Università Degli Studi di Milano - Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea Magistrale in Data Science
Progetto per il corso Streaming Data Management and Time Series Analysis

PREVISIONE DI DATI ELETTRICI CON MODELLI ARIMA, UCM E RETI NEURALI

Project Team
Ravazzi Lucia - 852646

Anno Accademico 2020-2021

Indice

1 Introduzione

Il progetto del corso *Streaming Data Management and Time Series Analysis* prevede di implementare diversi modelli ARIMA, UCM e di machine learning per prevedere una serie storica inerente a dati di natura elettrica. Per ogni casistica verranno implementati diversi modelli per proclamare il migliore intraclasse ed infine, i migliori verranno messi a confronto per definire il modello finale. In particolare, quest'ultimo verrà utilizzato per prevedere i dati dei **due mesi** successivi rispetto a quelli forniti.

2 Dataset

Il dataset si compone di una serie storica univariata. Di conseguenza, a meno dei regressori esterni, i dati futuri saranno previsti partendo dai valori passati. I dati sono stati raccolti dal $2018-09-01\ 01:00:00$ fino al $2020-08-31\ 24:00:00$ e sono stati campionati ogni ora. In totale, la serie si compone di 17518 dati.

2.1 Preprocessing

I dati forniti non presentano particolari criticità.

L'unico aspetto che bisogna tenere in considerazione è il cambio dell'ora: infatti, le giornate $2019-03-31$ e $2020-03-29$ sono formate da 23 ore perché la terza ora non è presente. Per uniformare tutte le giornate, si è deciso di inserire questi due valori mancanti il cui valore sarà uguale a quello dell'ora precedente. Questa è una scelta arbitraria e, dato che il numero di dati a disposizione è sufficientemente grande, non influirà sul resto del lavoro.

Ci si potrebbe aspettare anche giorni nel mese di ottobre con 25 ore ma, in questo dataset, non sono presenti.

Non sono presenti duplicati.

2.2 Esplorazione

In generale, una serie storica potrebbe essere caratterizzata da stagionalità, trend, cicli e l'impatto delle festività: esplorando i dati sarà possibile capire la presenza di questi pattern.

Osservando la serie storica in figura 1, si osserva che i dati sono caratterizzati da un trend globalmente decrescente: potrebbe essere legato alla presenza del covid. Inoltre, si nota una crescita significativa nella stagione estiva per entrambi gli anni, in particolare nei mesi di luglio e agosto, ed una crescita meno spiccata, ma comunque significativa, nei mesi di gennaio e febbraio. In seguito a queste crescite si osserva sempre una decrescita. È chiaro che la serie ha un trend locale che cambia nel tempo alternando fasi di crescita e decrescita.

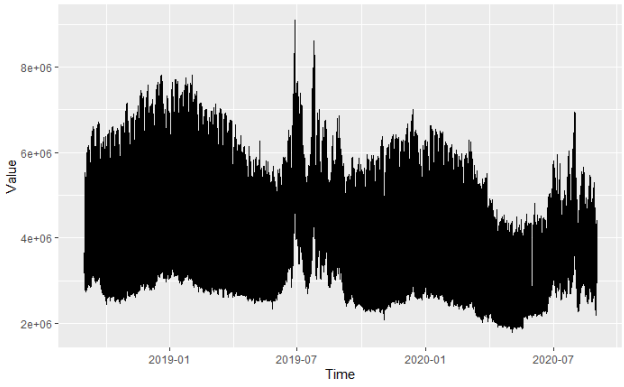


Figura 1: Serie storica fornita.

Alcune stagionalità possono essere intuite analizzando la figura 1 ma, per ottenere delle considerazioni più quantitative, la trasformata di fourier potrebbe aiutare. Per individuare le frequenze più rilevanti è possibile utilizzare il periodigramma in figura 2, i.e. una stima della densità spettrale per le diverse frequenze individuate. Le frequenze associate alla maggior densità spettrale sono legate alle stagionalità dei dati forniti.

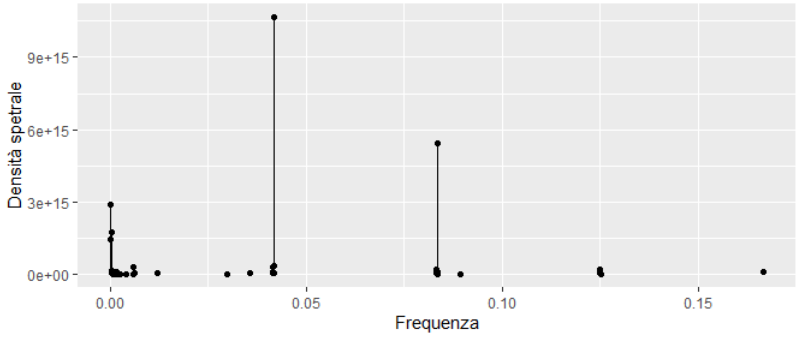


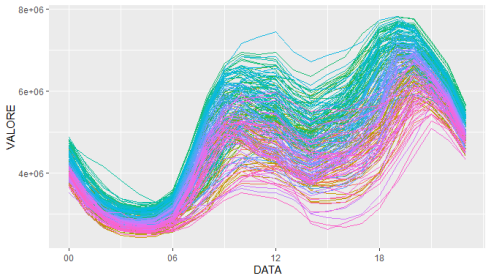
Figura 2: Periodigramma della serie storica.

Nel periodigramma in figura 2 si osserva che vi sono tre picchi principali. In ordine decrescente di densità spettrale, si associano alle frequenze caratterizzate da tali densità le seguenti stagionalità:

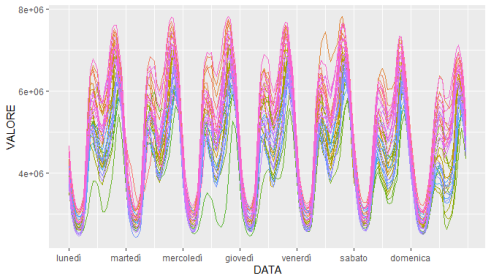
1. Stagionalità giornaliera: ogni 24 ore.
2. Stagionalità annuale: ogni $365 * 24$ giorni
3. Stagionalità settimanale: ogni $7 * 24$ giorni.

Ulteriori stagionalità residue sono multiple di quelle di cui sopra.

Per cercare di approfondire questi pattern, ognuno di essi verrà analizzato e commentato.



(a) *Profili giornalieri.*



(b) *Profili settimanali.*

(c) *Profili annuali.*

Figura 3: Per i profili giornalieri e settimanali è stato utilizzato solo un terzo del dataset (per problemi computazionali), mentre per il profilo annuale sono stati utilizzati tutti i dati.

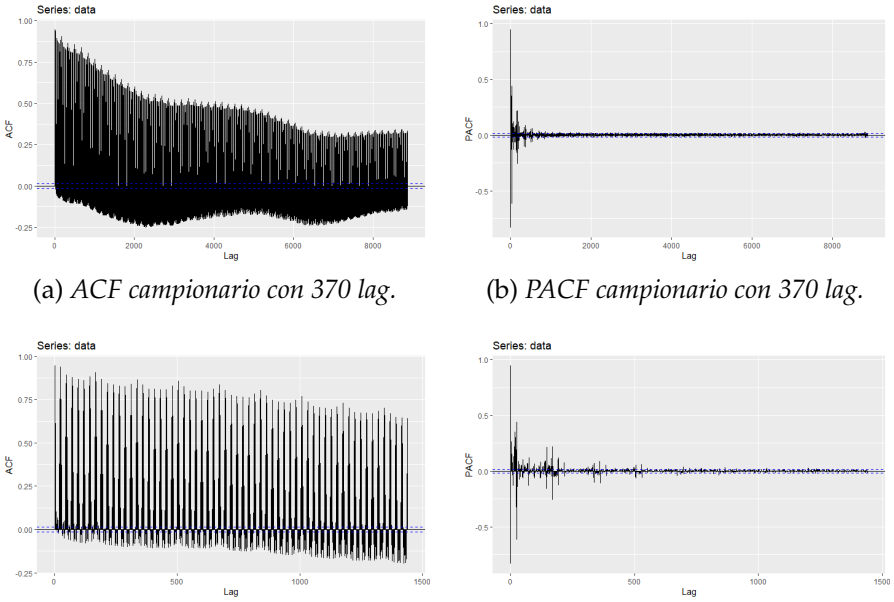
Si osserva dalla figura 3a che la stagionalità giornaliera non è deterministica. Tale profilo si caratterizza da due massimi locali di cui quello pomeridiano risulta anche essere globale nella stessa giornata.

Per quanto riguarda la stagionalità settimanale in figura 3b, anch'essa non è deterministica e si manifesta con gli ultimi due giorni della settimana con picchi meno pronunciati rispetto agli altri.

Infine, la stagionalità annuale è raffigurata in 3c: non è possibile analizzare questa stagionalità per periodi lunghi perché si hanno a disposizione solo due anni di dati. Anche se non si hanno a disposizione molti dati per catturare quest'ultima stagionalità, è necessario tenerla in considerazione perché le previsioni due mesi avanti non possono risultare rappresentative solo considerando il trend, la stagionalità giornaliera e settimanale.

Per quanto riguarda le correlazioni della serie storica, i grafici di ACF e PACF possono essere utili per capirne di più.

Dalla figura 4 si nota che entrambi tutti grafici mostrano diverse complessità e sono abbastanza rumorosi sia perché stiamo usando un campione, che quindi introdurrà del rumore nelle stime delle autocorrelazioni, ed anche, perché il processo generatore dei dati è abbastanza articolato: si compone di tre stagionalità, un trend e bisogna prestare attenzione all'effetto dei regressori. Inoltre, la presenza del covid ha un effetto



(a) *ACF campionario con 370 lag.* (b) *PACF campionario con 370 lag.*

(c) *ACF campionario con 60 lag.* (d) *PACF campionario con 60 lag.*

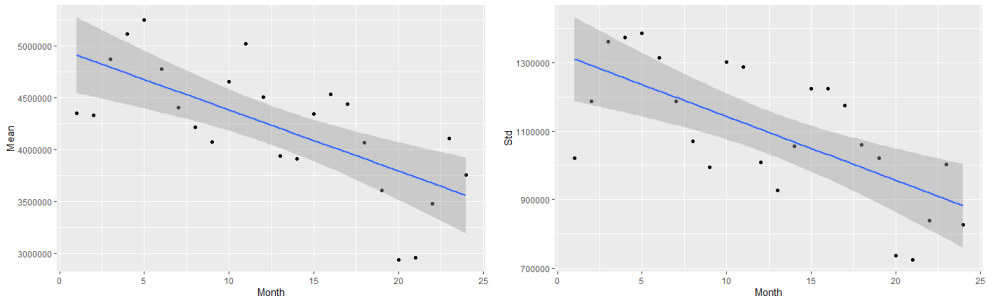
Figura 4: ACF e PACF della serie storica per diversi lag.

importante che potrebbe portare ad una modifica del processo generatore dei dati.

Si nota che il grafico 4a è piuttosto articolato: di conseguenza, fare uno zoom per analizzarlo meglio è opportuno. Analizzando la figura 4c si osservano distintamente la presenza di una stagionalità giornaliera e settimanale: questo è confermato dalla presenza dei picchi per tali lag in figura 4d. Si osserva dalla figura 4b che i lag più importanti sono quelli vicini all'origine e non vi sono picchi particolarmente significativi al lag annuale: probabilmente questo è dovuto al fatto che solo due anni di dati sono disponibili ed inoltre, poiché i dati sono giornalieri, il lag da considerare è molto lungo, ovvero $365 \cdot 24$.

Per capire se la serie è debolmente stazionaria, la media e la deviazione standard dei dati possono fornire informazioni cruciali.

In particolare, si nota dalle figure 5a e 5b un trend decrescente per entrambi i casi: una differenza prima e la trasformazione logaritmica dovrebbero essere prese in considerazione. Per entrambi i segnali si osserva una spiccata oscillazione dovuta alla stagionalità annuale.



(a) *Media dei dati per ogni mese.* (b) *Deviazione standard dei dati per ogni mese.*

Figura 5: Analisi della media e della deviazione standard dei dati per visualizzare la costanza nel tempo dei primi due momenti della distribuzione.

3 Regressori esterni

Come citato in precedenza, i dati a disposizione rappresentano una serie univariata: di conseguenza, le previsioni future dipenderanno dai valori passati. Tuttavia, è possibile integrare ulteriori informazioni grazie a dei regressori esterni. In particolare, i seguenti eventi e festività verranno considerati:

- Festività natalizie, i.e. 24, 25 e 26 di dicembre del 2018 e 2019.
- Capodanno: $2019-01-01$ e $2020-01-01$.
- Epifania: $2019-01-06$ e $2020-01-06$.
- Festività pasquali: $2019-04-21$, $2019-04-22$, $2020-04-12$, $2020-04-13$.
- Ferragosto: $2019-08-15$ e $2020-08-15$.
- Immacolata: $2018-12-08$ e $2019-12-08$.
- Tutti i santi: $2018-11-01$ e $2019-11-01$.
- Festa della repubblica: $2020-06-02$ e $2020-06-02$.
- Festa dei lavoratori: $2019-05-01$ e $2020-05-01$.
- Festa della liberazione: $2019-04-25$ e $2020-04-25$.
- L'effetto del *covid* sarà preso in considerazione a partire dal $2020-03-09$ fino alla fine dei dati a disposizione. Questa scelta è dovuta al fatto che, oltre al primo periodo di lockdown che ha sicuramente influito sui dati, anche il periodo successivo è stato intaccato dall'effetto che ha avuto il virus sulla nostra società e di conseguenza, sui dati energetici.

4 Addestramento e validazione

Per selezionare il miglior modello e modificare gli iperparametri, è necessario conservare parte dei dati per la **validazione**, mentre i rimanenti saranno utilizzati per l'**addestramento**. In particolare, la funzione **mean absolute error** verrà utilizzata come misura dell'errore. Nello specifico, si è deciso di suddividere il dataset nel seguente modo:

1. Il dataset di training è costituito da tutti i dati raccolti prima del $2020-04-01\ 23:00:00$ (con tale ora inclusa), ovvero circa l'80% dei dati. La scelta di tale data è stata effettuata cercando di considerare l'effetto del covid sia nel dataset di training sia di test.
2. Il dataset di test è invece definito a partire da $2020-04-02\ 00:00:00$, ovvero circa il 20% dei dati.

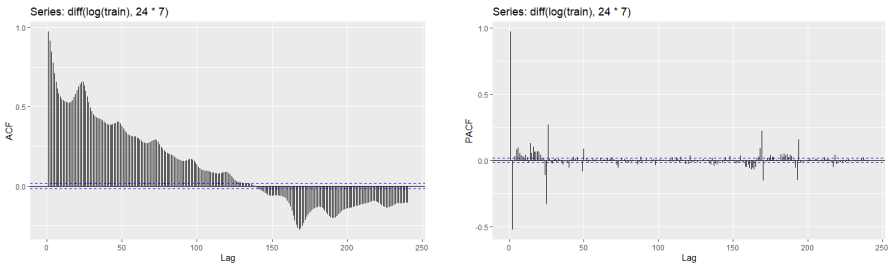
Per una certa casistica di modelli, i.e. ARIMA, UCM o reti neurali, dopo aver trovato il modello migliore attraverso questa strategia, tutti i dati verranno utilizzati per addestrare il miglior modello che prevederà i dati due mesi in avanti come richiesto.

5 Modelli

5.1 ARIMA

Il primo modello richiesto è un modello *ARIMA*. La strategia implementata è stata quella di partire da modelli semplici, in modo anche da avere una baseline, ed aumentare la complessità. La seguenti strategie sono state implementate:

1. *Modello ARIMA con stagionalità $s = 24*7$ stocastica*. Prima di giungere a questo modello, diverse altri modelli sono stati implementati considerando modelli ARIMA senza stagionalità e con stagionalità stocastica pari a 24. A partire dalle considerazioni fatte per quest'ultimi modelli, che verranno omesse per brevità, è stato evidenziato che la scelta di utilizzare una stagionalità stocastica $24 * 7$ potrebbe essere la chiave vincente per tenere in considerazione sia l'andamento giornaliero sia quello settimanale (dato che quest'ultima è un multiplo della prima) senza doverle modellare separatamente. In figura 6a si osserva un'oscillazione dei valori delle autocorrelazioni campionarie i quali evidenziano la presenza di un processo autoregressivo di ordine almeno pari a 2, come suggerisce anche il grafico 6b. Potrebbero esserci dei processi MA non stagionali. Per quanto riguarda la parte stagionale $s = 24*7$, sia il grafico 6a sia la figura 6b mostrano picchi significativi per i suoi multipli. Sicuramente vi è una componente MA stagionale dovuta alla differenza che viene evidenziata dal picco negativo a $s = 24*7$ in figura 6a e dai picchi



(a) *ACF campionario con* $24 * 10$ *lag.* (b) *PACF campionario con* $24 * 10$ *lag.*

Figura 6: ACF e PACF per un massimo numero di lag pari a $24 * 10$ della serie $log(diff(ts, 24 * 7))$.

sistematici negativi in figura 6b. I grafici suggeriscono anche la presenza di una componente autoregressiva stagionale. Questi modelli sono sicuramente migliori rispetto a quelli con la sola stagionalità giornaliera ma sono ancora incompleti: è necessario tenere in considerazione il susseguirsi delle stagioni.

2. *Modello ARIMA con stagionalità stocastica* $s = 24 * 7$ *e deterministica* $24 * 365.25$. I modelli ARIMA possono considerare solo una stagionalità stocastica che sarà quella settimanale e non quella annuale per i seguenti motivi:

   - Poiché sarebbe necessario applicare una differenza prima stagionale, considerare la stagionalità annuale come stocastica comporterebbe la perdita di un anno di dati, ovvero la metà del totale dei dati a disposizione.
   - Considerare tale stagionalità annuale come stocastica porterebbe ad instanziare modelli molto complessi che richiederebbero risorse computazionali non indifferenti, soprattutto in termini di tempo, per poter convergere alla soluzione.

   Di conseguenza, dei regressori esterni si occuperanno di modellare questa stagionalità: in particolare, quest'ultima verrà introdotta attraverso la somma di funzioni oscillanti i cui coefficienti verranno ottenuti attraverso una procedura di ottimizzazione. Il periodo è fisso, mentre l'iperparametro che può essere modificato è il numero di funzioni oscillanti da considerare.

   Verrà quindi preso in considerazione il miglior modello con stagionalità stocastica $24 * 7$ per poi aggiungere i regressori rappresentativi della stagionalità $24 * 365.25$.

3. *Modello ARIMA con stagionalità stocastica* $s = 24 * 7$ *e deterministica* $24 * 365.25$ *con regressori esterni*. Come regressori sono stati scelti sono quelli quelli che risultano significativi, i.e. la stima del parametro è maggiore di due volte il valore assoluto della deviazione standard, ovvero quelli che identificano la festa dei lavoratori, tutti i santi e il periodo covid. Questo è il modello finale, completo ed anche il migliore:

```
p = 3; d = 0; q = 2;
P = 1; D = 1; Q = 1;

mod = Arima(train,
    c(p,d,q),
    list(order = c(P,D,Q), period = 24*7),
    xreg = reg_tot_train,
    include.constant = TRUE,
    lambda = 0,
    method = "CSS")
```

dove i regressori esterni prendono in considerazione sia la stagionalità annuale con le prime 16 frequenze fondamentali e le festività significative citate in precedenza.

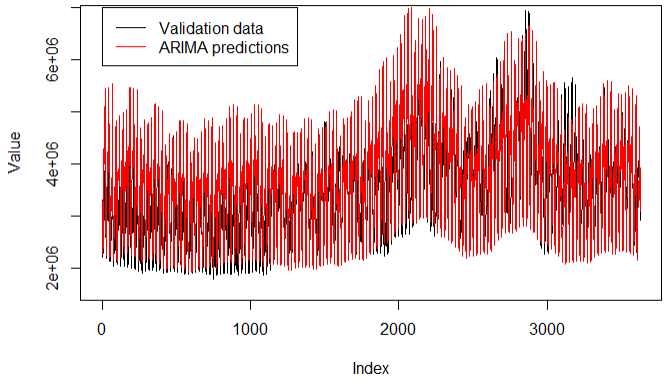


Figura 7: Validation set e previsioni dei suoi valori attraverso il miglior modello ARIMA ottenuto.

Dalla figura 7 si osserva che le previsioni del set di validazioni seguono i pattern presenti dati: la misure di performance sono riportate in tabella 1. Il modello non è perfetto: il correlogramma mostra che gli errori ottenuti non seguono un processo di tipo white noise e la loro distribuzione non è propriamente gaussiana. Cercare di perfezionare questo modello è possibile ma non è semplice a causa della complessità del meccanismo generatore dei dati. Tuttavia, anche se queste condizioni non sono rispettate perfettamente, le previsioni sembrano essere ragionevoli e potrebbero dare delle buone previsioni per il mese di settembre ed ottobre.

Tutti questi modelli sono stati implementati in R.

5.2 UCM

La seconda tipologia di modelli richiesti sono gli UCM (Unobserved Component Model). Per i dati a disposizione, le componenti da modellare sono chiare:

- Local linear trend.
- Stagionalità giornaliera modellata attraverso variabili dummy.
- Stagionalità settimanale modellata attraverso funzioni oscillanti.
- Stagionalità annuale modellata attraverso funzioni oscillanti.
- Regressori esterni.

La scelta di utilizzare le variabili dummy per la stagionalità giornaliera deriva dal fatto che quest'ultima non è propriamente liscia e per modellarla servirebbero diverse funzioni oscillanti: di conseguenza, la scelta delle dummy è appropriata in questo contesto. Le rimanenti stagionalità son invece più lisce e per questo motivo, sono state scelte le funzioni oscillanti per modellarle.

L'approccio per implementare questi modelli segue un processo analogo a quello implementato per gli ARIMA: partire dal modello più semplice, aumentare di complessità e modificare gli iperparametri associati ai valori iniziali.

Si riscontra che anche nei modelli più semplici, l'ottimizzatore numerico presenta dei problemi di overflow: probabilmente questo è dovuto all'ordine di grandezza dei dati messi a disposizione. Per risolvere questo problema è sufficiente scalare i dati di un fattore che è stato scelto pari a 100000.

Inoltre, per dare stabilità all'ottimizzatore numerico, non sono state utilizzate delle condizioni diffuse ma, attraverso la media e varianza della serie numerica scalata, sono state specificate.

Dopo diversi tentativi, si è giunti al miglior modello:

```
nh_annual = 16
nh_week = 8

mod = SSModel(as.numeric(y) ~ (SSMtrend(2, list(NA, NA)) +
                SSMseasonal(24, NA, "dummy") +
                SSMseasonal(24*7, NA, "trig",
                            harmonics = 1:nh_week) +
                SSMseasonal(24*365, NA, "trig",
                            harmonics = 1:nh_annual)),
              H = NA)

fit1 = fitSSM(mod,
              log(c(vy/10000000000,   # level
```

```
        vy/10000000000,    # slope
        vy/100000,         # stag. giornaliera
        vy/10000,          # stag. settimanale
        vy/1000000,        # stag. annuale
        vy/10              # epsilon
        )
    ),
updt,
update_args =  list(nh_annual = nh_annual,
                    nh_week = nh_week),
 control = list(maxit = 500)
)
```

La serie storica scalata y si caratterizza dei dati di training più valori nulli per tutta la lunghezza del set di validation: in questo modo, i valori previsiti saranno proprio quelli mancanti grazie all'utilizzo del filtro di kalman. Dal modello precedente si ricavano le previsioni per il set di validation, mostrate in figura 8.

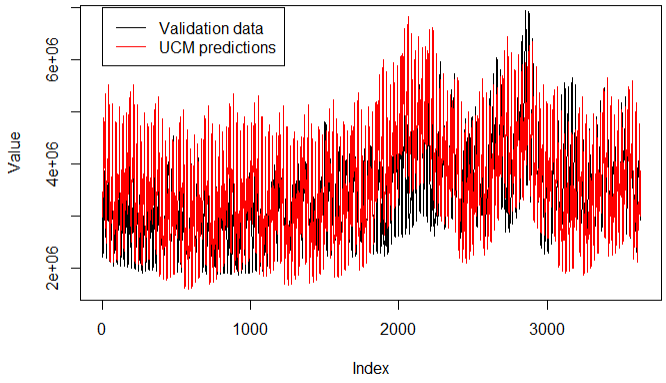


Figura 8: Validation set e previsioni dei suoi valori attraverso il miglior modello UCM ottenuto.

5.3 Reti neurali

Per esplorare più metodi possibili per la previsione della variabile d'interesse verranno utilizzate anche tecniche di deep learning. In particolare, le reti LSTM e GRU saranno progettate, addestrate e validate. Bisogna tenere in considerazione che tali architetture si rivelano particolarmente efficaci nel momento in cui molti dati sono disponibili: in questo modo, la rete potrà automaticamente estrarre i pattern d'interesse. In questo caso specifico, alcuni pattern si ripetono poche volte, e.g. le festività, perché solo due anni di dati sono a disposizione e di conseguenza, il modello non riuscirà a catturare in modo ottimale alcuni andamenti.

Prima della fase di addestramento i dati sono stati preprocessati: sia la standardizzazione sia la normalizzazione sono state effettuate ma la seconda si è rivelata migliore in termini di errore in fase di addestramento e validazione.

A partire da un set di dati $\{X_1, ..., X_N\}$, l'approccio utilizzato è di tipo supervisionato e prevede di creare delle finestre $\{X_k, ..., X_{k+n_{in}}\}$ di dimensione n_{in} e la rispettiva finestra traslata $\{X_{k+n_{out}}, ..., X_{k+n_{in}+n_{out}}\}$ $\forall k$ di un lag n_{out}: in questo modo, data una finestra di punti di dimensione n_{in}, si vuole cercare di trovare quella funzione f che preveda n_{out} passi in avanti. Questo ragionamento viene iterato per tutti gli N dati partendo dal primo indice e spostando la finestra di input di una sola unità fino a quando la fine della finestra di output non raggiunge la fine del dataset.

Nel caso d'interesse, l'iperparametro $n_{out} = 24 * (30 + 31)$ perché si vuole prevedere il mese di settembre e ottobre, mentre per il valore di n_{in} sarà necessario trovare quello ottimale. In particolare, si è deciso di creare le seguenti finestre di input con dimensioni:

- 2 mesi.
- 2 mesi e 2 settimane.
- 1 mese.
- 2 settimane.

Per addestrare i modelli è deciso di utilizzare google colab e sfruttare i metodi *cuDNNLSTM* e *cuDNNGRU* che accelerano in modo significativo l'addestramento della rete. Tuttavia, a causa della limitata disponibilità della RAM, non è stato possibile implementare modelli complessi che avrebbero potuto fornire risultati migliori.

Una volta definita la dimensione della finestra di input, è possibile definire diverse architetture: in generale, sono state definite delle reti neurali con uno o due blocchi LSTM o con uno o due blocchi GRU. Sono stati definiti gli stessi modelli, e.g. numero di neuroni, tasso di dropout, learning rate, batch size etc., per ogni rete LSTM e GRU. La strategia *early stopping* è stata utilizzata per controllare l'overfitting dei modelli con una pazienza pari a 5. In totale sono stati implementati circa 50 modelli.

Utilizzando finestre di 2 mesi o più, si riscontra che, sia per i modelli LSTM sia per le GRU, solo il pattern giornaliero viene catturato, mentre quello settimanale è praticamente assente. Si osserva che aumentando il numero di neuroni di output delle celle, i risultati migliorano perché il pattern giornaliero è sempre più definito ma a causa delle limitazioni dell'architettura utilizzata, non si è potuto utilizzare più di 1000 neuroni di output per le reti con una sola cella. Anzi, per avere modelli è stato necessario diminuire la dimensione della batch size.

D'altro canto, considerando delle finestre di dimensione pari a 2 settimane, i risultati sono più soddisfacenti: in particolare, oltre al pattern giornaliero, si osserva l'oscillazione dovuta alla stagionalità settimanale ed annuale. Probabilmente, dato che le reti LSTM e GRU sono delle reti ricorrenti, riescono a ricordare parte del passato e non

hanno bisogno di finestre di input eccessivamente grandi per catturare i pattern fondamentali. Inoltre, non vi sono delle differenze significative tra GRU e LSTM per i modelli implementati.

Il miglior modello ottenuto è il seguente:

```
n_in, n_out = 24*(15), 24*(30+31)

# modello LSTM.
model = Sequential()
model.add(CuDNNLSTM(2000, input_shape=(n_in, n_features)))
model.add(Dropout(0.2))
model.add(Dense(trainY.shape[1]))

model.compile(optimizer = Adam(learning_rate=LR,
                               clipvalue=CLIPVALUE),
              loss = 'mae')

callback = [EarlyStopping(monitor = 'val_loss',
                          patience = 5,
                          restore_best_weights = True
                         ),
              PlotLossesKeras()
            ]
# addestramento e validazione.
history = model.fit(trainX, trainY,
                    epochs = 100,
                    batch_size = 32,
                    validation_data = (valX, valY),
                    verbose = 1,
                    shuffle = False,
                    callbacks=[callback]
                    )
```

In questo caso specifico non è possibile mostrare le previsioni sul set di validazione perché le previsioni non sono puntuali ma un set di finestre.

5.4 Tabella riassuntiva dei risultati

Dopo aver addestrato e validato i modelli di cui sopra, è necessario riassumere i risultati per capire il modello migliore. Si osserva dalla tabella 1 che il miglior modello è proprio l'ARIMA. Il miglior modello UCM non si discosta molto dal risultato dell'ARIMA, in termini di errore di validazione, ed anche questo può essere considerato un buon modello per le previsioni. Le reti neurali invece non sembrano performare molto bene in

Modello	MAE TRAIN	MAE validation
ARIMA	59699	480520
UCM	107765	513543
ML	4496664	3604482

Tabella 1: Tabella riassuntiva dei migliori modelli ottenuti sul set di validation

.

questo contesto: probabilmente questo è dovuto al fatto che due anni di dati non sono sufficienti per apprendere in modo ottimale i pattern dei dati. Inoltre, l'architettura hardware messo disposizione da Colab non ha permesso di costruire reti sufficientemente complesse che avrebbero potuto dare risultati più soddisfacenti. Si osserva che il modello migliore ottenuto con le reti neurali riesce comunque a catturare il pattern giornaliero, settimanale ed annuale come mostra la figura 9.

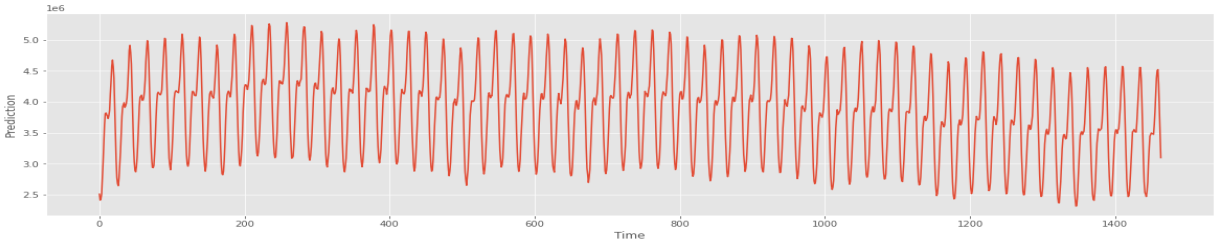


Figura 9: Previsioni due mesi in avanti ottenuto con le reti neurali.

I risultati finali ottenuti sono riportati in figura 10.

6 Conclusioni

Per questi dati, la miglior strategia per la previsione dei dati elettrici forniti è l'ARIMA. Anche i modelli UCM sono in grado di catturare i pattern principali dei dati, mentre le reti neurali si rivelano le peggiori.

Questa è solo una possibile scelta dei modelli dato che è possibile sfruttare altre librerie come Prophet, NeuralProphet o KNN.

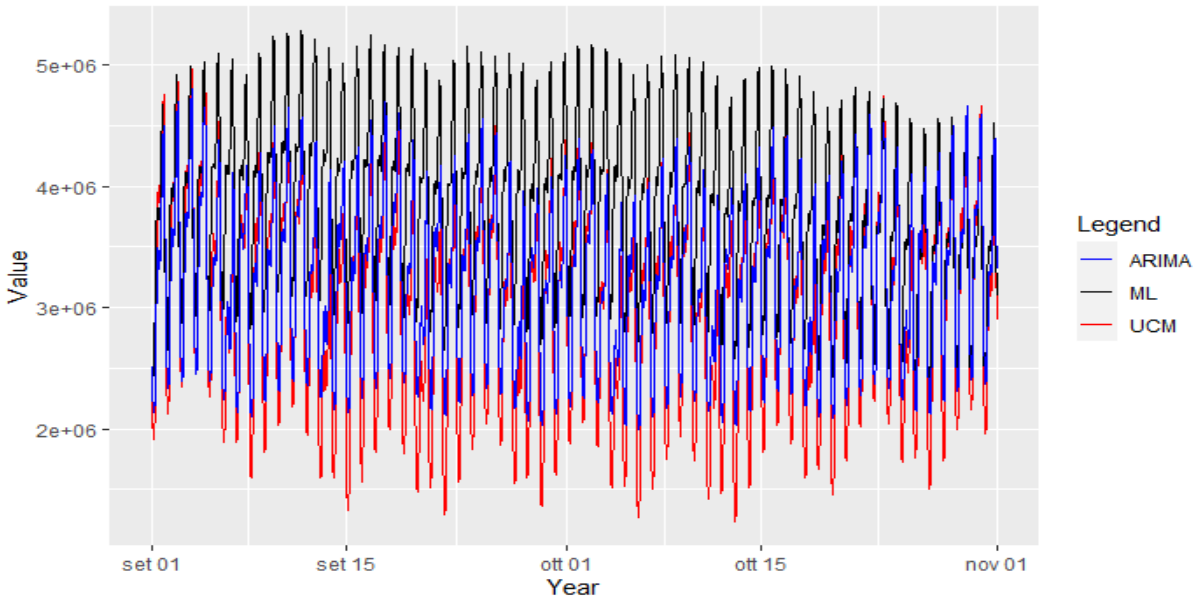


Figura 10: Previsioni due mesi in avanti con i tre modelli migliori per ciascuna strategia.