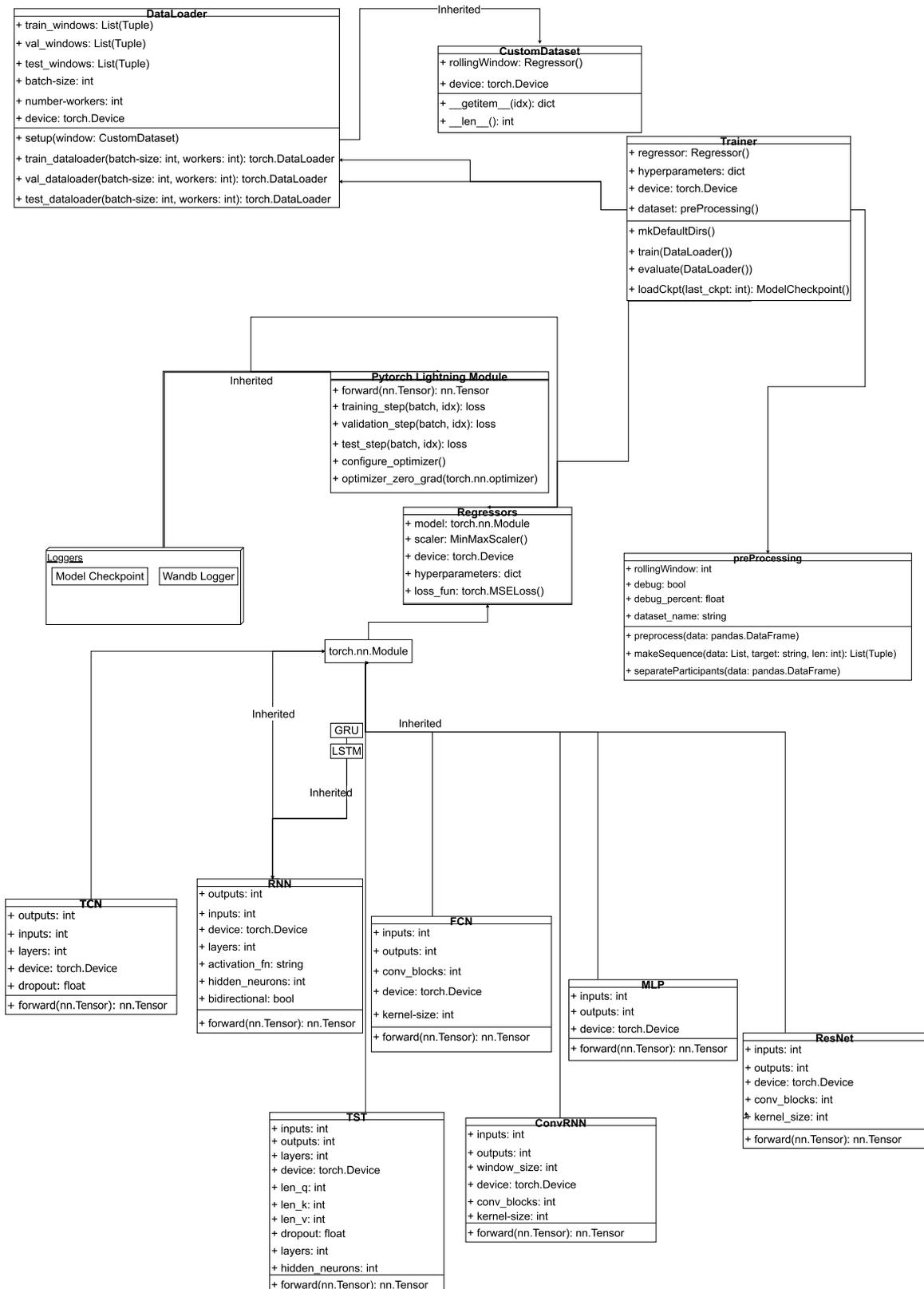
Fonte: Elaborada pelo autor (2023).

Figura 58 – Mapa de calor da correlação existente entre característica para o banco de dados HUE, em resolução de 1 hora.



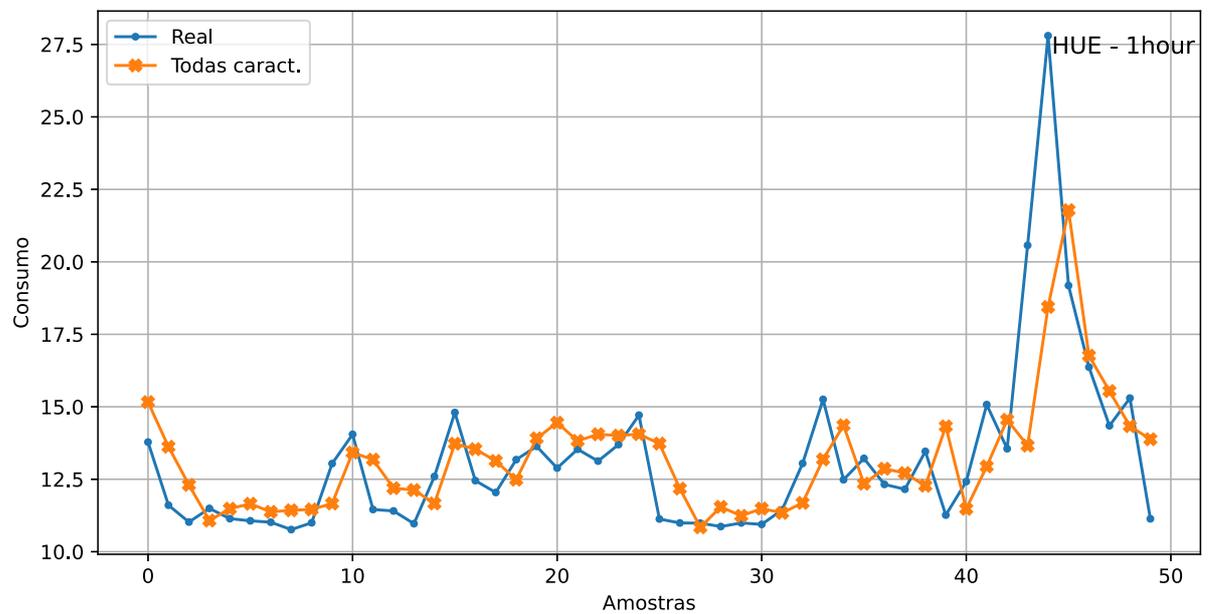
Fonte: Elaborada pelo autor (2023).

Figura 59 – Diagrama de classes do projeto de treinamento dos modelos.



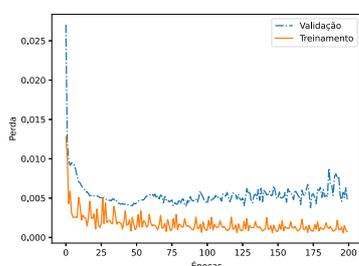
Fonte: Elaborada pelo autor (2023).

Figura 60 – Inferência utilizando o modelo SARIMAX e os dados horários da HUE.

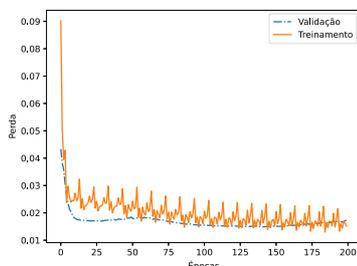


Fonte: Elaborada pelo autor (2023).

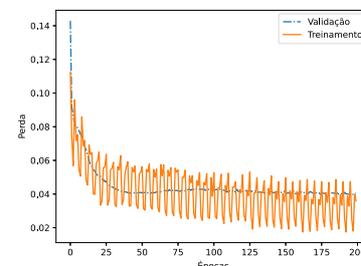
Figura 61 – Os subgráficos (a) à (i) demonstram a melhora no treinamento por meio das curvas de treinamento e validação em relação às épocas, utilizando os dados da *Pecan Street*. Já os sub-gráficos (j) à (l) há a utilização dos dados da *HUE Dataverse*.



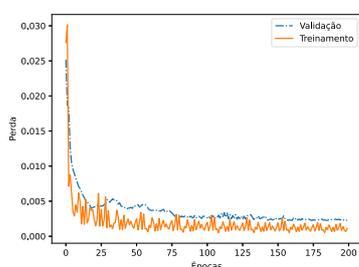
(a) RNN - Resolução de 1 minuto.



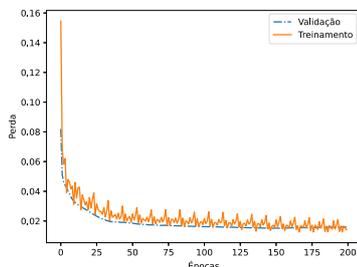
(b) RNN - Resolução de 15 minutos.



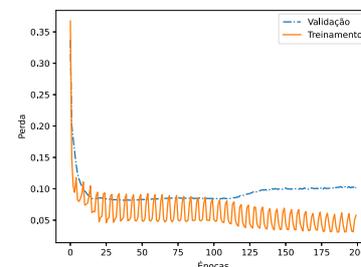
(c) RNN - Resolução de 1 hora.



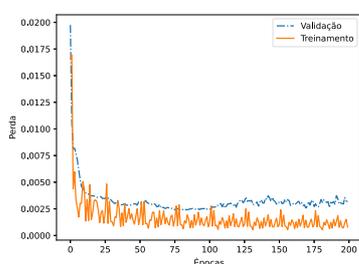
(d) LSTM - Resolução de 1 minuto.



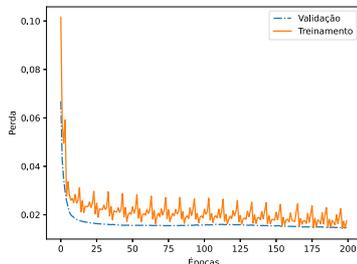
(e) LSTM - Resolução de 15 minutos.



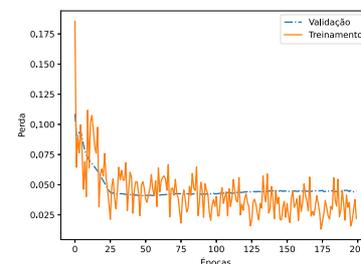
(f) LSTM - Resolução horária.



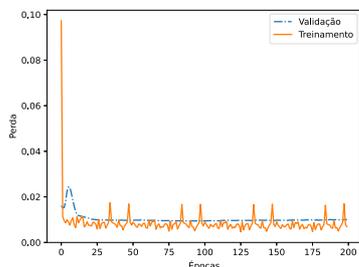
(g) GRU - Resolução de 1 minuto.



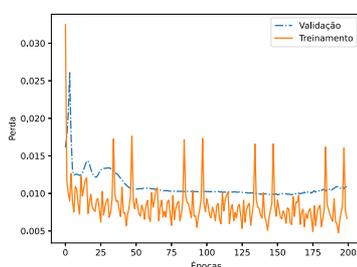
(h) GRU - Resolução de 15 minutos.



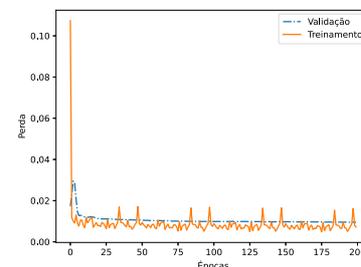
(i) GRU - Resolução de 1 hora.



(j) RNN - Resolução de 1 hora.



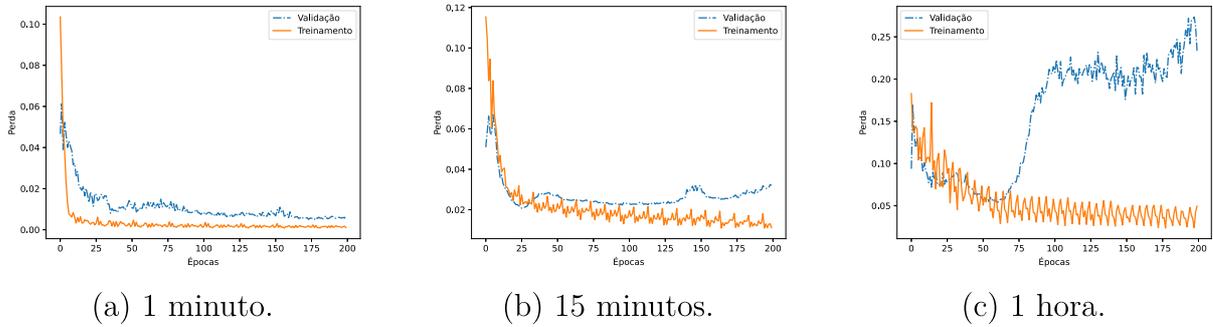
(k) LSTM - Resolução de 1 hora.



(l) GRU - Resolução de 1 hora.

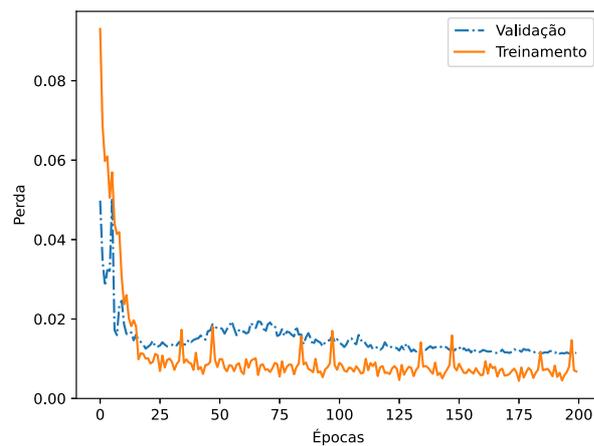
Fonte: Elaborada pelo autor (2023).

Figura 62 – Curva de treinamento da arquitetura TST utilizando os dados da *Pecan Street*.



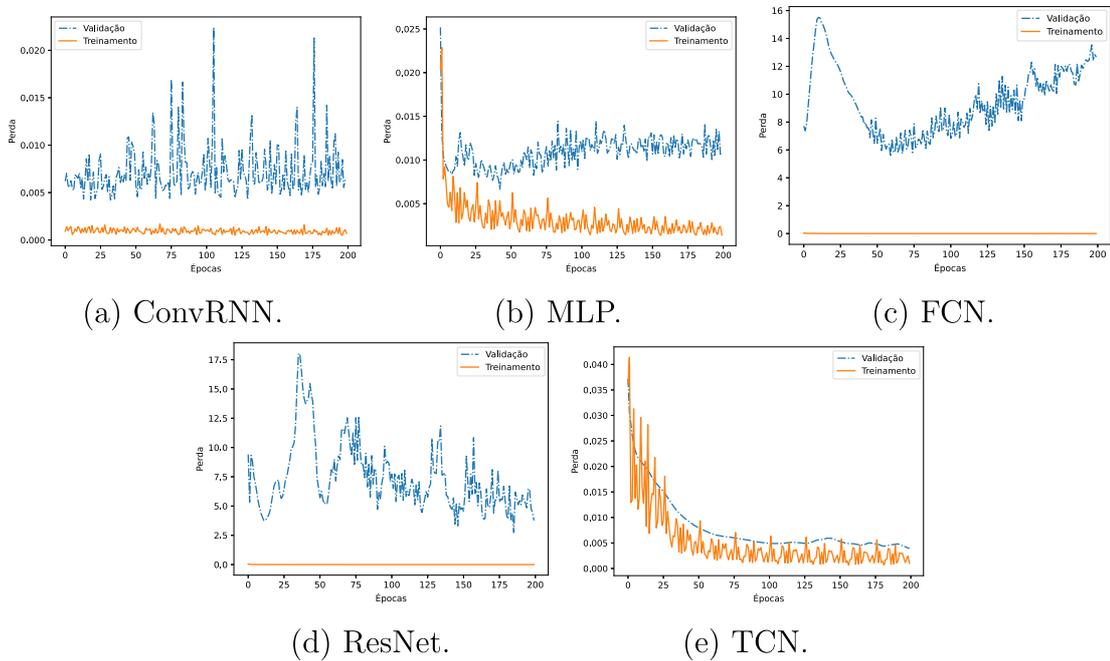
Fonte: Elaborada pelo autor (2023).

Figura 63 – Avanço do treinamento do modelo TST utilizando a resolução horária da *HUE Dataverse*.



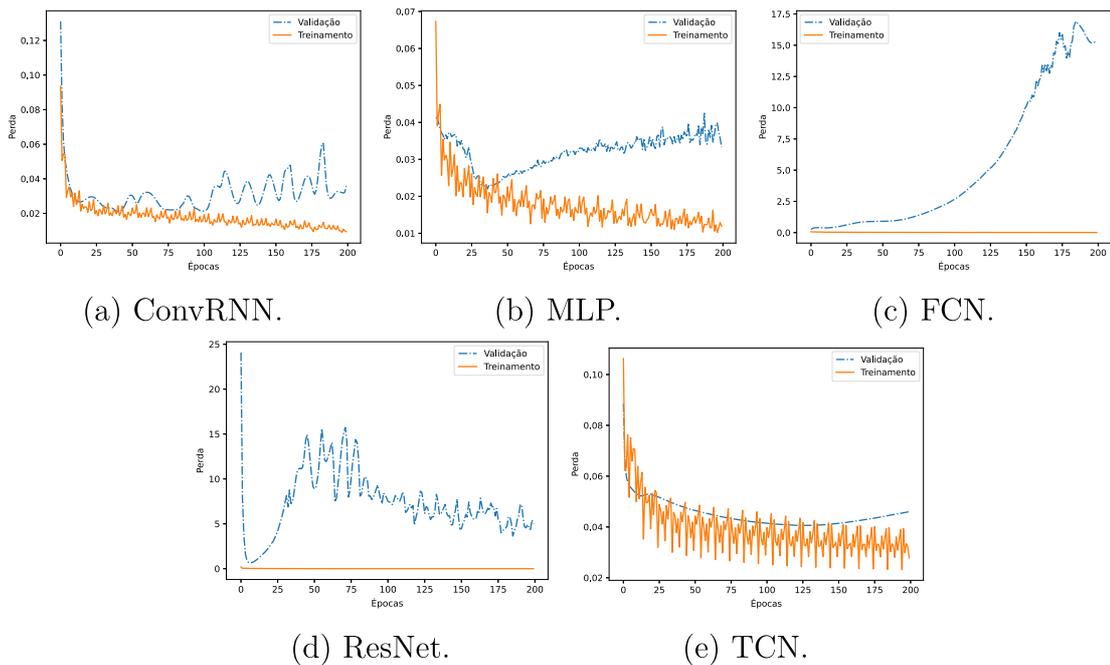
Fonte: Elaborada pelo autor (2023).

Figura 64 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da *Pecan Street*, na resolução de 1 minuto.



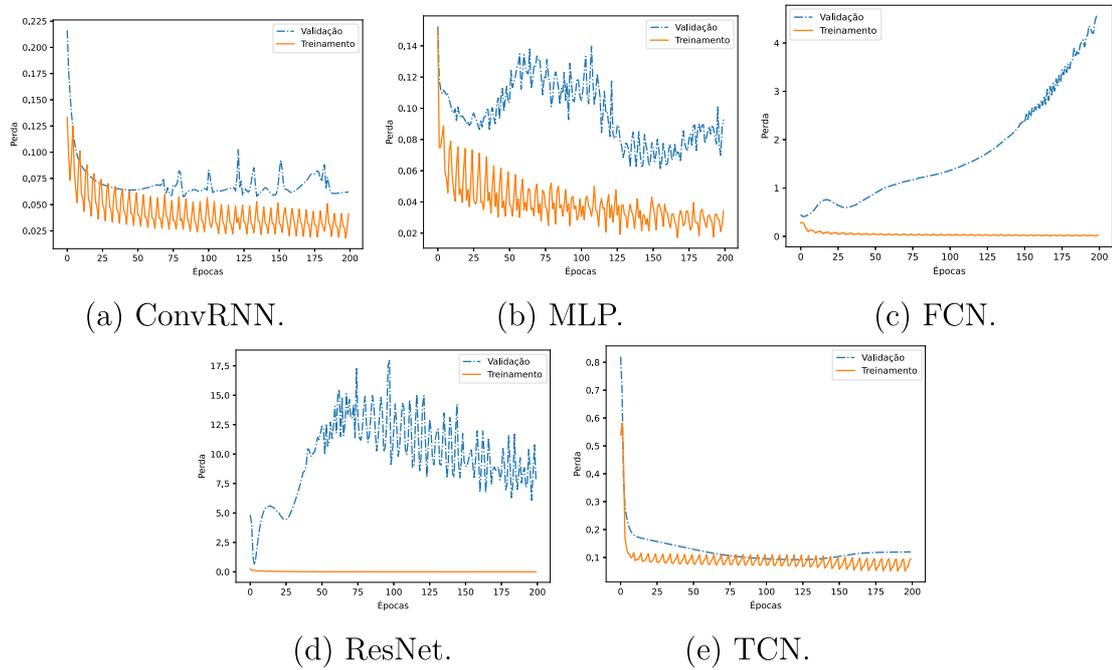
Fonte: Elaborada pelo autor (2023).

Figura 65 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da *Pecan Street*, na resolução de 15 minutos.



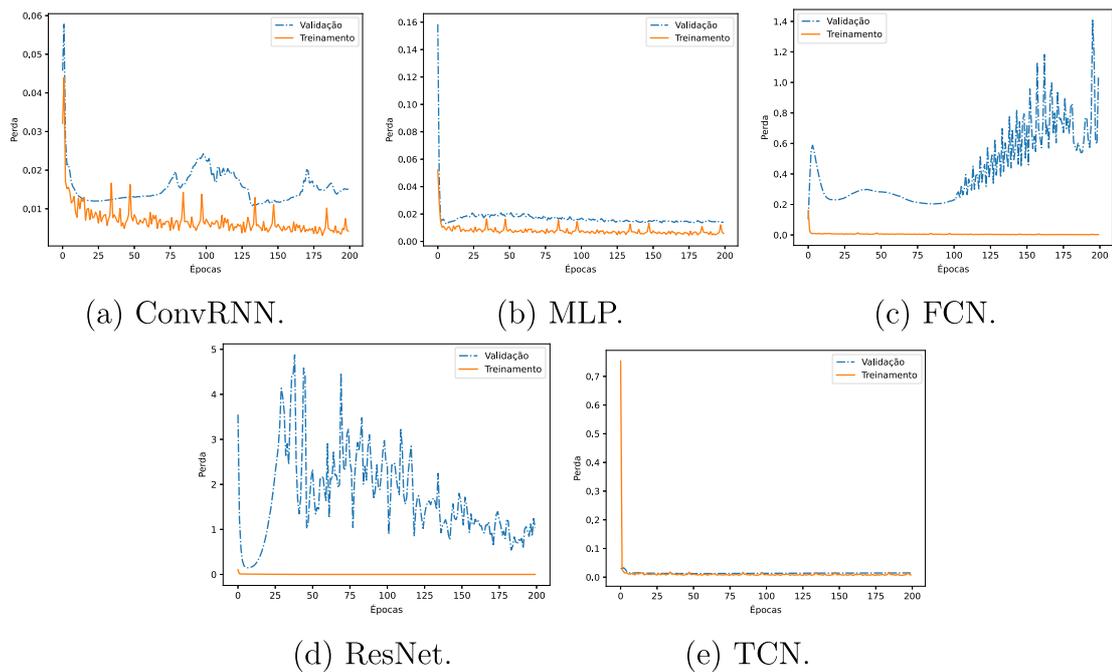
Fonte: Elaborada pelo autor (2023).

Figura 66 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da *Pecan Street*, na resolução de horária.



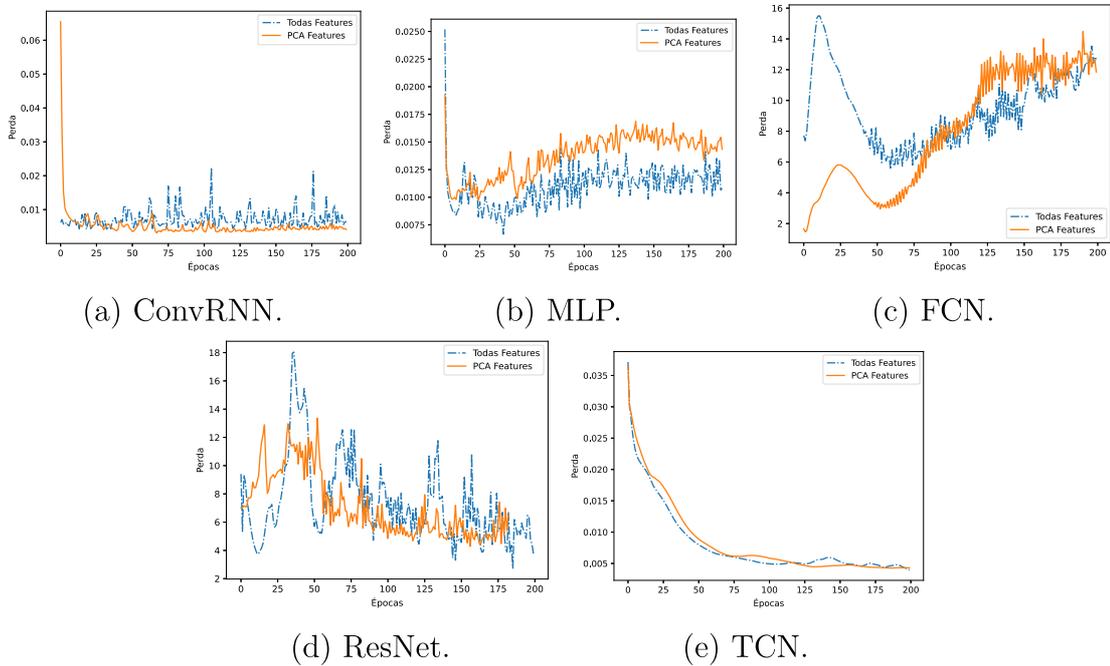
Fonte: Elaborada pelo autor (2023).

Figura 67 – Curvas de treinamento e validação dos modelos secundários utilizando todas as características da *HUE*, na resolução horária.



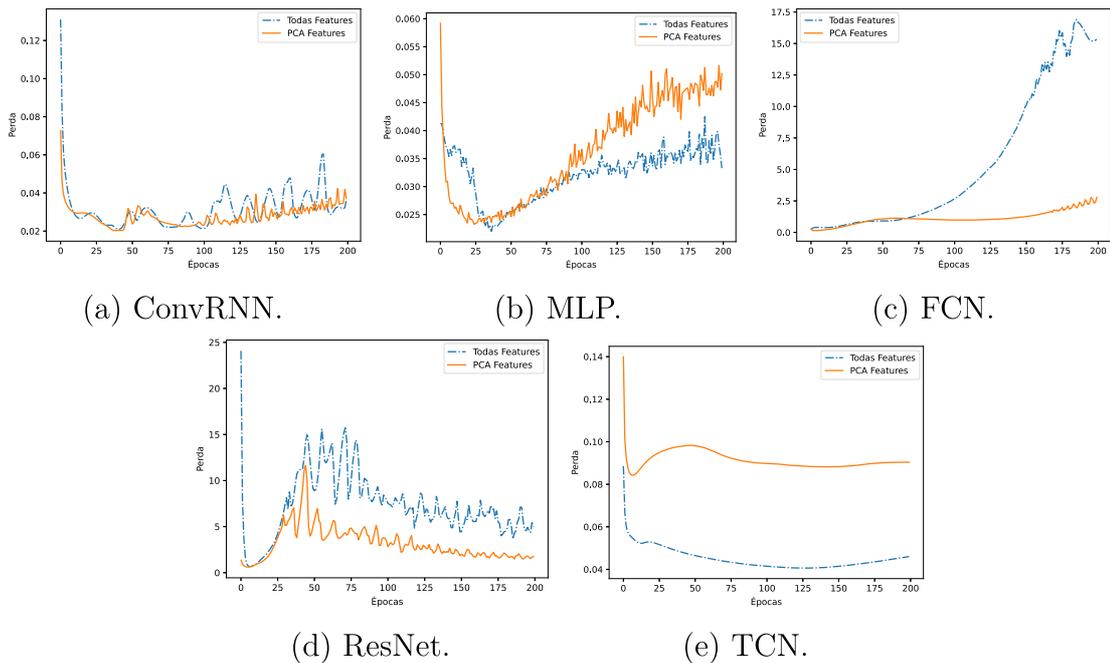
Fonte: Elaborada pelo autor (2023).

Figura 68 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da *Pecan Street*, em resolução de 1 minuto.



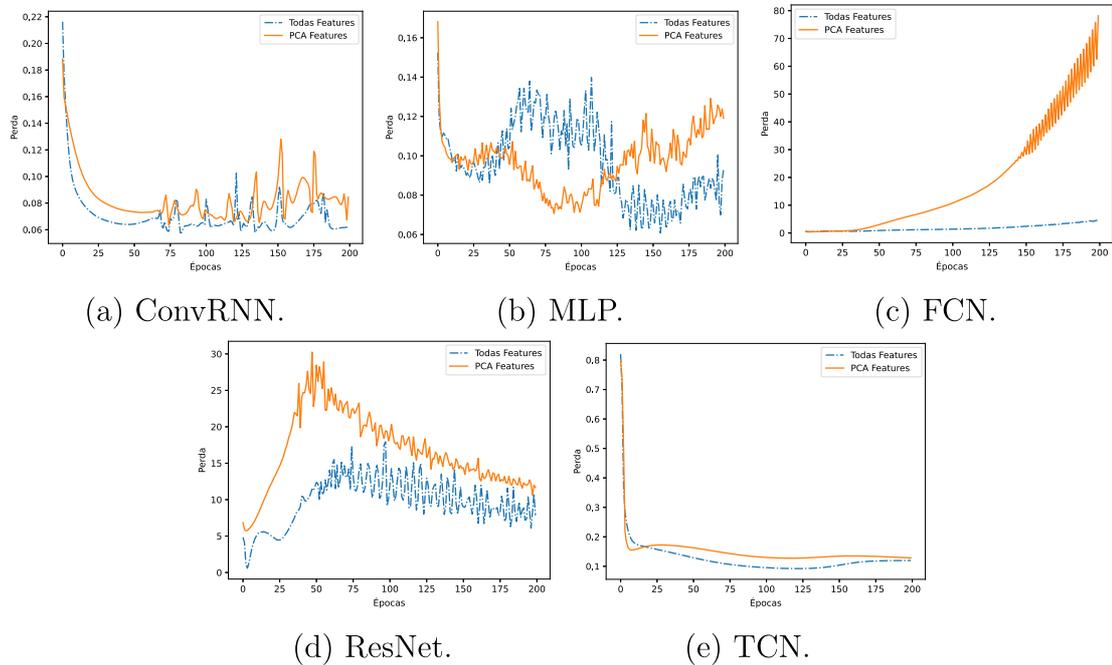
Fonte: Elaborada pelo autor (2023).

Figura 69 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da *Pecan Street*, em resolução de 15 minutos.



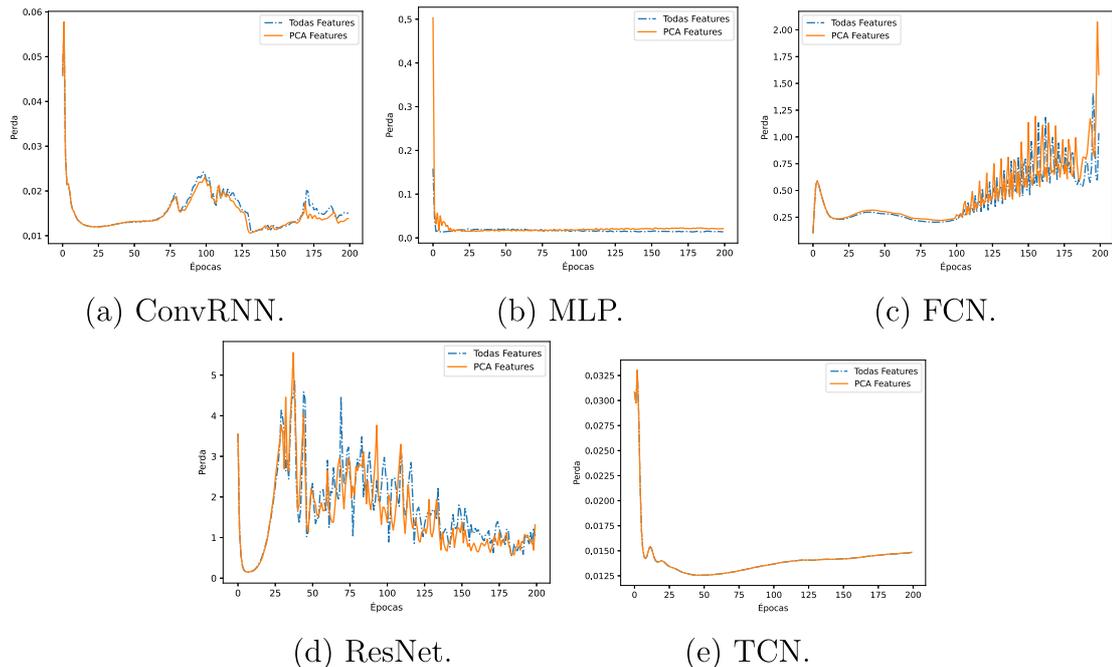
Fonte: Elaborada pelo autor (2023).

Figura 70 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da *Pecan Street*, em resolução horária.



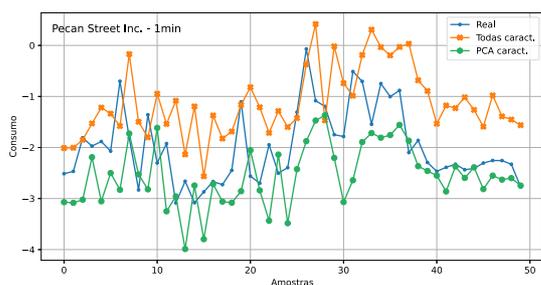
Fonte: Elaborada pelo autor (2023).

Figura 71 – Comparativo das métricas de avaliação utilizando as bases com todas as características e por pré-processamento por PCA, tendo como eixo a base de dados da HUE, em resolução horária.

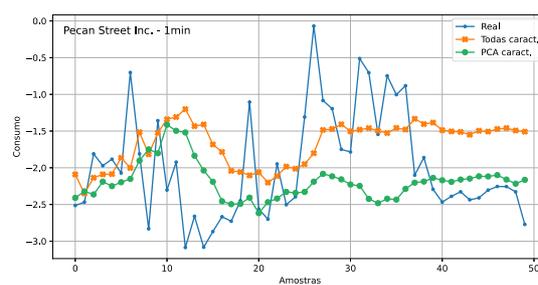


Fonte: Elaborada pelo autor (2023).

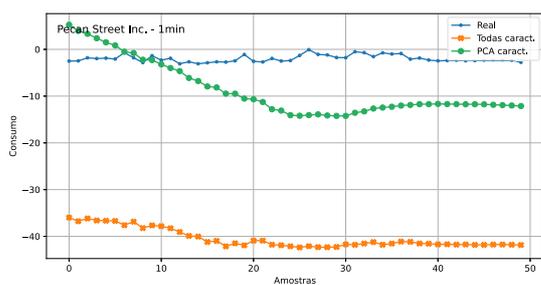
Figura 72 – Inferências de 50 amostras tendo como eixo a base de dados da *Pecan Street*, em resolução de 1 minuto.



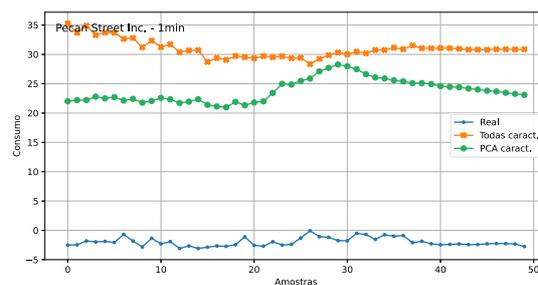
(a) ConvRNN.



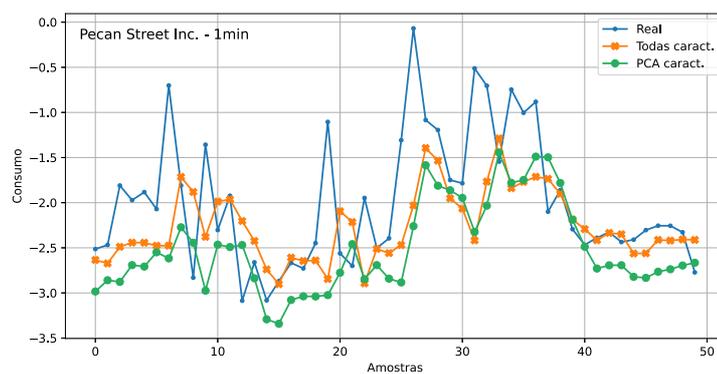
(b) MLP.



(c) FCN.



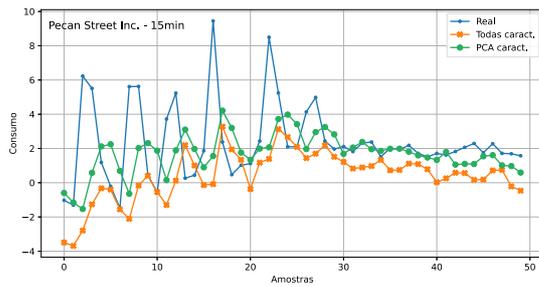
(d) ResNet.



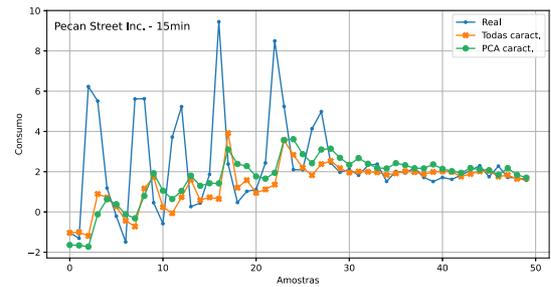
(e) TCN.

Fonte: Elaborada pelo autor (2023).

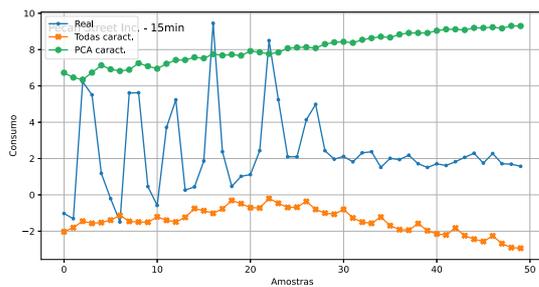
Figura 73 – Inferências de 50 amostras tendo como eixo a base de dados da *Pecan Street*, em resolução de 15 minutos.



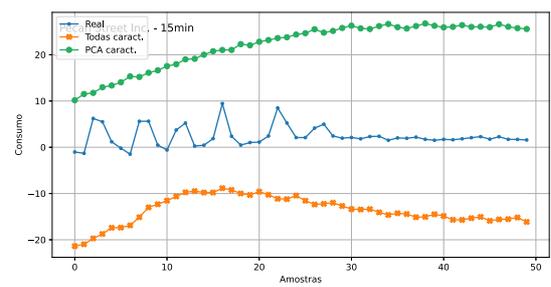
(a) ConvRNN.



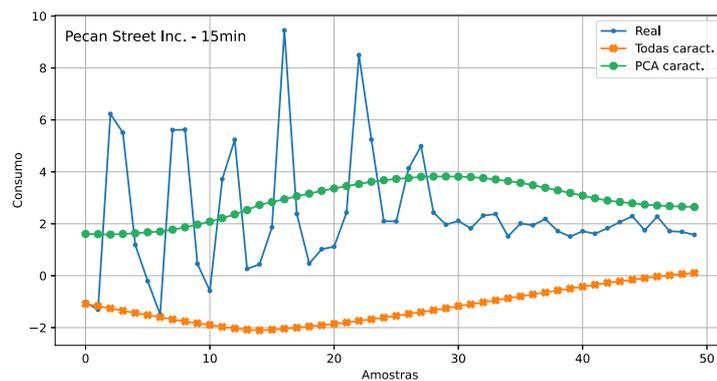
(b) MLP.



(c) FCN.



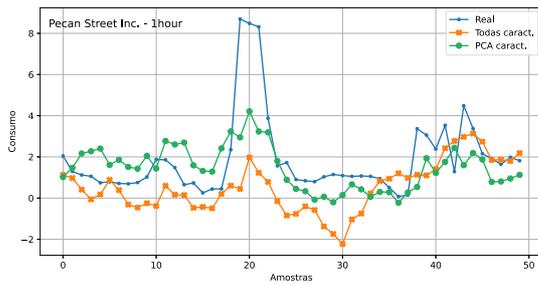
(d) ResNet.



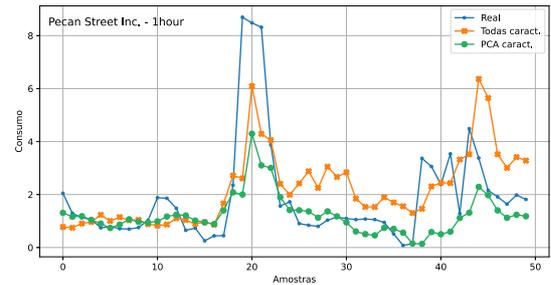
(e) TCN.

Fonte: Elaborada pelo autor (2023).

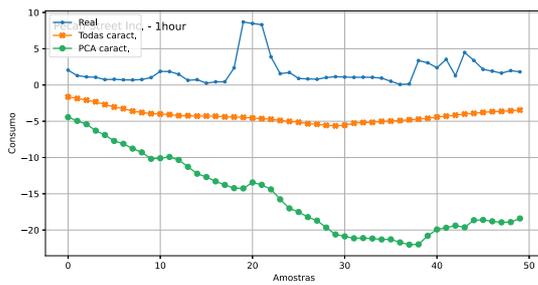
Figura 74 – Inferências de 50 amostras tendo como eixo a base de dados da *Pecan Street*, em resolução horária.



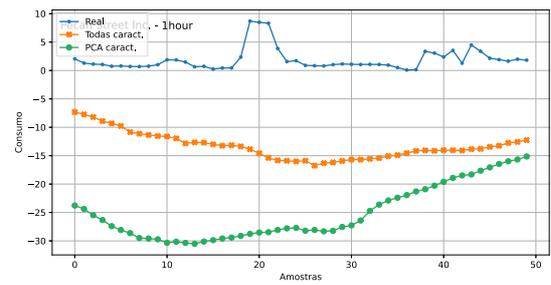
(a) ConvRNN.



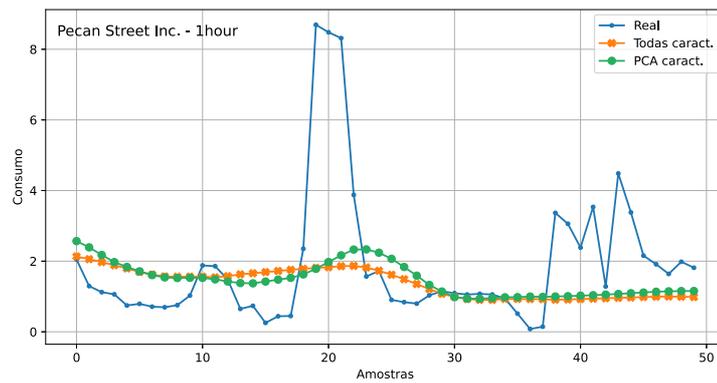
(b) MLP.



(c) FCN.



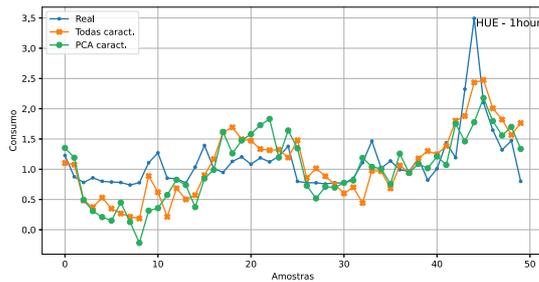
(d) ResNet.



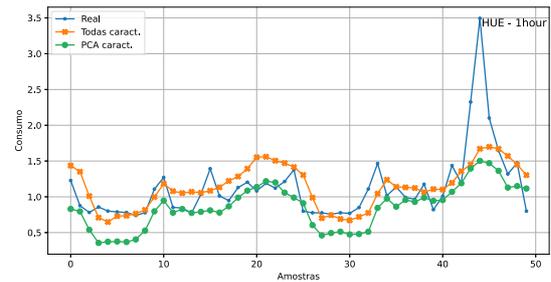
(e) TCN.

Fonte: Elaborada pelo autor (2023).

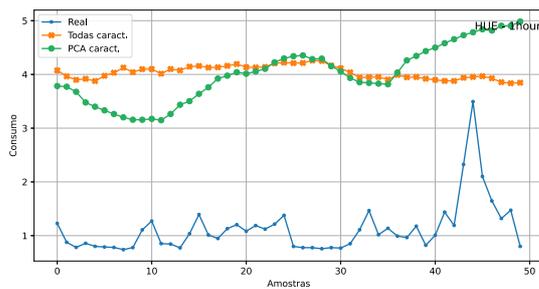
Figura 75 – Inferências de 50 amostras tendo como eixo a base de dados da HUE, em resolução horária.



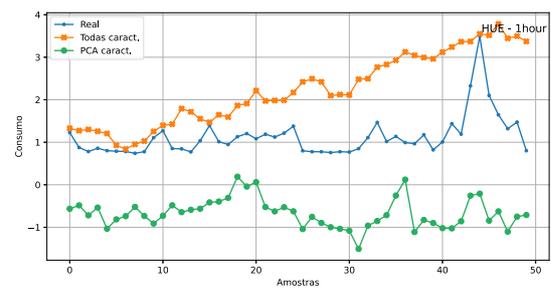
(a) ConvRNN.



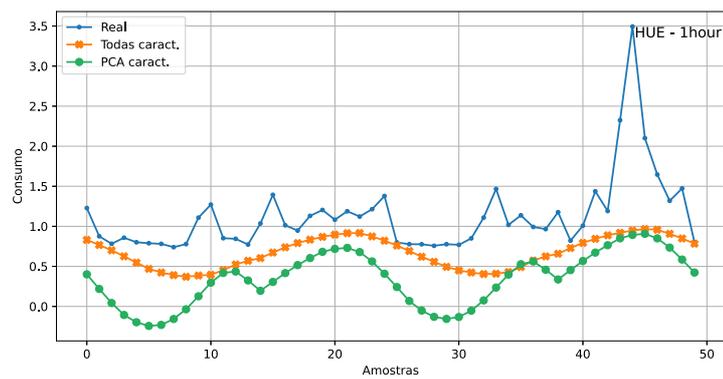
(b) MLP.



(c) FCN.



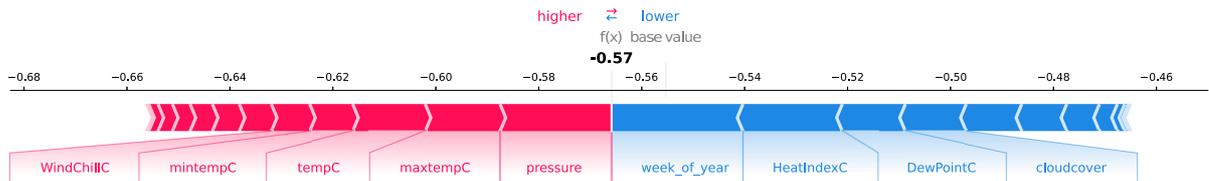
(d) ResNet.



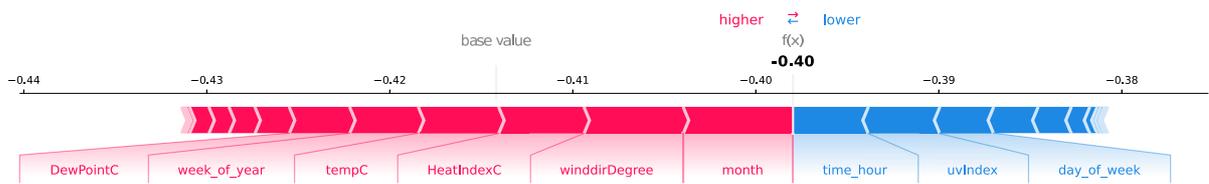
(e) TCN.

Fonte: Elaborada pelo autor (2023).

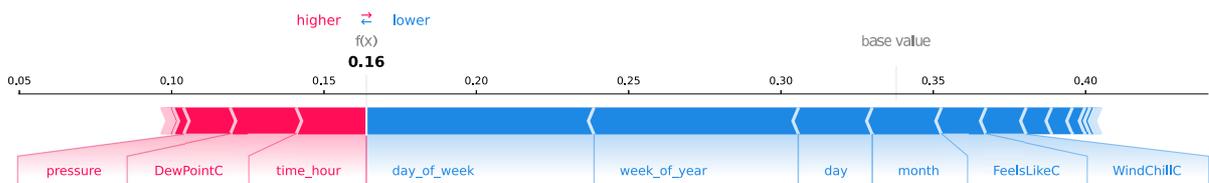
Figura 76 – Impacto de característica para a inferência de um único valor utilizando os dados da *Pecan Street*, na resolução de 1 minuto.



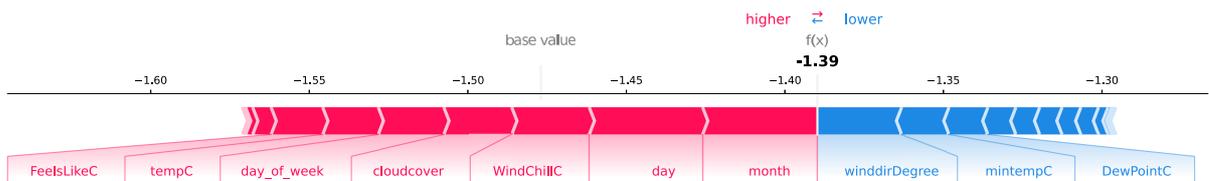
(a) ConvRNN.



(b) MLP.



(c) FCN.



(d) ResNet.



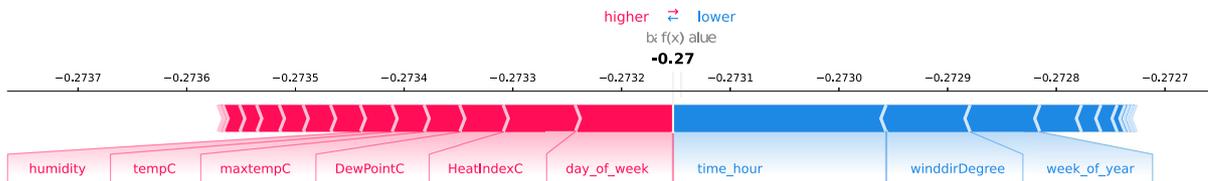
(e) TCN.

Fonte: Elaborada pelo autor (2023).

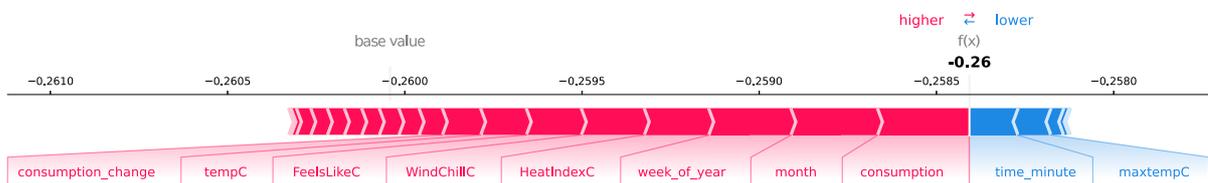
Figura 77 – Impacto de característica para a inferência de um único valor utilizando os dados da *Pecan Street*, na resolução de 15 minutos, para os modelos principais.



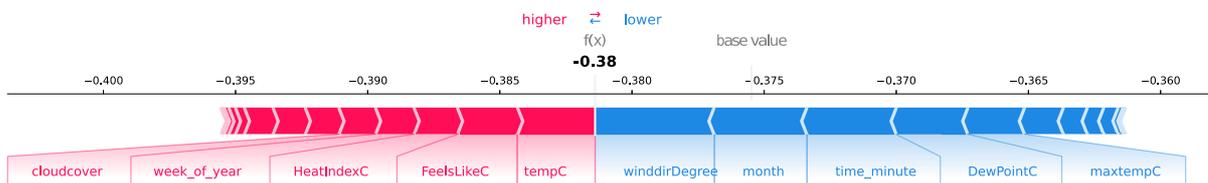
(a) RNN.



(b) LSTM.



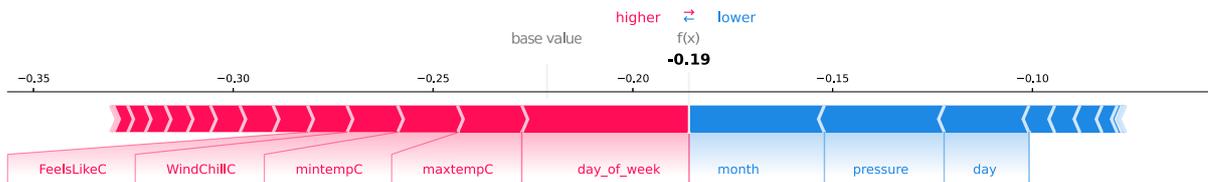
(c) GRU.



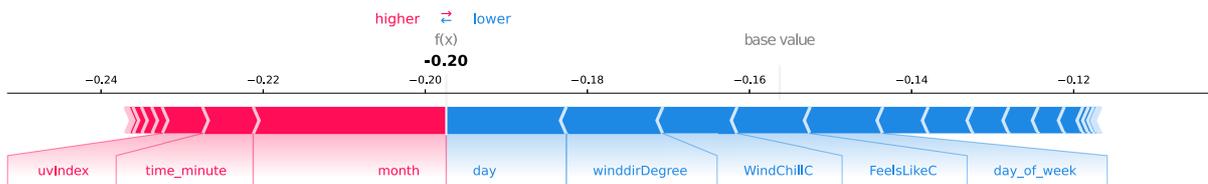
(d) TST.

Fonte: Elaborada pelo autor (2023).

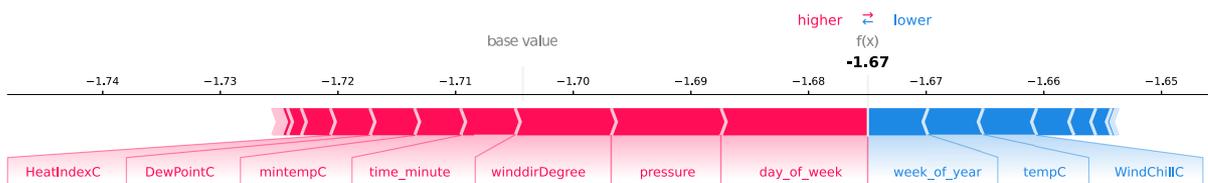
Figura 78 – Impacto de característica para a inferência de um único valor utilizando os dados da *Pecan Street*, na resolução de 15 minutos, para os modelos complementares.



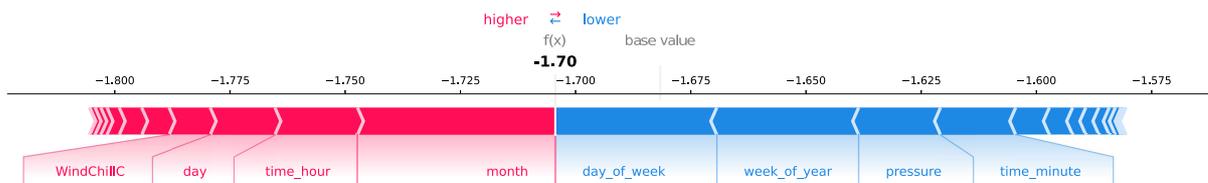
(a) ConvRNN.



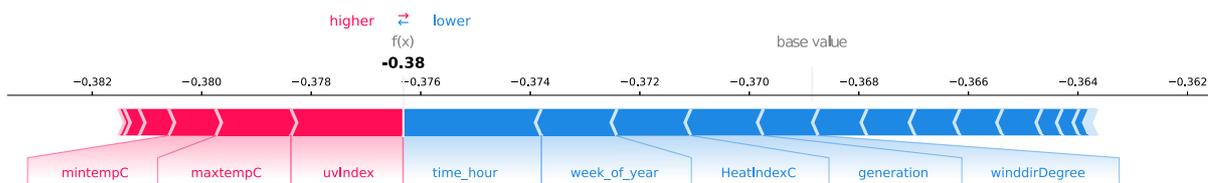
(b) MLP.



(c) FCN.



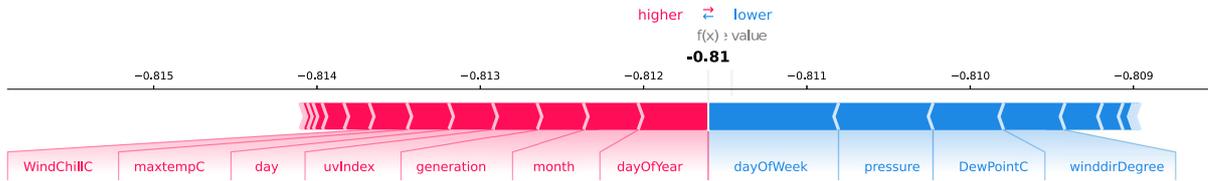
(d) ResNet.



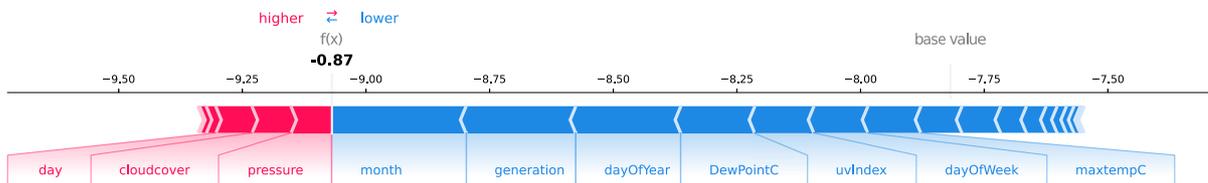
(e) TCN.

Fonte: Elaborada pelo autor (2023).

Figura 79 – Impacto de característica para a inferência de um único valor utilizando os dados da *Pecan Street*, na resolução horária, para os modelos principais.

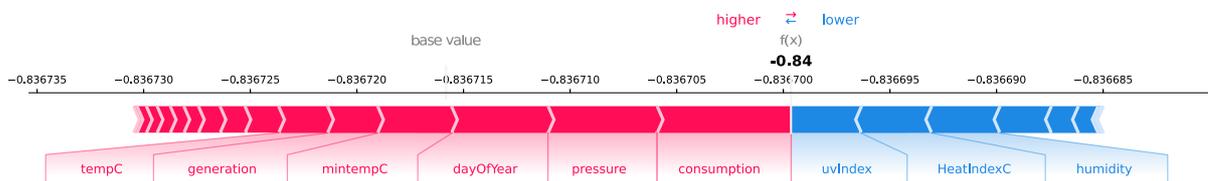


(a) RNN.

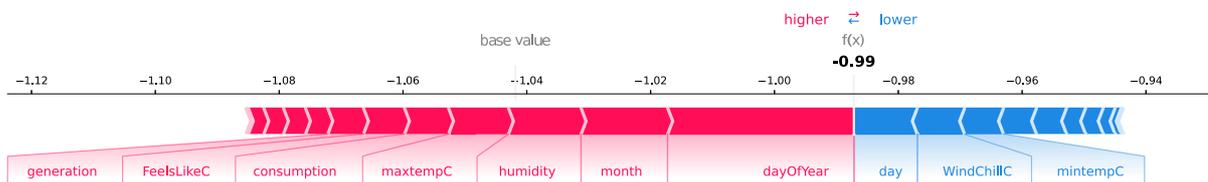


1e-8-8,674259e-1

(b) LSTM.



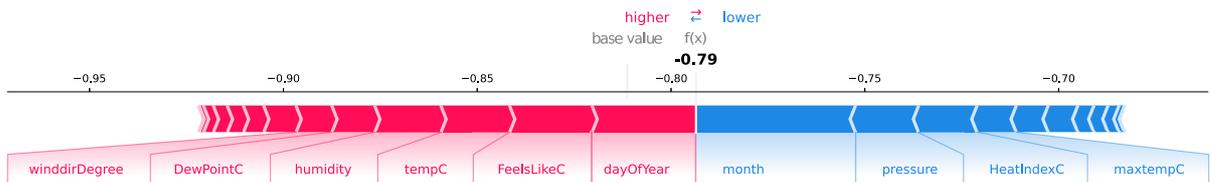
(c) GRU.



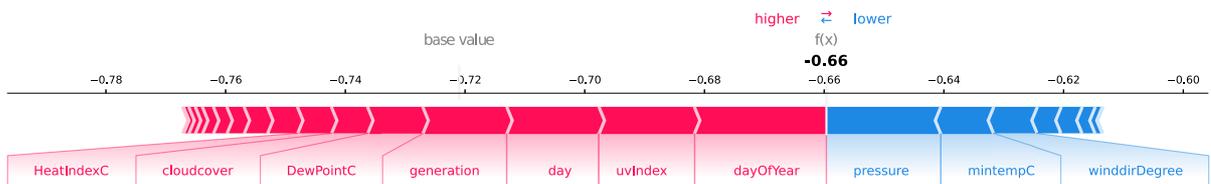
(d) TST.

Fonte: Elaborada pelo autor (2023).

Figura 80 – Impacto de característica para a inferência de um único valor utilizando os dados da *Pecan Street*, na resolução horária, para os modelos complementares.



(a) ConvRNN.



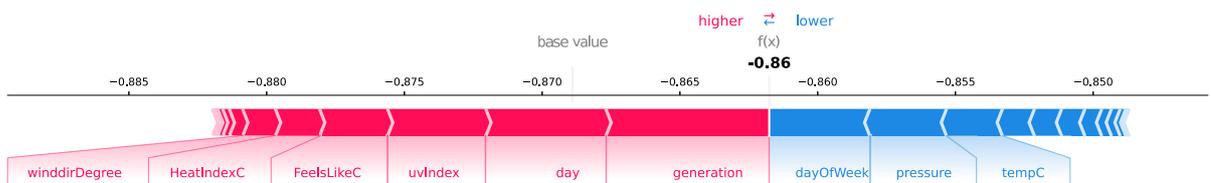
(b) MLP.



(c) FCN.



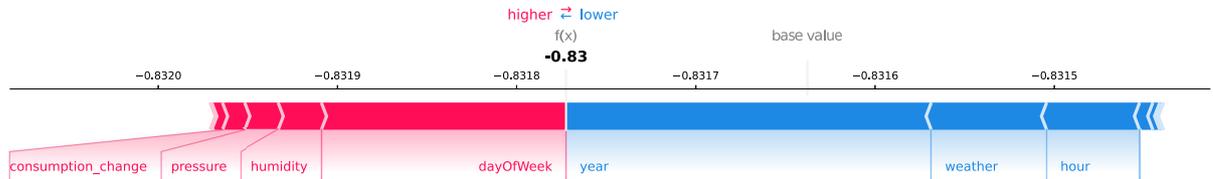
(d) ResNet.



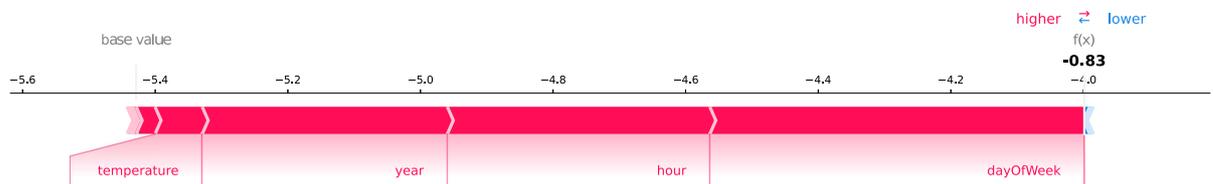
(e) TCN.

Fonte: Elaborada pelo autor (2023).

Figura 81 – Impacto de característica para a inferência de um único valor utilizando os dados da HUE, na resolução horária, para os modelos principais.



(a) RNN.



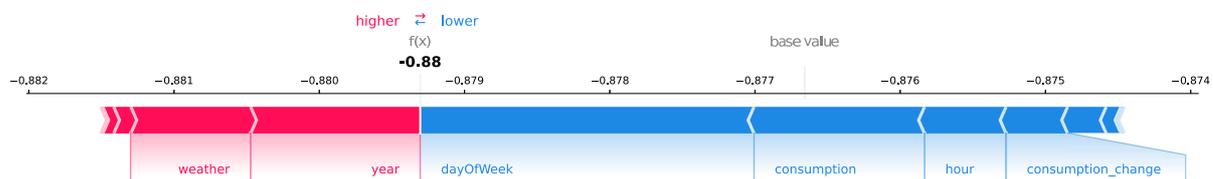
$1e-5-8.329e-1$

(b) LSTM.



$1e-7-8.08478e-1$

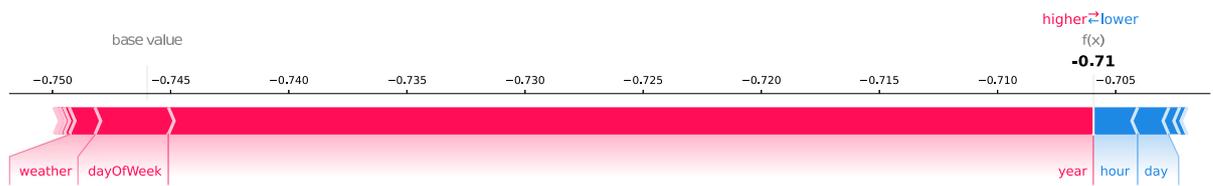
(c) GRU.



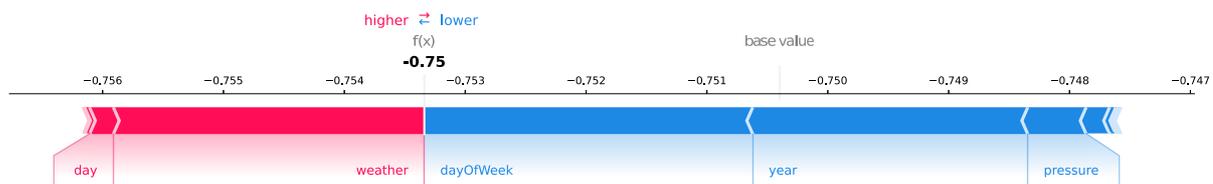
(d) TST.

Fonte: Elaborada pelo autor (2023).

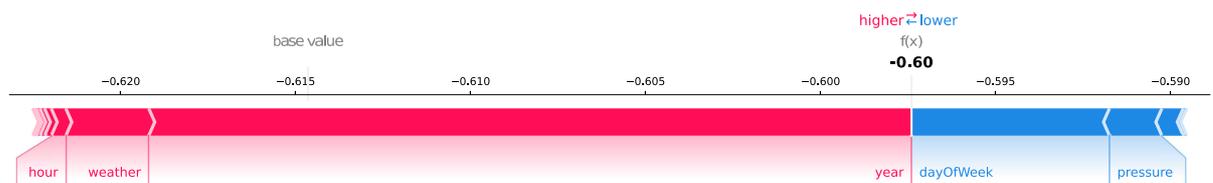
Figura 82 – Impacto de característica para a inferência de um único valor utilizando os dados da HUE, na resolução horária, para os modelos complementares.



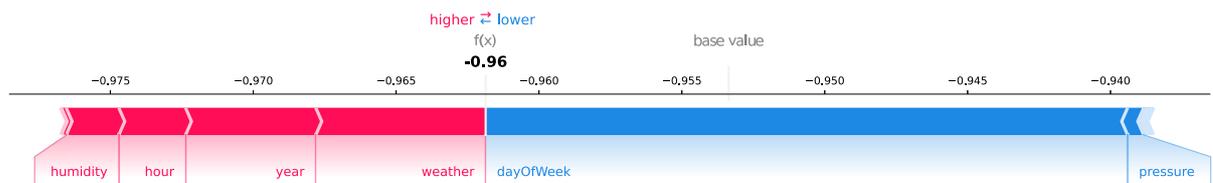
(a) ConvRNN.



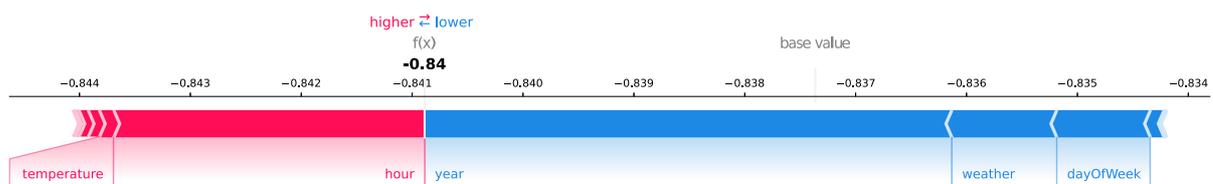
(b) MLP.



(c) FCN.



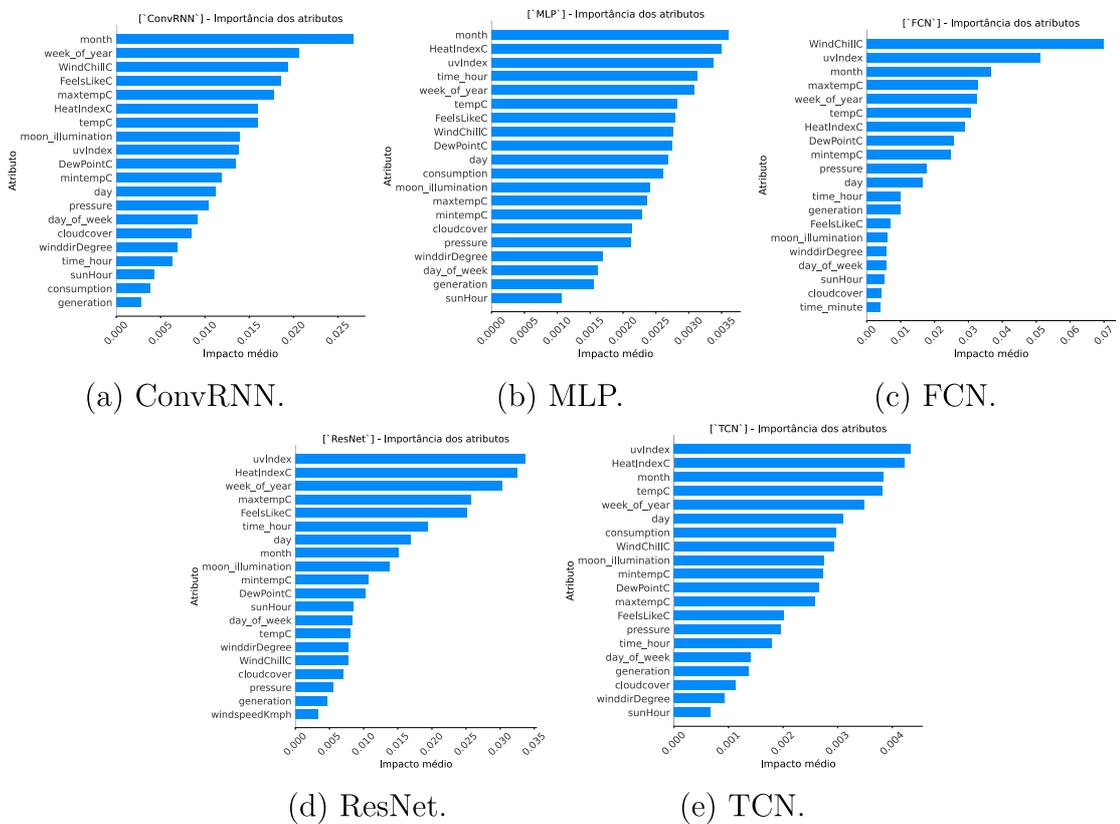
(d) ResNet.



(e) TCN.

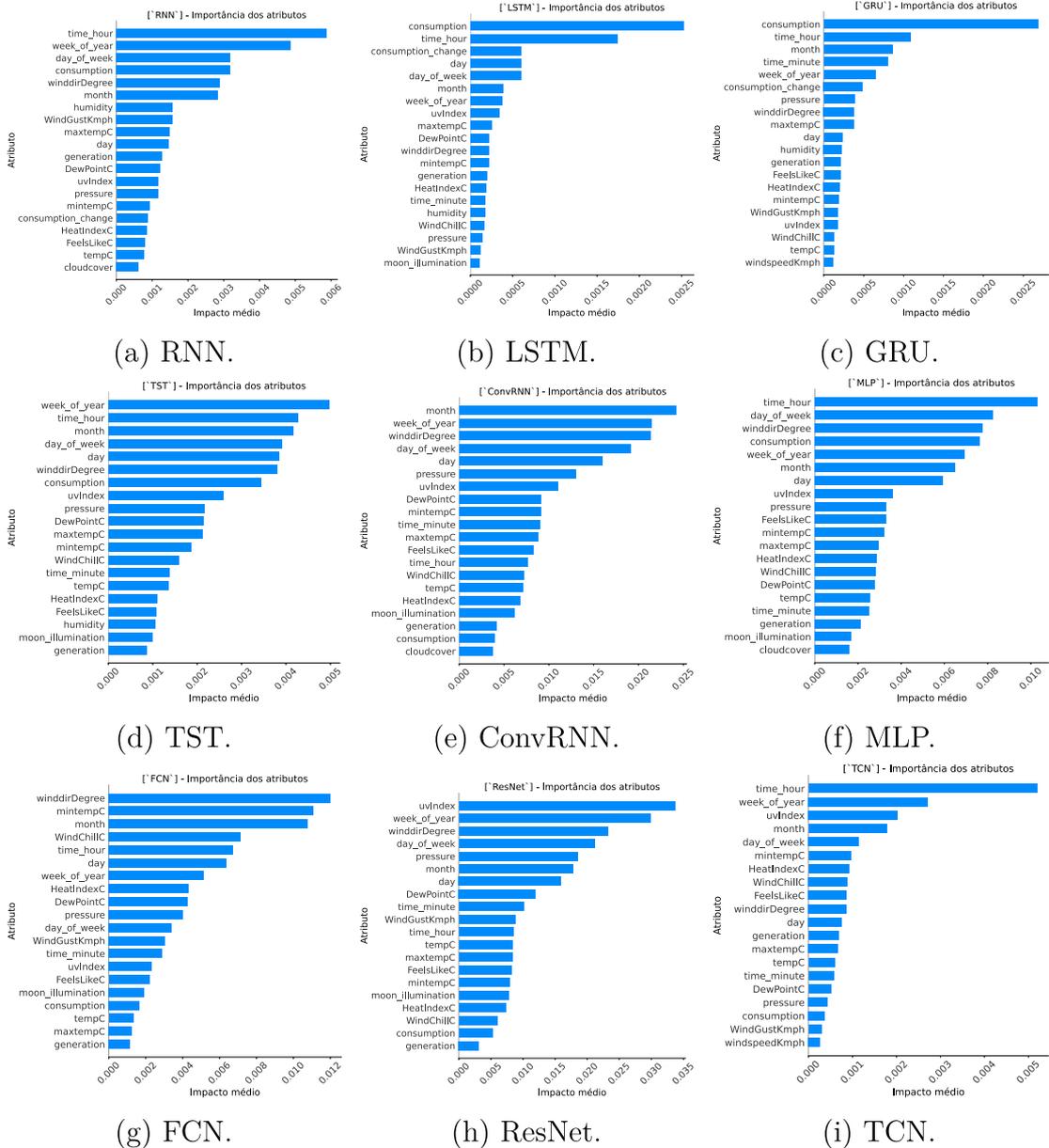
Fonte: Elaborada pelo autor (2023).

Figura 83 – Importância média das características para a inferência dos valores de demanda utilizando os dados da *Pecan Street*, na resolução de 1 minuto, para os demais modelos.



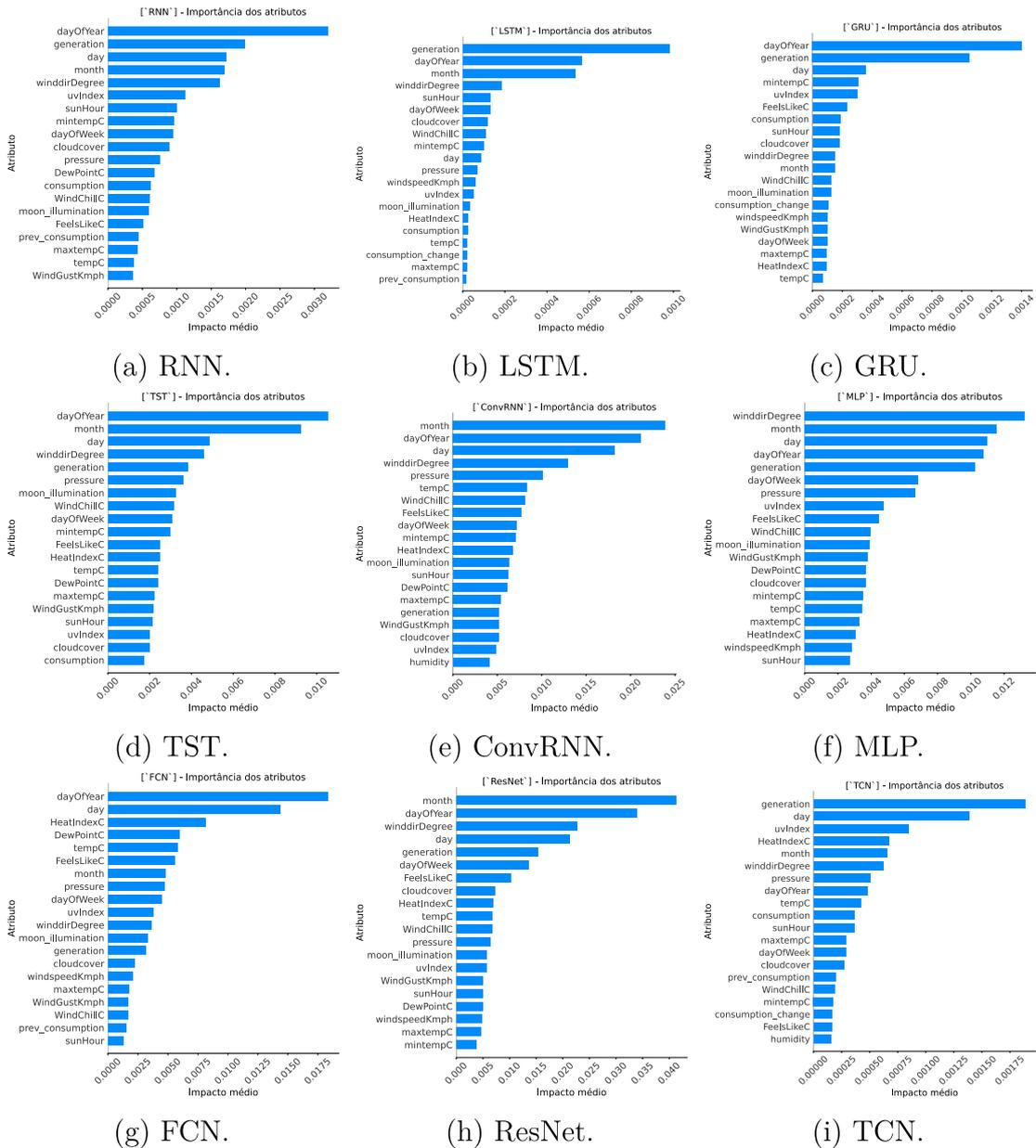
Fonte: Elaborada pelo autor (2023).

Figura 84 – Importância média das características para a inferência dos valores de demanda utilizando os dados da *Pecan Street*, na resolução de 15 minutos.



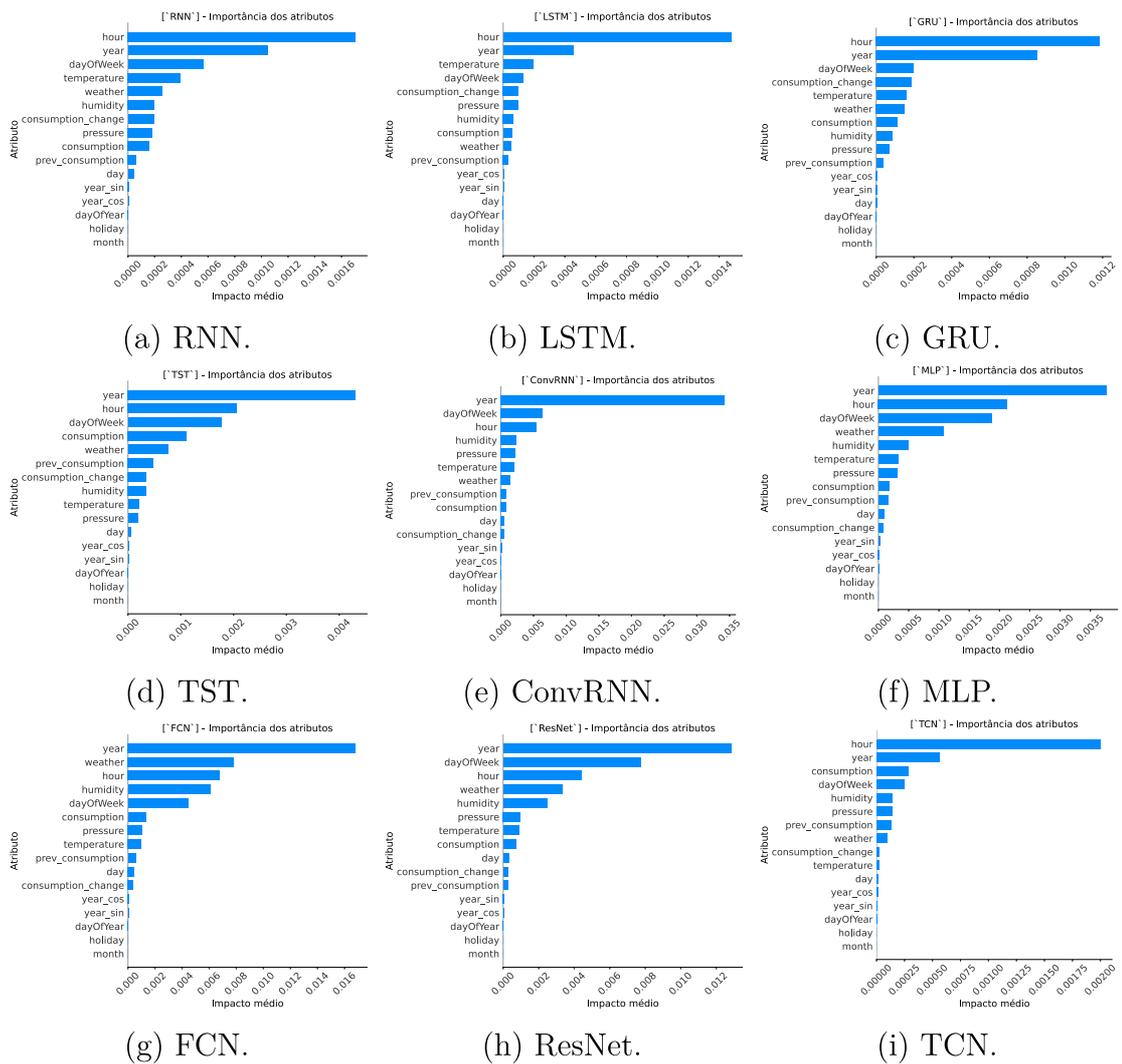
Fonte: Elaborada pelo autor (2023).

Figura 85 – Importância média das características para a inferência dos valores de demanda utilizando os dados da *Pecan Street*, na resolução horária.



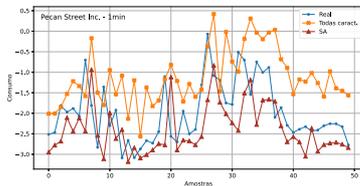
Fonte: Elaborada pelo autor (2023).

Figura 86 – Importância média das característica para a inferência dos valores de demanda utilizando os dados da HUE, na resolução horária.

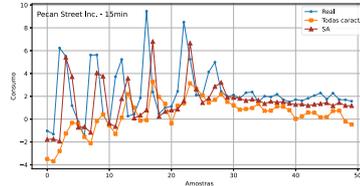


Fonte: Elaborada pelo autor (2023).

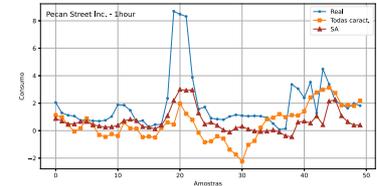
Figura 87 – Comparativo dos modelos isolados ao modelo por fusão tardia por votação, utilizando os dados da *Pecan Street*.



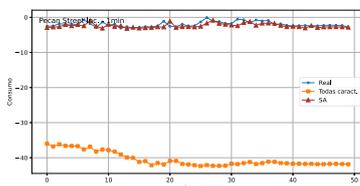
(a) Arquitetura ConvRNN - Resolução de 1 minuto.



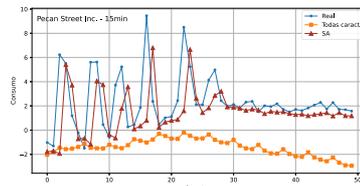
(b) Arquitetura ConvRNN - Resolução de 15 minutos.



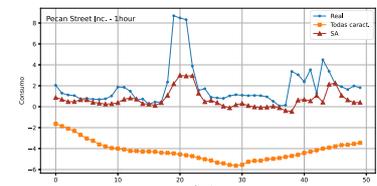
(c) Arquitetura ConvRNN - Resolução horária.



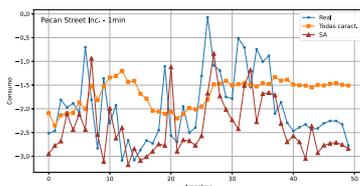
(d) Arquitetura FCN - Resolução de 1 minuto.



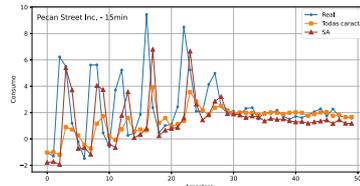
(e) Arquitetura FCN - Resolução de 15 minutos.



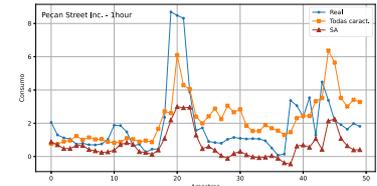
(f) Arquitetura FCN - Resolução horária.



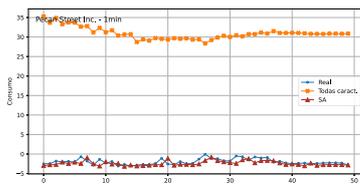
(g) Arquitetura MLP - Resolução de 1 minuto.



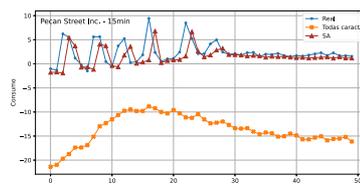
(h) Arquitetura MLP - Resolução de 15 minutos.



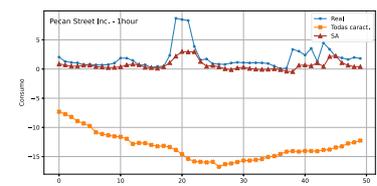
(i) Arquitetura MLP - Resolução horária.



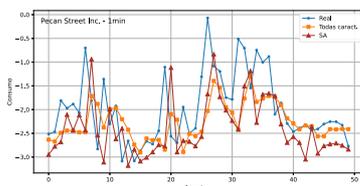
(j) Arquitetura ResNet - Resolução de 1 minuto.



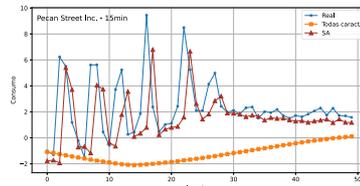
(k) Arquitetura ResNet - Resolução de 15 minutos.



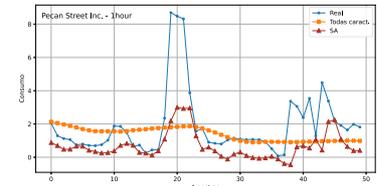
(l) Arquitetura ResNet - Resolução horária.



(m) Arquitetura TCN - Resolução de 1 minuto.



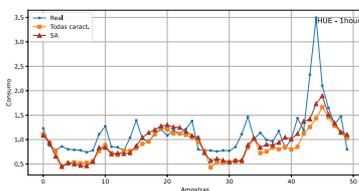
(n) Arquitetura TCN - Resolução de 15 minutos.



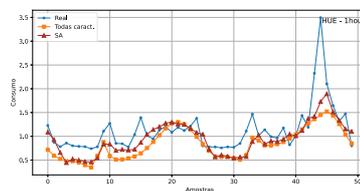
(o) Arquitetura TCN - Resolução horária.

Fonte: Elaborada pelo autor (2023).

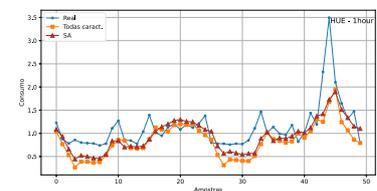
Figura 88 – Comparativo dos modelos isolados ao modelo por fusão tardia por votação, utilizando os dados da HUE.



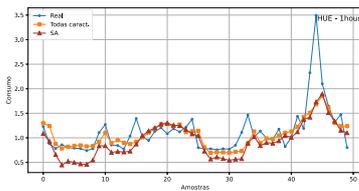
(a) Arquitetura RNN - Resolução horária.



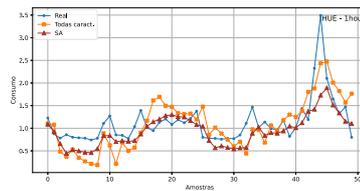
(b) Arquitetura LSTM - Resolução horária.



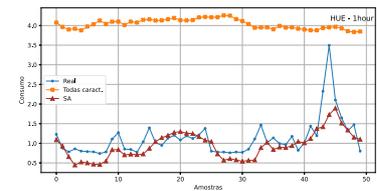
(c) Arquitetura GRU - Resolução horária.



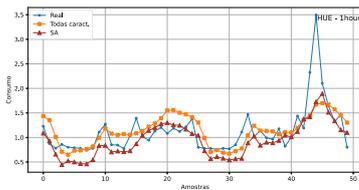
(d) Arquitetura TST - Resolução horária.



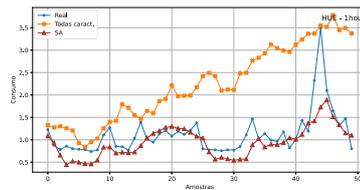
(e) Arquitetura ConvRNN - Resolução horária.



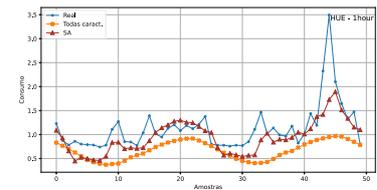
(f) Arquitetura FCN - Resolução horária.



(g) Arquitetura MLP - Resolução horária.



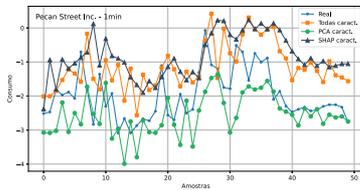
(h) Arquitetura ResNet - Resolução horária.



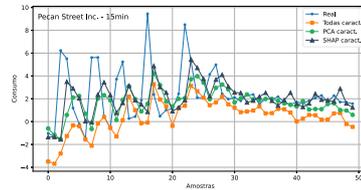
(i) Arquitetura TCN - Resolução horária.

Fonte: Elaborada pelo autor (2023).

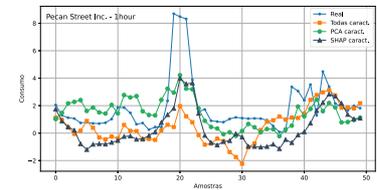
Figura 89 – Inferências realizadas utilizando características realçadas por *SHAP Values*, utilizando os dados da *Pecan Street* para os demais modelos treinados.



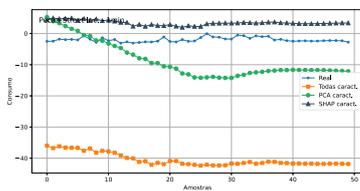
(a) Arquitetura ConvRNN - Resolução de 1 minuto.



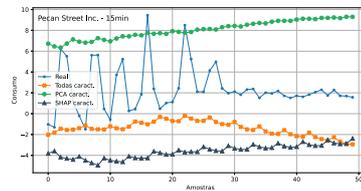
(b) Arquitetura ConvRNN - Resolução de 15 minutos.



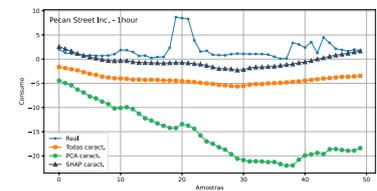
(c) Arquitetura ConvRNN - Resolução horária.



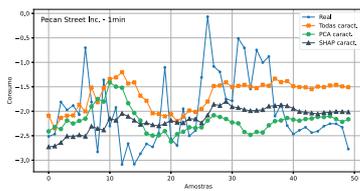
(d) Arquitetura FCN - Resolução de 1 minuto.



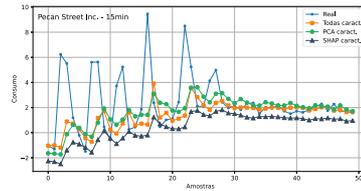
(e) Arquitetura FCN - Resolução de 15 minutos.



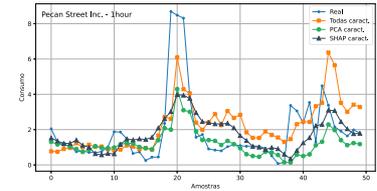
(f) Arquitetura FCN - Resolução horária.



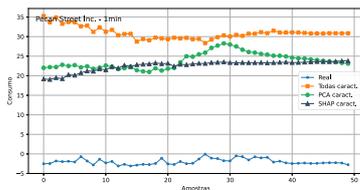
(g) Arquitetura MLP - Resolução de 1 minuto.



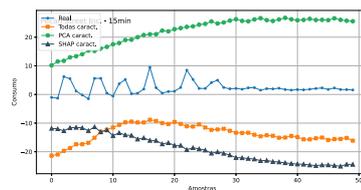
(h) Arquitetura MLP - Resolução de 15 minutos.



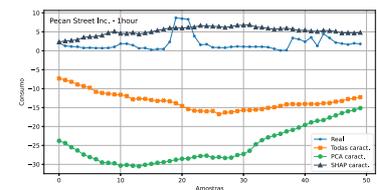
(i) Arquitetura MLP - Resolução horária.



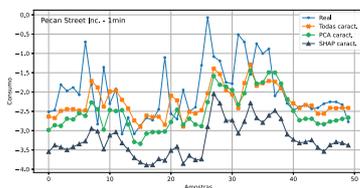
(j) Arquitetura ResNet - Resolução de 1 minuto.



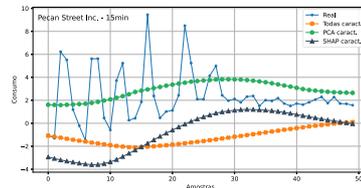
(k) Arquitetura ResNet - Resolução de 15 minutos.



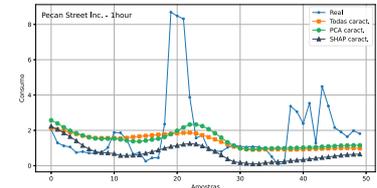
(l) Arquitetura ResNet - Resolução horária.



(m) Arquitetura TCN - Resolução de 1 minuto.



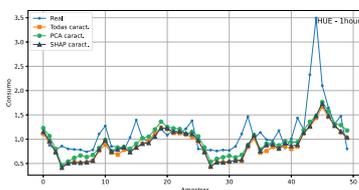
(n) Arquitetura TCN - Resolução de 15 minutos.



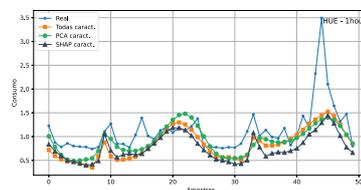
(o) Arquitetura TCN - Resolução horária.

Fonte: Elaborada pelo autor (2023).

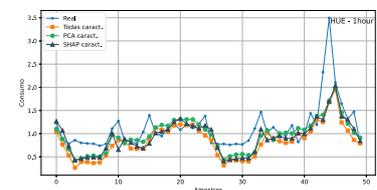
Figura 90 – Inferências realizadas utilizando características realçadas por *SHAP Values*, utilizando os dados da HUE para os demais modelos treinados.



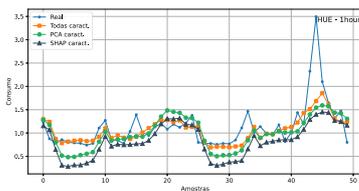
(a) Arquitetura RNN - Resolução horária.



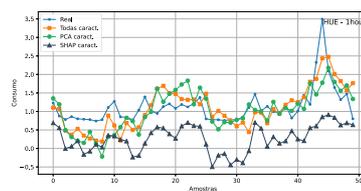
(b) Arquitetura LSTM - Resolução horária.



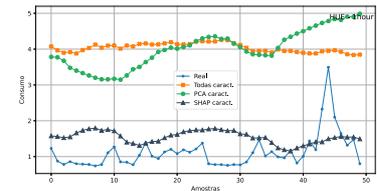
(c) Arquitetura GRU - Resolução horária.



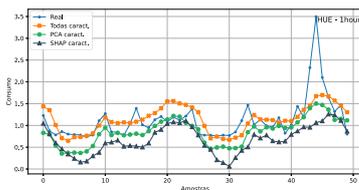
(d) Arquitetura TST - Resolução horária.



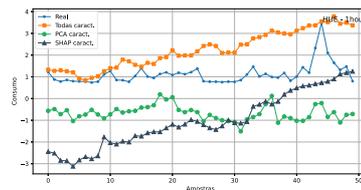
(e) Arquitetura ConvRNN - Resolução horária.



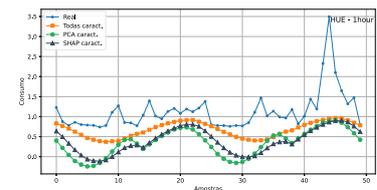
(f) Arquitetura FCN - Resolução horária.



(g) Arquitetura MLP - Resolução horária.



(h) Arquitetura ResNet - Resolução horária.



(i) Arquitetura TCN - Resolução horária.

Fonte: Elaborada pelo autor (2023).