

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ENGENHARIA GEOGRÁFICA, GEOFÍSICA E ENERGIA



Participação da geração renovável no mercado de reservas de um sistema eléctrico

João Pedro Passagem dos Santos

Mestrado em Engenharia da Energia e Ambiente

Dissertação orientada por:
Professora Doutora Ana Estanqueiro
Doutor Hugo Algarvio

2023

Resumo

A crescente penetração das fontes de energia renovável nos mercados de produção energética, aliada à mudança de paradigma nos tamanhos e centralização dos produtores, trás instabilidades e variabilidade não coompreendidas nos modelos actuais de mercado.

Por sua vez, devido à sua natureza menos constante, estes produtores têm maiores dificuldades em conseguir lucros na situação corrente.

As tecnologias de previsão estão a migrar cada vez mais para modelos de *machine learning*, com resultados mais precisos.

Este trabalho propõe a aplicação de tecnologias de redes neuronais para previsão das bandas de alocação de reserva secundária, usando os dados do mercado ibérico.

Palavras chave: sistemas de reserva, mercados de energia, redes neurais, previsões

Abstract

The growing penetration of renewable energy sources, in the energy production markets, together with the paradigm shift in size and centralization of producers, brings instabilities and variability not in the scope of current market models.

In turn, due to their less constant nature, these renewable producers have greater difficulties in making a profit in current models.

Forecast technologies are increasingly migrating to *machine learning* models, with more accurate results.

This work proposes the application of neural network technologies to forecast secondary reserve allocation bands, using data from the Iberian market.

Palavras chave: reserve systems, energy markets, neural networks, forecast

Agradecimentos

Opcional, embora no caso de dissertações que decorram no âmbito de projetos financiados por exemplo pela FCT ou programas Europeus devem ser mencionados aqui a referência e nome do projeto, e mais alguma informação de acordo com as regras de publicitação do projeto em questão.

Nome do Autor

Nomenclatura

Lista de siglas, acrónimos, abreviaturas e simbologia apresentadas por ordem alfabética.

Abreviaturas

(A/F)	Relação mássica ar/combustível
pme	Pressão média efectiva
vol	Volume

Siglas e acrónimos

ODS	Objetivos de Desenvolvimento Sustentável
vRES	variable Renewable Energy Systems
MIBEL	Mercado Ibérico de Eletricidade
APA	Agência Portuguesa do Ambiente
EDP	Eletricidade De Portugal
RNN	Recurrent neural network
CNN	Convolutional neural network
LSTM	Long short-term memory
AR	dec
MA	dec
ARMA	dec
ARIMA	dec
SARIMA	dec

Simbologia

A	Área
η	Eficiência
p	Pressão
T	Temperatura

Índice

Resumo	i
Abstract	ii
Agradecimentos	iii
Nomenclatura	iv
List of Figures	viii
List of Tables	ix
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos e perguntas de investigação	2
1.3 Organização do documento	2
2 Revisão bibliográfica	3
3 Contextos	4
3.1 Mercado de Serviços de Sistema	4
4 Dados	5
4.1 Dados Utilizados	5
4.1.1 Aquisição dos Dados	5
4.2 Estudo dos dados	5
4.2.1 Correlações	7
4.2.1.1 Correlações entre atributos	7
4.2.1.2 Correlações Temporais	9
4.2.2 Agrupamento	10
4.3 Tratamento dos dados	12
4.4 Considerações adicionais	14
5 Arquitecturas de Modelos	15
5.1 Camadas	15
5.1.1 Dense	15
5.1.2 Convolution	15
5.1.3 MaxPooling	15
5.1.4 Dropout	15

5.2	Blocos	16
5.2.1	Bloco Dense	17
5.2.2	Blcoo CNN	17
5.2.3	Blcoo LSTM	17
5.3	Arquiteturas	18
5.3.1	Vanilla	18
5.3.2	Stacked	18
5.3.3	MultiHead	18
5.3.4	MultiTail	18
5.3.5	UNET	18
5.4	Considerações adicionais	19
6	Métodos	20
6.1	Benchmark	20
6.2	Modelos estatísticos	21
6.2.1	Univariate Analysis	21
6.2.1.1	AR	21
6.2.1.2	MA	22
6.2.2	ARMA	22
6.2.3	ARIMA	23
6.2.4	SARIMA	24
6.2.5	Resultados	26
6.3	Forecat	26
6.3.1	Construtor de modelos	26
6.3.2	Gerador de dados	26
6.4	Treino e Resultados	27
6.4.1	Arquiteturas e numeros de epocas	27
6.4.2	Funções de Perda (Loss)	28
6.4.3	Hiperparametrização	28
6.4.3.1	Activação	28
6.4.3.2	Optimizadores	28
6.4.4	Janelas Temporais	28
6.4.5	Classificação	28
6.4.6	Pesos	29
6.4.6.1	Modelos lineares	29
6.4.6.2	Modelos Lineares e de Classificação	29
6.5	Considerações adicionais	29
7	Resultados e discussão	30
7.1	Métricas	30
7.2	Experiências	30
7.2.1	Arquiteturas e numeros de epocas	31
7.2.1.1	Arquiteturas	31
7.2.1.2	Epocas	31
7.2.2	Funções de Perda (Loss)	31

7.2.3	Hiperparametrização	31
7.2.3.1	Activação	31
7.2.3.2	Optimizaçao	32
7.2.4	Janelas Temporais	32
7.2.5	Classificação	32
7.2.6	Pesos	32
8	Conclusões e sugestões futuras	34
9	Referências	35
A	Anexos	37

List of Figures

Figura 1.1 Objetivos desenvolvimento sustentável ONU	1
Figura 4.1 Serie Temporal dos dados alvo	6
Figura 4.2 Janelas Temporais dos dados alvo	6
Figura 4.3 Frequência dos dados alvos	7
Figura 4.4 Correlação entre atributos	8
Figura 4.5 Valores de correlação entre atributos	9
Figura 4.6 Serie Temporal dos dados alvo	9
Figura 4.7 Serie Temporal dos dados alvo	10
Figura 4.8 Serie Temporal dos dados alvo	11
Figura 4.9 Histograma das classes	12
Figura 4.10 Outliers	13
Figura 4.11 Dados em falta	14
Figura 5.1 Ilustração do efeito da camada de dropout	16
Figura 5.2 Bloco Densa	17
Figura 6.1 Serie Temporal do benchmark	20
Figura 6.2 Serie Temporal do benchmark 2021	21
Figura 6.3 Previsões 2021 com modelo MA	22
Figura 6.4 Previsões 2021 com modelo ARMA	23
Figura 6.5 Previsões 2021 com modelo ARIMA	24
Figura 6.6 Previsões 2021 com modelo SARIMA	25
Figura 6.7 Previsões 2021 com modelo VAR	25
Figura 7.1 Exemplo de como considerar um gráfico	33
Figura A.1 Exemplo de como considerar um gráfico nos anexos.	37

List of Tables

Tabela 6.1 Dados Benchmark	20
Tabela 6.2 Dados Benchmark de validação	21
Tabela 6.3 Resultados modelos Estatísticos	26
Tabela 7.1 Resultados Arquitecturas	31
Tabela 7.2 Resultados Funções de perda	31
Tabela 7.3 Resultados Ativação	31
Tabela 7.4 Resultados Optimizadores	32
Tabela 7.5 Resultados Janelas Temporais	32
Tabela 7.6 Resultados Janelas Temporais	32
Tabela 7.7 Resultados Janelas Temporais	32
Tabela A.1 Isto é um exemplo de uma tabela. Se fôr igual(copiada) a outro autor deve ser pedido autorização para reproduzir.	37

Introdução

1.1 Enquadramento

Esta dissertação encontra-se no âmbito do projecto [TradeRES](#), o qual estuda um sistema de mercado eléctricos que consiga dar resposta às necessidades da sociedade num sistema quase todo renovável. Tendo as características para se integrar nos [ODS 1.1](#).

O estudo da acessibilidade das energias renováveis ao mercado vigente integra-se nos ODS nº7, “Energia Renováveis e Acessíveis”, indo directamente de encontro a um dos pontos deste objectivo: 7.2.1 “Peso das energias renováveis no consumo total final de energia”. Por meio deste objectivo, a participação das renováveis no mercado faz também cumprir, embora indirectamente, o objectivo nº8 “Trabalho Digno e Crescimento Económico”, através do ponto 8.4, onde é dada primazia à eficiência dos recursos globais no consumo e na produção. Indirectamente, pois, ao haver um melhor uso das renováveis, o uso de energias não limpas vai diminuir, melhorando a gestão de recursos, e baixando o consumo de recursos naturais não renováveis.

Por último podemos incluir o objectivo nº13, “Acção Climática”, no qual, referimos de novo a diminuição de consumo de recursos finitos, mas mais importante, a melhor gestão de recursos renováveis. Promovendo o planeamento e estratégias de combate a emissões de gases de efeito estufa.



Figure 1.1: Objetivos desenvolvimento sustentável ONU

1.2 Objetivos e perguntas de investigação

Foram aprovadas a nível europeu (2020), medidas de alteração aos serviços de sistema, que serão seguidas pelos Estados-Membros. Nesta dissertação far-se-á a aplicação dessas medidas, identificando as melhorias face ao desenho actual e, avaliando se as novas medidas serão suficientes para assegurar a operação de um sistema eléctrico 100% renovável, eventualmente identificando acções adicionais que garantam a robustez e segurança do sistema eléctrico sem recurso a combustíveis fósseis.

- a) É positivo para as vRES participar no mercado de reserva?
- b) Como configurar essa participação para otimizar o lucro do ponto de vista das vRES?
- c) Essa participação é positiva para o sistema eléctrico num todo?

Como forma de responder a estas questões vamos tentar utilizar dados de previsão de geradores renováveis para prever a energia necessária para alocação secundária.

Os valores de previsão deste mercado estão actualmente longe do que realmente é utilizado, fazendo com que a alocação no dia anterior não seja em conformidade com o necessário, logo não existindo o optimo aproveitamento de recursos.

O objectivo neste trabalho é perceber se utilizando técnicas de *Machine Learning* (Aprendizagem automática) conseguimos obter previsões mais próximas da energia utilizando, possibilitando assim uma melhor gestão das alocações, logo um menor gasto de recursos, energéticos mas especialmente financeiros.

1.3 Organização do documento

Este documento está dividido em capítulos. Sendo que os primeiros apresentam uma introdução às ideias e temas 1.3, o estado de arte do temas na literatura publicada 2, e por fim uma contextualização dos temas abordados 3.1.

Segue um capítulo de explicação dos dados usado 4.4, onde se apresentam os mesmos juntamente com alguns estudos preliminares para compreender a natureza e qualidades dos mesmos.

O capítulo seguinte entra no âmbito experimental da dissertação, onde se apresenta as diferentes arquitecturas 5.4 utilizadas, incluindo uma explicação dos componentes das mesmas.

Os capítulos 6 e 7 são bastante paralelos, sendo que o sexto 6.5 apresenta a metodologia do trabalho, e explica todas as experiências, e o sétimo 7.2.6 apresenta apenas os resultados e conclusões experiência a experiência.

Termina com um capítulo conclusivo 8 onde são avaliadas as experiências como um todo, e o seu impacto no âmbito dos mercados de reserva.

Revisão bibliográfica

A análise de séries temporais é um tema recorrente em pesquisa. Especialmente para previsões.

Desde as previsões para mercados de acções[1], fenómenos meteorológicos[2], e especialmente mercados energéticos, onde se quer ter em consideração o impacto das gerações mais voláteis.

As energias renováveis, devido à sua natureza, são as produções mais voláteis, logo alvo de estudo ideal para estas tecnologias[3], energia eólica[4], energia solar[5], aplicabilidade dos vários sistemas[6], procura[7]. Sendo que cada problema já apresenta arquiteturas e soluções diferentes, como a geração de energia fotovoltaica em casas pode ser melhor prevista com LSTM[8] mas também com uso de SVM[9] As várias faces destas tecnologias estão optimamente apresentadas em[10]

Para o estudo de previsões de séries temporais chega a ser o caso se pesquisar primeiramente com *deep learning*, antes de procurar outras soluções.

Em [11] é visto o impacto dessa decisão, e se realmente compensa emergir em machine learning. O trabalho conclui que modelos simples, com alguma engenharia de atributos inteligente, consegue competir, ou mesmo passar as qualidades de redes neuronais profundas.

Esta conclusão mostra também que por vezes a procura por modelos mais complexos não compensa, e que cada problema/dataset deve ter a sua própria investigação e conclusão, consoante a quantidade/qualidade de recursos disponíveis.

Os sistema de reserva de frequência do mercado espanhol já foram alvo de análise predictivas com modelos de redes neuronais profundas [12].

Neste trabalho procurou-se prever o preço da banda de reserva secundaria, sendo que os melhores modelos atingiram metricas de MASE de cerca de 64%, o que não foi considerado um bom resultado.

Para o trabalho presente, isto mostra que a resolução linear destes dados pode ser difícil. O trabalho utilizou 32 variáveis abertas pela TSO espanhola, e também conclui que o aumento de numero de variáveis não melhorou os modelos. Para este trabalho o impacto deste conhecimento é pouco, visto estudarmos com um conjunto fixo de variáveis, e sendo que queremos perceber se conseguimos prever com esse conjunto.

Embora estes dados tragam um estudo parecido, usando dados da mesma fonte, o objectivo é ligeiramente diferente.

Contextos

3.1 Mercado de Serviços de Sistema

O mercado de serviço de sistema é parte integrante dos mercados de energia e mantém responsabilidade sobre a segurança do mesmo.[13]

Serve para garantir o equilíbrio entre a energia produzido e a consumida. Esta qualidade e segurança é controlada através da frequência e da potência activa, controlo de tensão e potência reactiva, arranque automático e outras técnicas de sistemas [14] [15].

Neste caso de estudo estamos interessados nos serviços de controlo de frequência. A nível europeu estes serviços são impostos pela ENTSO-E (*European Network of Transmission System Operators for Electricity*), e a operação dos mesmos é da responsabilidade dos TSO (*Transmission System Operator* ou *Operador da Rede de Transporte*) nacionais.

Para manter o controlo de frequência o gestor de sistema deverá manter reservas para responder às diferenças entre a energia consumida e produzida na rede, que deve ser mantida em equilíbrio. Quando o serviço de sistema precisa de actuar para manter a frequência no seu valor nominal, 50Hz na Europa, isto é feito alterando a potência activa dos geradores.

Quando é necessário um aumento na potência chama-se a isto Banda de Reserva/Regulação a Subir, e quando é necessária uma diminuição chama-se à mesma a Descer.

Para isto, no mercado ibérico, a tarefa é dividida em três reservas, primária, secundária e terciária. Esta divisão assenta no tempo de resposta que os sistemas precisam de ter, e na capacidade de actuação (MWh/Hz).

A reserva secundária, como sistema de segurança à reserva primária, regula-se pelo mercado de banda das reservas secundárias, que decorre no dia anterior ao que será necessário utilização da mesma.

Este valor alocado tem um custo para as operadoras, como tal a previsão do mesmo é importante para a gestão destes sistemas de segurança. Estas previsões são feitas através de estatísticas dos sistemas, e tendo em conta as áreas de balanço que o mesmo têm.

Uma melhor previsão deste valor poderia levar a uma poupança, tanto financeira, como de recursos.

Estas previsões são feitas ao uso de formulas. Que por si só não preveem a variabilidade dos sistemas de produção de energia renovável. Esta variabilidade sendo dificilmente previsível, tem sido alvo de estudo com modelos de *machine learning* [**<empty citation>**]

Com bons resultados apresentados em estudo de energias renováveis, a aplicação dos mesmos métodos para as reservas de sistema parece um passo natural.

Dados

4.1 Dados Utilizados

Os dados em estudo são do mercado energético espanhol, retirados do site da [ESIOS](#).

indicators	names	units
632	SecondaryReserveAllocationAUpward	MW
633	SecondaryReserveAllocationADownward	MW
680	UpwardUsedSecondaryReserveEnergy	MWh
681	DownwardUsedSecondaryReserveEnergy	MWh
1777	WindD+1DailyForecast	MWh
1779	PhotovoltaicD+1DailyForecast	MWh
1775	DemandD+1DailyForecast	MWh
10258	TotalBaseDailyOperatingSchedulePBFGeneration	MWh
14	BaseDailyOperatingSchedulePBFSolarPV	MWh
10073	BaseDailyOperatingSchedulePBFWind	MWh
10186	BaseDailyOperatingShedulePBFTotalBalanceInterconnections	MWh

4.1.1 Aquisição dos Dados

No âmbito da automatização destes dados foi modificado o repositório [ESIOS](#) para ser usado como uma biblioteca de python, aberta, em pypi.

Sendo uma ferramenta mais facilmente acessível para a extrair dados do mercado espanhol, [pyesios](#):

4.2 Estudo dos dados

Os dados que propunho a prever são: "UpwardUsedSecondaryReserveEnergy",
"DownwardUsedSecondaryReserveEnergy"

4. DADOS

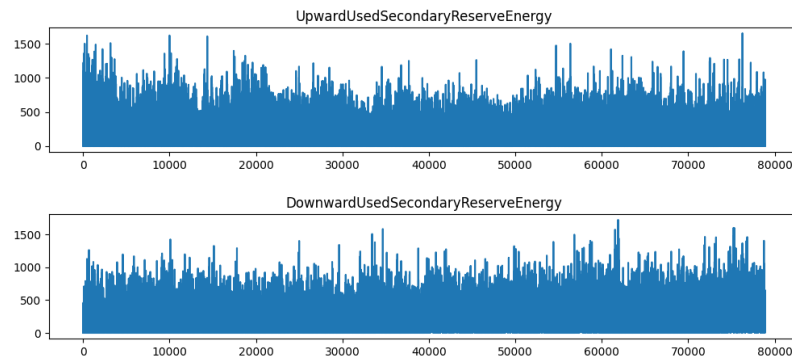


Figure 4.1: Serie Temporal dos dados alvo

Para termos uma melhor percepção dos mesmos segue algumas janelas temporais mais pequenas.

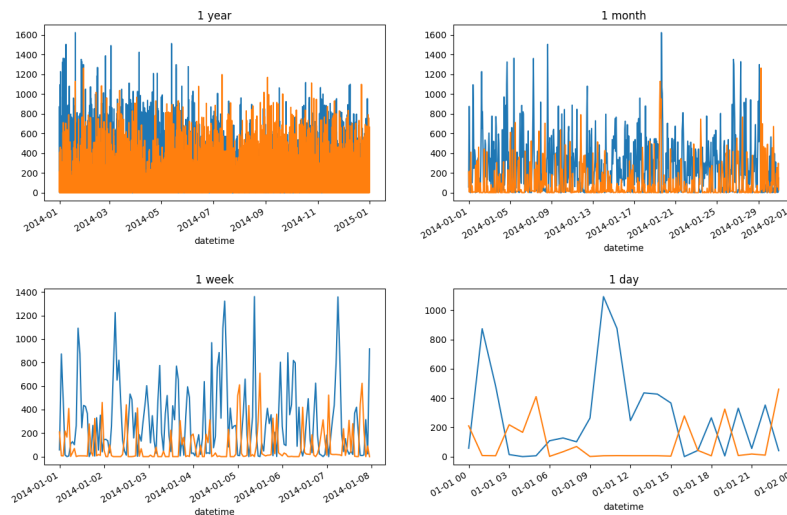


Figure 4.2: Janelas Temporais dos dados alvo

Estas mostram claramente que ambos os atributos mantêm um comportamento tanto discreto, como linear, isto é, que ou existe algum valor, ou é zero, e se existe valor este tem comportamento linear.

A distribuição destes dados é claramente exponencial. O que é importante para a escolha de alguns parametros no modelação

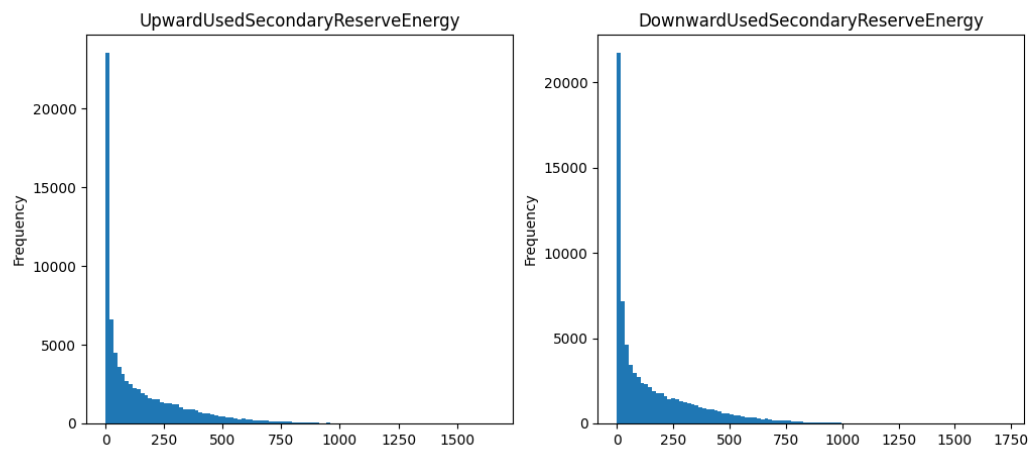


Figure 4.3: Frequência dos dados alvos

4.2.1 Correlações

4.2.1.1 Correlações entre atributos

Os modelos vão depender bastante de correlação entre variáveis.

Nesta secção queremos tentar identificar se há visíveis relações entre as variáveis, e se há relações temporais visíveis nas colunas alvo.

4. DADOS

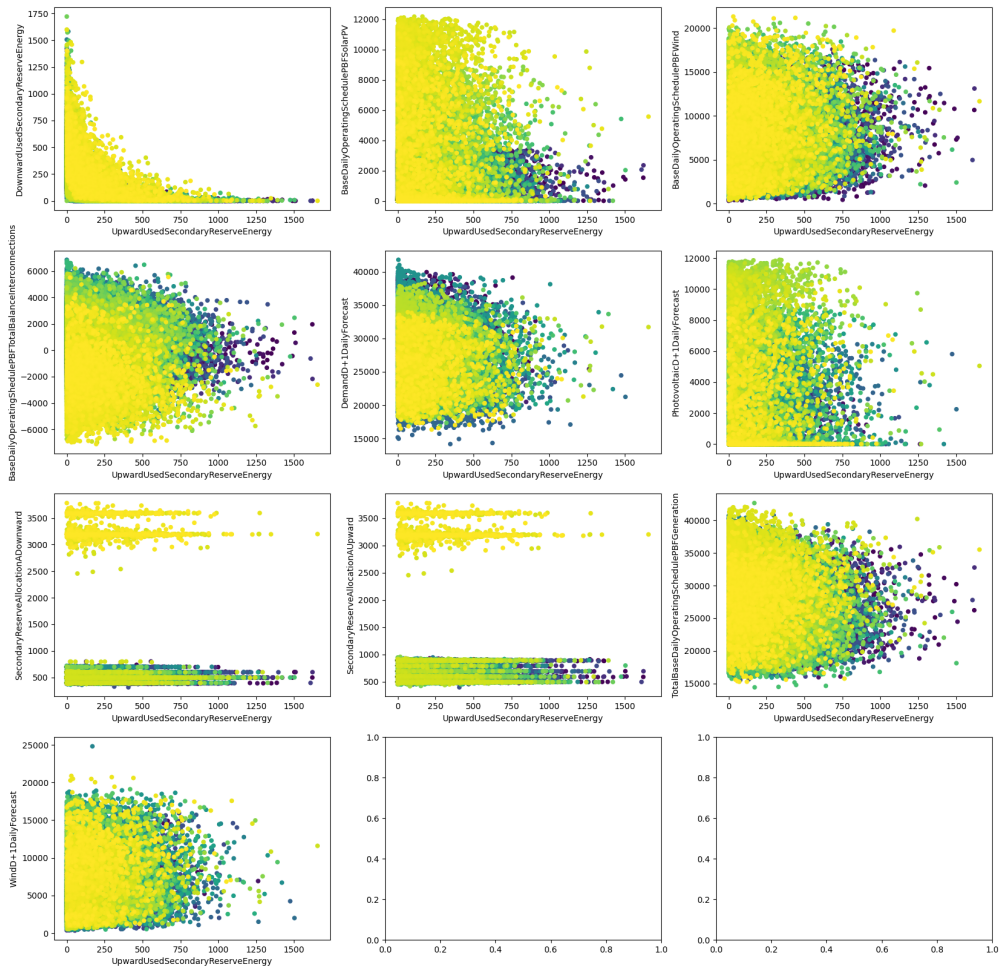


Figure 4.4: Correlação entre atributos

As correlações entre variáveis parecem muito escassas o que apresenta já que a previsão deste dados usando estas variáveis vai ser um problema difícil.

Por norma é feito uma seleção de atributos baseado nestas correlações, eliminando assim os atributos que ajudam menos, ou até prejudicam os modelos.

Segue os valores de correlação onde podemos ver numericamente que existe muito pouca correlação entre os atributos.

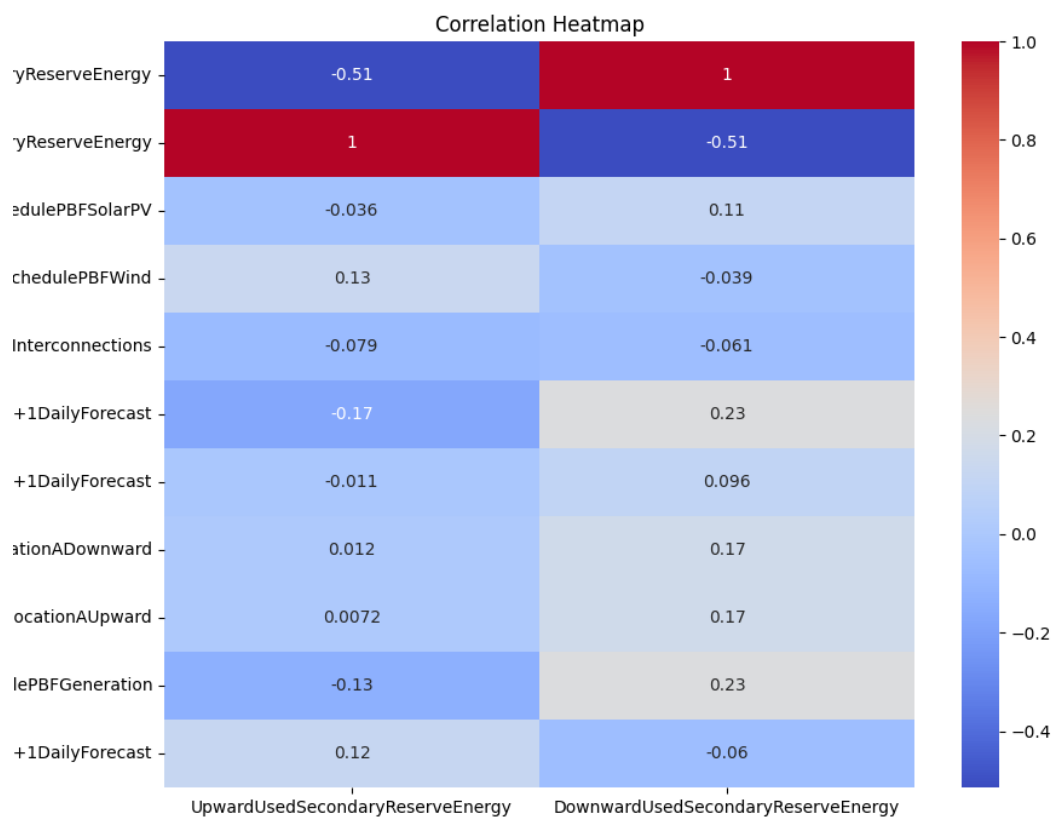


Figure 4.5: Valores de correlação entre atributos

4.2.1.2 Correlações Temporais

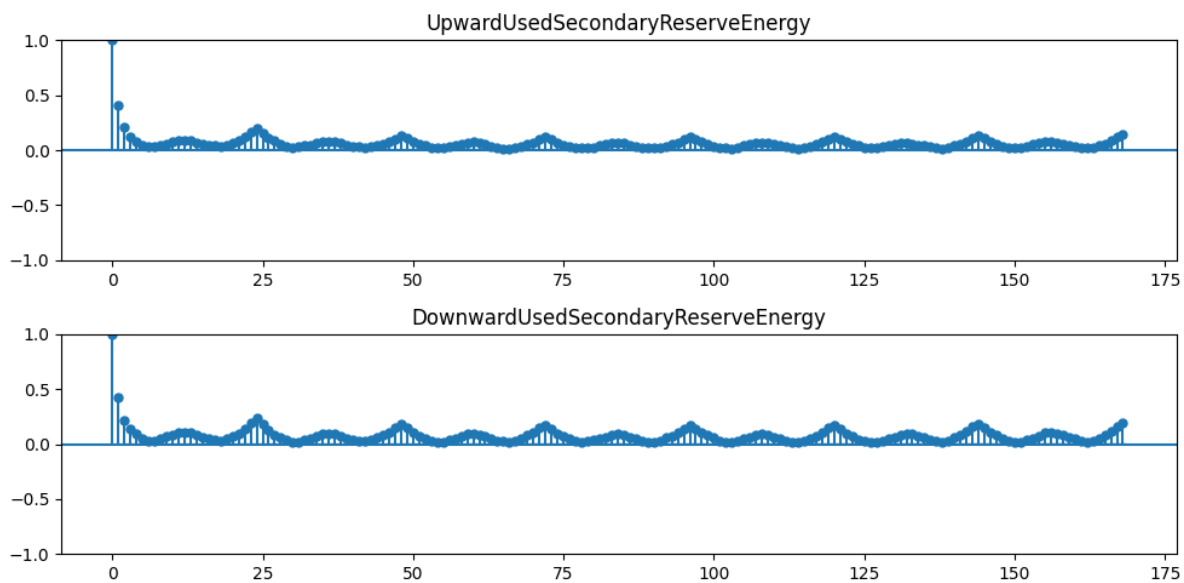


Figure 4.6: Serie Temporal dos dados alvo

4. DADOS

A autocorrelação, em ambos os alvos, é mais forte nas 3 horas mais próximas, e nos pontos com diferença de 12 e 24 horas.

É de notar que estes valores são baixos, prometendo já também uma baixa regressividade temporal.

Outro ponto a denotar é que os objectos não têm um comportamento completamente linear, i.e., parece existir um comportamento discreto na questão ser alocado ou não estas reservas secundárias, e caso seja alocado, aí existir alguma linearidade.

Logo qualquer tipo de modelação terá de resolver primeiramente este problema.

Estas relações mostram que em termos de atributos usados vai ser um desafio complicado para qualquer tipo de modelo.

No âmbito desta dissertação queremos verificar a qualidade das previsões usando estes mesmos atributos, logo, não será feita seleção dos mesmos.

A nível da relação temporal, a maior parte dos modelos que iremos testar aplica uma janela na dimensão temporal, usando todos os valores nessa janela, e aplicando os pesos nessas distâncias que mais se enquadram. Logo também não é relevante escolher apenas as distâncias temporais com maior correlação, pois os modelos vão fazer essa pesagem.

4.2.2 Agrupamento

Uma das possibilidades na modelação será a utilização de grupo de valores, classes, em conjunto com a linearidade.

Devido ao comportamento não exclusivamente linear é também estudado as possíveis agregações (clustering) em que podemos dividir os valores em classe.

Tendo por base que uma das classes é o valor zero, devido ao comportamento não linear desta série, vamos apenas testar quantos, e quais, as melhores classes em que podemos dividir os dados.

É realizado o teste de silhueta (ref) e o teste do cotovelo (ref), que se baseiam nos resultados de silhueta do modelo GMM (ref) e nos valores de inércia do modelo K-Means (ref).

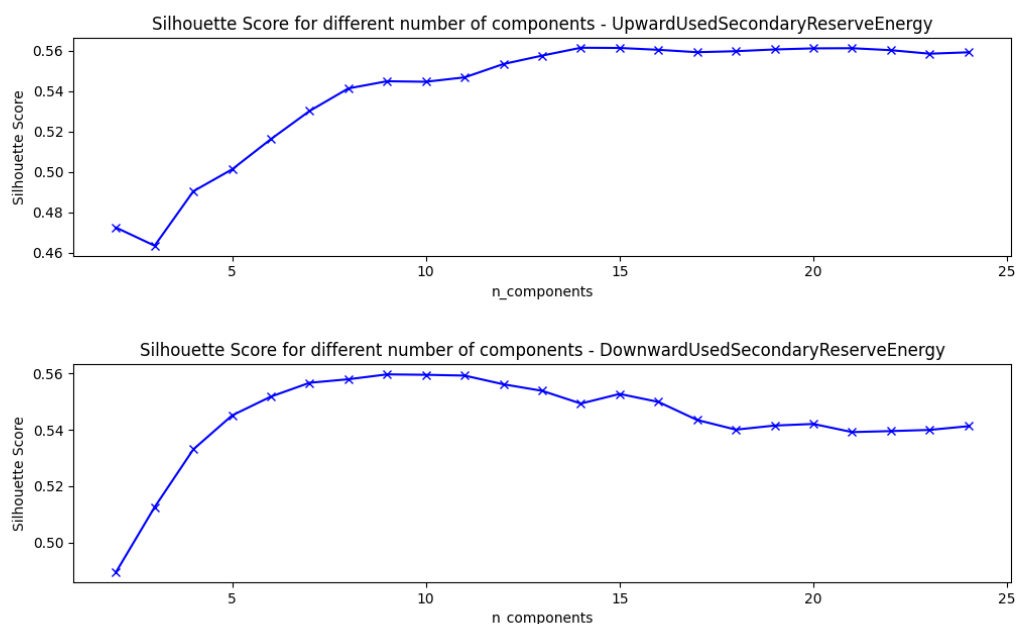


Figure 4.7: Serie Temporal dos dados alvo

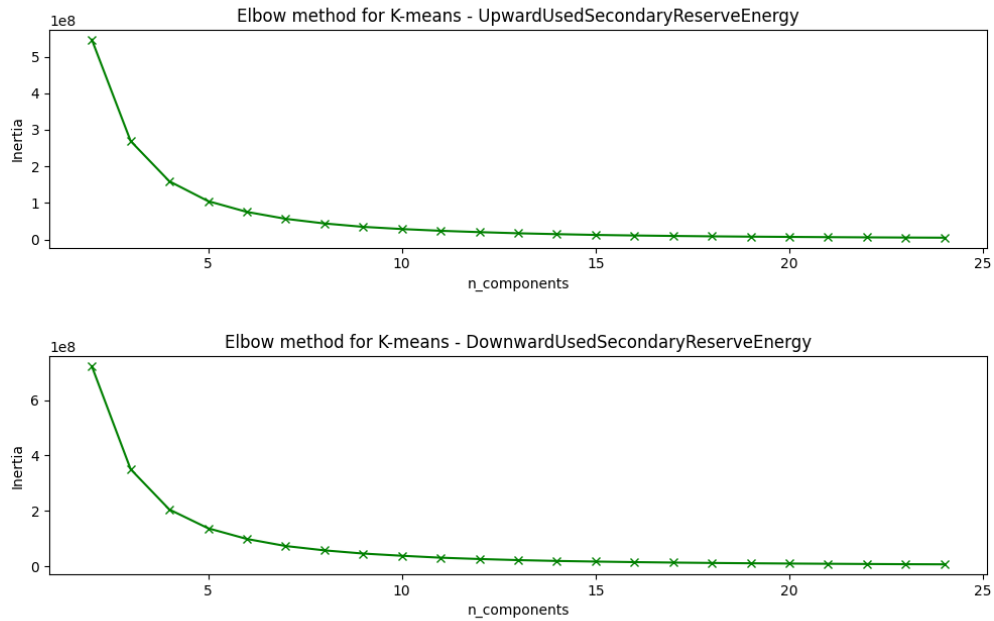


Figure 4.8: Serie Temporal dos dados alvo

Ambos os casos apontam um assintota na relação interna dos clusters, a partir de cerca de 5 clusters, sendo que o melhor valor dos verificados seria com 14 clusters em "UpwardUsedSecondaryReserveEnergy" e 9 em "DownwardUsedSecondaryReserveEnergy".

Para a nossa questão, queremos algumas classes, mas quanto menos classes mais facil será para os modelos correctamente identificar a que classe pertence. Logo para os valores apresentados, escolhemos 5 clusters, sendo que este já pode ser um numero elevado de classes, logo usemos 3 clusters se os modelos tiverem muita dificuldade com 5.

O método do cotovelo apresenta como numero ideal de clusters 4, mas sendo que K-Means é um metodo mais apropriado para distribuições normais, em que a distancia nos limites dos clusters não varia, deixamos apenas informativo.

O histograma das divisões pode ser visto em baixo. O valor é a sua própria classe para além das apresentadas.

4. DADOS

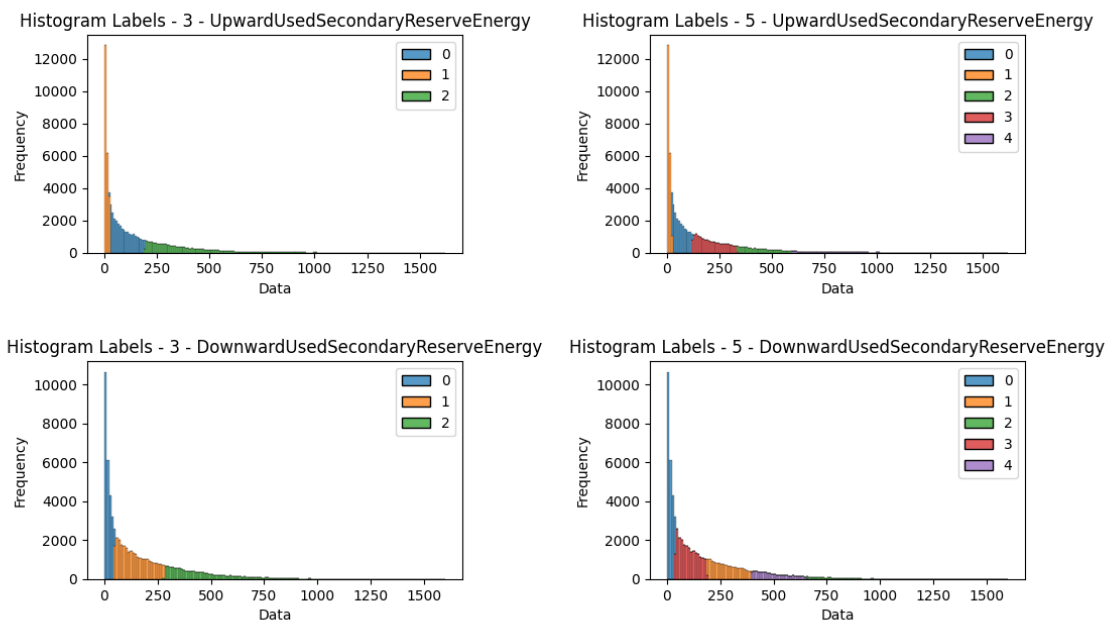


Figure 4.9: Histograma das classes

Os limites retirados são:

Atributo	Nº Clusters	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5
UpwardUsedSecondaryReserveEnergy	3	26.2	"26.3-194.9"	195.0		
UpwardUsedSecondaryReserveEnergy	5	20.2	"20.3-119.2"	"119.3-331.9"	"332.0-592.6"	592.7
DownwardUsedSecondaryReserveEnergy	3	47.0	"47.1-281.9"	282.0		
DownwardUsedSecondaryReserveEnergy	5	38.9	"39.0-186.9"	"187.0-393.1"	"393.2-658.0"	658.2

4.3 Tratamento dos dados

Normalização

A normalização foi deixada por ser aprendida nos modelos, sendo que todos têm como segunda camada, uma de normalização.

Limpeza

Podemos ver pelos graficos seguintes que a existem alguns outliers, sendo estes definidos como 3 desvios padrão de distância à média.

Estes graficos mostram também que existe uma variação do que são os valores normais de cada atributo a nível temporal. Logo um método de limpeza não se poderia basear apenas numa definição geral de outliers, mas teria de ser feito em janelas temporais.

Pelo mesmo argumento e visto que os outliers fazem parte do que queremos também descobrir, não é aplicada nenhum método de remoção dos mesmo, sendo os dados passados a cru para os modelos.



Figure 4.10: Outliers

Outra análise desta variação dos atributos a nível temporal leva-nos a que qualquer divisão dos dados para treino e teste deva levar as variações em consideração. Isto sendo que o treino deve ter representatividade de todas, ou maior parte, das condições diferentes.

Visto estarmos a usar dados de 2014 a 2022, argumento até que podemos nem fazer divisão no treino, e usar como teste e validação dados de 2023.

Dados em falta (Missing Data)

Estudemos também o caso de dados em falta. Alguns destes atributos têm certas entradass vazias, e como podemos ver alguns não têm alguns anos inteiros.

Como queremos usar o maximo de dados possiveis iremos usar tecnicas de imputing nesses dados.

Podemos ver que temos dados em falta de varios anos, em três atributos, e um tem algumas horas esporadicas em falta nos primeiros anos.

4. DADOS

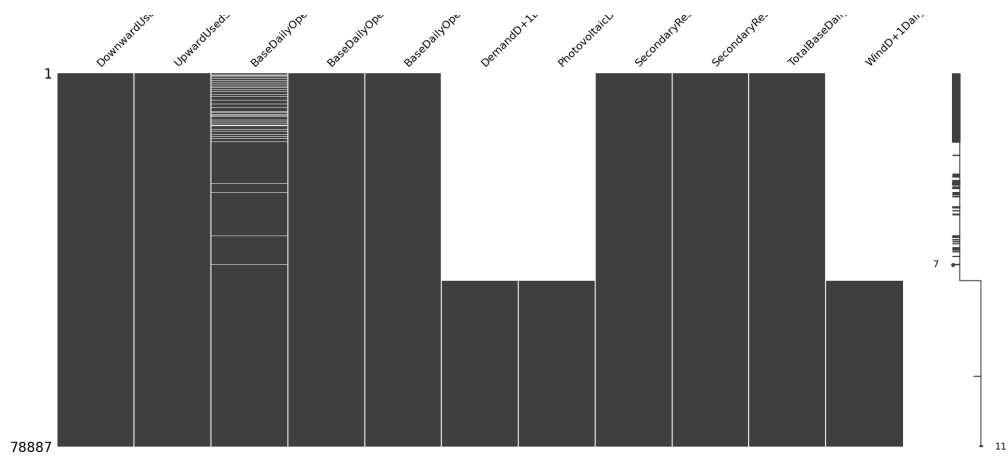


Figure 4.11: Dados em falta

Vamos aplicar o método experimental [IterativeImputer](#) da biblioteca de python [sklearn](#).

Este método é baseado nos trabalhos de[16] e de[17]

Por último foi adicionado aos dados mais atributos, sendo eles todos de cariz temporal. É adicionado atributos correspondentes à hora, ao dia do ano, ao dia da semana, ao dia do mês, mês, ano.

4.4 Considerações adicionais

Talvez aqui uma secção para finalizar e mostrar algumas coisas

Arquitecturas de Modelos

Grande parte da literatura sobre previsões em modelos de aprendizagem apresenta as mesmas arquiteturas, sendo que são depois aprimoradas conforme os dados e o problema.

Apresento aqui as arquiteturas mais usadas em previsões, como também algumas usadas noutros ramos tentando prever a compatibilidade neste problema.

As arquiteturas irão seguir um esquema lógico comum, um bloco de camadas de entrada, um bloco principal e um por fim um bloco interpretativo.

As dimensionalidades destas camadas é o que irá formar as diferentes arquiteturas em estudo.

5.1 Camadas

Para uma construção de modelos usando a ferramenta [keras](#) a unidade básica são as camadas. Estas representam uma operação, com uma entrada, e uma saída, e com possíveis parametrizações específicas.

Estas camadas ligadas entre si, perfazem um *profundo* de camadas neuronais, chamado profundo pois tem mais que uma camada.

Apresento aqui as camadas utilizadas nos modelos aplicados.

5.1.1 Dense

A camada dense pega num input,

5.1.2 Convolution

5.1.3 MaxPooling

5.1.4 [Dropout](#)

Dropout é uma camada que elimina/ignora alguns dos neurónios da camada anterior. Este procedimento impede o overfitting, ajudando na generalização.

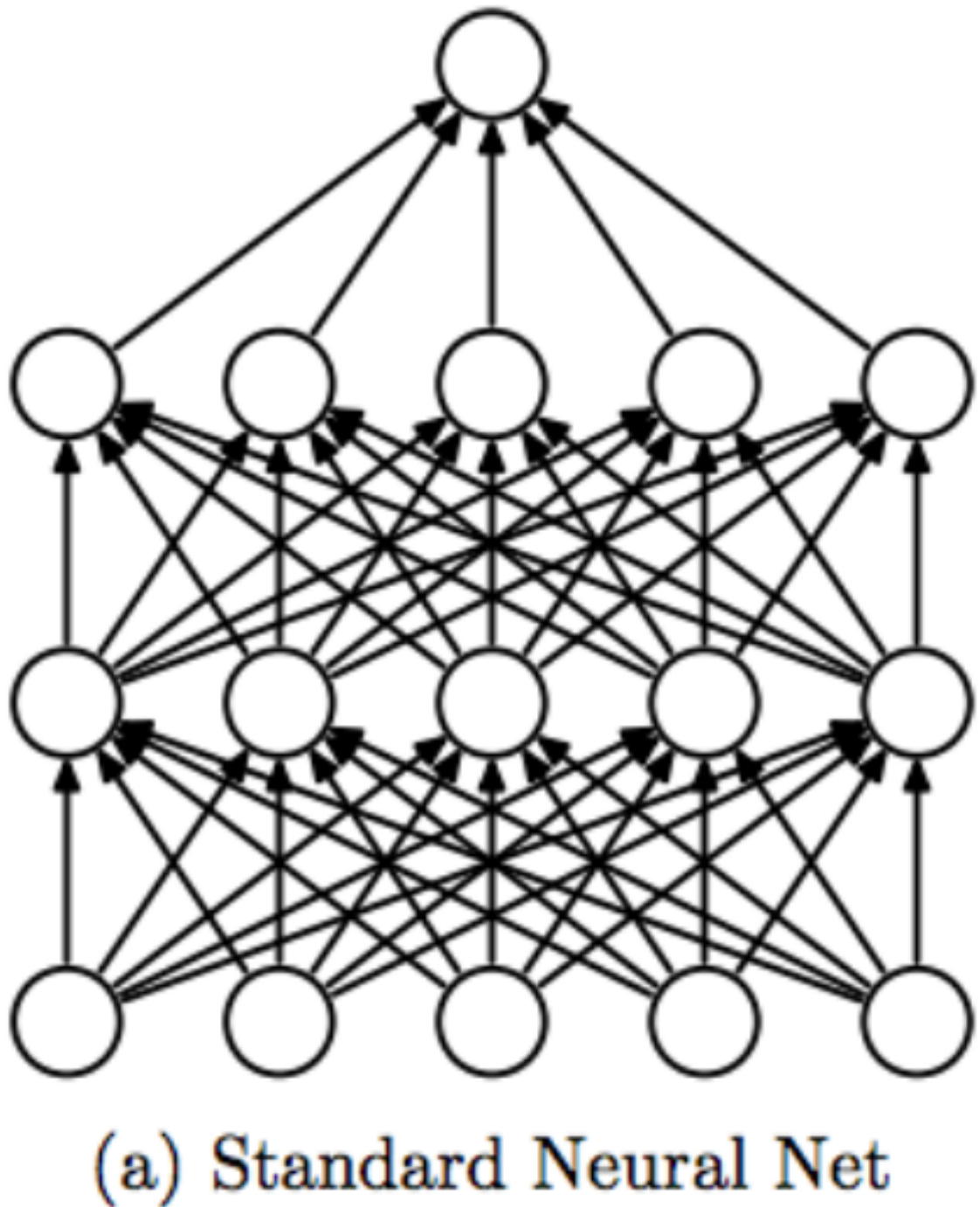


Figure 5.1: Ilustração do efeito da camada de dropout

5.2 Blocos

Todas as arquiteturas em análise irão ter por base um bloco de camadas neurais. A formação dessas arquiteturas passa pelas diferentes maneiras que se pode utilizar o bloco principal. Repetições em serie ou em paralelo são um exemplo.

5.2.1 Bloco Dense

O bloco dense sendo ele o mais simples é formado por duas camadas Dense [[empty citation](#)], em que a primeira apresenta um numero maior de filtros que a segunda.

Estas camadas não são mais do que uma criação de filtros aleatórios combinando as entradas, para criar todos os filtros de saída. São a base das camadas interpretativas. A acumulação em série (stacked) de camadas de dense está ligada a melhorias nas capacidades predictivas dos modelos [[18](#)].

Exemplo ilustrativo do nosso bloco basico onde entrariam 16 filtros na primeira camada e para finalizar o bloco com 2 filtros

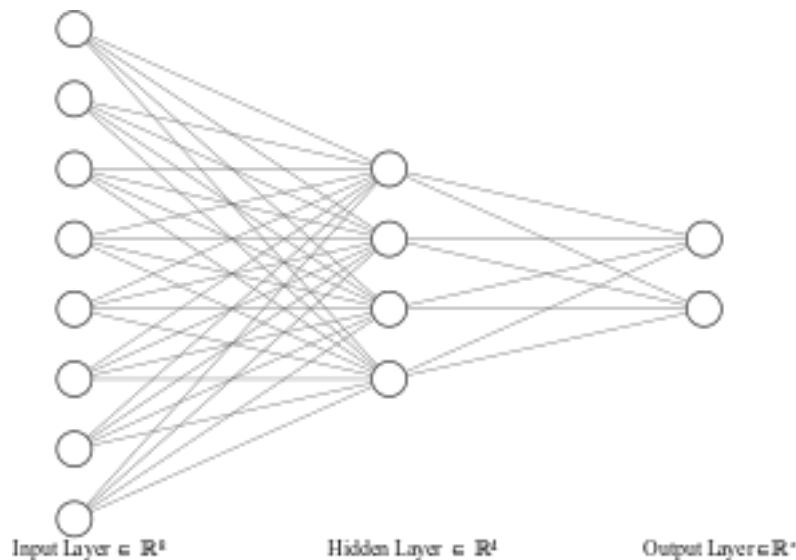


Figure 5.2: Bloco Dense

5.2.2 Bloco CNN

Bloco de CNN é aqui definido como uma convolução na dimensão temporal seguido de camadas para combater o overfitting, MaxPooling e Dropout.

Normalmente usada em processamentos de imagens, o uso de convoluções temporais é também por si mesmo uma ideia forte.

explicar om que e CNN imagem

Usada também as ideias de attention, residual e o que eu chamei broad

5.2.3 Bloco LSTM

O uso de LSTM para previsões é uma area comum, mas aqui é seguido através das ideias partilhadas em [[19](#)], e reforçado pelo uso em previsões energéticas demonstrados em [[8](#)]

O bloco LSTM é a aplicação das RNN, aqui sendo apenas definido como uma camada de LSTM.

Estes blocos mantêm dentro de si ligações a diferentes camadas temporais, e cada filtro criado, mantêm uma "memória" dos filtros passados.

Bastante utilizado em modelação de linguagem.

imagem

5.3 Arquiteturas

5.3.1 Vanilla

O termo "Vanilla" aqui é aplicado para aquitecturas que apenas usam um bloco de cada, um de entrada, um principal, e um interpretativo.

Como exemplo a arquitetura de "VanillaCNN"
imagem da mesma

5.3.2 Stacked

Stacked refere-se a "amontoadado" onde se utiliza o bloco principal várias vezes em série. E apenas um bloco de entrada e um interpretativo.

Como exemplo a arquitetura de "StackedCNN"
imagem da mesma

5.3.3 MultiHead

Multihead é o termo para quando os blocos de entrada e principais são repetidos paralelamente, um caminho para cada atributo, ou uma outra paralelização à escolha. Sendo depois concatenadas essas camadas e passadas juntas para a camada interpretativa.

Aqui foi usado sempre a paralelização por atributos, e ao invés de fazer Multihead no sentido de múltiplas entradas, para simplicidade de programação, foi feita uma paralelização interna no modelo, após a camada de entrada, onde a mesma é repetida para cada atributo.

Foi testado a diferença, e para os dados usados não havia diferenças de qualidade, mas sim em tempo de treino, logo a mais rápida foi a escolhida.

Como exemplo a arquitetura de "MultiheadCNN"
imagem da mesma

5.3.4 MultiTail

Esta arquitectura tem o mesmo conceito que a anterior a nível de paralelização, mas neste caso esta é feita apenas na camada interpretativa. Sendo que o resultado do bloco principal é repetido para criar a paralelização.

Neste caso foi paralelizado com o número de tempos a prever, 24 horas, 24 objectos de saída destas modelos.

A grande diferença desta arquitectura para a "Vanilla" que prevê 24 horas, é que aqui cada hora tem o seu próprio valor de função de perda, logo o modelo como que está a treinar 24 modelos diferentes, e no caso "Vanilla" a função de perda é única e é a média do erro das horas todas.

Como exemplo a arquitetura de "MultiTailCNN"
imagem da mesma

5.3.5 UNET

Normalmente usando em modelação de imagens, a arquitectura UNET passa por criar uma rede de expansão dos filtros, usando convoluções, e de seguida uma rede de contracção dos mesmos, até aos tamanhos pretendidos.

O bloco principal contextualmente o mesmo que o CNN.

Nas suas ligações UNET junta informação de filtros passados (não de nível temporal mas de rede neuronal) para realçar informação já trabalhada, e assim identificar padrões de vários contextos diferentes.

É habitual também adicionar aos blocos principais portões de atenção, portões residuais. Estas duas técnicas são também estudadas aqui.

É chamada assim pois é uma rede (NET) que forma um U na sua expansão e contração.

Como exemplo a arquitetura de "UNET"

imagem da mesma

5.4 Considerações adicionais

Aqui e dizer que os modelos utilizados para teste são as combinações destes blocos nestas arquiteturas.

Imagens de layers criadas com [dense http://alexlenail.me/NN-SVG/index.html](http://alexlenail.me/NN-SVG/index.html)

Métodos

Neste capítulo percorremos as experiências realizadas. Estas foram feitas através dos usos dos programas criados para o efeito, disponíveis no repositório GitHub do [projecto](#).

6.1 Benchmark

Como modelo benchmark iremos usar a alocação feita. Pois são estes valores que procuramos melhorar no caso prático.

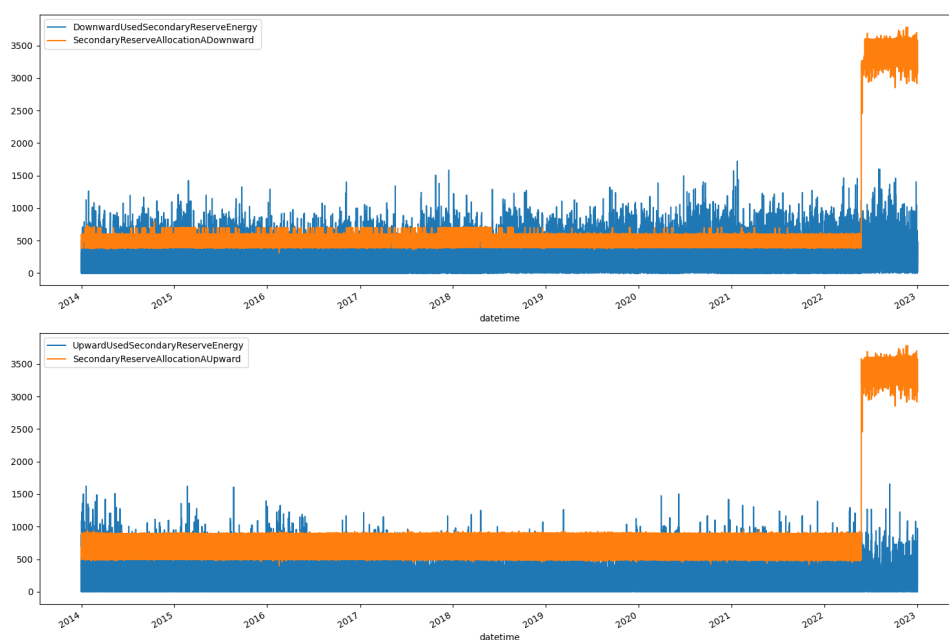


Figure 6.1: Serie Temporal do benchmark

Table 6.1: Dados Benchmark

rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus
963.1909591866699	53254091.2	False	-26.147637046658637	90.44161200506936	430902.00000000006	52823189.2

Para validação dos mesmo, vamos usar o ano 2021, devido aquele salto nos valores de alocação em 2022.

Para esses temos os seguintes dados:

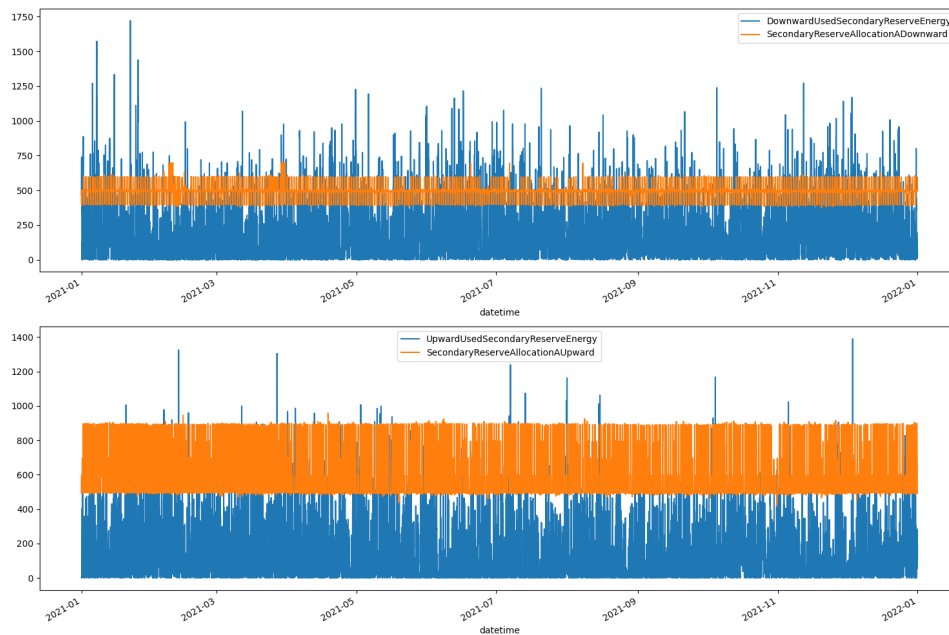


Figure 6.2: Serie Temporal do benchmark 2021

Os metodos em estudo vão ser comparados a esta medida. Sendo que o principal é baixar tanto a alocação perdida, como a alocação a mais. Que se traduzem no erro absoluto.

Table 6.2: Dados Benchmark de validação

rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus
488.55368458838365	3898946.9000000004	False	-6.008942605247387	85.98375795472323	57046.4	3841900.5

6.2 Modelos estatísticos

Antes de entrar para o desenvolvimento de modelos vamos usar métodos e modelos abertos para usar comparativamente.

Os modelos estatísticos recorrentes em previsões são AR, MA, ARMA, ARIMA, SARIMA para previsões so com um atributo, e para múltiplos atributos VAR.

O modelo AR e o VAR não obtiveram resultados aplicáveis, logo foram desconsiderados.

6.2.1 Univariate Analysis

Estas análises apenas aplicam uma fórmula à variável em questão.

6.2.1.1 AR

$$y_t = \beta_0 + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \epsilon_t \quad (6.1)$$

onde:

- y_t : O valor da série no tempo t .
- p : O número de atrasos.

6. MÉTODOS

- ε_t : O barulho no tempo t .
- β : O coeficiente dos valores em atrasado.

6.2.1.2 MA

MA - Moving Average

A MA

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \quad (6.2)$$

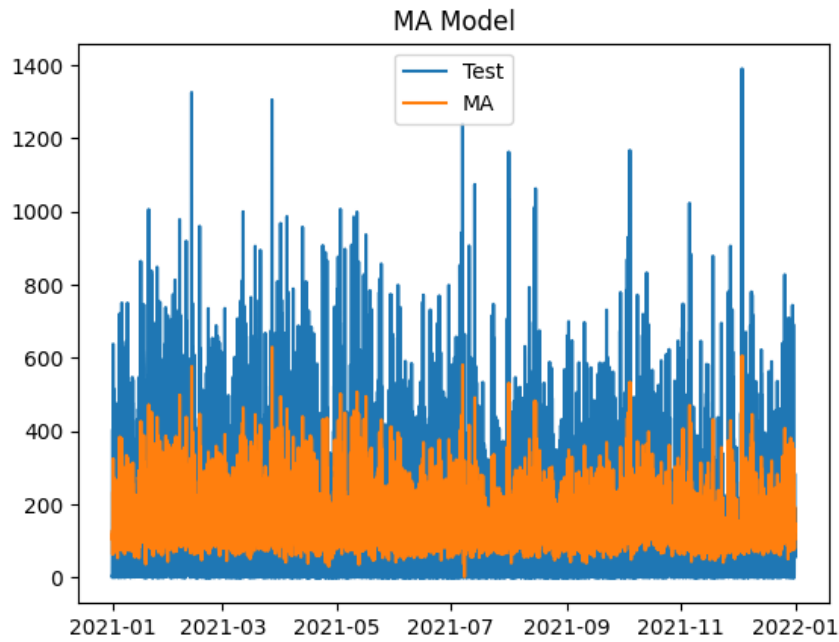


Figure 6.3: Previsões 2021 com modelo MA

6.2.2 ARMA

AR é blabla

$$y_t = \beta_0 + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \quad (6.3)$$

ARMA (Autoregressive Moving Average) Model:

- y_t : The value of the time series at time t .
- p : The number of time lags to regress on (AR part).
- q : The number of time lags of the error term to regress on (MA part).
- ε_t : The error term at time t .
- β : The coefficients of the lagged values (AR part).

- θ : The coefficients of the lagged error terms (MA part).

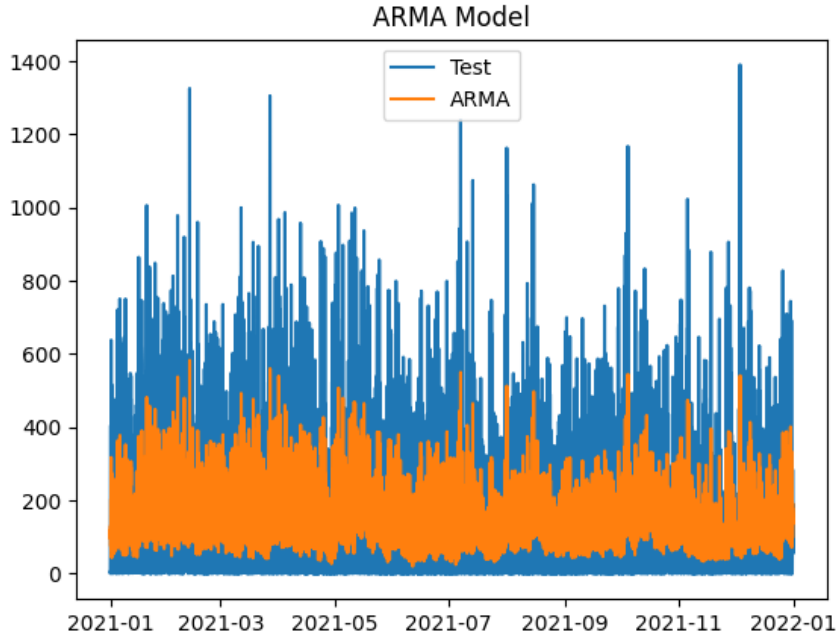


Figure 6.4: Previsões 2021 com modelo ARMA

6.2.3 ARIMA

AR é blabla

$$y_t^d = \beta_0 + \beta_1 y_{t-1}^d + \cdots + \beta_p y_{t-p}^d + \varepsilon_t^d + \theta_1 \varepsilon_{t-1}^d + \cdots + \theta_q \varepsilon_{t-q}^d \quad (6.4)$$

ARIMA (Autoregressive Integrated Moving Average) Model:

- $y_t^{[d]}$: The differenced value of the time series at time t .
- p : The number of time lags to regress on (AR part).
- d : The order of differencing.
- q : The number of time lags of the error term to regress on (MA part).
- $\varepsilon_t^{[d]}$: The differenced error term at time t .
- β : The coefficients of the lagged values (AR part).
- θ : The coefficients of the lagged error terms (MA part).

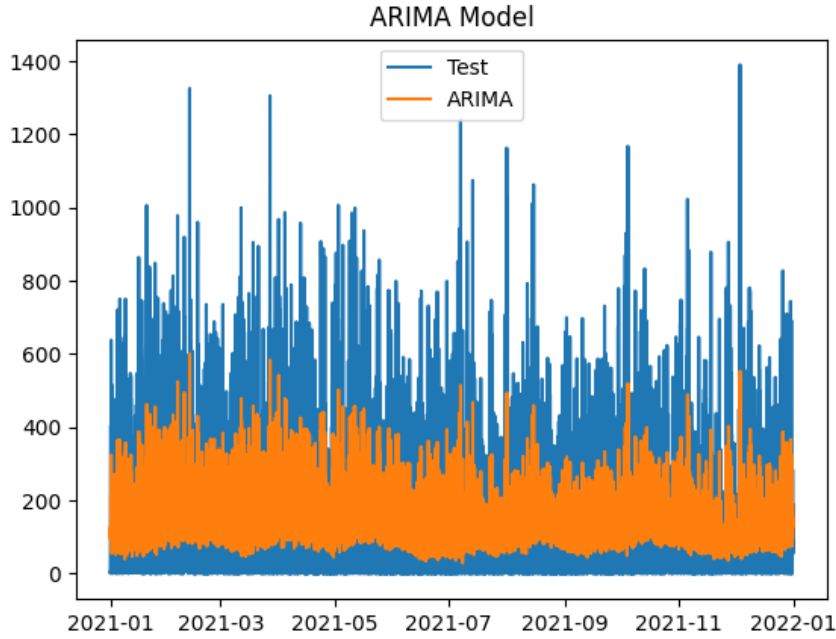


Figure 6.5: Previsões 2021 com modelo ARIMA

6.2.4 SARIMA

AR é blabla

AR é blabla

$$y_t^{[m]d} = \beta_0 + \beta_1 y_{t-m}^{[m]d} + \dots + \beta_p y_{t-pm}^{[m]d} + \epsilon_t^{[m]d} + \theta_1 \epsilon_{t-m}^{[m]d} + \dots + \theta_q \epsilon_{t-qm}^{[m]d} \quad (6.5)$$

SARIMA (Seasonal Autoregressive Integrated Moving Average) Model:

- $y_t^{[m]d}$: The differenced value of the time series at time t .
- p : The number of time lags to regress on (AR part).
- d : The order of differencing.
- q : The number of time lags of the error term to regress on (MA part).
- m : The number of time lags comprising one full period of seasonality.
- $\epsilon_t^{[m]d}$: The differenced error term at time t .
- β : The coefficients of the lagged values (AR part).
- θ : The coefficients of the lagged error terms (MA part).

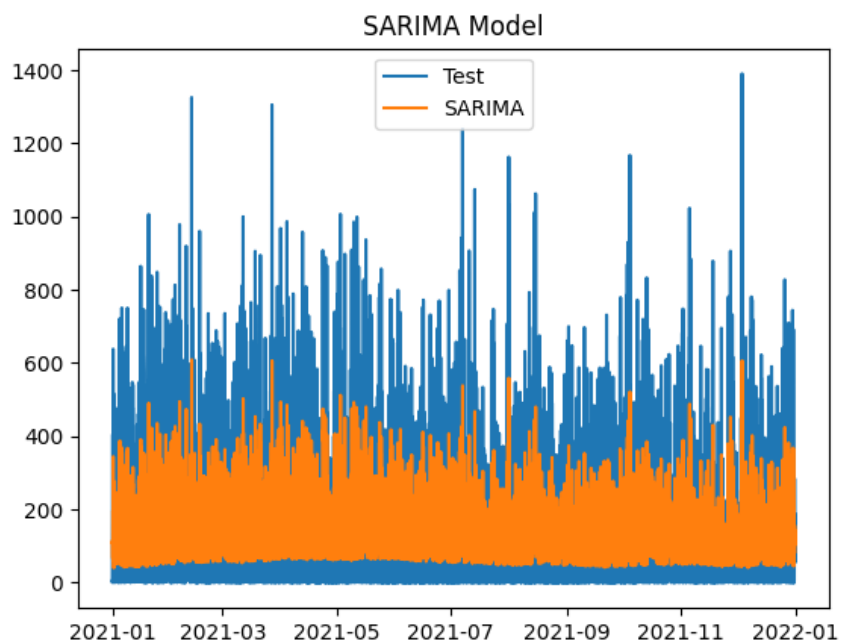


Figure 6.6: Previsões 2021 com modelo SARIMA

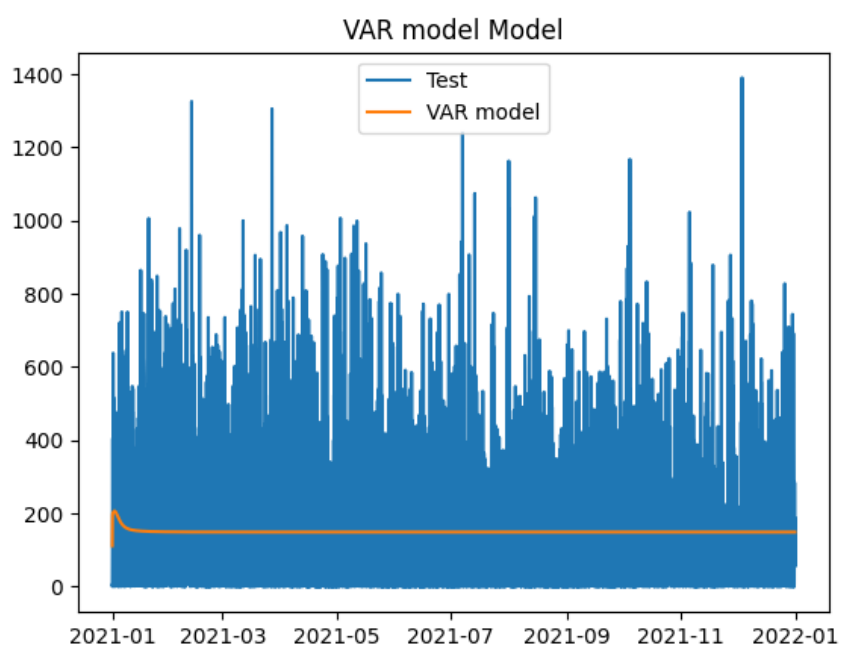


Figure 6.7: Previsões 2021 com modelo VAR

6. MÉTODOS

6.2.5 Resultados

Table 6.3: Resultados modelos Estatísticos

rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
NaN	0.00	True	-0.65	NaN	NaN	NaN	0.00	0.00	0.00
171.14	1123214.13	True	0.14	17.04	560507.02	562707.11	63.79	63.79	86.84
169.99	1108555.94	True	0.15	16.36	554443.34	554112.60	64.04	64.04	87.17
170.13	1111215.12	True	0.15	16.52	556281.62	554933.50	64.03	64.03	87.01
171.88	1115725.36	True	0.13	16.43	568538.03	547187.33	63.45	63.45	86.92
184.60	1253196.02	True	-0.00	21.63	619929.79	633266.23	64.82	64.82	86.07

Apenas pelos métodos estatísticos verificamos que no ano de 2021 teria havido uma melhoria de cerca de 80% das vezes, usando qualquer um dos métodos apresentados.

Embora a alocação em falta seja de uma ordem de grandeza superior.

6.3 Forecat

Com o propósito de desenvolver este estudo, e deixar ferramentas para a replicação do mesmo, foi criado uma biblioteca em python para desenhar as arquitecturas em estudo.

TODO: url

6.3.1 Construtor de modelos

Seguindo as arquitecturas descritas anteriormente esta ferramenta constroi os modelos automaticamente, sendo que precisamos apenas de fornecer os parametros variáveis.

O construtor assenta na ideia de três camadas abstratas de redes neuronais. A camada de entrada, a camada de bloco, e a camada interpretativa.

A camada de entrada recebe os dados e normaliza, podendo também fazer outras operações de preparação para a camada de bloco.

A camada de bloco é a camada descritiva da arquitetura, é a que tem as operações fundamentais.

A camada interpretativa é a que recebe o sinal de múltiplas redes neuronais internas, e traduz para o objectivo, usando Dense layers.

Esta abstracção segue sempre esta ordem. As variações dentro de cada arquitetura dependem das hiperparametrizações das mesmas, ou de cada camada, ou então da repetição do circuito, em paralelo ou em série. ou uma combinação destes.

6.3.2 Gerador de dados

O gerador construido trata da formatação dos dados para entrada nos modelos. Formatação esse que se baseia nos valores de janelas temporais a usar, e na divisão treino/teste.

Esta ferramenta agrega os dados em tensores de formato (N, t, a) , onde N é o numero do casos, t é a janela temporal, e a é o numero de atributos.

A ferramenta permite também definir o tempo de salto entre cada entrada.

Usando como exemplo uma janela temporal de 168 (horas, uma semana) para treino, e 24 (horas) para o alvo. Com um salto temporal de 1 a primeira entrada teria como treino as primeiras 168 horas dos dados, e como alvo as 24 horas consequentes. A segunda entrada seria a partir da segunda hora dos

dados, e assim consecutivamente. Para um caso em que o tempo de salto seria 24, a primeira entrada mantinha-se, mas a segunda começaria 24 horas depois, e não apenas uma.

Como estamos também a lidar com dados desfasados, o gerador permite um TODO: shift em atributos a especificar. No caso em estudo temos que os atributos são de DA (day-ahead), logo estão desfasados 24 horas. O que implica termos de aplicar este shift nos dados que não são DA, nomeadamente os dados alvo. Esta propriedade permite também o fácil uso da ferramenta noutros dados desfasados, como as previsões a 3 ou 8 horas.

6.4 Treino e Resultados

Realizaram-se várias experiências, onde em cada uma se ia eliminando alguns dos objectos em estudo. Em cada experiência toda a parametrização era igual, à excepção do objecto de estudo.

6.4.1 Arquiteturas e numeros de epocas

Nesta experiência foi testado o resultado das várias arquiteturas em estudo, como também o impacto do número de épocas na qualidade dos modelos.

As arquiteturas estudadas foram:

- VanillaDense
- VanillaCNN
- VanillaLSTM
- StackedCNN
- StackedLSTM
- EncoderDecoder
- UNET

Os modelos foram treinados em 200 épocas, sendo que foram salvos a cada 10 épocas, de forma conseguirmos perceber os contextos nos saltos de épocas.

As parametrizações usadas:

- loss : mean squared error
- Metodo activação no meio : relu
- Metodo activação no fim : relu
- optimizador : Adam
- Janela temporal em X : 168 horas (1 semana)
- Janela temporal em Y : 24 horas (1 semana)
- Fracção de treino : 95%

6.4.2 Funções de Perda (Loss)

TODO: o que é a loss function?

Esta experiência consiste em rever que função de perda é melhor aplicável ao problema. Sendo um problema de regressão linear, de valores bastante oscilatórios e com uma distribuição exponencial, temos algumas loss functions que já são reconhecidas para o problema.

- mean absolute error
- mean squared error
- loss : mean absolute error

6.4.3 Hiperparametrização

6.4.3.1 Activação

6.4.3.2 Optimizadores

6.4.4 Janelas Temporais

Um dos pontos deste trabalho é perceber a fesiabilidade de usar dados de previsão do dia anterior (DA) para estes atributos energéticos.

Algo que pode ser também aplicado no futuro a outros dados que não DA, mas sim a 3 horas, ou a 8 horas.

Para perceber esta flexibilidade, mas especialmente para escolher as melhores janelas temporais a usar neste modelos, vamos testar várias combinações.

Mantendo em mente que o objectivo é prever 24 horas, para os casos onde o alvo não dá um previsão de 24 horas, é necessario criar um numero de modelos para fazer as 24 horas.

Para validação apenas é usado o espaço temporal previsto, e não multiplos modelos.

Dado as análises de autocorrelação iremos usar como janelas para treino o conjunto [24, 48, 98, 168] para prever o conjunto [1, 4, 8, 12, 24]

Para alem destes foram também testadas combinações com janelas de treino 8 e 12 horas. Estas mostraram rapidamente que janelas de treino menores que as de previsão funcionam muito mal.

6.4.5 Classificação

Como descrito em (ref)... existe também o uso de tanto classes como valores lineares para resolução de problemas de regressão, também chamado *cluster-wise regression*.

Para este teste mudamos um pouco o modelo em uso. Ao invés de apenas uma camada interpretativa, fazemos duas, em paralelo, sendo que uma resolve a regressão e a outra a classificação.

Outro caso, proposto aqui, é usar uma nova camada intrepertativa, que combina as duas saidas anteriores (linear e classificação), e resolve novamente para os valores lineares.

Estes modelos não teram apenas uma saida, mas varias, como as arquiteturas MultiTail, mas neste caso cada uma resolve para um problema diferente, com funções de perda, e activações diferentes.

TODO: desenho destas duas camadas intrepertaticas

6.4.6 Pesos

Por ultimo foi testado o impacto do uso de pesos nos modelos. Estes pesos são o peso que aquele alvo TODO: explicar pesos

6.4.6.1 Modelos lineares

Para os modelos lineares o peso que é adiciona ao modelo é a distância à média.

Este peso serve para dar mais importância a valores facilmente considerados outliers.

6.4.6.2 Modelos Lineares e de Classificação

Aqui o peso é dado por saída. Para as saídas lineares o peso dados é o mesmo que apresentado anteriormente, para a saída de classificação, o peso é o inverso da frequência da classe.

Distribuindo assim a importância de treino pela frequência das classes. Sendo um prática comum especialmente quando as distribuições são muito desiguais, como o caso em estudo.

É aqui estudada a aplicação destes pesos individualmente, e em conjunto.

Os pesos aqui são também normalizados de modo a que o maior peso em cada um deles seja 1, e logo a multiplicação dos dois esteja dentro das mesmas dimensões de relevância.

$$P_m = |y - mean| \quad (6.6)$$

6.5 Considerações adicionais

Foram realizados testes adicionais que não obtiveram resultados passivos de boa interpretação, e foram imediatamente descartados, como:

- Janela temporal em X : 96, 48, 24
- otimizador : todos os otimizadores disponiveis na biblioteca keras
- loss : todas as outra loss functions de regressão disponiveis.
- epocas : influência do numero de epocas nos modelos, foram treinados até 20000 epocas alguns modelos mas à medida que a perda ia estagnando na assintomta, o modelo ia apenas piorando.

Todos os metodos foram realizados utilizando código em python, que está aberto em <https://github.com/JotaFan/renewable-generation-into-reserve-markets>

Resultados e discussão

Os resultados das experiências são apresentados por experiência, sendo que cada um vai testando e eliminando parametros na modulação.

Após análise inicial na modelação foi concluído que usar tentar modelar apenas um dos atributos de cada vez leva a melhor resultados que tentar modelar os dois no mesmo modelo.

Foi também visto que os dois atributos em causa são analogos, logo os sistemas que melhor repretam um dos atributos tambem são semelhantemente eficazes no outro.

Assim todas as experiências foram realizadas usando apenas um dos atributos alvo, sendo este o "UpwardUsedSecondaryReserveEnergy", e em que o atributo de alocação comparativo é "SecondaryReserveAllocationAUpward".

7.1 Métricas

A nível de comparação efectiva de modelos iremos usar as seguintes metricas: RMSE, erro absoluto, r2 score, percentagem optima, alocação em falta, alocação a mais.

Sendo que as métricas de decisão final são as que representam melhor o objectivo: baixar os custo de alocação das reservas secundárias. logo serão as metricas de alocação em falta e em demasia, sendo que a soma das mesmas é o erro absoluto.

$$RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N}} \quad (7.1)$$

$$\text{Absolute Error} = \sum_{i=0}^{N-1} |y_i - \hat{y}_i| \quad (7.2)$$

$$R^2 \text{ score} = 1 - \frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1} (y_i - \bar{y})^2} \quad (7.3)$$

$$\text{alocação em falta} = \begin{cases} 0 & , \text{if } \hat{y} \geq y \\ y - \hat{y} & , \text{otherwise} \end{cases} \quad (7.4)$$

$$\text{optimal percentage} = \frac{1}{N} \sum_{i=1}^n 1[\hat{y}_i \geq y_i \& \hat{y}_i \leq alloc](\text{not: [20]}) \quad (7.5)$$

Onde \hat{y} são as previsões dos modelos, y são os valores reais utilizados, "UpwardUsedSecondaryReserveEnergy", e $alloc$ são os valores alocados "SecondaryReserveAllocationAUpward"

7.2 Experiências

Para comparação o erro absoluto para o ano de 2021 na alocação feita é de 3889367.4

7.2.1 Arquiteturas e numeros de epocas

7.2.1.1 Arquiteturas

Table 7.1: Resultados Arquitecturas

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
EncoderDecoder	177.07	1050078.76	True	0.08	11.13	754129.54	295949.22	53.40	53.40	84.80
StackedCNN	453.12	3717806.99	True	-5.03	82.67	52038.16	3665768.83	35.04	34.48	38.34
StackedLSTMA	NaN	0.00	True	-0.65	NaN	NaN	NaN	0.00	0.00	0.00
UNET	176.73	1138605.66	True	0.08	16.85	619845.54	518760.12	62.20	62.20	86.14
VanillaCNN	174.96	1124205.72	True	0.10	16.45	614355.90	509849.82	61.79	61.79	86.18
VanillaDense	1200.20	10073188.28	False	-41.27	192.50	487.24	10072701.04	1.13	0.77	1.17

7.2.1.2 Epocas

7.2.2 Funções de Perda (Loss)

As funções de perda foram estudas com duas arquitecturas diferentes, de modo a conseguir ter uma melhor noção do impacto das mesmas.

Table 7.2: Resultados Funções de perda

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
StackedCNNmsle	290.70	2299543.31	True	-1.48	54.80	193283.75	2106259.56	86.34	86.34	92.14
UNETmse	180.84	1269085.40	True	0.04	23.32	521000.39	748085.00	68.78	68.78	87.31
StackedCNNwl	175.87	1186767.36	True	0.09	20.02	572945.10	613822.26	64.92	64.92	86.81
VanillaCNNmse	162.44	1038022.80	True	0.23	14.30	568479.54	469543.26	61.02	61.02	87.08
StackedCNNmse	176.71	1103152.68	True	0.08	14.81	673158.53	429994.14	58.75	58.75	85.68
StackedCNNmsde	612.31	4978280.05	False	-10.00	81.86	3325165.78	1653114.27	18.27	18.06	30.14
UNETwl	30011.42	141637978.61	False	-26431.60	2558.08	10334.28	141627644.33	9.35	8.62	9.91
VanillaCNNwl	1071.18	8912349.22	False	-32.67	171.33	63940.63	8848408.59	1.01	0.62	3.89
VanillaCNNmsle	1250.10	10759371.90	False	-44.86	201.97	0.00	10759371.90	0.22	0.00	0.22
UNETmape	3545800.24	18933122401.84	False	-368973766.69	276314.32	0.00	18933122401.84	0.00	0.00	0.00

A nível de percentagem de modelo melhor, elas estão todas bastante renhidas, mas ha uma claro vantagem quando vemos a percentagem de melhor alocação ou de alocação optima.

Sendo que a perda que avança é a de MSLE (Mean Square Log Error).

Esta perda é tendicionalmente usada para distribuições exponencias, e onde temos bastantes outliers, que devem ser considerados. O que é o caso no nosso problema.

Esta experiência valida a observação feito na análise estatística sobre o mesmo.

7.2.3 Hiperparametrização

7.2.3.1 Activação

Table 7.3: Resultados Ativação

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
StackedCNN linear softplus	321.88	2577000.96	True	-2.04	59.88	149354.46	2427646.50	88.59	88.59	92.51
StackedCNN elu elu	329.79	2639978.00	True	-2.19	61.79	138194.90	2501783.10	87.76	87.76	91.33
StackedCNN linear elu	297.39	2368520.21	True	-1.60	55.86	177934.14	2190586.07	87.47	87.47	92.53
StackedCNN softplus selu	287.86	2249601.96	True	-1.43	53.57	218301.68	2031300.27	84.44	84.44	91.23
StackedCNN linear linear	238.22	1866499.25	True	-0.67	43.41	292863.60	1573635.64	81.02	81.02	90.41
StackedCNN tanh softsign	236.24	1289436.20	True	-0.64	0.85	1288663.18	773.02	11.00	11.00	80.39
StackedCNN softplus softsign	236.36	1291380.19	True	-0.64	0.90	1290725.89	654.30	9.39	9.39	80.40
StackedCNN softsign linear	1112.70	8936608.01	False	-35.34	178.94	8185.60	8928422.41	7.99	7.50	8.60
StackedCNN exponential softsign	250.31	353784.60	True	-0.64	0.90	NaN	NaN	1.58	1.58	19.45
StackedCNN softsign selu	2053.61	17501165.80	False	-122.77	323.70	73467.70	17427698.09	0.01	0.00	2.98

7. RESULTADOS E DISCUSSÃO

7.2.3.2 Optimizaçao

Table 7.4: Resultados Optimizadores

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
StackedCNN Adadelta	1946.60	16304588.30	False	-110.20	298.49	121783.10	16182805.20	0.37	0.37	6.57
StackedCNN Adafactor	209.16	1127287.01	True	-0.28	4.25	1036653.47	90633.54	39.16	39.16	82.23
StackedCNN Adagrad	2281.10	19250481.97	False	-151.71	354.33	50132.10	19200349.87	0.29	0.27	3.51
StackedCNN Adam	507.51	4103373.73	False	-6.56	90.96	44467.79	4058905.95	36.50	35.86	38.94
StackedCNN AdamW	208.26	1114855.55	True	-0.27	3.82	1033146.63	81708.92	38.60	38.60	82.34
StackedCNN Adamax	1085.65	8916100.25	False	-33.59	177.57	64682.73	8851417.52	3.48	2.92	6.39
StackedCNN Ftrl	242.89	1847009.05	True	-0.73	39.01	341650.54	1505358.52	77.52	77.52	89.31
StackedCNN Nadam	534.57	4388148.16	False	-7.39	92.41	35224.69	4352923.47	22.85	21.81	25.02
StackedCNN RMSprop	501.82	3995286.60	False	-6.39	87.47	165111.98	3830174.63	30.90	30.17	38.94
StackedCNN SGD	207.90	1125396.48	True	-0.27	4.88	1021725.80	103670.68	40.59	40.59	82.27

7.2.4 Janelas Temporais

Table 7.5: Resultados Janelas Temporais

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
StackedCNN 168X 8Y	277.45	2189089.58	True	-1.26	51.96	226599.69	1962489.89	84.47	84.47	91.59
StackedCNN 168X 4Y	247.11	1941513.71	True	-0.79	45.26	272577.02	1668936.69	82.07	82.07	90.72
StackedCNN 168X 24Y	352.28	2757367.10	True	-2.64	65.61	155940.39	2601426.71	70.78	70.65	75.40
StackedCNN 168X 12Y	404.20	3219562.53	True	-3.79	70.85	90454.58	3129107.95	67.74	67.30	71.01
StackedCNN 98X 8Y	401.18	3287580.95	True	-3.73	74.38	76725.24	3210855.71	51.78	51.65	55.36
StackedCNN 24X 24Y	468.78	3907629.32	True	-5.46	84.62	48042.45	3859586.87	26.35	25.65	29.52
StackedCNN 48X 1Y	510.34	4249569.03	False	-6.65	91.65	35151.96	4214417.07	23.58	22.63	26.05
StackedCNN 168X 1Y	568.14	4694571.86	False	-8.47	100.18	23253.65	4671318.21	20.47	19.49	22.15
StackedCNN 98X 12Y	622.31	5179835.43	False	-10.41	108.86	19985.94	5159849.49	17.78	16.78	19.33
StackedCNN 98X 24Y	1167.21	9770108.13	False	-39.09	191.54	1578.73	9768529.40	2.25	1.83	2.41

7.2.5 Classificação

Table 7.6: Resultados Janelas Temporais

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage	acc	acc alloc
StackedCNNClusterLinear 4 arch	207.47	1111659.96	True	-0.26	3.81	1027024.21	84635.75	38.66	38.66	82.34	0.40	0.29
StackedCNNClusterLinear 4 interpr	207.45	1111808.59	True	-0.26	3.84	1027318.24	84490.36	38.77	38.77	82.38	NaN	NaN
StackedCNNClusterLinear 6 arch	208.03	1113342.39	True	-0.27	3.72	1032371.26	80971.13	38.48	38.48	82.28	0.33	0.10
StackedCNNClusterLinear 6 interpr	208.23	1114478.16	True	-0.27	3.72	1034020.09	80458.06	38.55	38.55	82.26	NaN	NaN
StackedCNNClusters 4 arch	206.56	1106934.65	True	-0.25	4.11	1017487.36	89447.30	39.57	39.57	82.43	0.40	0.29
StackedCNNClusters 6 arch	206.94	1109766.69	True	-0.26	4.12	1019180.77	90585.93	39.49	39.49	82.45	0.33	0.10

7.2.6 Pesos

Table 7.7: Resultados Janelas Temporais

name	rmse	abs erro	erro comp	r2 score	mape score	alloc missing	alloc surplus	optimal percentage	better allocation	beter percentage
StackedCNN delta mean	177.29	1204388.33	True	0.08	19.99	508061.06	696327.28	66.69	66.69	87.72
StackedCNN no weight	178.20	1100714.22	True	0.07	14.09	697157.86	403556.35	57.49	57.49	85.36
VanillaCNN delta mean	4061.70	34669933.86	False	-483.15	634.39	0.00	34669933.86	0.00	0.00	0.00
VanillaCNN no weight	1146.52	9109107.48	False	-37.58	185.61	25950.43	9083157.05	11.90	11.55	12.81

Aqui devem constar gráficos e sua análise crítica e ligação com a secção 2 da revisão bibliográfica no sentido de comparar valores e discutir diferenças, por exemplo. A legenda dos gráficos deve seguir a das figuras, isto é, porque também são figuras.

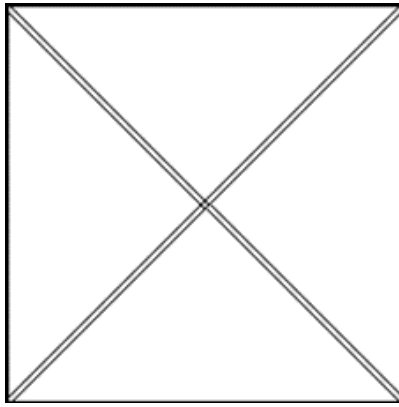


Figure 7.1: Exemplo de como considerar um gráfico

Conclusões e sugestões futuras

Primeiramente podemos ver pela análise estatística 6.3 e aplicando a ideia [11], simples modelos estatísticos conseguiriam resolver o problema em mãos, melhor do que o que é utilizado actualmente 6.2, e melhor do que muitos dos modelos profundos que testamos.

E se considerarmos ainda que os modelos estatísticos apresentados que apresentam estes resultados, utilizam apenas a variável em questão, e não todos os outros atributos, a nível de aplicabilidade já são um melhoria.

Aqui são dadas as respostas às perguntas de investigação formuladas na secção 1.2. Não fazer aqui a discussão dos resultados. Essa discussão deve ser feita no capítulo 7.2.6. Não esquecer de indicar sugestões futuras para que um colega possa dar continuidade ao trabalho desenvolvido.

Referências

- [1] Hum Nath Bhandari et al. “Predicting stock market index using LSTM”. In: *Machine Learning with Applications* 9 (Sept. 2022), p. 100320. ISSN: 2666-8270. DOI: [10.1016/J.MLWA.2022.100320](https://doi.org/10.1016/J.MLWA.2022.100320).
- [2] Bin Wang et al. “Deep uncertainty quantification: A machine learning approach for weather forecasting”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (July 2019), pp. 2087–2095. DOI: [10.1145/3292500.3330704](https://doi.org/10.1145/3292500.3330704). URL: <https://dl.acm.org/doi/10.1145/3292500.3330704>.
- [3] Siyuan Lu et al. “Machine learning based multi-physical-model blending for enhancing renewable energy forecast - Improvement via situation dependent error correction”. In: *2015 European Control Conference, ECC 2015* (Nov. 2015), pp. 283–290. DOI: [10.1109/ECC.2015.7330558](https://doi.org/10.1109/ECC.2015.7330558).
- [4] Haibo Sun, Li Li, and Haonan Wang. “Short-term Wind Power Forecasting Model Based on Stacking Fusion Learning”. In: *IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC) 2022-June* (2022), pp. 53–57. ISSN: 26932865. DOI: [10.1109/ITAIC54216.2022.9836510](https://doi.org/10.1109/ITAIC54216.2022.9836510).
- [5] T. Rajasundrapandiyanleebanon et al. “Solar Energy Forecasting Using Machine Learning and Deep Learning Techniques”. In: *Archives of Computational Methods in Engineering* 2023 30:5 30 (5 Feb. 2023), pp. 3059–3079. ISSN: 1886-1784. DOI: [10.1007/S11831-023-09893-1](https://doi.org/10.1007/S11831-023-09893-1). URL: <https://link.springer.com/article/10.1007/s11831-023-09893-1>.
- [6] Tanveer Ahmad and Huanxin Chen. “A review on machine learning forecasting growth trends and their real-time applications in different energy systems”. In: *Sustainable Cities and Society* 54 (Mar. 2020), p. 102010. ISSN: 2210-6707. DOI: [10.1016/J.SCS.2019.102010](https://doi.org/10.1016/J.SCS.2019.102010).
- [7] Ioannis Antonopoulos et al. “Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review”. In: *Renewable and Sustainable Energy Reviews* 130 (Sept. 2020), p. 109899. ISSN: 1364-0321. DOI: [10.1016/J.RSER.2020.109899](https://doi.org/10.1016/J.RSER.2020.109899).
- [8] Rogério Luís de C. Costa. “Convolutional-LSTM networks and generalization in forecasting of household photovoltaic generation”. In: *Engineering Applications of Artificial Intelligence* 116 (Nov. 2022), p. 105458. ISSN: 0952-1976. DOI: [10.1016/J.ENGAPPAI.2022.105458](https://doi.org/10.1016/J.ENGAPPAI.2022.105458).
- [9] R. Meenal and A. Immanuel Selvakumar. “Assessment of SVM, empirical and ANN based solar radiation prediction models with most influencing input parameters”. In: *Renewable Energy* 121 (June 2018), pp. 324–343. ISSN: 0960-1481. DOI: [10.1016/J.RENENE.2017.12.005](https://doi.org/10.1016/J.RENENE.2017.12.005).
- [10] Natei Ermias Benti, Mesfin Diro Chaka, and Addisu Gezahegn Semie. “Forecasting Renewable Energy Generation with Machine Learning and Deep Learning: Current Advances and Future Prospects”. In: *Sustainability (Switzerland)* 15 (9 May 2023). ISSN: 20711050. DOI: [10.3390/SU15097087](https://doi.org/10.3390/SU15097087).
- [11] Shereen Elsayed et al. “Do We Really Need Deep Learning Models for Time Series Forecasting?” In: ().

- [12] Javier Cardo-Miota, Emilio Pérez, and Hector Beltran. “Deep learning-based forecasting of the automatic Frequency Reserve Restoration band price in the Iberian electricity market”. In: *Sustainable Energy, Grids and Networks* 35 (Sept. 2023), p. 101110. ISSN: 2352-4677. DOI: [10.1016/J.SEGAN.2023.101110](https://doi.org/10.1016/J.SEGAN.2023.101110). URL: <https://linkinghub.elsevier.com/retrieve/pii/S2352467723001182>.
- [13] *Mercado de Serviço de Sistemas*. URL: <https://www.dgeg.gov.pt/pt/areas-transversais/mercados-e-mecanismos-de-capacidade/outros-mercados/mercado-de-servico-de-sistemas/>.
- [14] Hussein Ashraf Rassid, Orientador : Fernando, and Jorge Ferreira Lopes. *Mercado de Reservas: Simulador Multiagente SIMREEL e Caso de Estudo Envolvendo o Mercado Ibérico*. 2017. URL: <https://run.unl.pt/handle/10362/30815>.
- [15] Célia Carneiro. “Mecanismos de funcionamento do Mercado de Serviços de Sistema - aplicação ao MIBEL”. In: (Nov. 2016). URL: <https://core.ac.uk/download/pdf/302870251.pdf>.
- [16] Stef van Buuren and Karin Groothuis-Oudshoorn. “mice: Multivariate Imputation by Chained Equations in R”. In: *Journal of Statistical Software* 45 (3 Dec. 2011), pp. 1–67. ISSN: 1548-7660. DOI: [10.18637/JSS.V045.I03](https://doi.org/10.18637/JSS.V045.I03). URL: <https://www.jstatsoft.org/index.php/jss/article/view/v045i03>.
- [17] S F Buck. “A Method of Estimation of Missing Values in Multivariate Data Suitable for use with an Electronic Computer”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 22 (2 1960), pp. 302–306. ISSN: 00359246. URL: <http://www.jstor.org/stable/2984099>.
- [18] V L Helen Josephine, A.P. Nirmala, and Vijaya Lakshmi Alluri. “Impact of Hidden Dense Layers in Convolutional Neural Network to enhance Performance of Classification Model”. In: *IOP Conference Series: Materials Science and Engineering* 1131 (1 Apr. 2021), p. 012007. ISSN: 1757-8981. DOI: [10.1088/1757-899X/1131/1/012007](https://doi.org/10.1088/1757-899X/1131/1/012007).
- [19] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. “Recurrent Neural Networks for Time Series Forecasting: Current status and future directions”. In: *International Journal of Forecasting* 37 (1 Jan. 2021), pp. 388–427. ISSN: 01692070. DOI: [10.1016/j.ijforecast.2020.06.008](https://doi.org/10.1016/j.ijforecast.2020.06.008).
- [20] Kenneth E. Iverson. *A programming language*. 1962, p. 11. ISBN: 0471430145. URL: <https://www.jsoftware.com/papers/APL.htm>.

Anexos

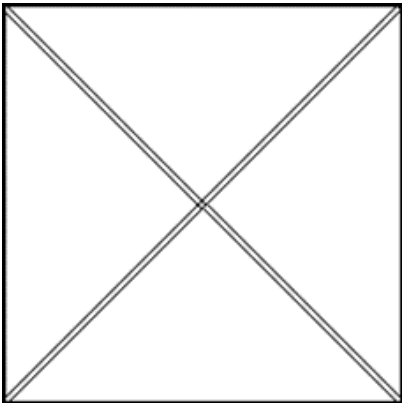


Figure A.1: Exemplo de como considerar um gráfico nos anexos.

Table A.1: Isto é um exemplo de uma tabela. Se fôr igual(copiada) a outro autor deve ser pedido autorização para reproduzir.

Title 1	Title 2	Title 3	Title 4
entry 1	data	data	data
	data	data	data
	data	data	data
entry 2	data	data	data
	data	data	data
entry 3	data	data	data
	data	data	data
	data	data	data
	data	data	data
entry 4	data	data	data