

ANÁLISIS DE DATOS SOBRE DEMANDA DE ENERGÍA

Jorge Meza, Rodolfo Moran, Victor Rojas
Electromecánica

INTRODUCCIÓN

Un análisis de demanda de energía a lo largo de los años es un estudio fundamental para comprender las tendencias y los patrones de consumo energético en una determinada región o país. Por lo que con este trabajo buscamos **determinar el mejor conjunto** para poder **predecir la demanda de energía**.

OBJETIVOS:

- Interpretar los datos de consumo a través de los años 2008-2022.
- Cuantificar el impacto de las variables.
- Determinar el mejor conjunto de datos para la predicción.
- Comparar las predicciones de los conjuntos seleccionados.
- Mejorar los hiperparametros y las capas de la red neuronal.

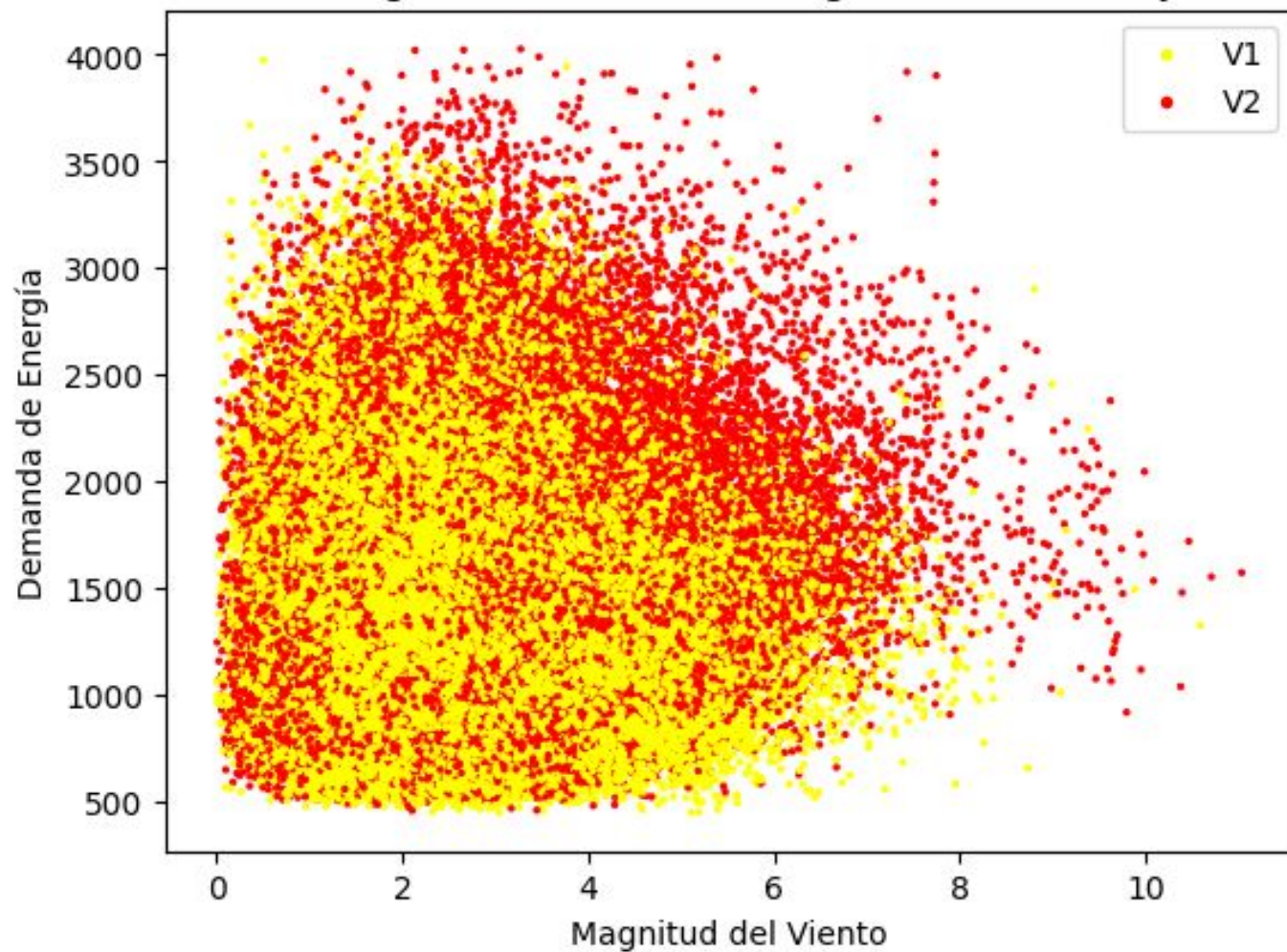
DATASET:

	Fecha	Potencia	Temperatura	Humedad	Presión	Precipitación	U	V
	2008-01-01 00:00:00-03:00	877.658352	24.463579	96.798320	999.585909	0.000629	-1.613880	-1.142420
	2008-01-01 01:00:00-03:00	855.606858	24.314531	96.065988	999.059392	0.000419	-1.624135	-1.256589
	2008-01-01 02:00:00-03:00	803.357910	24.165483	95.333657	998.532875	0.000210	-1.634390	-1.370759
	2008-01-01 03:00:00-03:00	750.530366	24.016435	94.601325	998.006359	0.000000	-1.644645	-1.484929
	2008-01-01 04:00:00-03:00	729.029900	23.667572	94.257455	998.982867	0.000000	-1.273453	-1.629541
	2008-01-01 05:00:00-03:00	717.195424	23.318710	93.913586	999.959376	0.000000	-0.902261	-1.774154
	2008-01-01 06:00:00-03:00	636.497140	22.969847	93.569716	1000.935885	0.000000	-0.531068	-1.918767
	2008-01-01 07:00:00-03:00	566.063757	24.355974	88.873767	1001.222567	0.000000	-0.609305	-2.025998
	2008-01-01 08:00:00-03:00	543.269824	25.742102	84.177818	1001.509249	0.000000	-0.687541	-2.133229
	2008-01-01 09:00:00-03:00	564.513991	27.128229	79.481869	1001.795931	0.000000	-0.765778	-2.240461
	2008-01-01 10:00:00-03:00	564.095336	28.404613	73.903403	1001.211104	0.000059	-0.835915	-2.511545
	2008-01-01 11:00:00-03:00	612.422595	29.680997	68.324936	1000.626277	0.000119	-0.906053	-2.782630
	2008-01-01 12:00:00-03:00	601.973939	30.957381	62.746470	1000.041449	0.000178	-0.976190	-3.053715
	2008-01-01 13:00:00-03:00	601.452977	30.560316	65.994571	999.509598	0.001045	-1.136838	-2.682597
	2008-01-01 14:00:00-03:00	591.757538	30.163251	69.242672	998.977746	0.001912	-1.297487	-2.311480
	2008-01-01 15:00:00-03:00	624.594143	29.766186	72.490773	998.445894	0.002779	-1.458135	-1.940362
	2008-01-01 16:00:00-03:00	632.386352	29.355445	75.390794	998.542574	0.002470	-1.577949	-1.433771
	2008-01-01 17:00:00-03:00	619.220616	28.944704	78.290815	998.639254	0.002161	-1.697763	-0.927180
	2008-01-01 18:00:00-03:00	596.283884	28.533962	81.190836	998.735934	0.001852	-1.817577	-0.420588

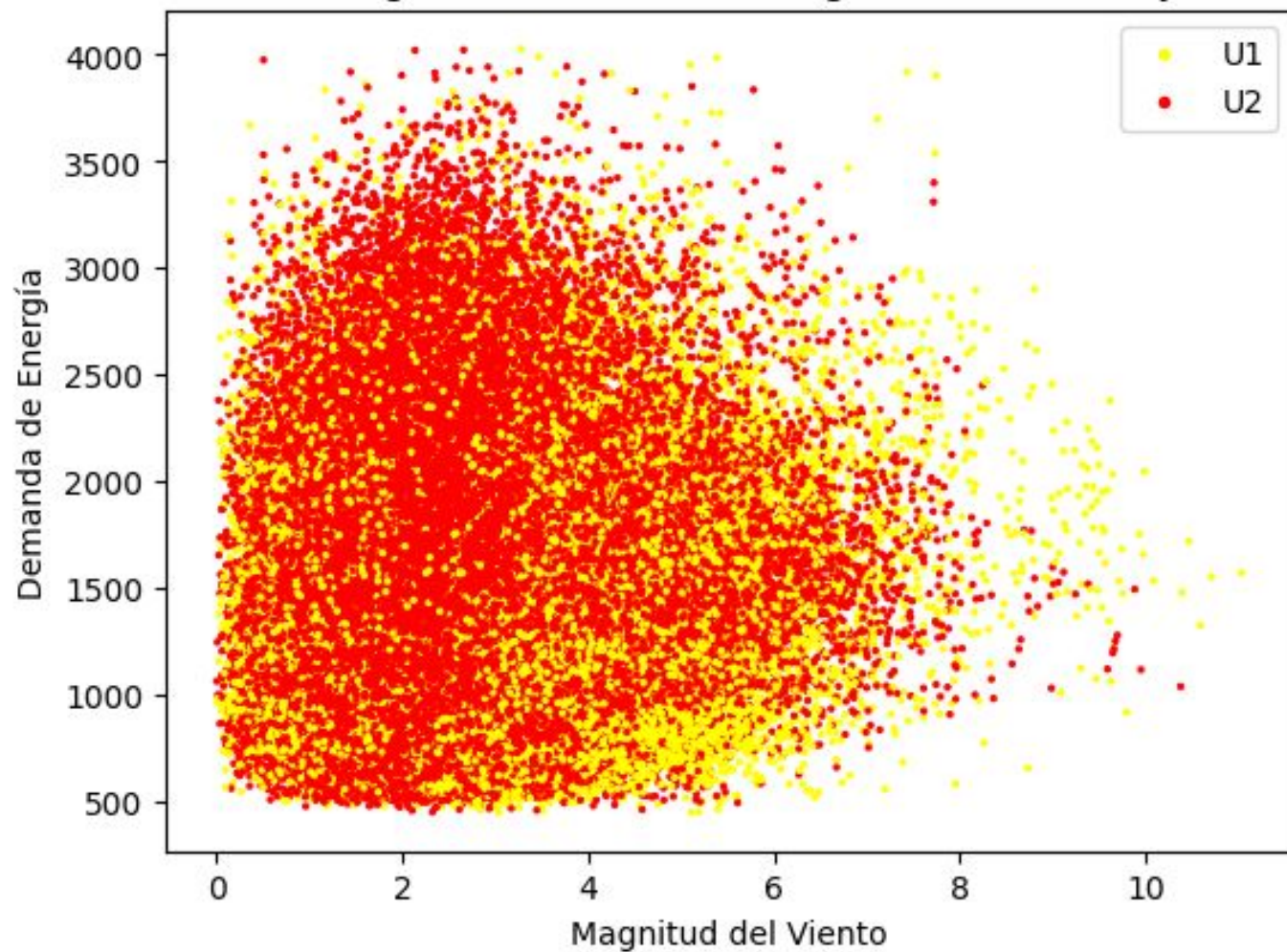


GRÁFICOS DE DATOS

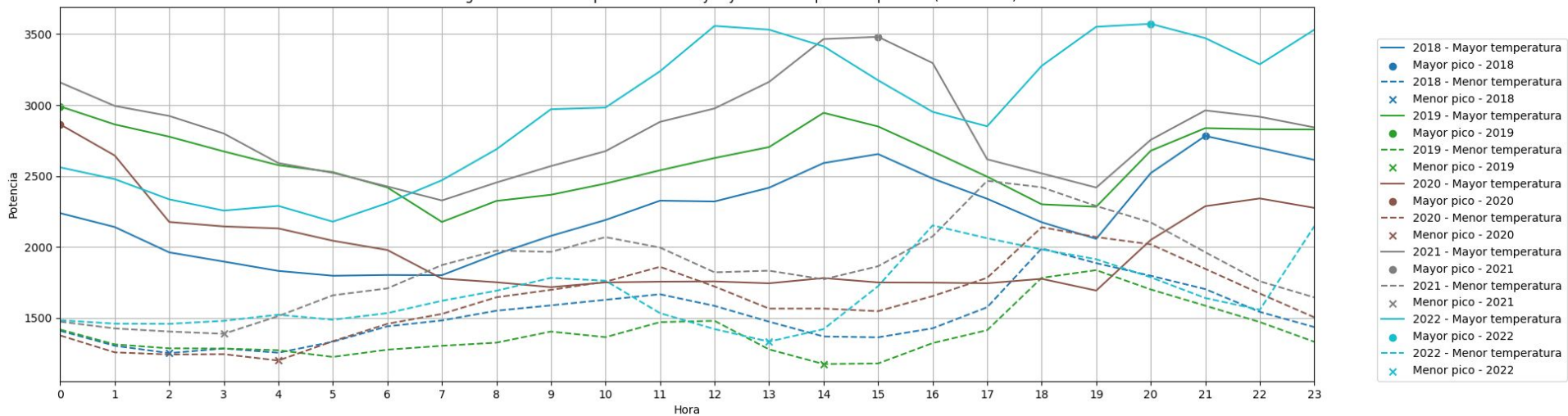
Demanda de Energía en función de la Magnitud del Viento y la Dirección



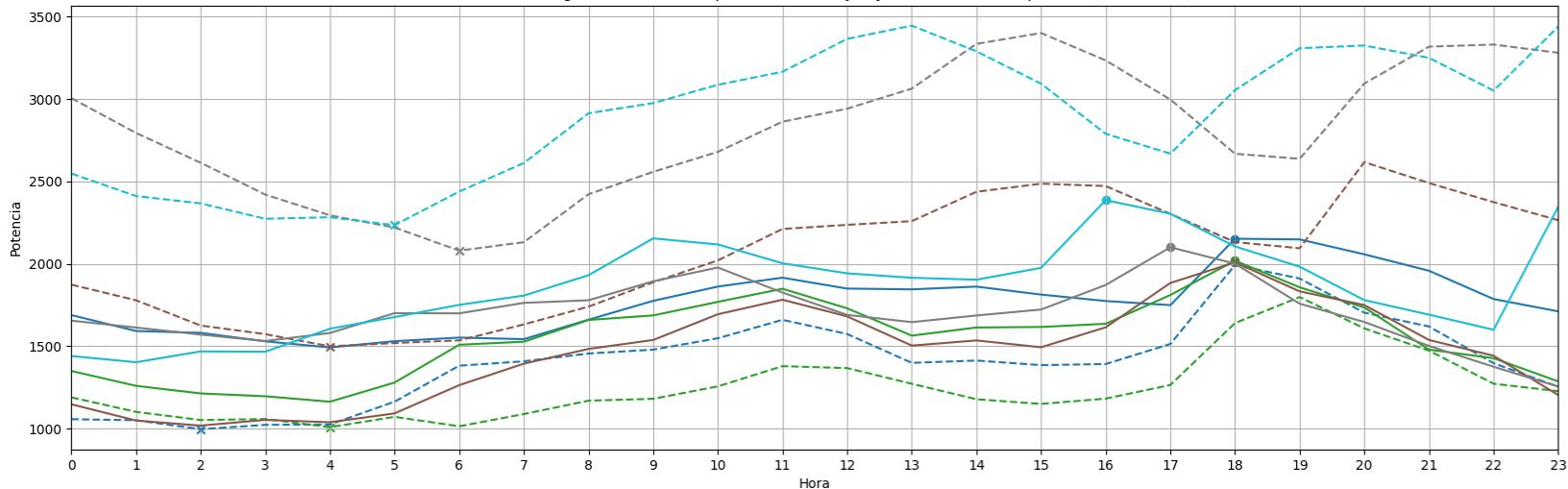
Demanda de Energía en función de la Magnitud del Viento y la Dirección



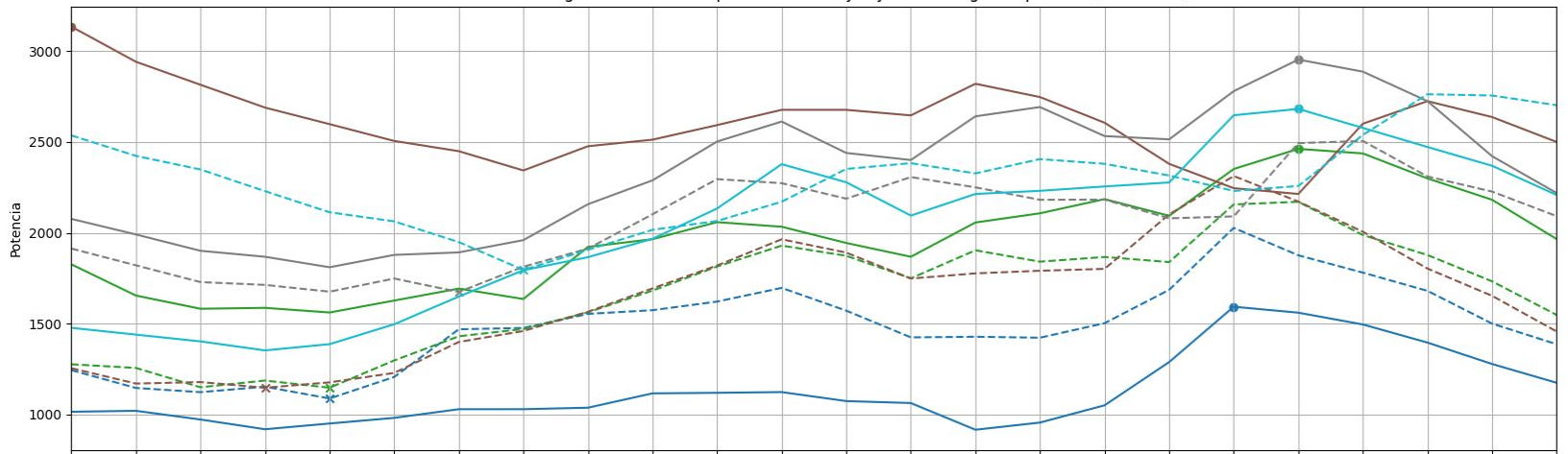
Potencia a lo largo de las 24 horas para Días de Mayor y Menor Temperatura por Año (2018-2022)



Potencia a lo largo de las 24 horas para Días de Mayor y Menor Humedad por Año (2018-2022)



Potencia a lo largo de las 24 horas para Días de Mayor y Menor Magnitud por Año (2018-2022)



Luego de interpretar los gráficos armamos 5 conjuntos con los parámetros que mejor podrían predecir la demanda de energía.

```
features_options = [
```

```
    ["Potencia", "Mes_sin", "Año", "Temperatura", "Magnitud", "dia_cos", "Humedad"],
```

```
    ["Potencia", "Mes_cos", "dia_sin", "Magnitud", "Temperatura", "Año"],
```

```
    ["Potencia", "Mes", "Humedad", "Temperatura", "Año", "dia_sin"],
```

```
    ["Potencia", "Mes_sin", "Temperatura", "Magnitud", "dia_cos", "Año", "Humedad"],
```

```
    ["Potencia", "Mes", "Temperatura", "dia_sin", "dia_cos", "Año", "Humedad"]
```

```
]
```

RED NEURONAL

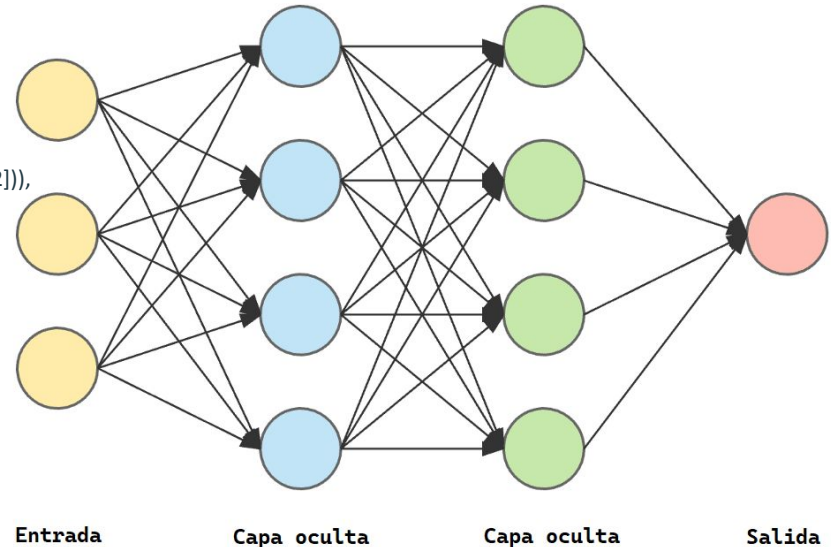
Modelo de red neuronal

```
model = Sequential([  
    LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),  
    LSTM(50, return_sequences=False),  
    Dense(25),  
    Dense(pred_window)  
])
```

```
model.compile(optimizer='adam', loss='mean_squared_error')
```

Entrenamiento del modelo

```
model.fit(X_train, y_train, epochs=5, batch_size=30, verbose=0)
```

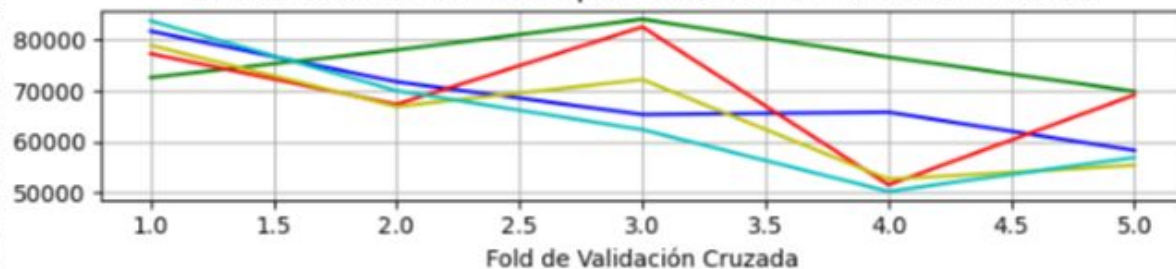


Utilizando una red neuronal con series temporales y validación cruzada, obteniendo los siguientes resultados:

Conjunto de Características	Promedio MSE	Promedio RMSE	Promedio MAX ERROR	Promedio hsup
Conjunto 1	68560,068	261,8397754	1564,458	1564,458
Conjunto 2	76223,336	276,0857403	1606,073	1606,073
Conjunto 3	69550,205	263,7237285	1498,261	1498,261
Conjunto 4	65192,233	255,3276973	1503,423	1503,423
Conjunto 5	64634,207	254,2325845	1481,652	1481,652

Error Cuadrático Medio (MS)

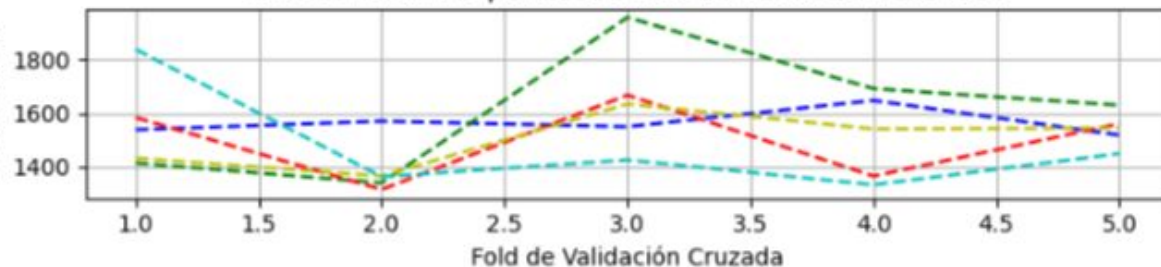
Errores Cuadráticos Medios para cada Fold de Validación Cruzada



- Conjunto 1 (MSE)
- Conjunto 2 (MSE)
- Conjunto 3 (MSE)
- Conjunto 4 (MSE)
- Conjunto 5 (MSE)

Error Máximo

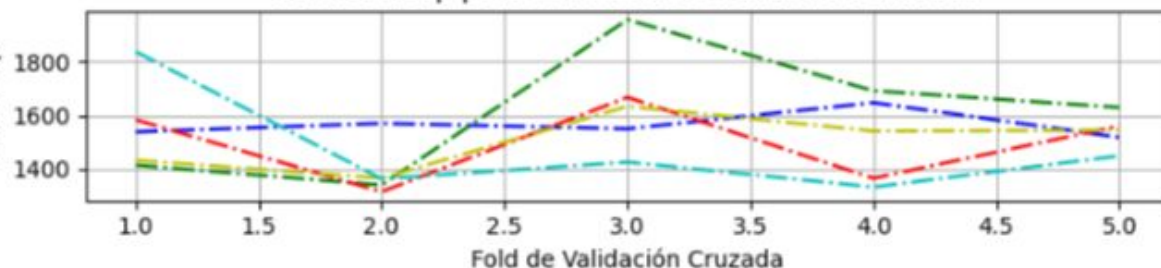
Errores Máximos para cada Fold de Validación Cruzada



- Conjunto 1 (Max Error)
- Conjunto 2 (Max Error)
- Conjunto 3 (Max Error)
- Conjunto 4 (Max Error)
- Conjunto 5 (Max Error)

Valor hsup

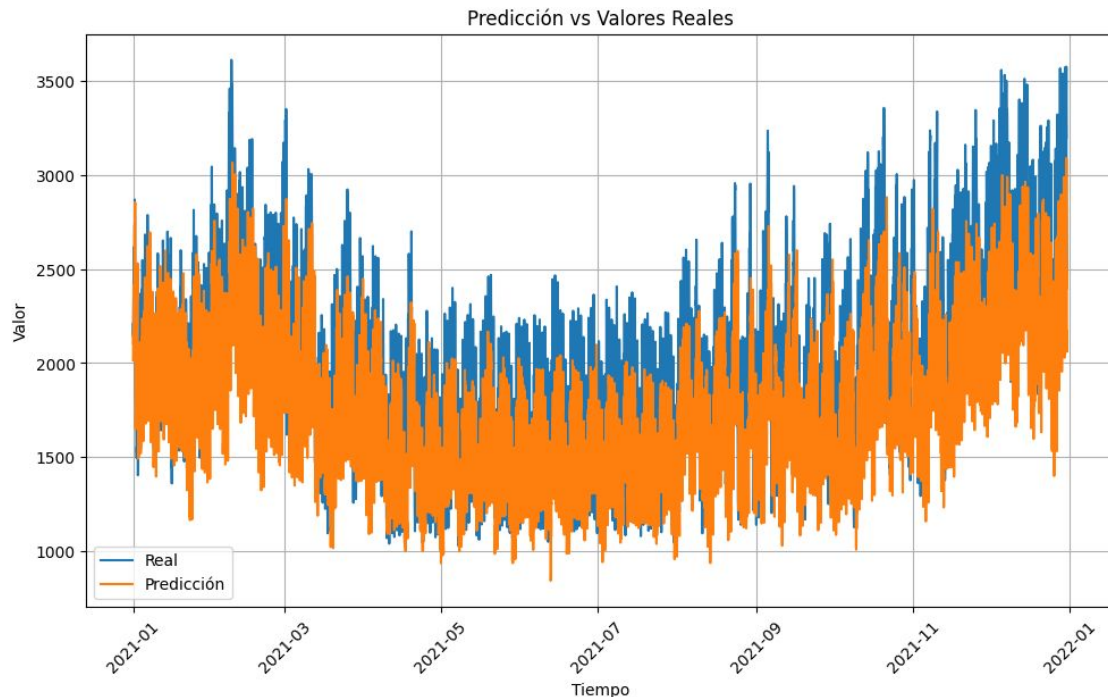
Valores hsup para cada Fold de Validación Cruzada



- Conjunto 1 (hsup)
- Conjunto 2 (hsup)
- Conjunto 3 (hsup)
- Conjunto 4 (hsup)
- Conjunto 5 (hsup)

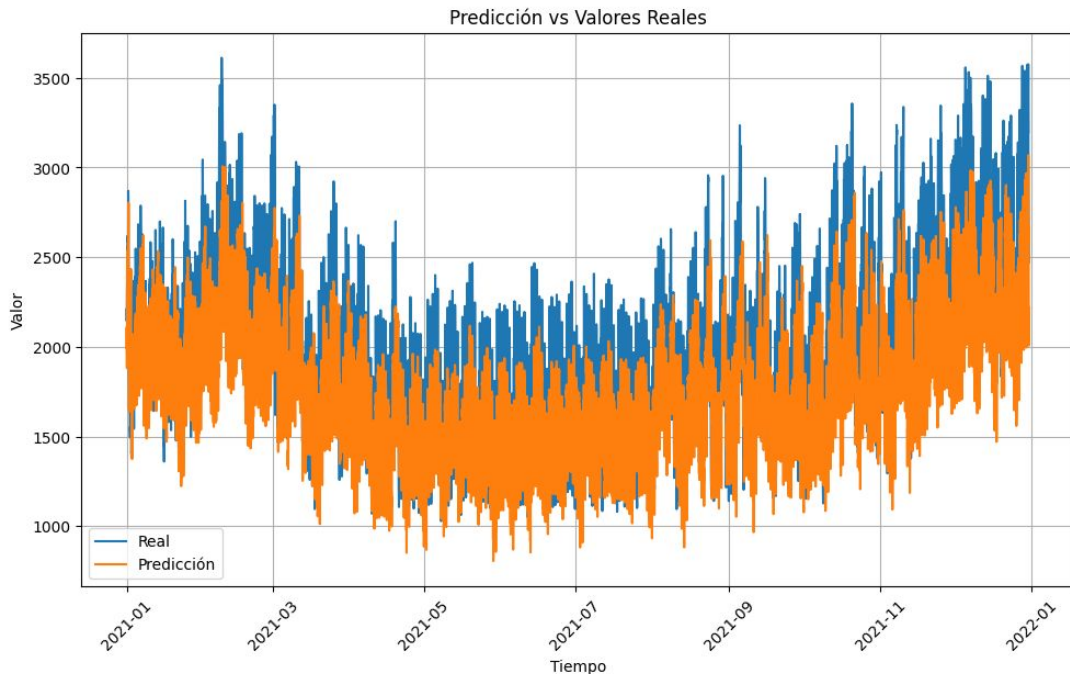
Con los resultados obtenidos podemos concluir que el conjunto 5 tuvo el mejor rendimiento

Por lo que procedemos a realizar predicciones con el conjunto 5 del año 2021 para observar su desempeño:

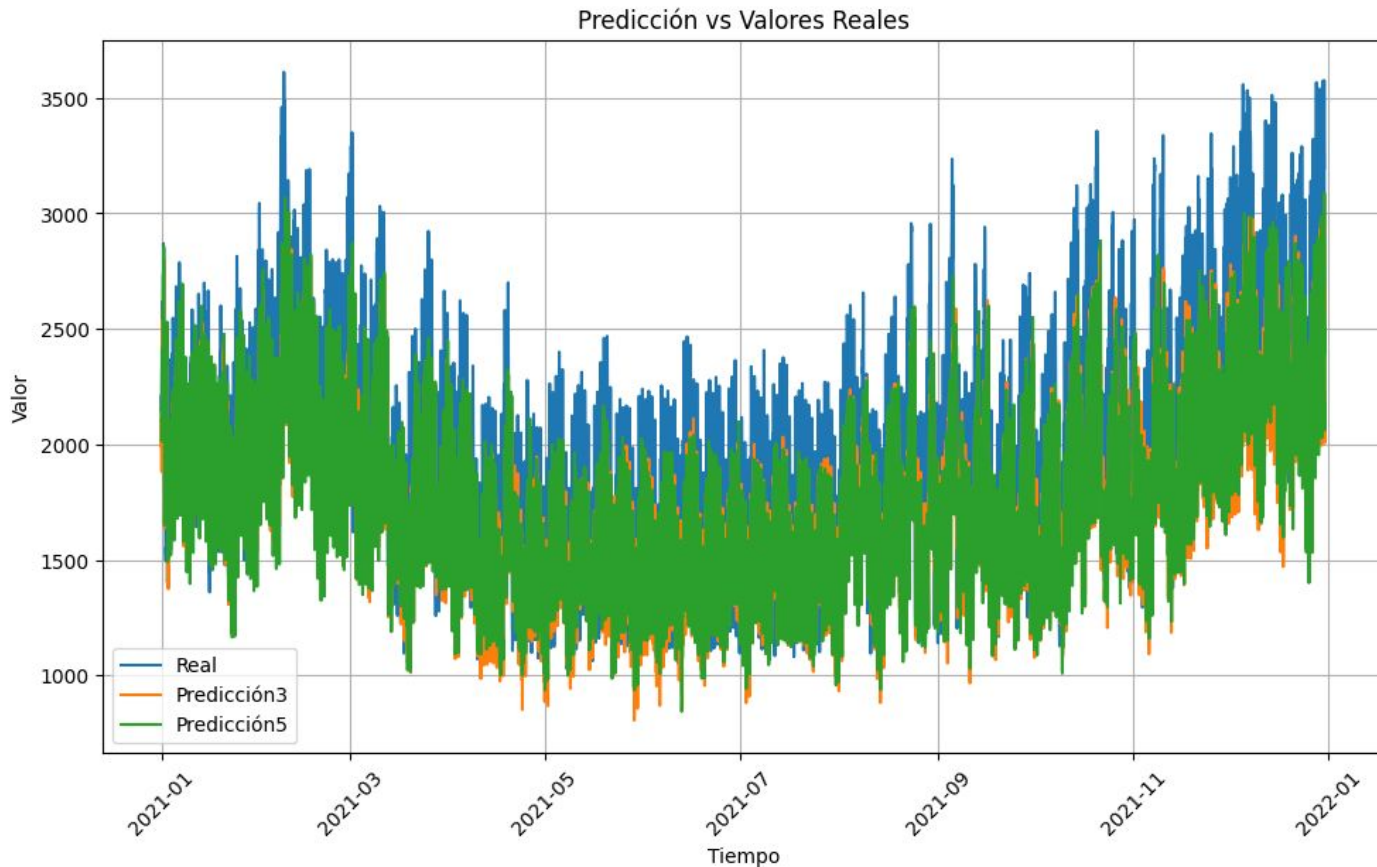


Con los resultados obtenidos también observamos que el conjunto 3 tuvo buenos resultado

Por lo que procedemos a realizar predicciones con el conjunto 3 del año 2021 para observar su desempeño:



PREDICCIONES conjunto 5 vs conjunto 3; año 2021



Con la intención de mejorar las predicciones modificamos los hiperparametros de la red neuronal

```
# Nuevo Modelo de red neuronal
model = Sequential([
    LSTM(100, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    LSTM(100, return_sequences=False),
    Dropout(0.2), # Capa Dropout del 20%
    Dense(pred_window) # Capa Dense de salida
])
model.compile(optimizer='adam', loss='mean_squared_error')
# Entrenamiento el modelo
model.fit(X_train, y_train, epochs=50, batch_size=150, verbose=0)
```

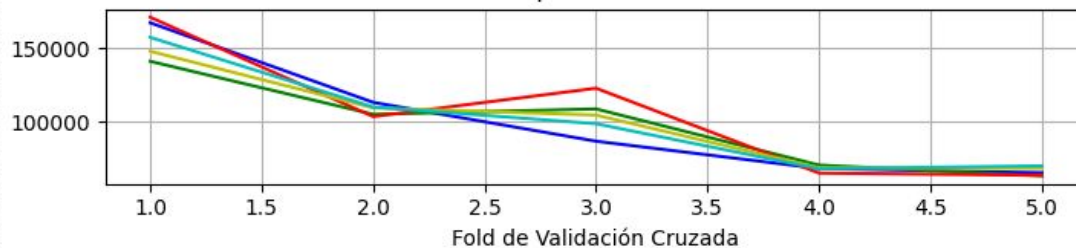
Obteniendo los siguientes resultados para determinar el mejor conjunto

Conjunto de Características	Promedio MSE	Promedio RMSE	Promedio MAX ERROR	Promedio hsup
Conjunto 1	100072,3739	316,3421785	1610,133998	1610,133998
Conjunto 2	97632,28843	312,4616591	1507,304719	1507,304719
Conjunto 3	105132,1098	324,2408207	1579,810898	1579,810898
Conjunto 4	99686,90062	315,7323243	1675,597938	1675,597938
Conjunto 5	100741,5588	317,3981077	1617,749491	1617,749491

Podemos observar que con el nuevo modelo se destaca el conjunto 2

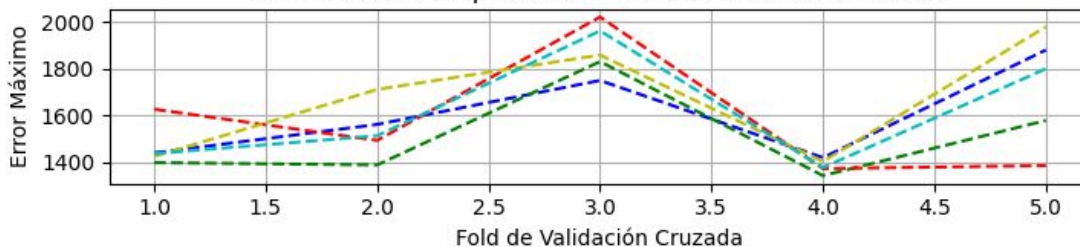
Error Cuadrático Medio (MSE)

Errores Cuadráticos Medios para cada Fold de Validación Cruzada



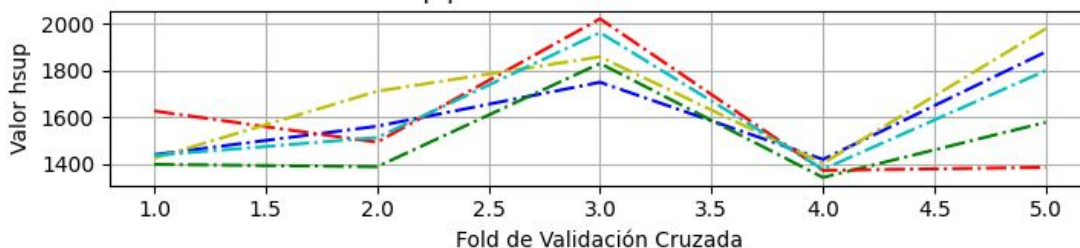
- Conjunto 1 (MSE)
- Conjunto 2 (MSE)
- Conjunto 3 (MSE)
- Conjunto 4 (MSE)
- Conjunto 5 (MSE)

Errores Máximos para cada Fold de Validación Cruzada



- Conjunto 1 (Max Error)
- Conjunto 2 (Max Error)
- Conjunto 3 (Max Error)
- Conjunto 4 (Max Error)
- Conjunto 5 (Max Error)

Valores hsup para cada Fold de Validación Cruzada



- Conjunto 1 (hsup)
- Conjunto 2 (hsup)
- Conjunto 3 (hsup)
- Conjunto 4 (hsup)
- Conjunto 5 (hsup)

También creamos modelos específicos para el conjunto 5 y 3 que se destacaron de la prueba original

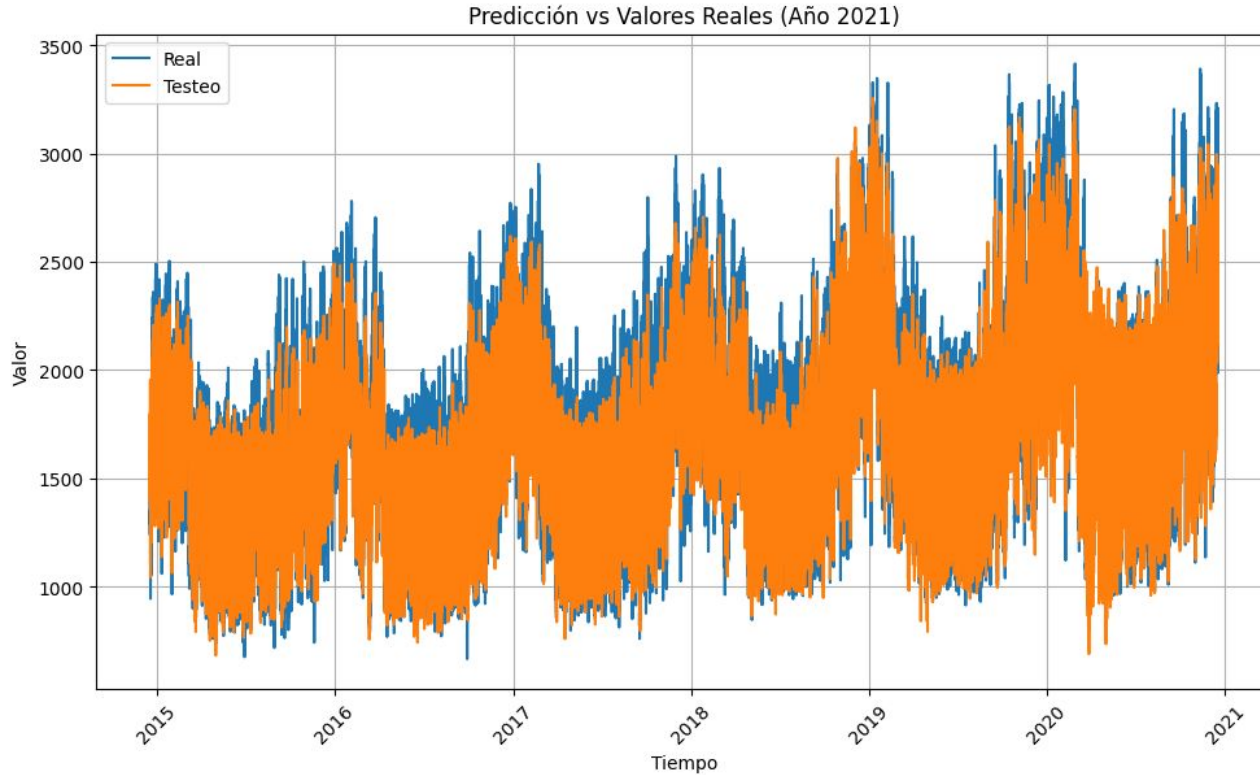
Modelo para el conjunto 5:

```
model_c5 = Sequential([
    LSTM(100, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    LSTM(100, return_sequences=False),
    Dropout(0.2), # Capa Dropout del 20%
    Dense(pred_window) # Capa Dense de salida
])
model_c5.compile(optimizer='adam', loss='mean_squared_error')
# Entrenamiento del modelo
model_c5.fit(X_train, y_train, epochs=100, batch_size=150, verbose=0)
```

Modelo para el conjunto 3:

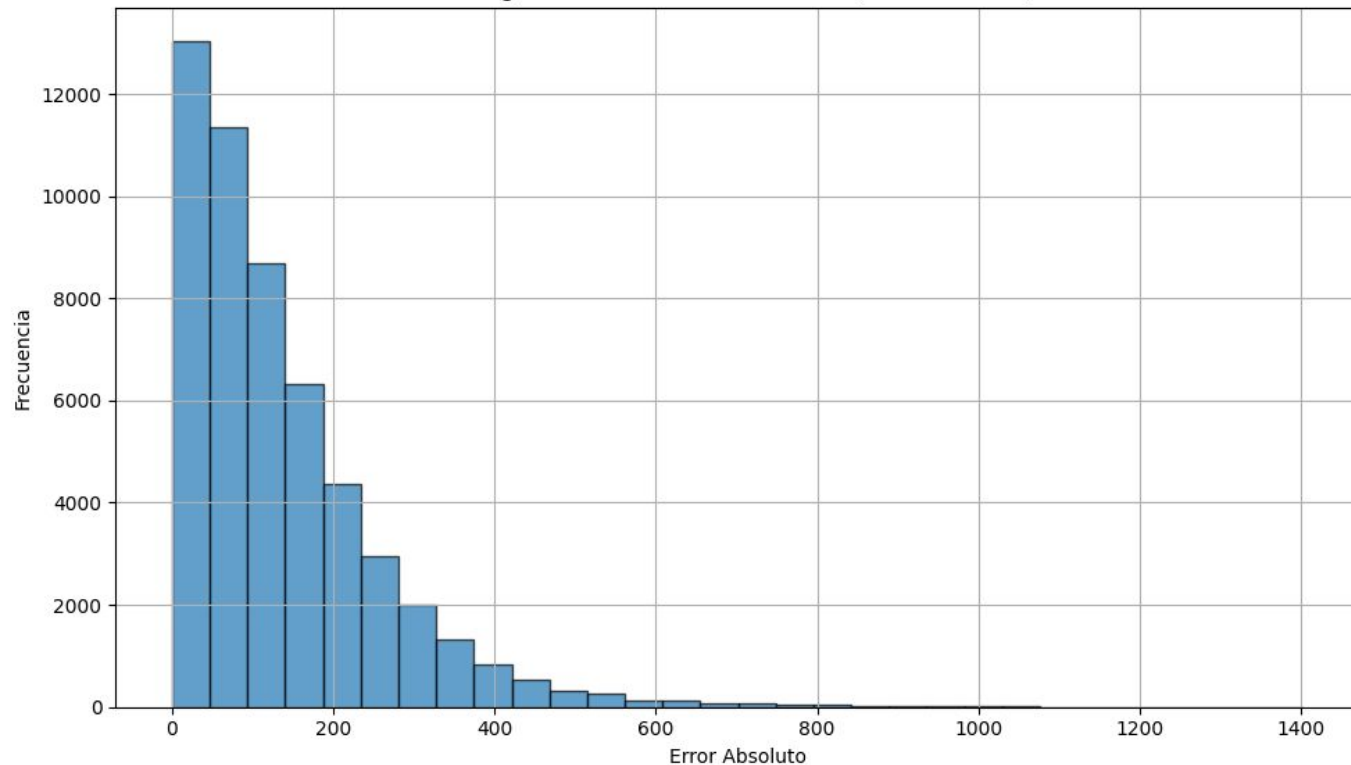
```
model_c3 = Sequential([
    LSTM(100, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    LSTM(100, return_sequences=False),
    Dropout(0.2), # Capa Dropout del 20%
    Dense(pred_window) # Capa Dense de salida
])
model_c3.compile(optimizer='adam', loss='mean_squared_error')
# Entrenamiento del modelo
model_c3.fit(X_train, y_train, epochs=epok_as, batch_size=batch_s, verbose=0)
```


Test del Modelo del C5



Errores

Histograma de la Función de Costo (Error Absoluto)



Promedio MSE

33679,2502

Promedio RMSE

183,5190731

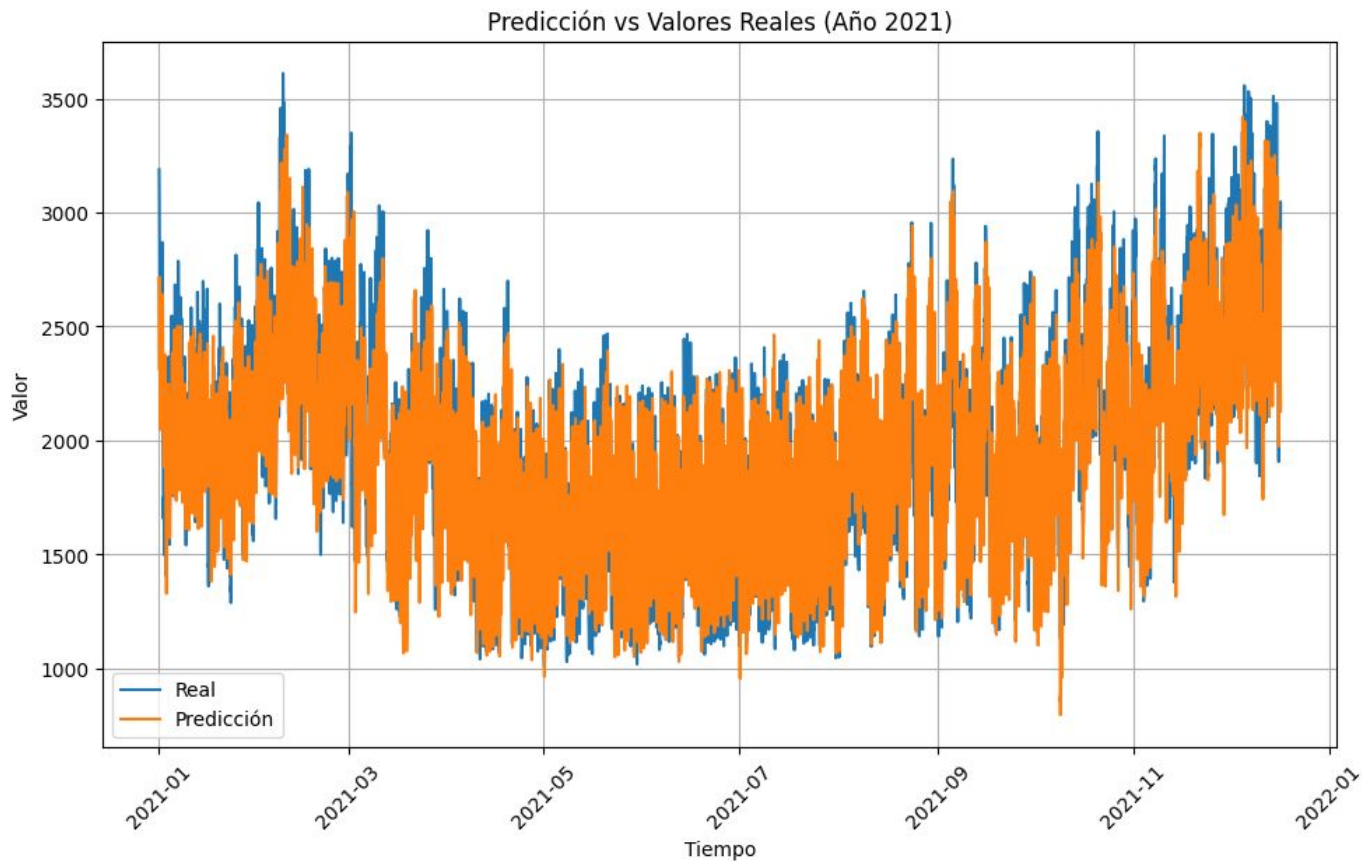
**Promedio MAX
ERROR**

1403,476468

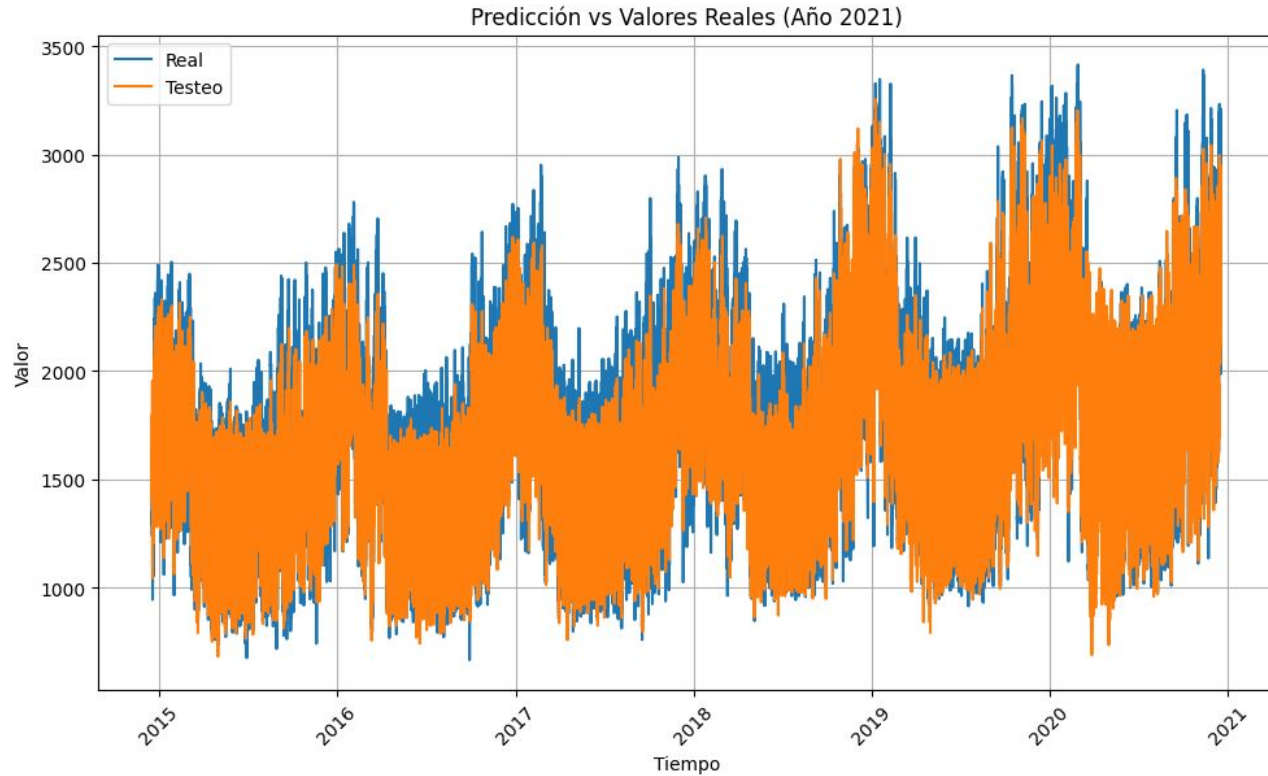
Promedio hsup

1403,476468

PREDICCIONES conjunto 5 año 2021

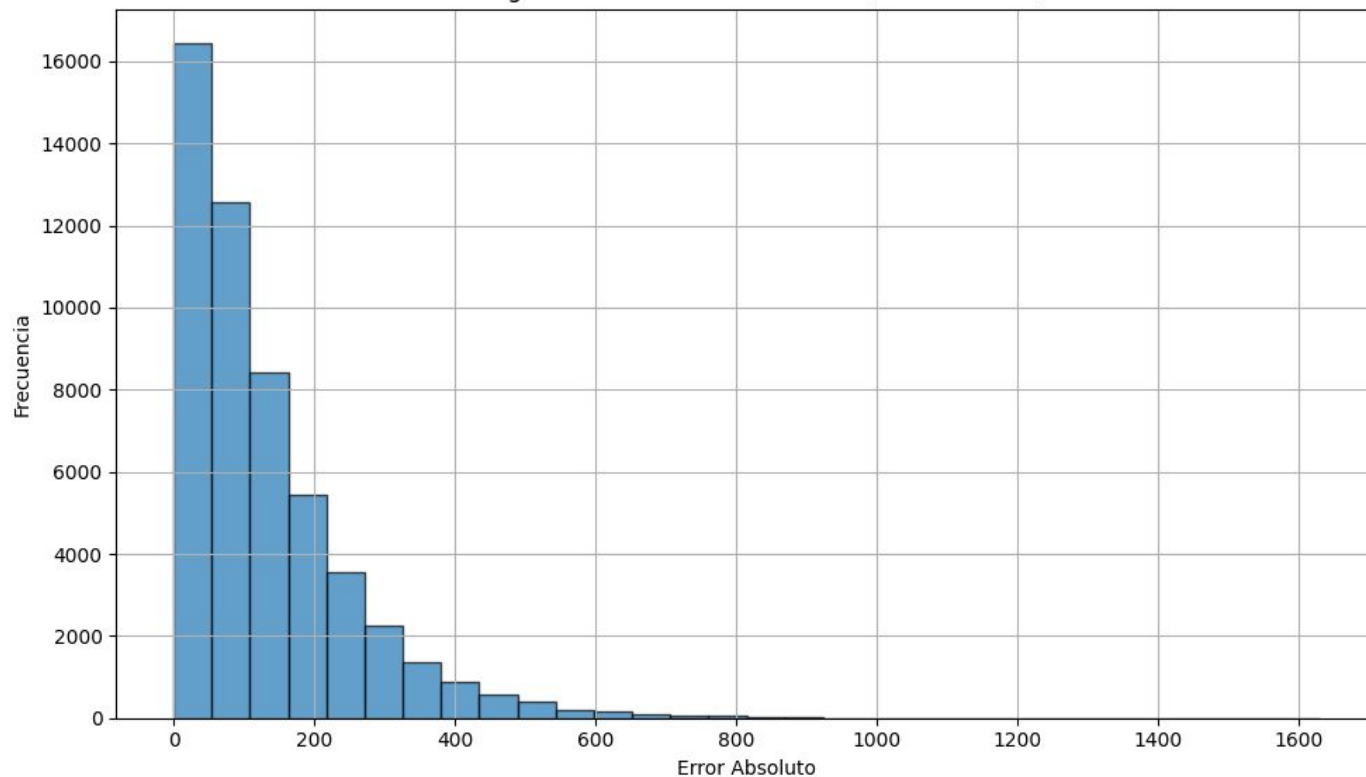


Test del Modelo del C3



Errores

Histograma de la Función de Costo (Error Absoluto)



Promedio MSE

32326,77296

Promedio RMSE

179,7964765

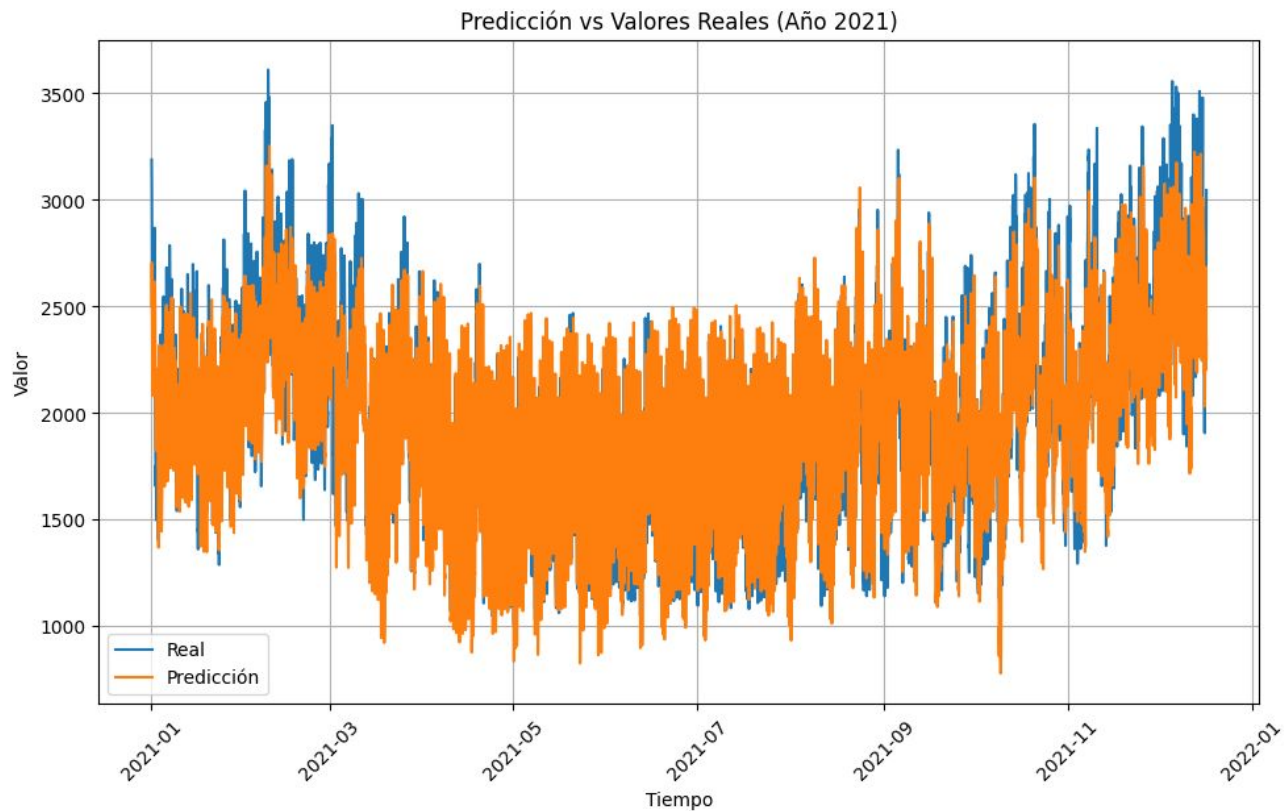
**Promedio MAX
ERROR**

1630,208022

Promedio hsup

1630,208022

PREDICCIONES conjunto 3 año 2021



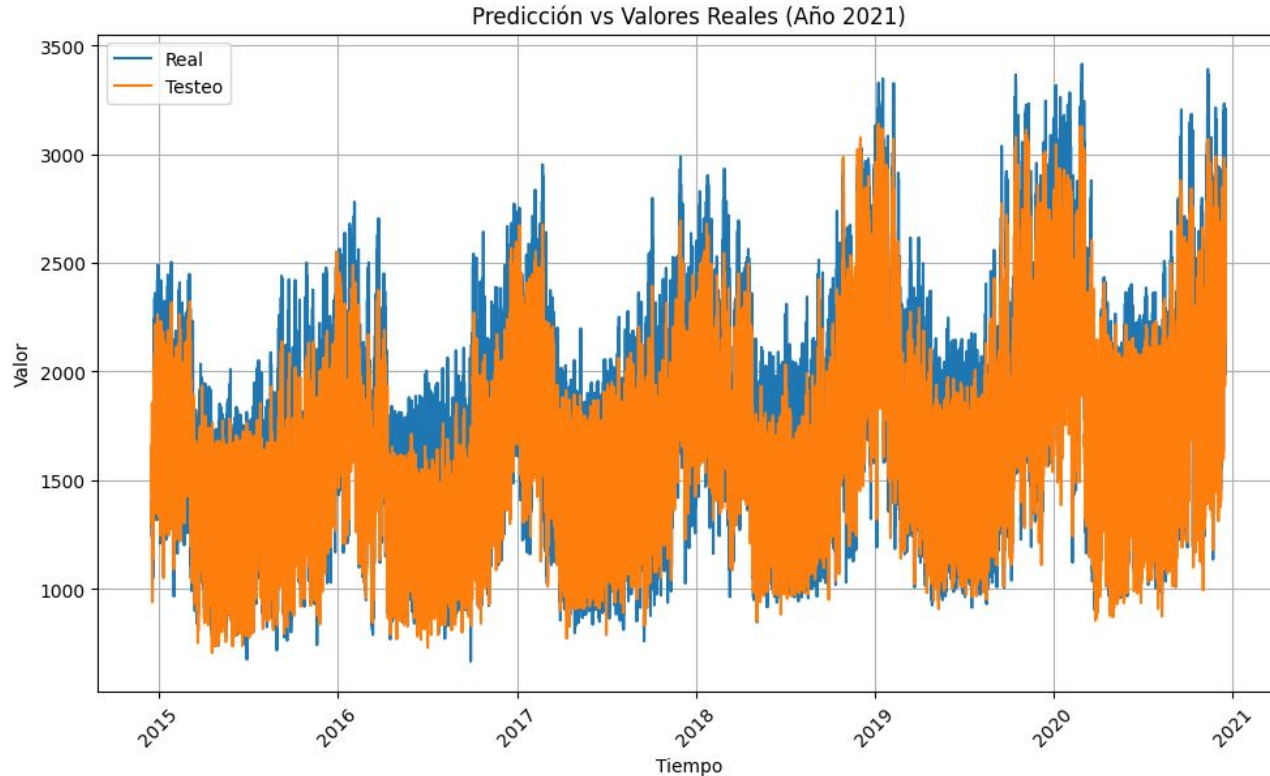
Además tuvimos curiosidad por cómo se desempeñaría los datos de magnitud del viento

Por lo que desarrollamos un conjunto con la magnitud del viento:

```
selected_features = ["Potencia", "Mes", "Humedad", "Temperatura", "Año",  
"dia_sin", "Magnitud", "DirecciónU", "DirecciónV"]
```

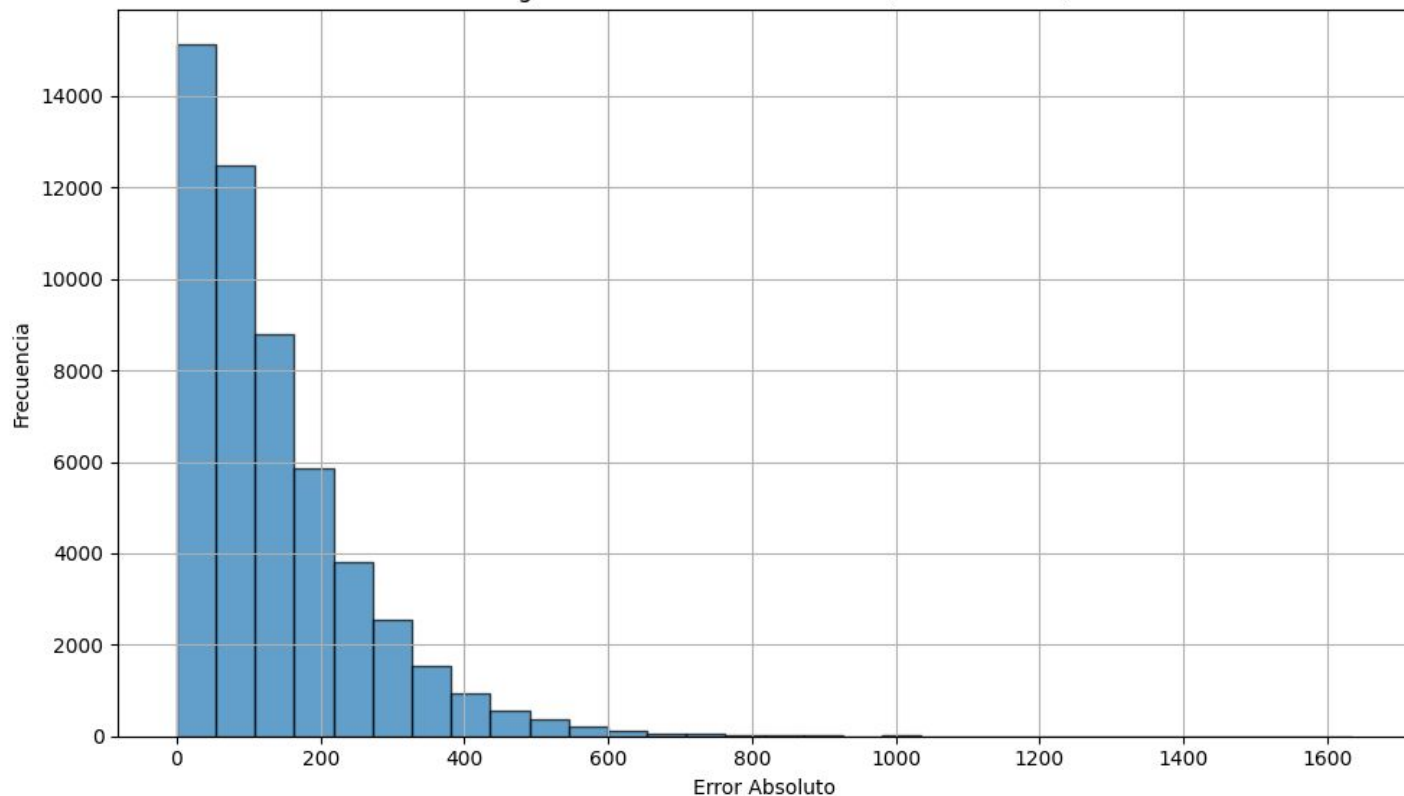
Y su propia red para determinar predicciones del año 2021.

Test del Modelo del Conjunto Viento



Errores

Histograma de la Función de Costo (Error Absoluto)



Promedio MSE

33142,290024

Promedio RMSE

182,0502404

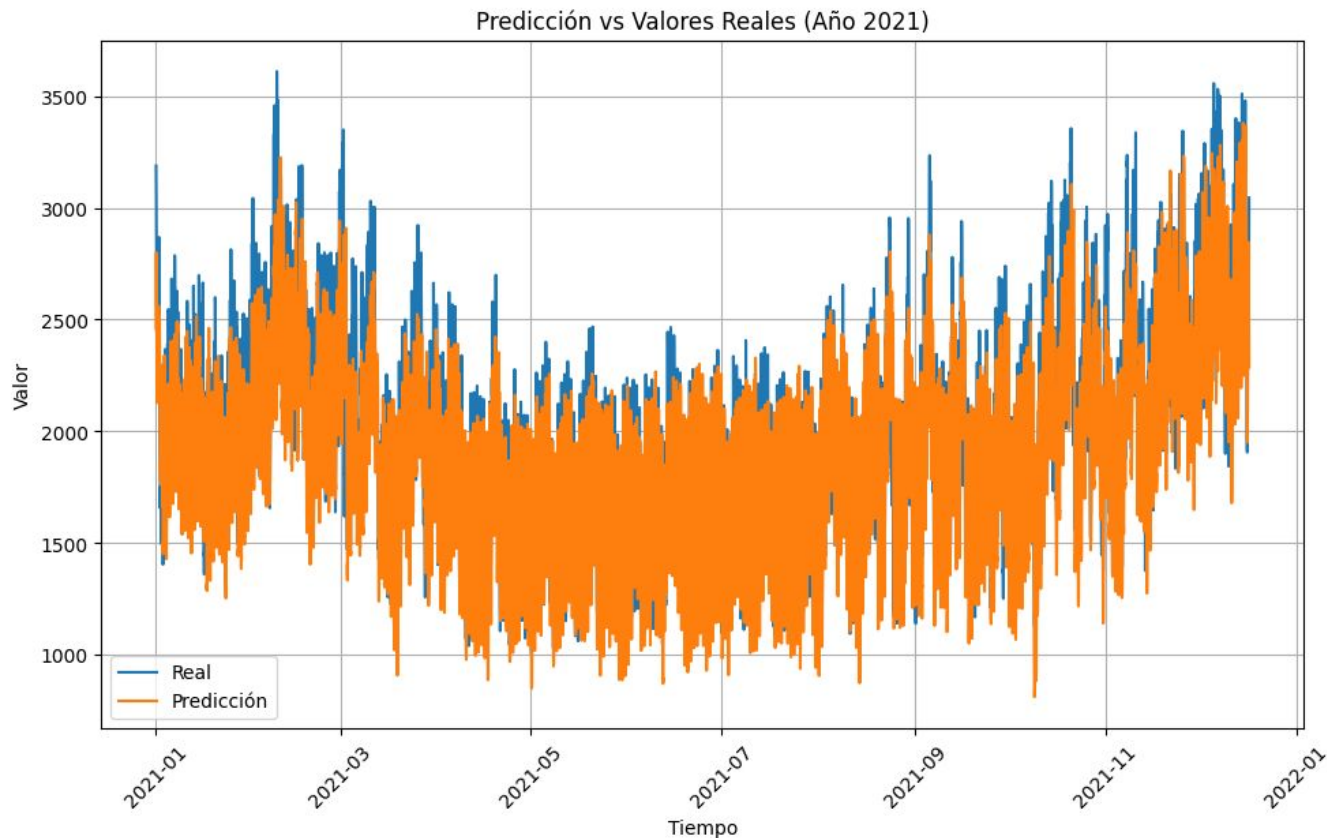
**Promedio MAX
ERROR**

1634,501283

Promedio hsup

1634,501283

PREDICCIONES conjunto Viento año 2021



CONCLUSIONES

Gracias a este trabajo hemos tenido la oportunidad de aumentar nuestras habilidades para desarrollar redes neuronales para realizar predicciones.

Con el dataset estudiado hemos podido reafirmar que la temperatura y el tiempo son factores muy influyentes en la demanda de energía, además de observar que la dirección del viento puede ser un factor relevante para la predicción de la demanda de energía.

La red neuronal original ha determinado que el mejor conjunto está compuesto por: [**"Mes", "Temperatura", "dia_sin", "dia_cos", "Año", "Humedad"**].

Mientras que al mejorar los hiperparametros de la red neuronal el conjunto que se destacó fue: [**"Mes_cos", "dia_sin", "Magnitud", "Temperatura", "Año"**].

Mediante el desarrollo de redes específicas para cada conjunto hemos podido observar el impacto de los hiperparametros ya que no solo se pudieron reducir los errores sino que también reducir el tiempo de ejecución del programa.