
https://www.meetup.com/DSDTmtl
Sept
29
2021
Data Science | Design | Technology

Data Science | Design | Technology

https://www.meetup.com/DSDTmtl

Please, don't forget to mute yourself

JL Maréchaux

DSDT Co-Organizer
(Google)

Christophe Pere

Chief Data Scientist
La Capitale Insurance

https://www.meetup.com/DSDTmtl

Agenda

DSDT Meetup - Sept 29, 2021

3:45 - 4:00 Arrival & Networking

4:00 - 4:15 News & Intro

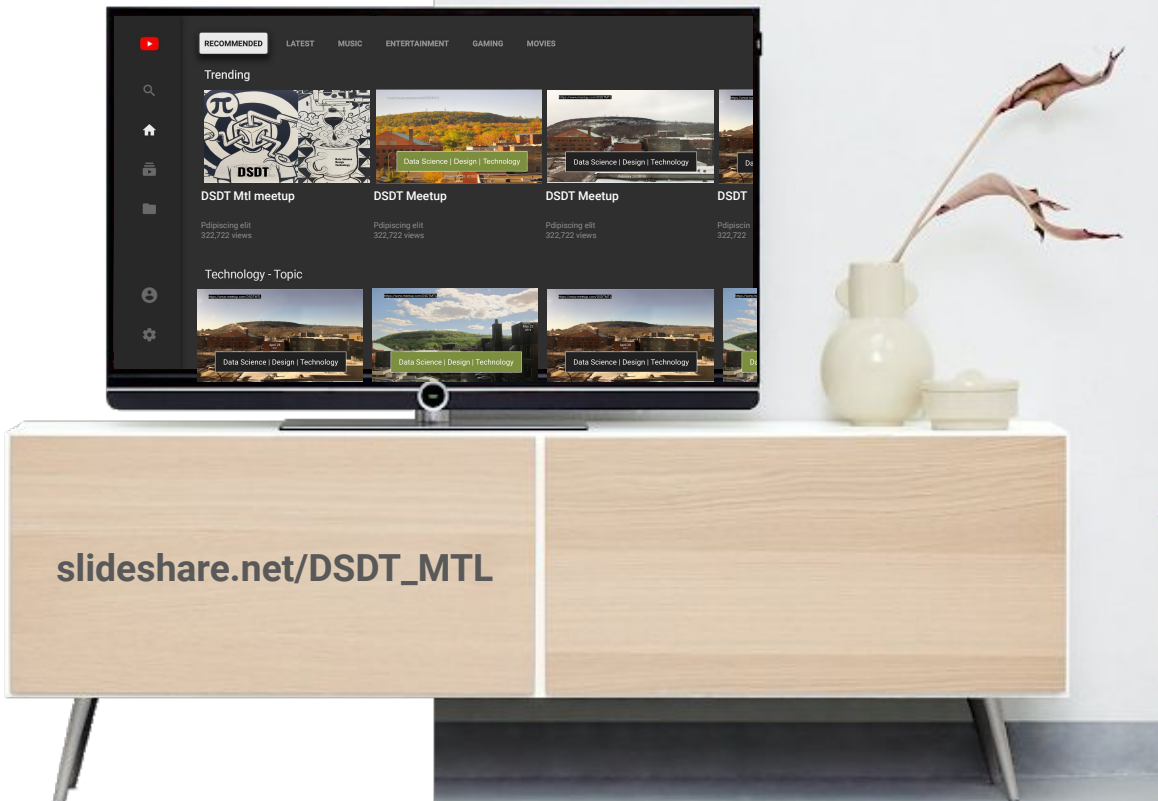
4:15 - 5:15 Time series in financial domain: A deep learning approach

5:15 - 5:30: Virtual Snack & Networking

Virtual Meetups

Until we can do in-person events again in Montreal…

Past (and future) presentations available on Slideshare.



DSDT meetups in 2021

Monthly cadence, on Wednesdays.
Incredible sessions already planned. Contact us with your expectations & ideas.

April 28 — ML Validation

May 26 — Reinforcement Learning

July 21 — Explainable AI

Aug 25 — Rodent brain & NN

Sept 29 — RNN & Time Series

http://bit.ly/DSDTsurvey2021

Your ideas, your meetup.

Christophe Pere
Chief Data Scientist
La Capitale Insurance

Data Science | Design | Technology

Time series in financial domain

A deep learning approach

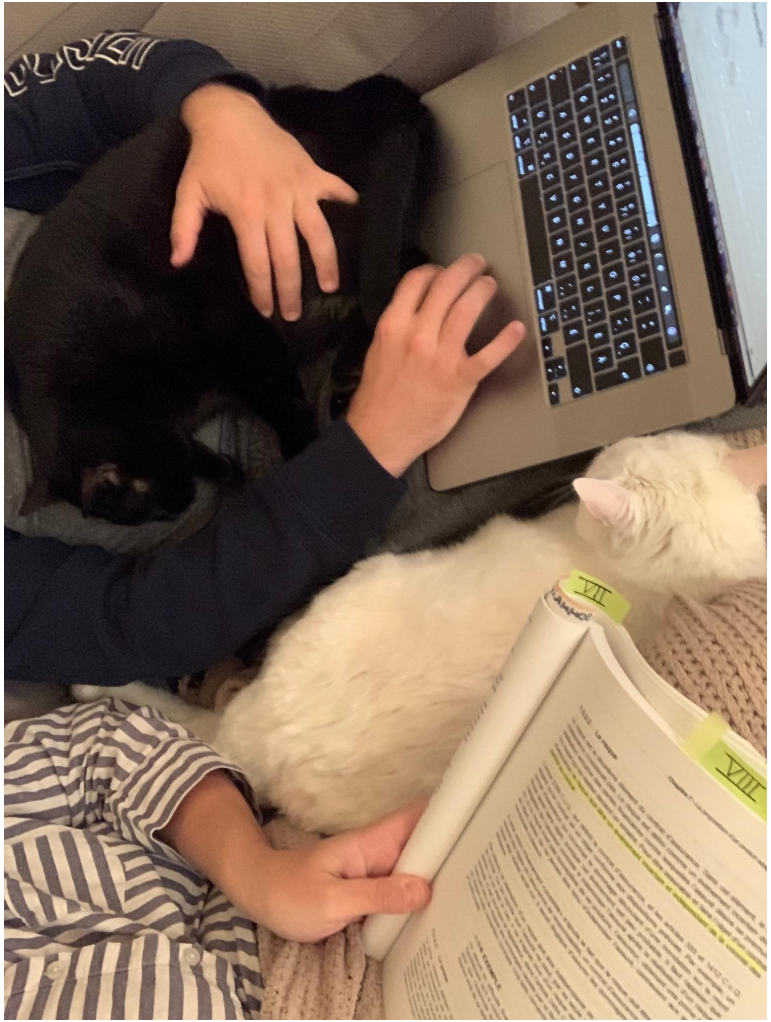
Content

- Who am I
- Time Series
- RNN Family
- Applications with real data
- Discussion
- Conclusion
- Future

Who am I

- French guy
- PhD in Astrophysics
- Researcher since 2016
  - Automotive market
  - Autonomous vehicle
  - NLP
  - Synthetic data (imbalanced data)
  - Knowledge Representation
  - Reasoning
- Passion for AI
- Passion for Quantum

When you try to work
on your presentation...



Time Series

> *In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*
>
> Wikipedia[1]

[1] https://en.wikipedia.org/wiki/Time_series

Time Series

> *In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*
>
> Wikipedia[1]

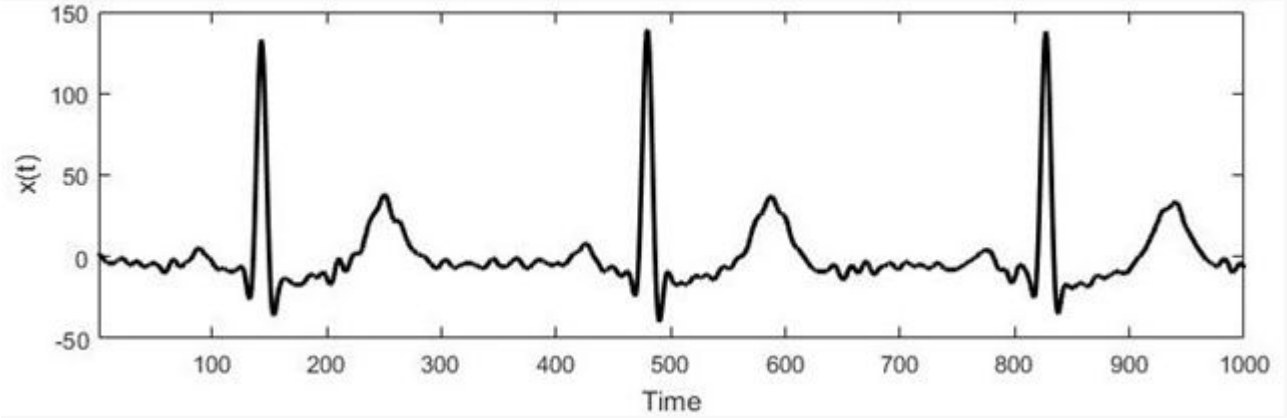
Ok, but in practice?

[1] https://en.wikipedia.org/wiki/Time_series

Time Series

*In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*

Wikipedia[1]



Electrocardiogram[2]

[1] https://en.wikipedia.org/wiki/Time_series

[2] Čepulionis Paulius, Lukoševičiūtė Kristina Electrocardiogram time series forecasting and optimization using ant colony optimization algorithm. Mathematical Models in Engineering, Vol. 2, Issue 1, 2016, p. 69-77

Time Series

*In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*

Wikipedia[1]

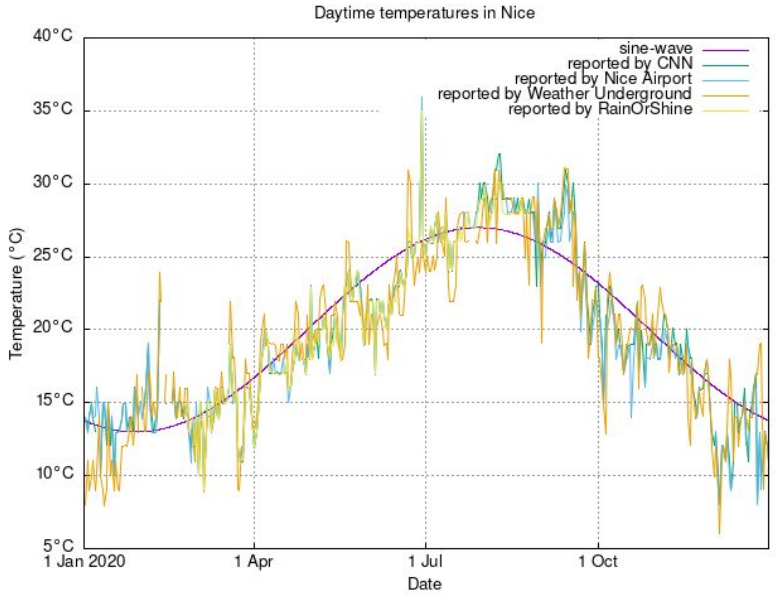
Stock Market[2]

[1] https://en.wikipedia.org/wiki/Time_series
[2] https://ionides.github.io/531w16/final_project/Project13/final_project.html

Time Series

*In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*

Wikipedia[1]



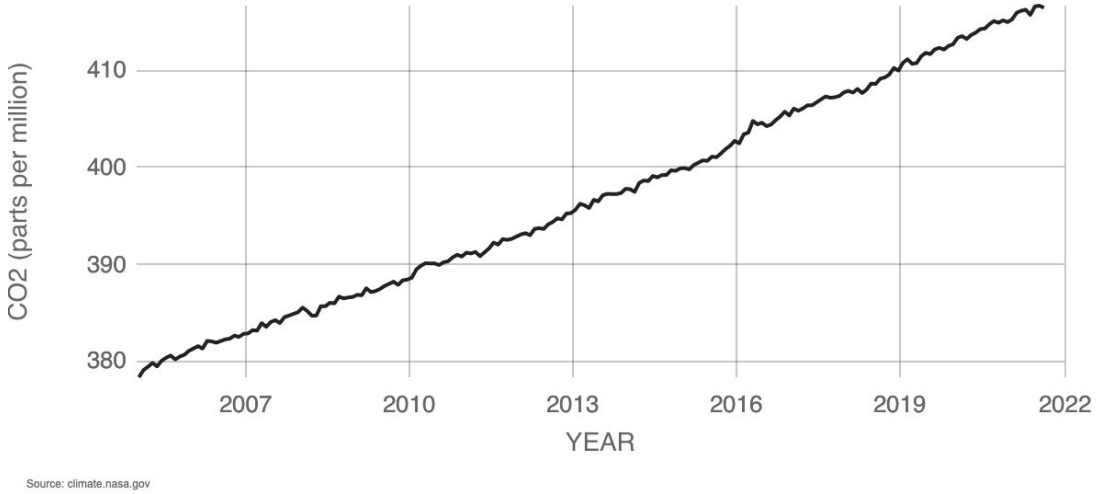
Temperatures in Nice city, France[2]

[1] https://en.wikipedia.org/wiki/Time_series
[2] https://www.w3.org/People/Bos/Nice/tempgraph

Time Series

*In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*

Wikipedia[1]



Carbon Dioxide concentration[2]

[1] https://en.wikipedia.org/wiki/Time_series
[2] https://climate.nasa.gov/vital-signs/carbon-dioxide/

Time Series

> *In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*
>
> Wikipedia[1]

Is that all? Could we use other kind of data with another representation of “time”?

[1] https://en.wikipedia.org/wiki/Time_series

Time Series

> *In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*
>
> Wikipedia[1]

Is that all? Could we use other kind of data with another representation of “time”?

What do we call *Time* ?

[1] https://en.wikipedia.org/wiki/Time_series

Time Series

> *In mathematics, a **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.*
>
> Wikipedia[1]

Is that all? Could we use other kind of data with another representation of “time”?

What do we call *Time* ?

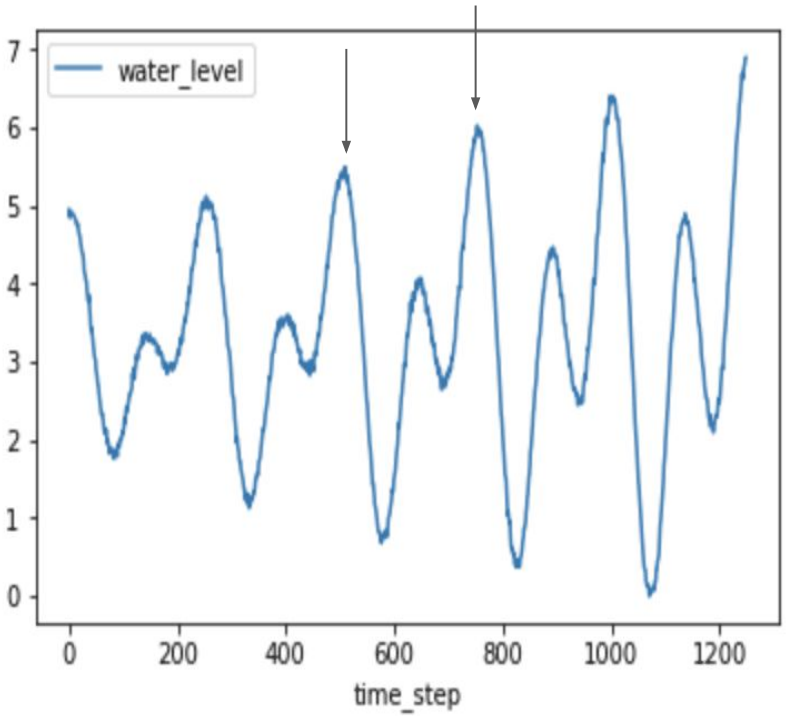
[1] https://en.wikipedia.org/wiki/Time_series

Time Series

Source: How DNA works, Craig Freudenrich, Ph.D., https://science.howstuffworks.com/life/cellular-microscopic/dna.htm

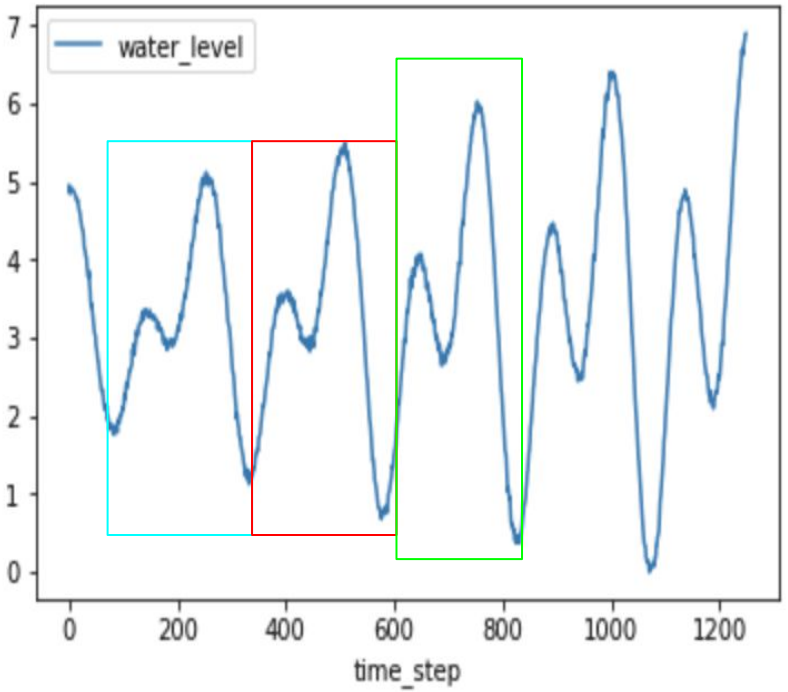
Time Series - Composition

Autocorrelation



Time Series - Composition

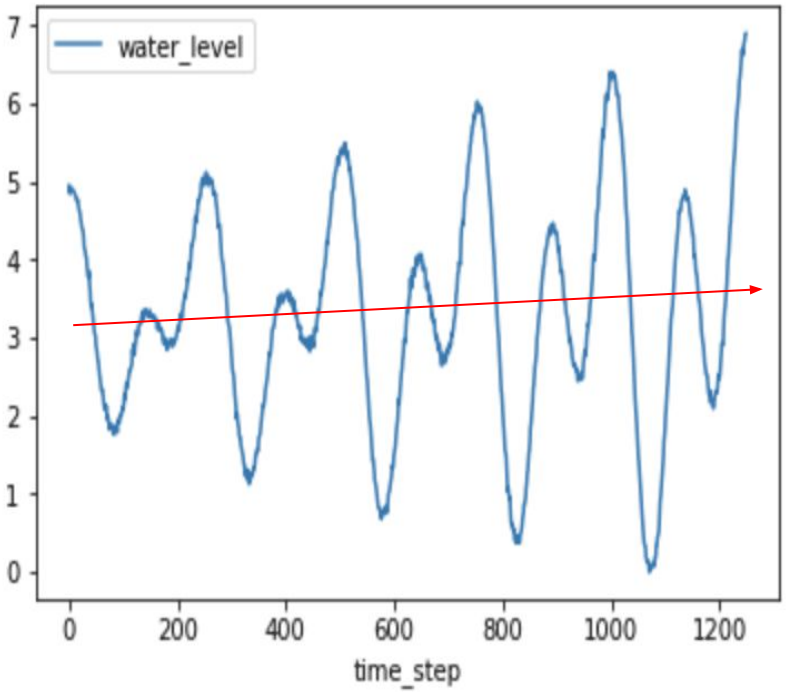
Seasonality



Time Series - Composition

Trend

Movement of a series (high and low values) over a long period of time



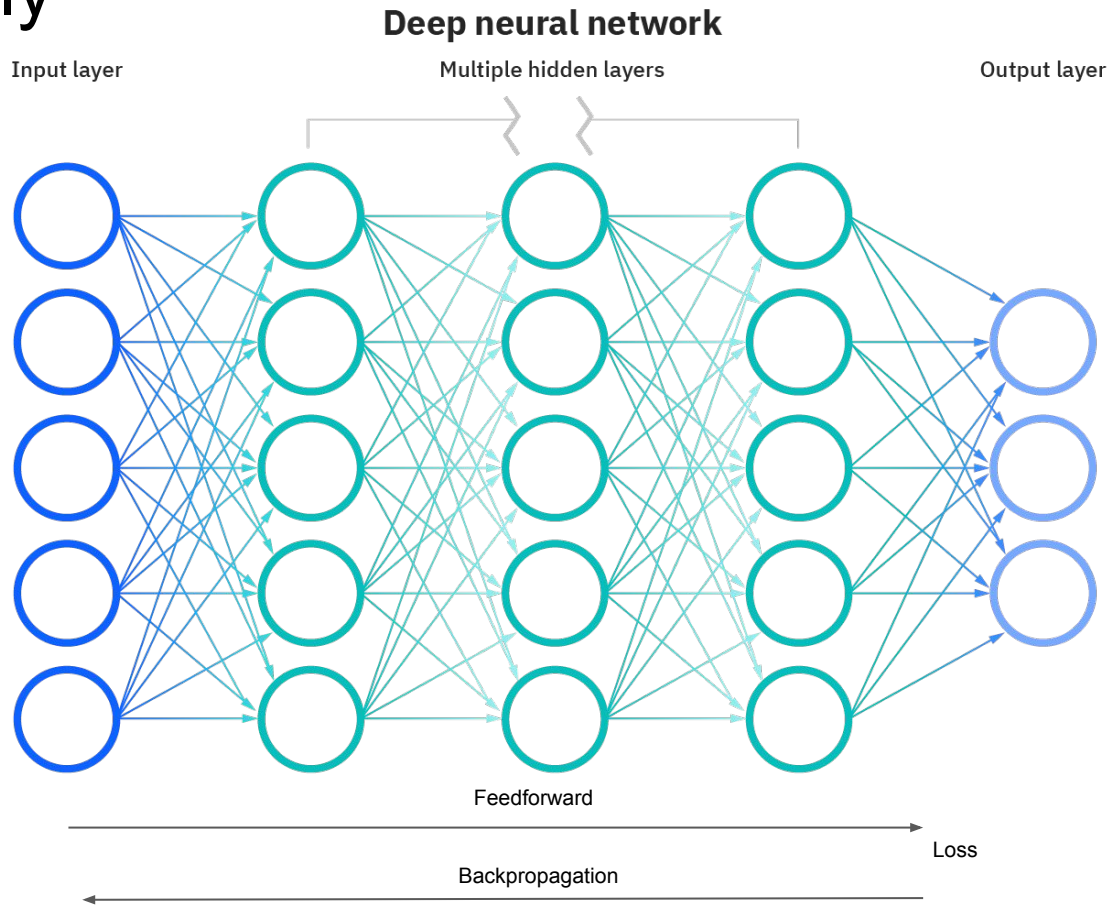
Time Series

Stationarity

Constant mean
Constant variance
Covariance independent of time

Source: https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

RNN Family


Deep neural network
Input layer
Multiple hidden layers
Output layer
Feedforward
Loss
Backpropagation

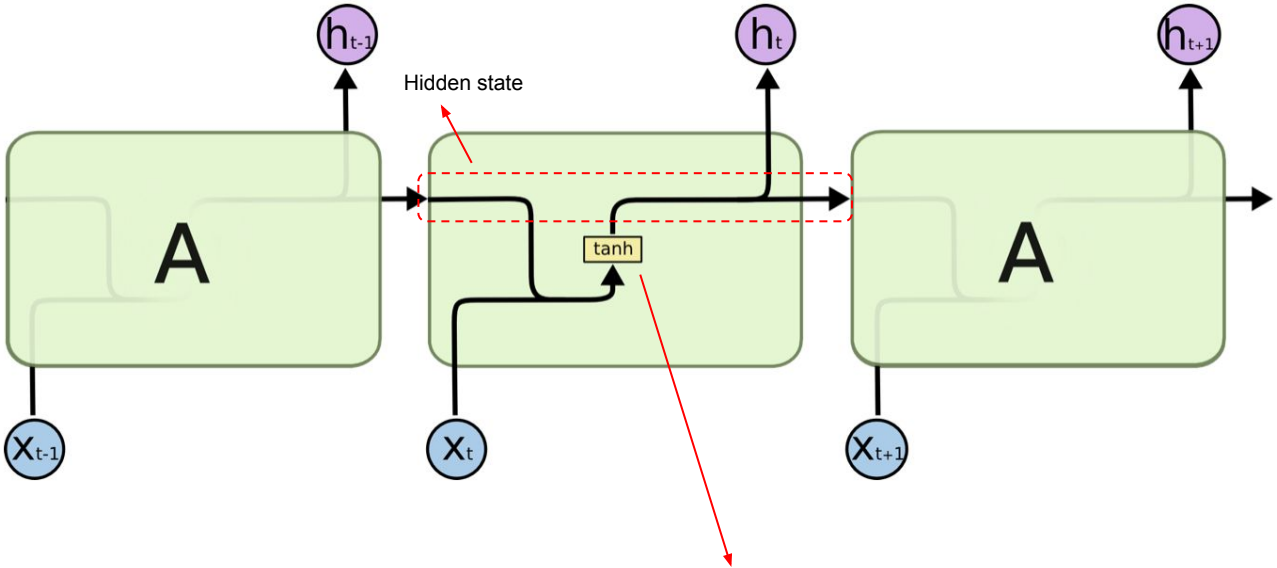
RNN Family

Sequence model
- text
- video
- sound

Persist information
- Memory
- Hidden state

RNN Family

Simple RNN



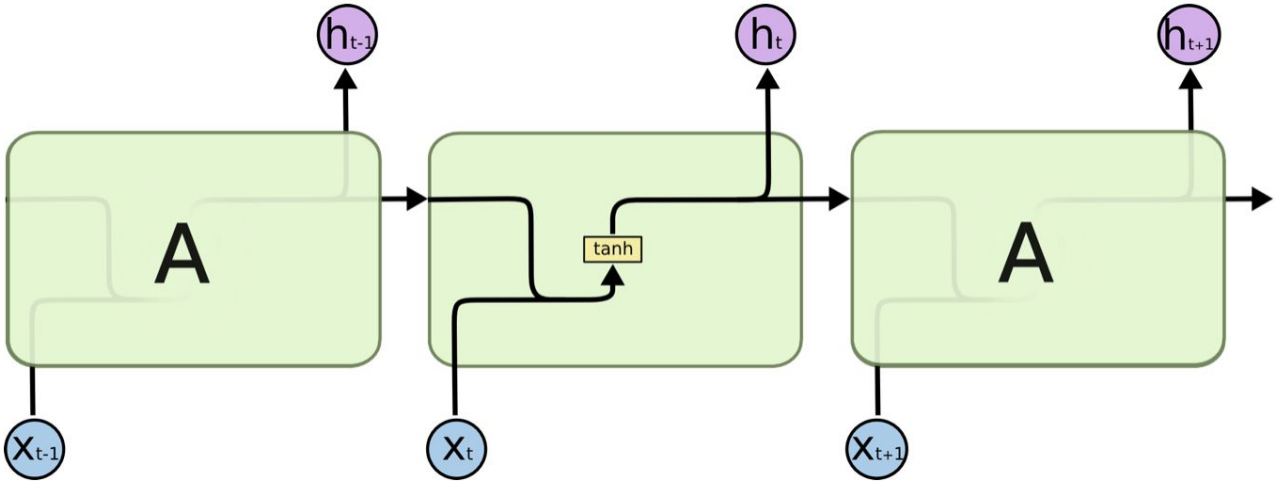
Advantage:

- Short-term memory

Source: colah's blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN Family

Simple RNN



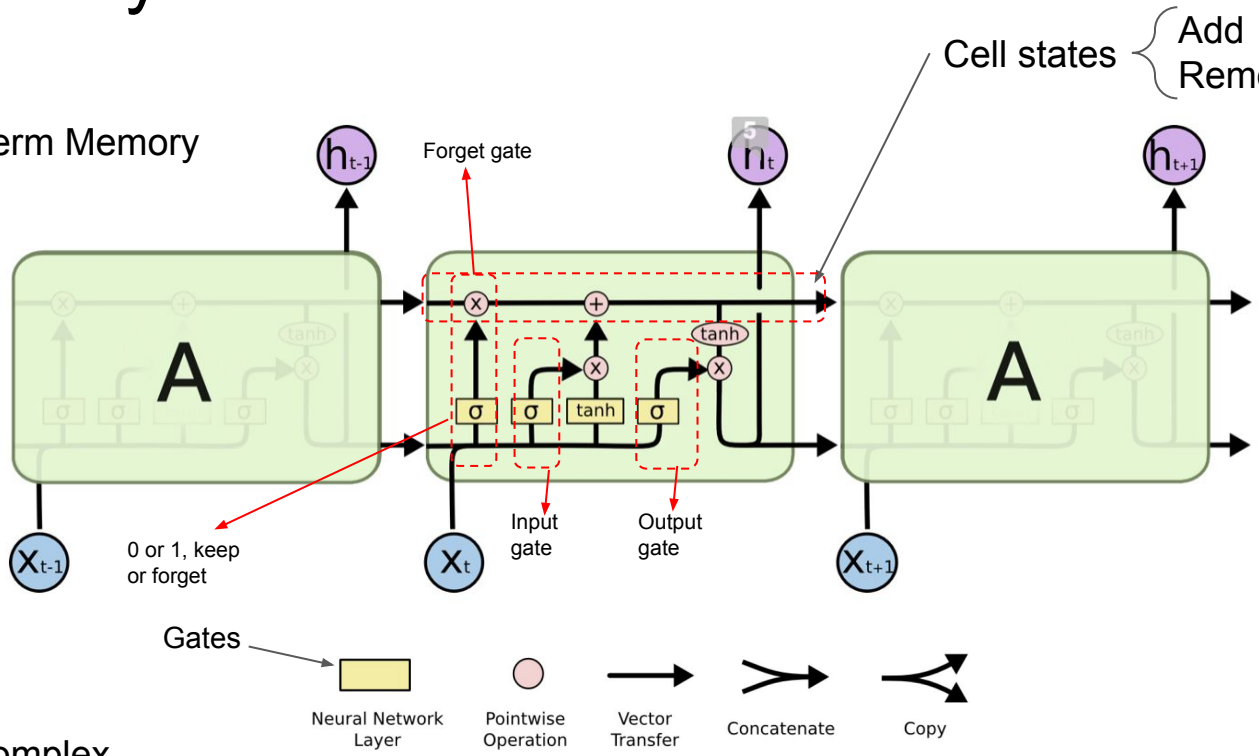
Problems:

- Vanishing gradient
- Exploding gradient
- Long memory (only remember the previous output)

Source: colah's blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN Family

Long Short-Term Memory (LSTM)

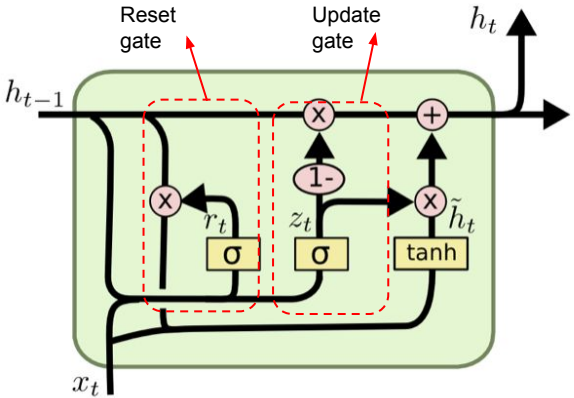


A little bit more complex...

Source: colah's blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/ ← Excellent resource to know how LSTM works

RNN Family

Gated Recurrent Unit (GRU)



Combines the forget and input gates into a single “update gate”

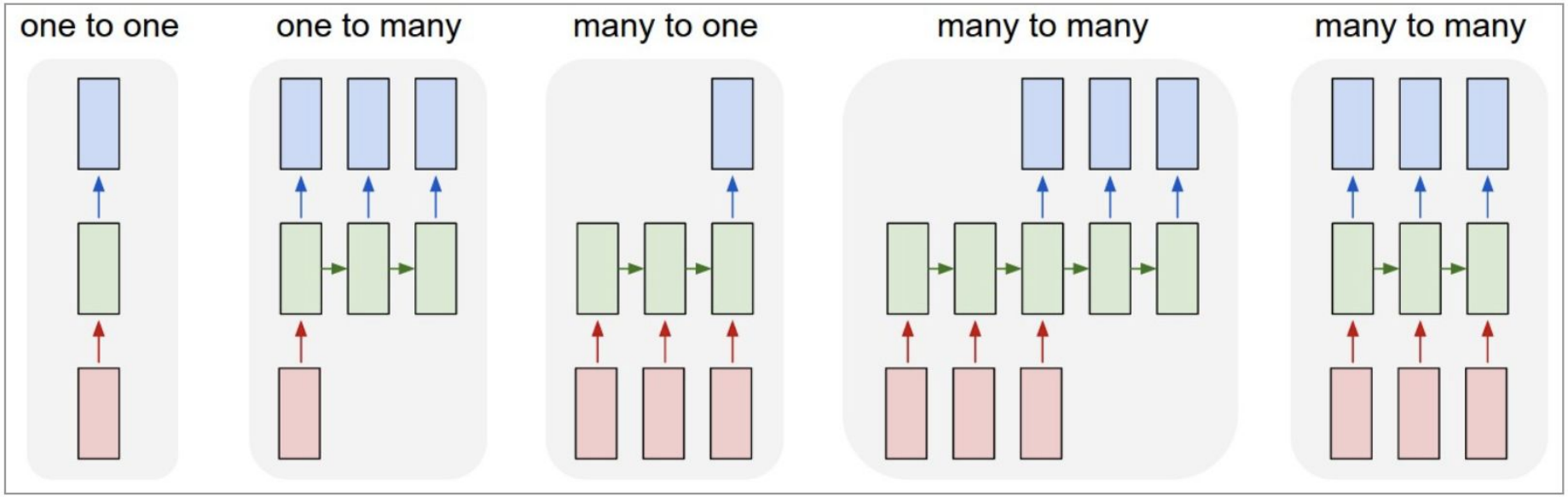
Source: colah’s blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN Family

Applications:

- Prediction problems (ex: forecasting)
- Language modelling
- Text Generation
- Machine Translation
- Speech Recognition
- Image description generation
- Video Tagging
- Text summarization
- Call center Analysis
- Face detection
- OCR
- Image Recognition
- Music Generation
- ...

RNN Family - Architectures



Classical neural network

Music Generation

Sentiment analysis

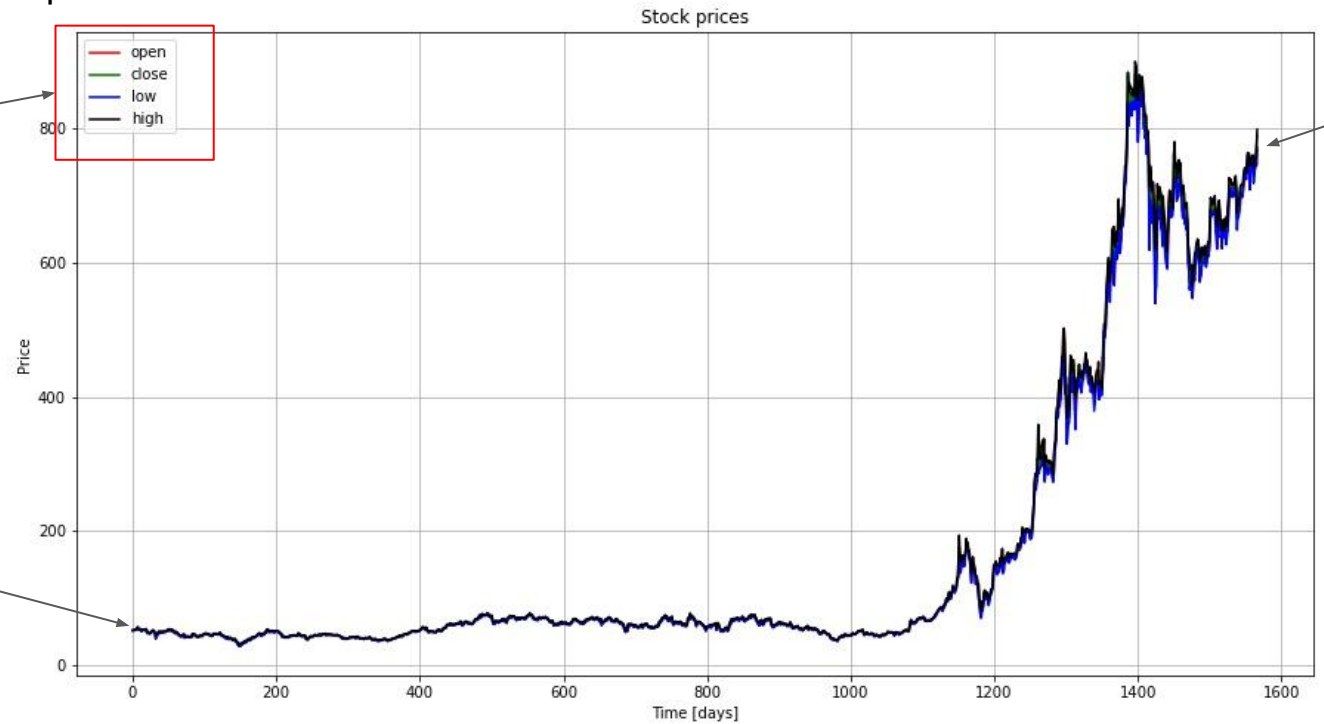
Machine Translation

Name entity recognition

Andrej Karpathy, 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Application

TESLA stock prices



Features
+ Date

2021-09-27

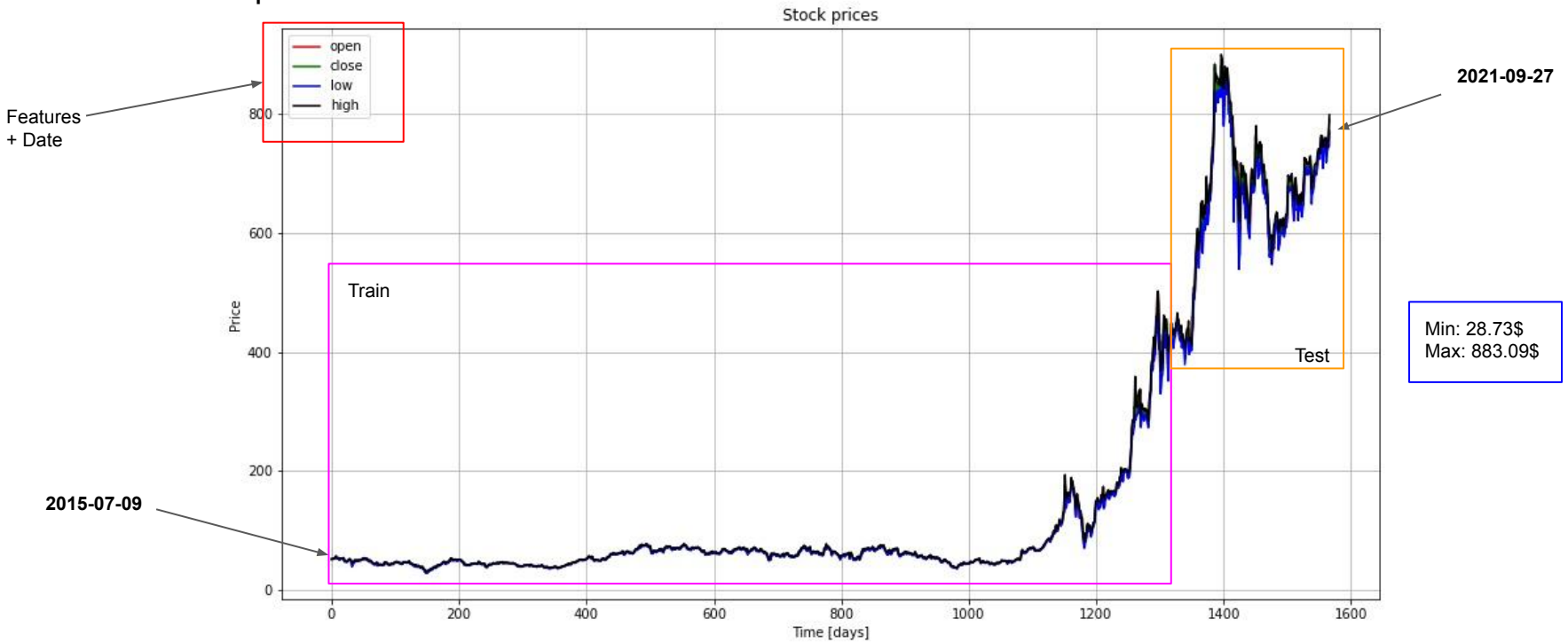
Min: 28.73$
Max: 883.09$

2015-07-09

Source: yahoo finance

Application

TESLA stock prices

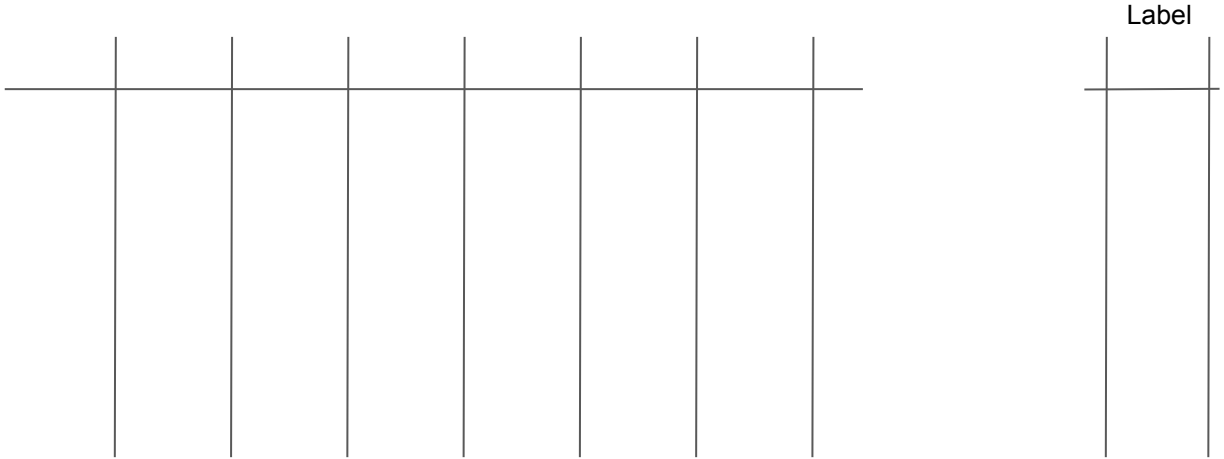


Source: yahoo finance

Application

Forecasting?

How to use the data?

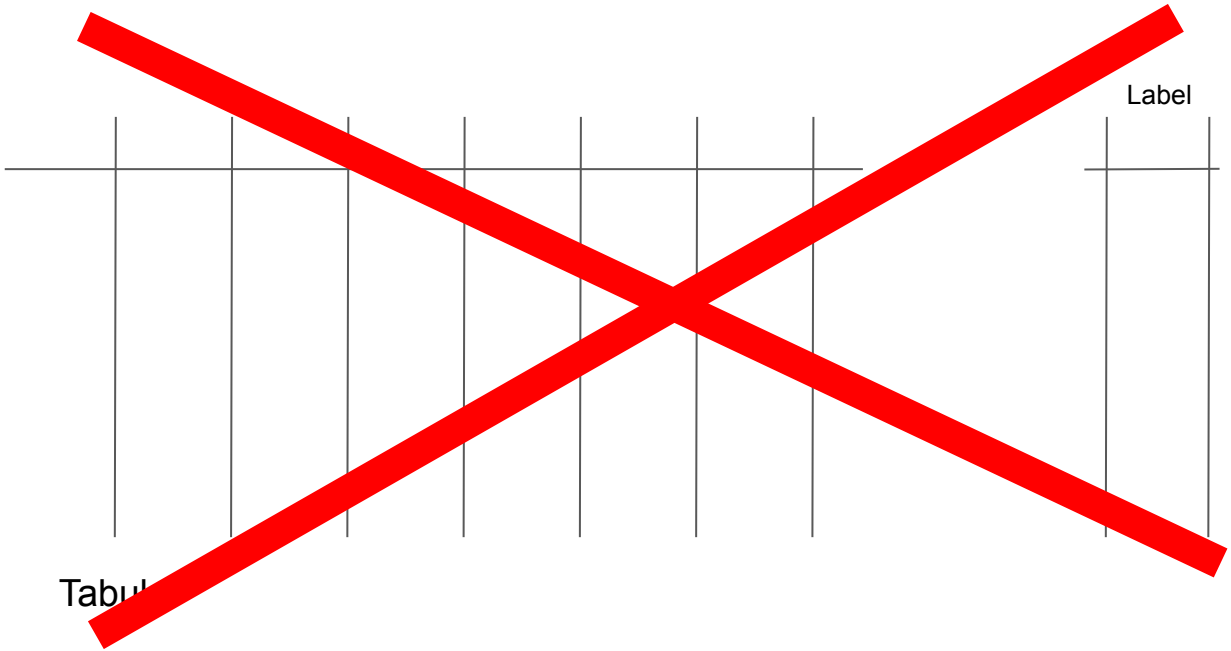


Tabular

Application

Forecasting?

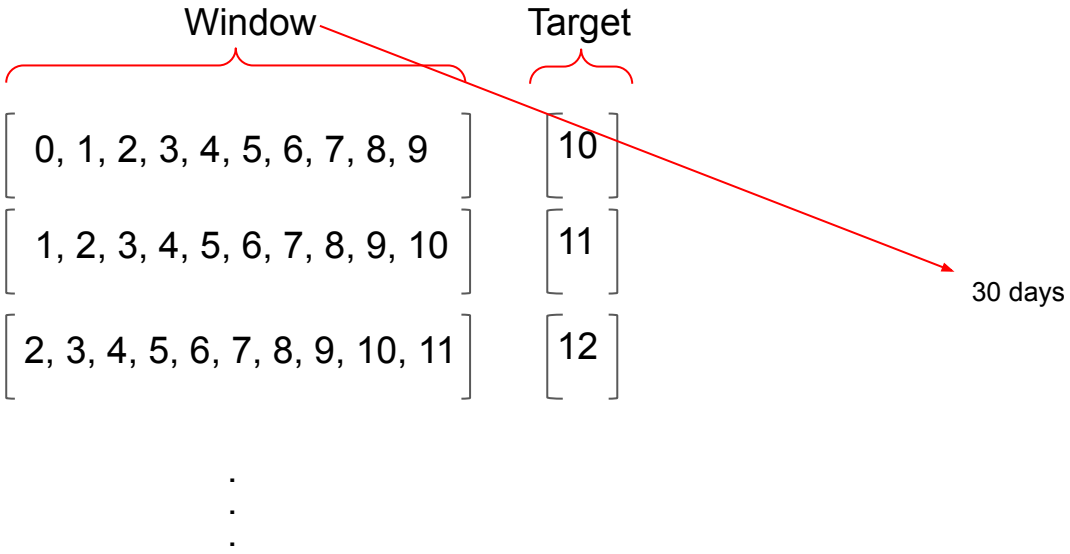
How to use the data?



Application

Forecasting?

How to use the data?



.
.
.

Normalization of data points:

- Subtract the mean computed on the training set and divide by the standard deviation

Or

- Scale the data between 0 and 1

Process to do for:

- Train
- Test

Application

Forecasting?

A little bit of code

```
model_name = "rnn.keras"

inputs = keras.Input(shape=(X_train.shape[1],1))
x = layers.SimpleRNN(250, recurrent_dropout=0.2)(inputs)
outputs = layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

callbacks = [
    keras.callbacks.ModelCheckpoint(model_name,
                                    save_best_only=True)
]
model.compile(optimizer="NAdam", loss="mse", metrics=["mae"])
history = model.fit(X_train, y_train,
                    epochs=50,
                    validation_split=0.2,
                    callbacks=callbacks, verbose = 0)

predicted_stock_price = eval_model(model_name, X_test, y_test)

loss = history.history["mae"]
val_loss = history.history["val_mae"]
epochs = range(1, len(loss) + 1)
plt.figure(figsize=(12,8))
plt.plot(epochs, loss, "bo", label="Training MAE")
plt.plot(epochs, val_loss, "b", label="Validation MAE")
plt.title("Training and validation MAE SimpleRNN + dropout")
plt.grid(True)
plt.legend()
plt.show()
```

Only this part will change

One neuron

RNN model

Save the best model

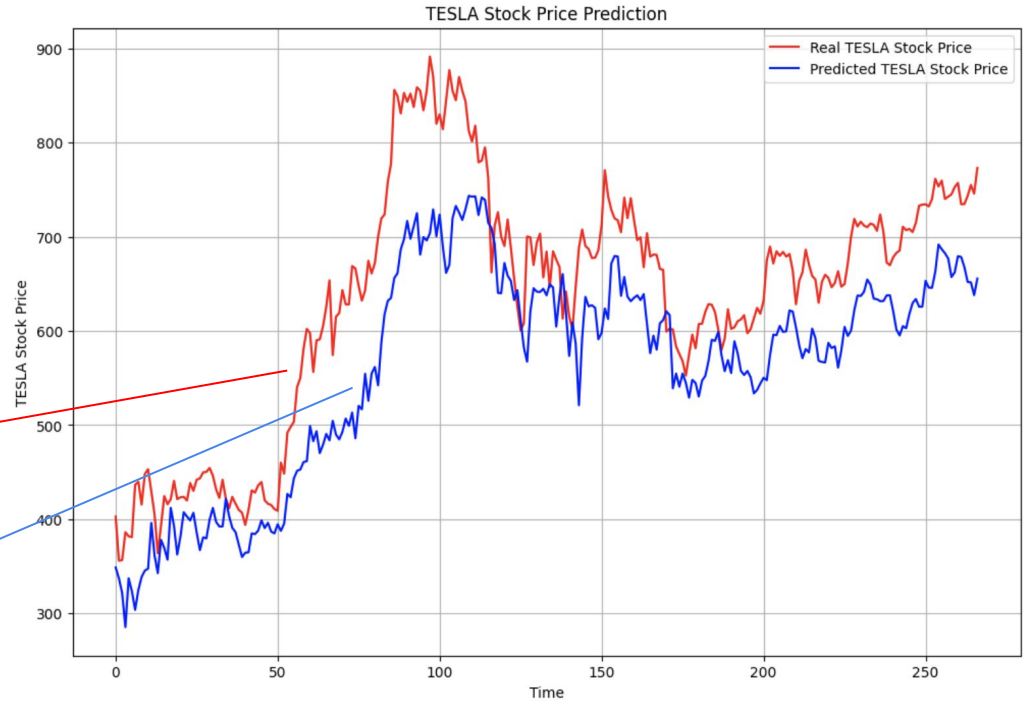
Will compute the MAE

Visualization of the loss for each epoch

Application - Results

- First we need a **baseline**
- Simple neural network (50 neurons)
- Test MAE: 75.24

Computed on the denormalized data



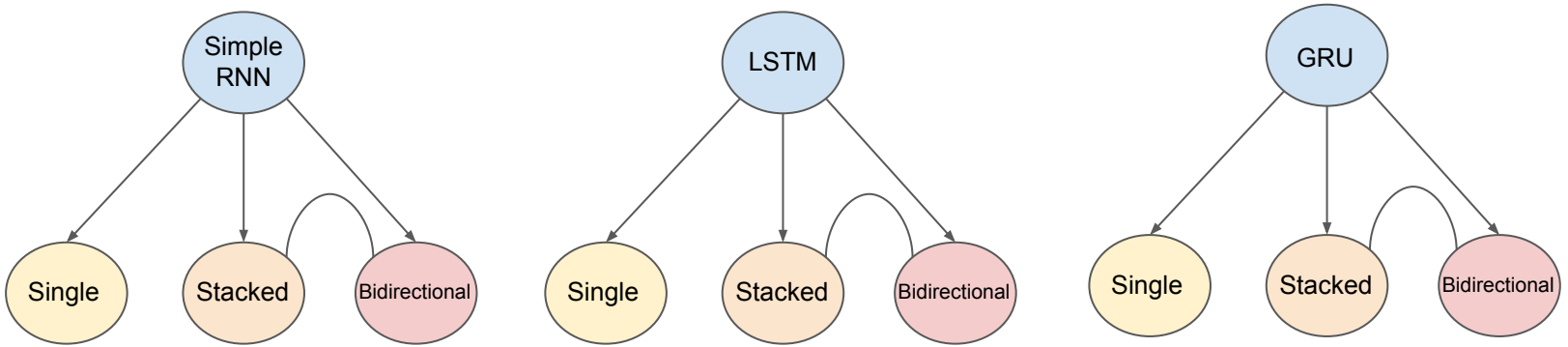
Real close value

Predict close value

2021

Application - Results

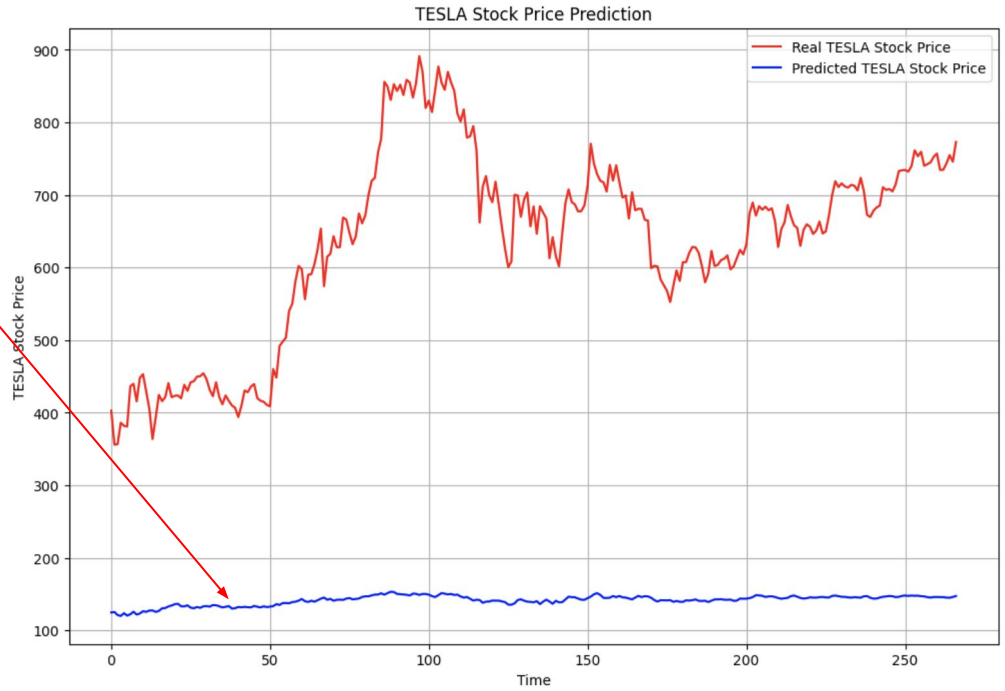
Models tested



Variations with recurrent dropout have been tested

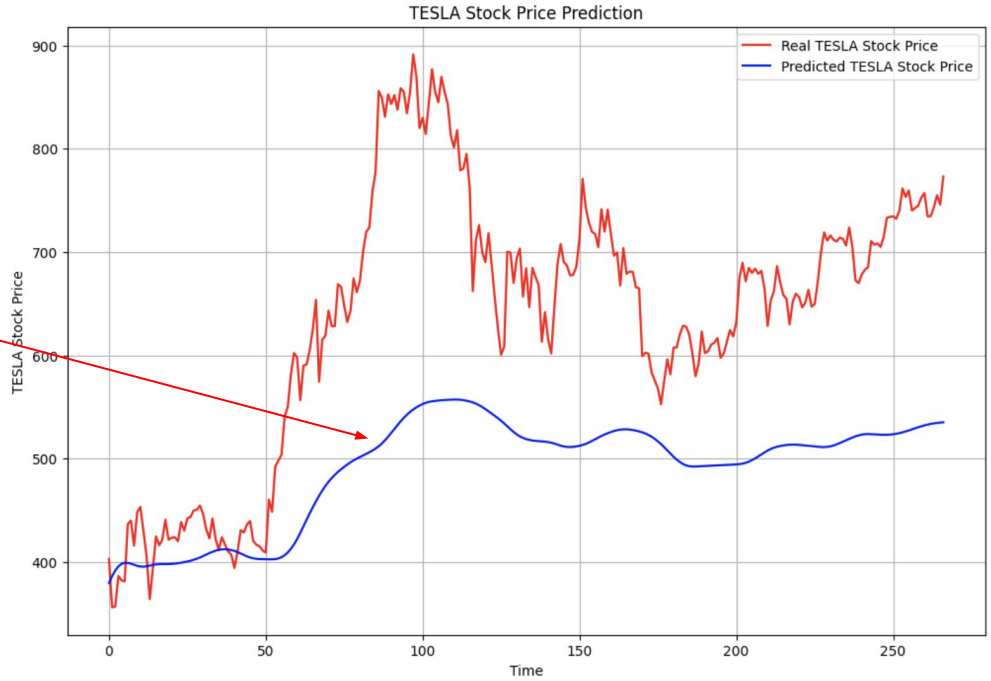
Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24



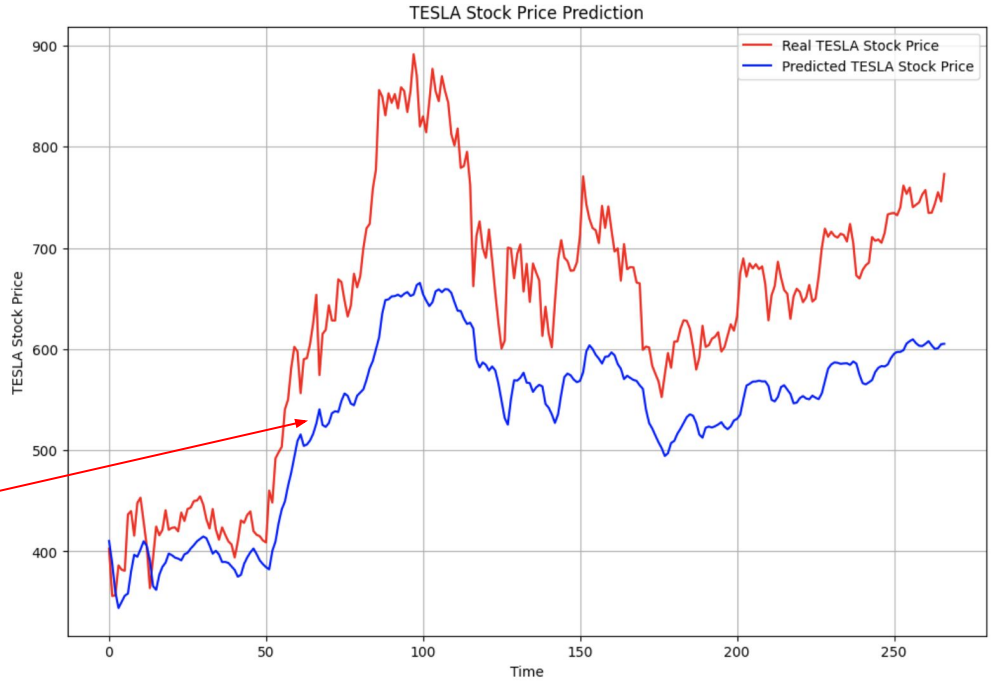
Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24



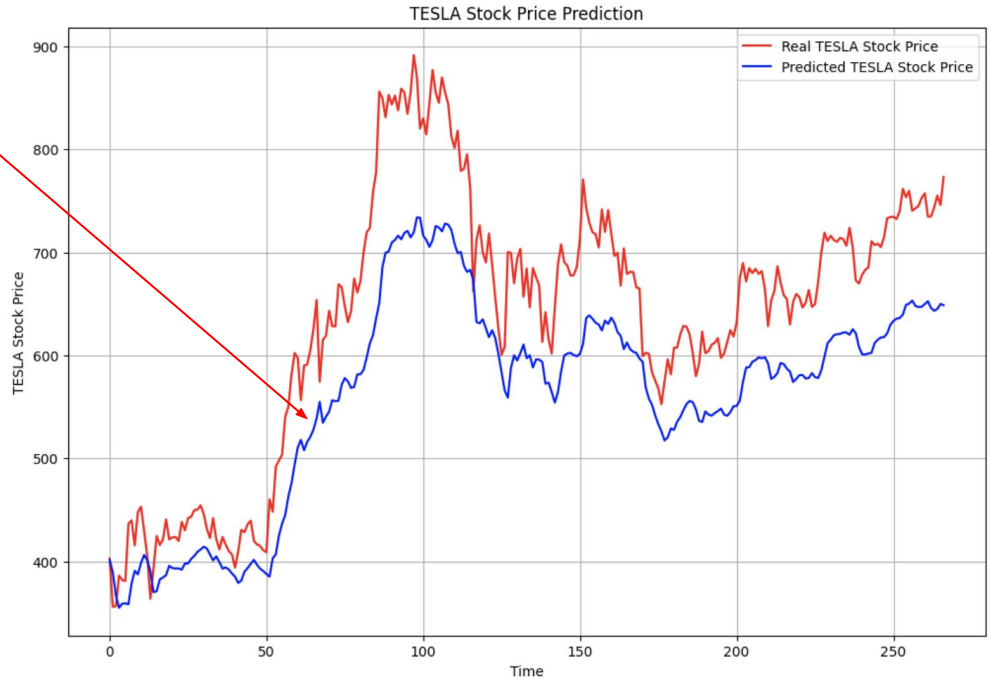
Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24



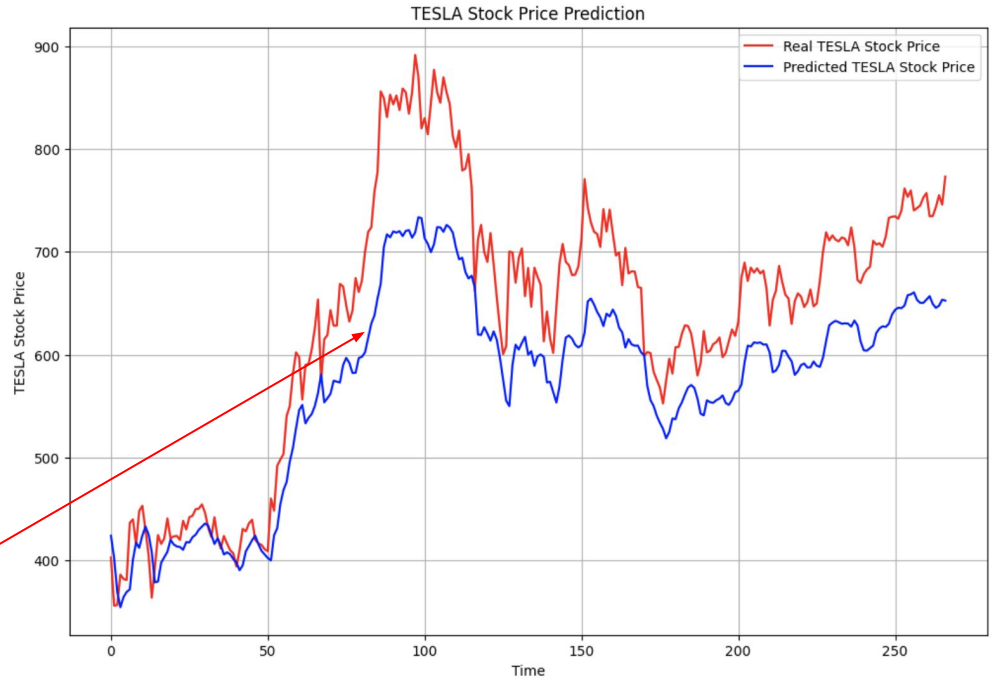
Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24



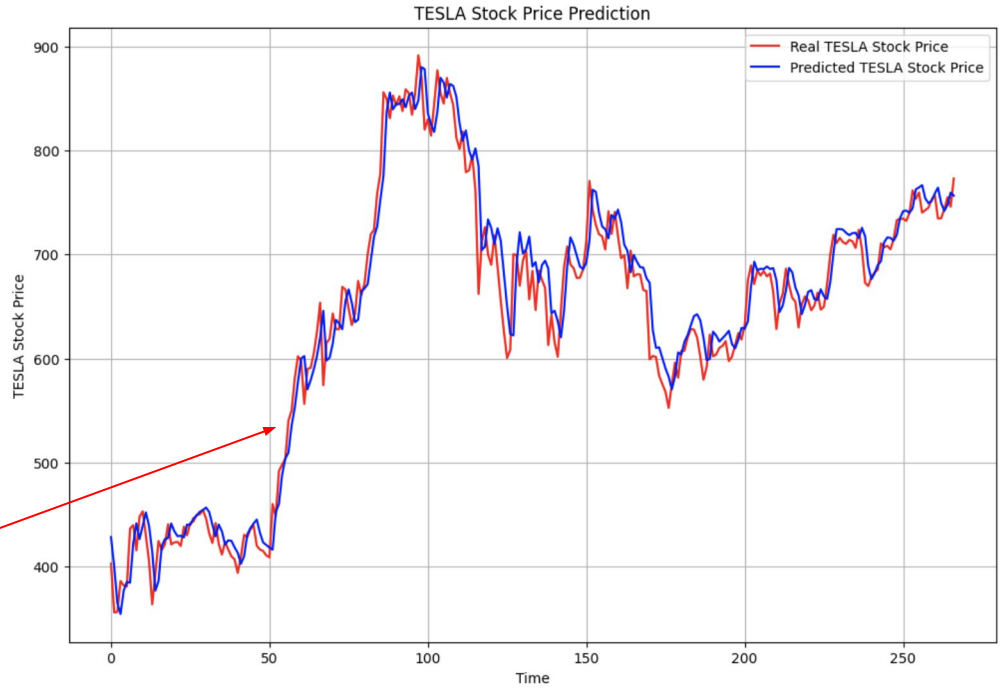
Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24

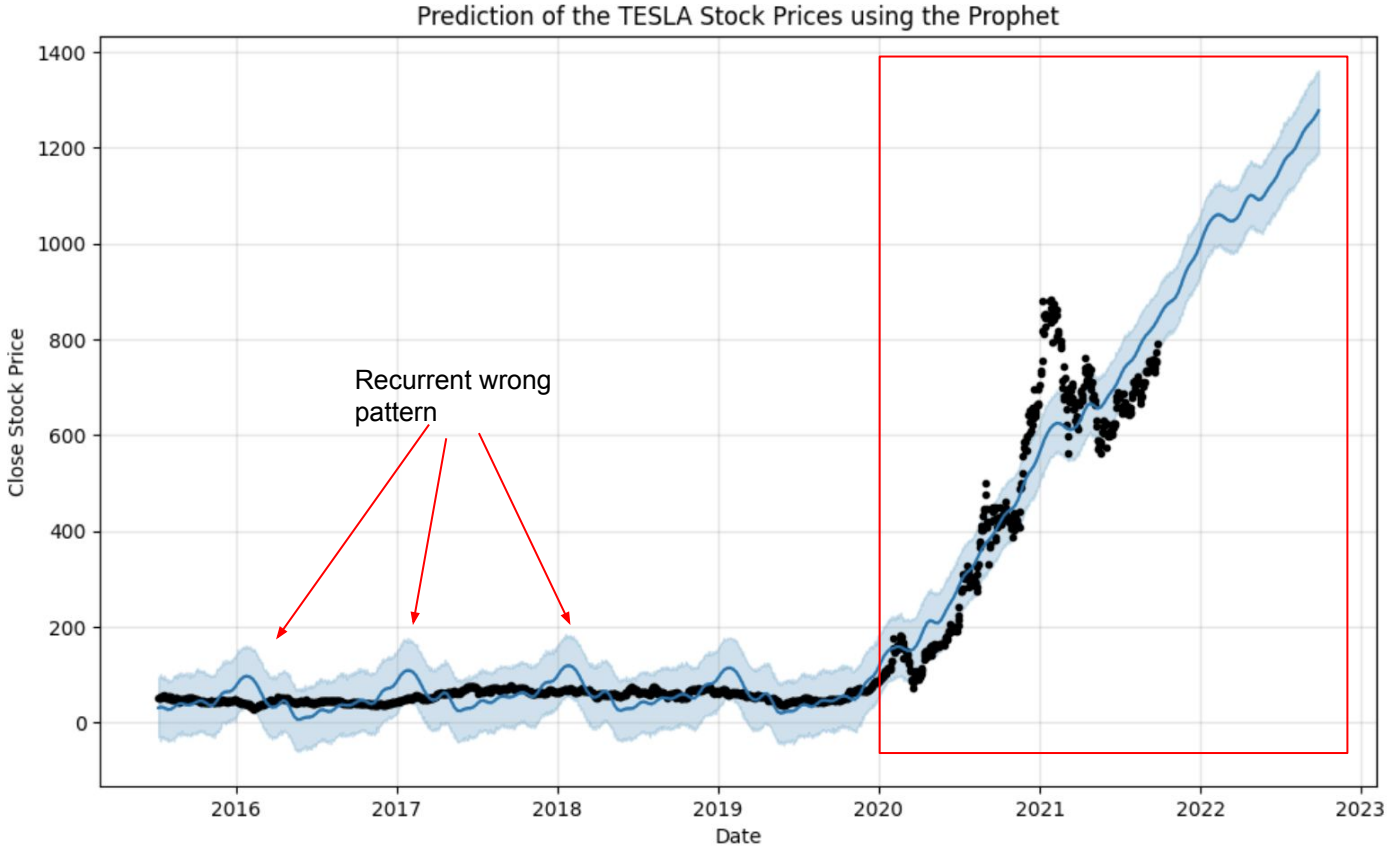


Application - Results

Models	MAE
SimpleRNN	76.46
Stacked SimpleRNN	448.94
Bidirectional SimpleRNN	235.19
Stacked Bidirectional SimpleRNN	494.51
LSTM	22.27
Stacked LSTM	146.99
Bidirectional LSTM	19.20
Stacked Bidirectional LSTM	124.32
GRU	20.70
Stacked GRU	103.21
Bidirectional GRU	**18.94**
Stacked Bidirectional GRU	66.88
Baseline (NN)	75.24



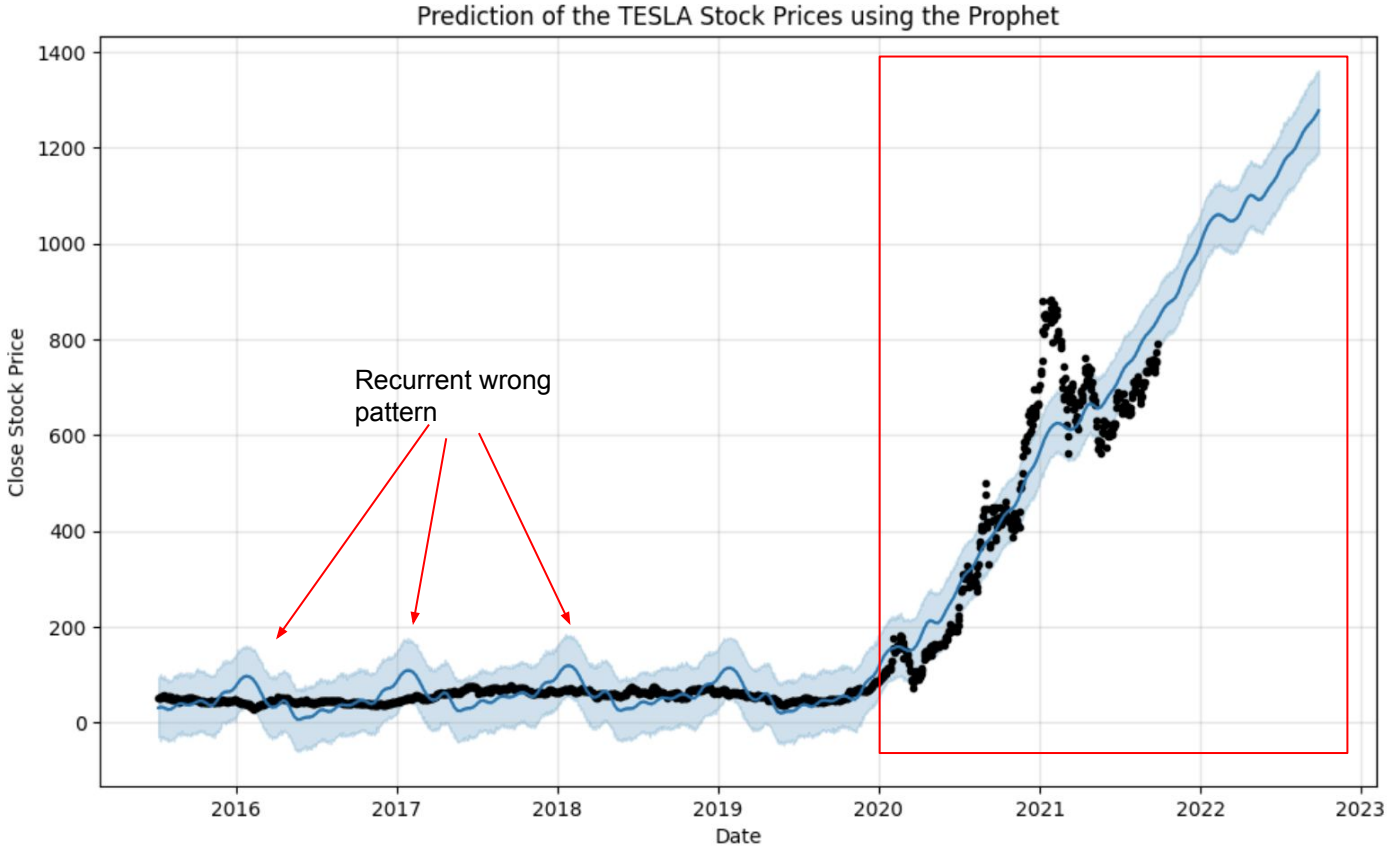
Application - Results



Prediction with Prophet[1] (Facebook)
- additive model
- modular regression model

[1] https://facebook.github.io/prophet/

Application - Results

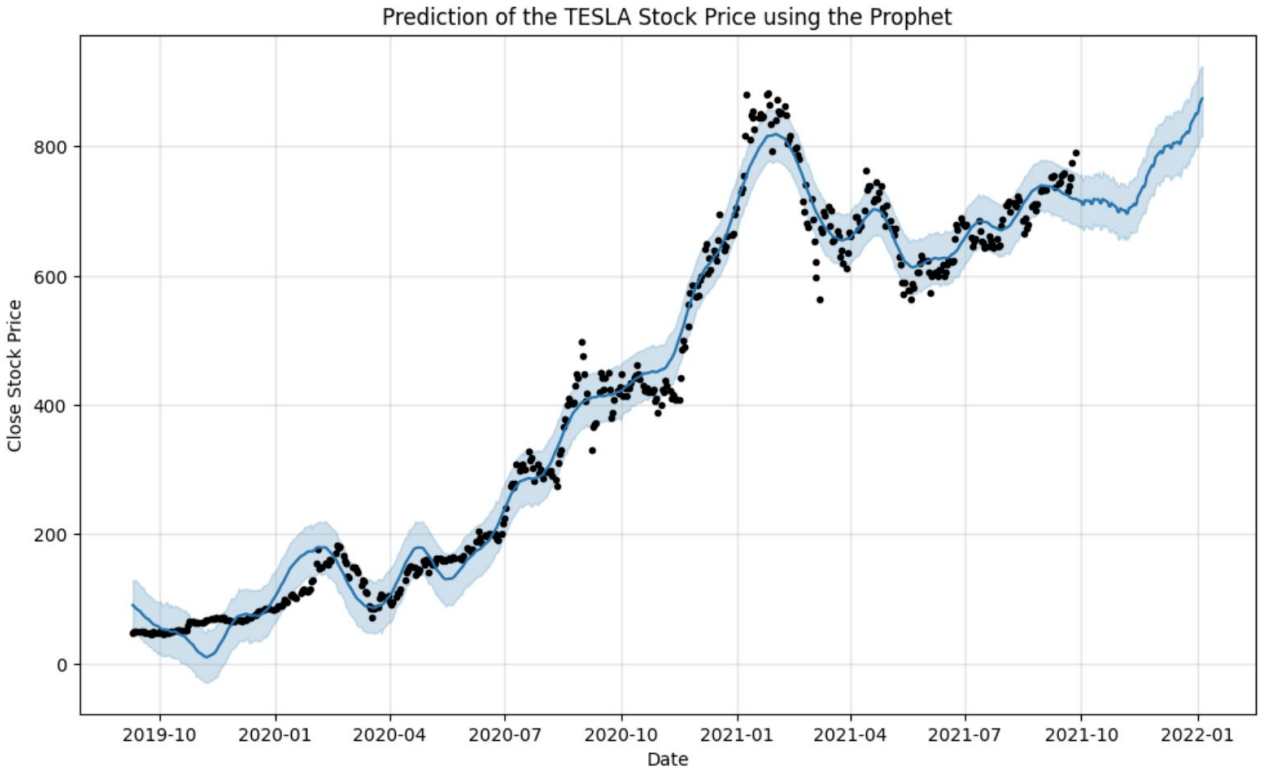


Prediction with Prophet[1] (Facebook)
- additive model
- modular regression model

Could we improve it?

[1] https://facebook.github.io/prophet/

Application - Results



Prediction with Prophet[1] (Facebook)
- additive model
- modular regression model

Reducing the history the model could focus on the extreme part

[1] https://facebook.github.io/prophet/

Discussion/Conclusion

The results show that model **complexity does not improve performance**.
Training a deep learning model is not simple because it requires **hyperparameters optimization**.
The choice of the **window size** has a strong impact on the performance of the models.
The **data history** must be chosen with relevance.
Do not forget the more classical statistical models like ARMA, ARIMA, SARIMA...
It is important to always take a **baseline**.
Memory models (LSTM, GRU) remain the most efficient in single layer or bidirectional (only considering Deep Learning).
Stock prices are **hard to predict**, or **even impossible** if the market fluctuates too sharply.

Conclusion

Marco Peixeiro, 2019. The Complete Guide to Time Series Analysis and Forecasting.
https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

Future reading

Lara-Benitez & Carranza-Garcia & Riquelme, 2021. *An Experimental Review on Deep Learning Architectures for Time Series Forecasting.* ArXiv.
https://arxiv.org/pdf/2103.12057.pdf

→ Comparison between Deep Learning and statistical approaches

Future

- Self-Supervised learning
- Quantum Machine Learning

Github

You can access to the notebook, data and presentation here:

- https://github.com/Christophe-pere/DSDT-timeseries-rnn

Merci / Thank You

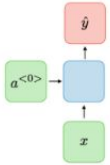
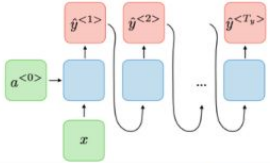
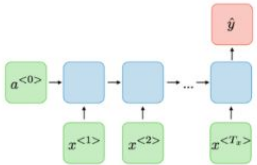
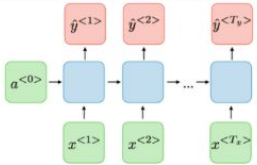
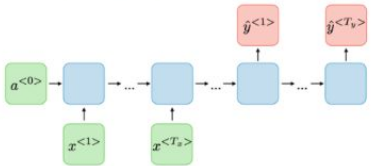
Data Science | Design | Technology

(Check for next DSDT meetup at meetup.com/DSDTmtl)

@DsdtMtl

http://bit.ly/dsdtmtl-in

RNN Family - Architectures

Type of RNN	Illustration	Example
One-to-one $T_x = T_y = 1$		Traditional neural network
One-to-many $T_x = 1, T_y > 1$		Music generation
Many-to-one $T_x > 1, T_y = 1$		Sentiment classification
Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation

CS 230 - Stanford
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks