Apply learning to time series analysis from human expert to automation using data

Presented by Assistant Professor Krung Sinapiromsaran

9 July 2021, 9:00 - 9:30

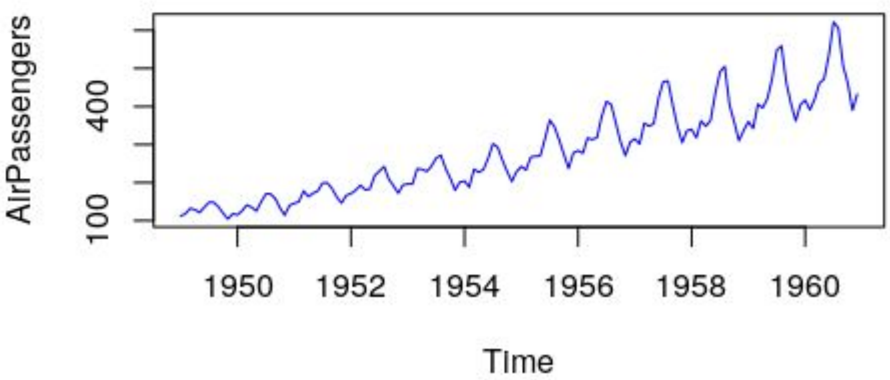
Applied Mathematics and Computational Science, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University

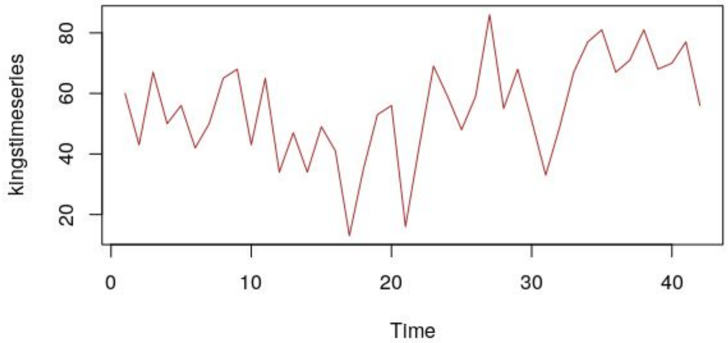
Agenda

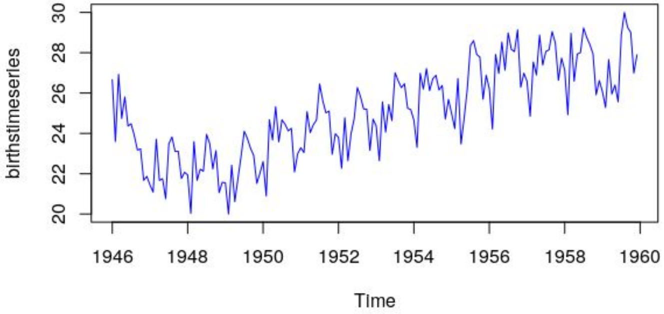
- Motivation
- Time series analysis
- Literature survey
- Gentle introduction to deep learning
- Current methodology
- Future work

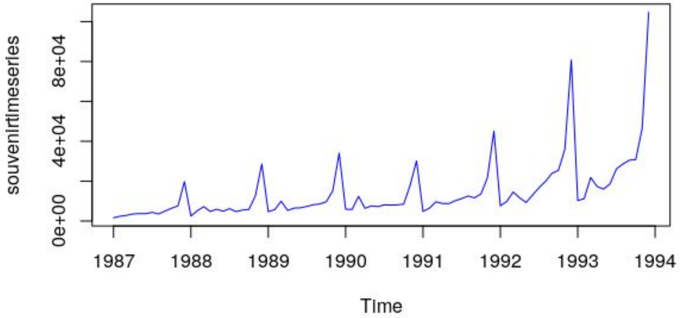
Motivation

Time series data and forecast



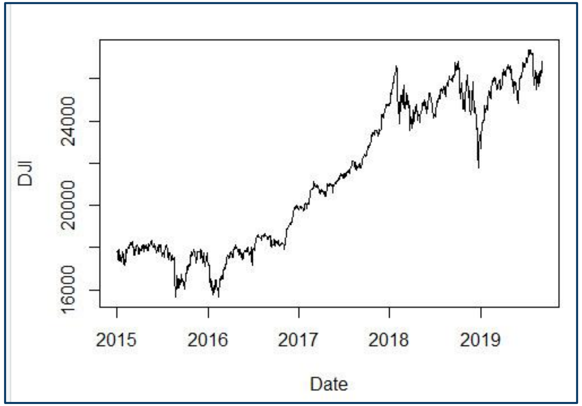




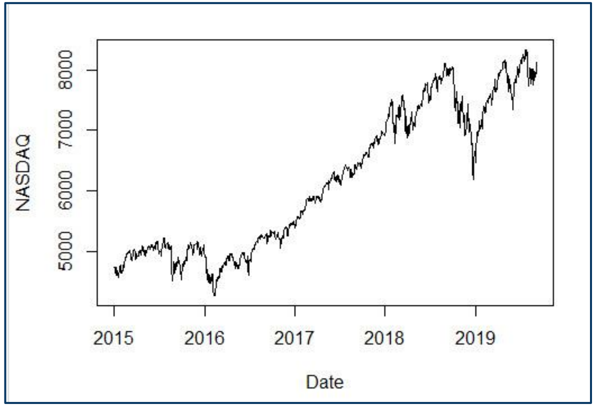


Financial time series data

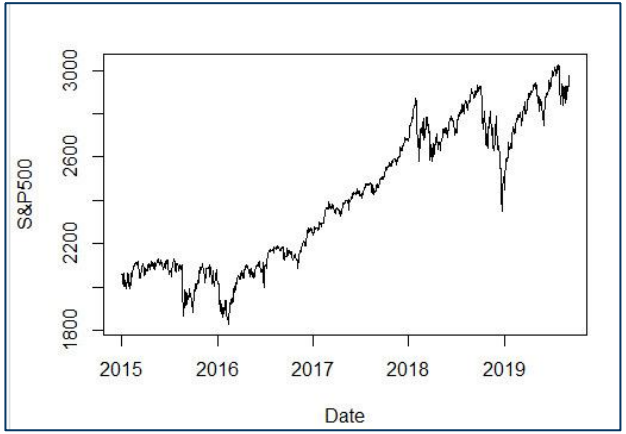
Dow Jones



NASDAQ



S&P500



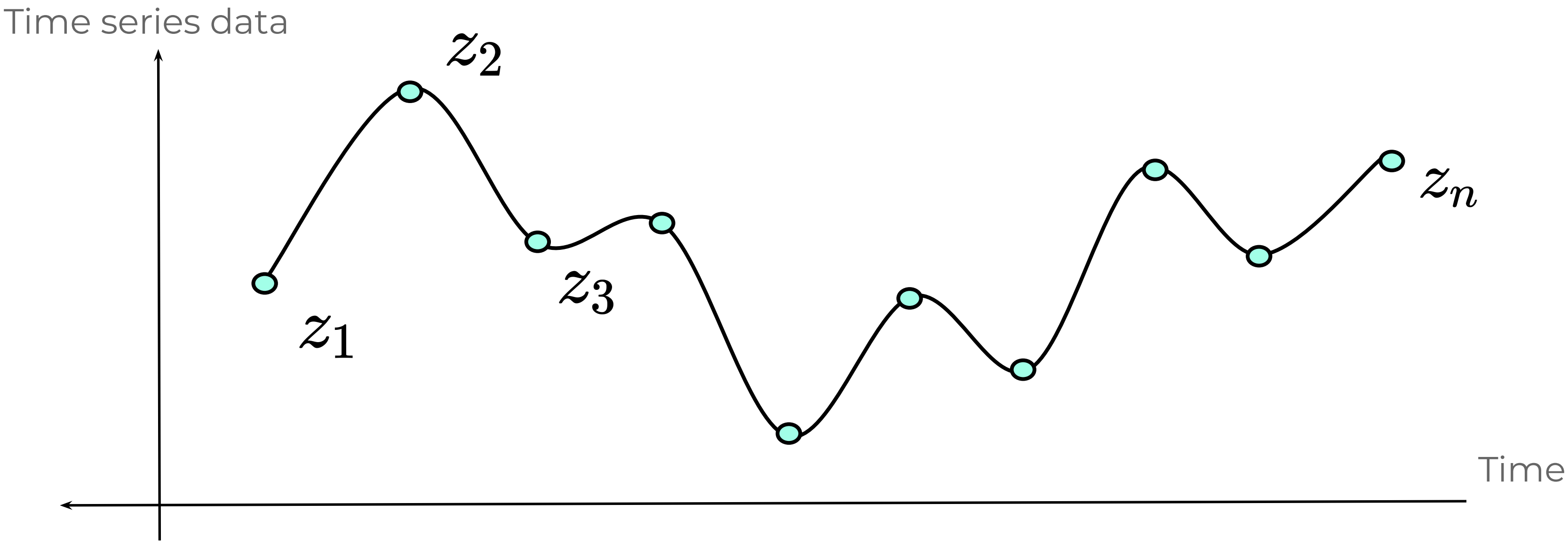
Forecast in organization

Objectives of time series analysis

- Compact description of data
- Interpretation
- Forecasting
- Control
- Hypothesis testing
- Simulation

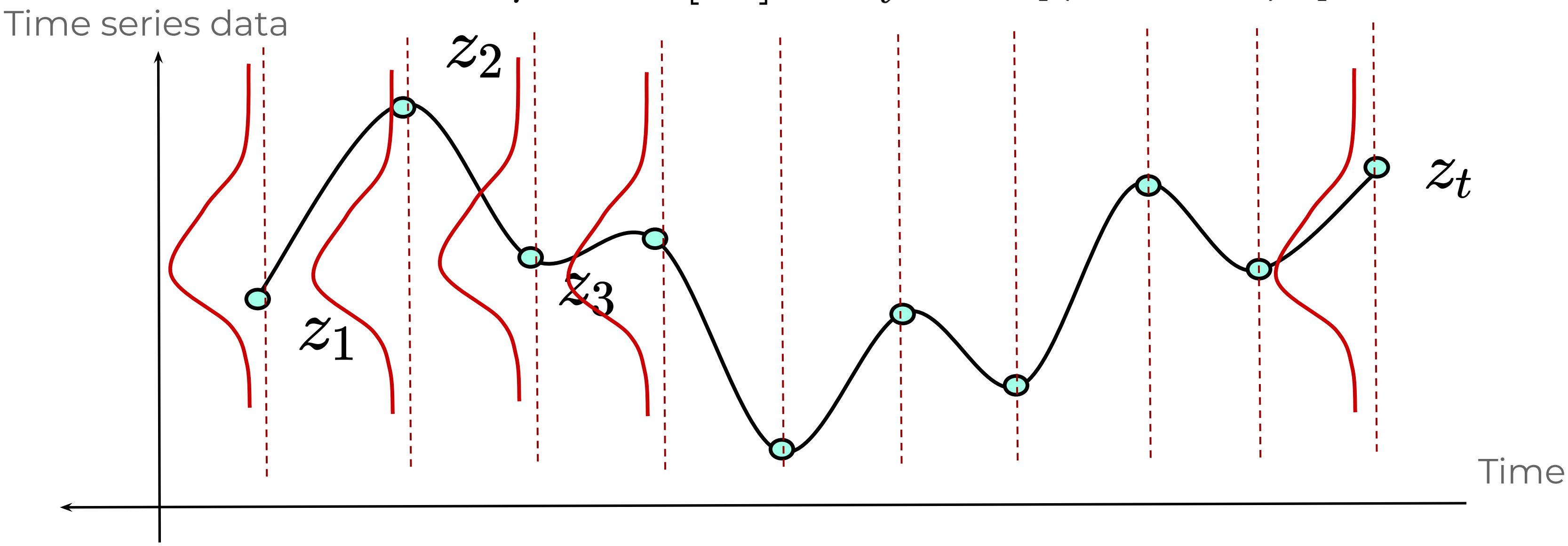
Time series analysis

Time Series Analysis



Stochastic process (Discrete time)

$$\mu_t = E[Z_t] \qquad \sigma_t^2 = E[(Z_t - \mu_t)^2]$$



$$\gamma(t_1, t_2) = E[(Z_{t_1} - \mu_{t_1})(Z_{t_2} - \mu_{t_2})]$$

First-order and second-order stationary in distribution

First-order stationary in distribution

A process is said to be first-order stationary in distribution if its 1-dimensional function is time invariant , i.e., if for any t_1, k

$$F_{Z_{t_1}}(x) = F_{Z_{t_1+k}}(x)$$

Second-order stationary in distribution

A process is said to be second-order stationary in distribution if its 2-dimensional function is time invariant , i.e., if for any t_1, t_2, k

$$F_{Z_{t_1},Z_{t_2}}(x_1, x_2) = F_{Z_{t_1+k},Z_{t_2+k}}(x_1, x_2)$$

Weakly stationary process

Mean function of the process

The process $\{Z_t, t = 0, 1, -1, 2, -2, \ldots\}$ is said to be weakly stationary or covariance-stationary if

- The first moment of Z_t is independent of t
- The second moment of Z_t is finite for all t
- The cross moment (covariance) of Z_{t1} and Z_{t2} depends only on $t1 - t2 = h$.

Autoregressive moving average process

Summary ARMA(p,q), If ϕ and θ are the coefficients of the ARMA model

$$
\begin{aligned}
x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} \\
+ \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}
\end{aligned}
$$

where B is the lag operator where $Bx_t = x_{t-1}$

Autoregressive Integrated Moving Average Process

The ARIMA(p,d,q) model

$$\phi(B)(B-1)^d x_t = \theta(B)\epsilon_t$$
$$\phi(B) \bigtriangledown^d x_t = \theta(B)\epsilon_t$$

where $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ and $\theta(B) = 1 - \theta_1 B - \ldots -\theta_q B^q$ and $d \in \mathbb{Z}^+$

Seasonal autoregressive integrated moving average process or SARIMA $(p,d,q)\times(P,D,Q)$

Suppose that, $\{x_t\}$ contains seasonal component.

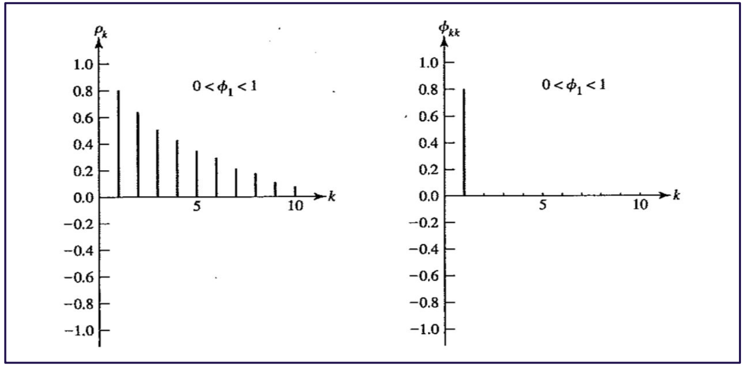
$$\Phi(B^s)\phi(B)(1-B)^d(1-B^s)^D x_t = \Theta(B^s)\theta(B)\epsilon_t$$

where $\mathbf{\Phi}(B) = 1 - \mathbf{\Phi}_1 B - \ldots - \mathbf{\Phi}_P B^{Ps}$ $\quad$ $\mathbf{\Theta}(B) = 1 - \mathbf{\Theta}_1 B - \ldots - \mathbf{\Theta}_Q B^{Qs}$

$\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ $\quad$ $\theta(B) = 1 - \theta_1 B - \ldots -\theta_q B^q$

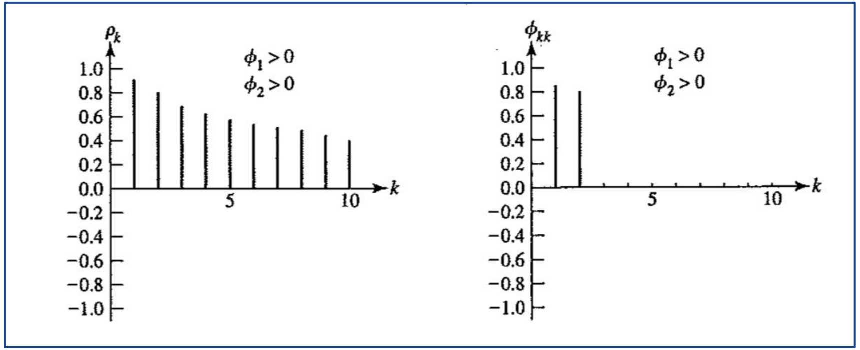
Sample ACF and PACF of AR(1) and AR(2) processes

- ❖ $AR(1)$ process : ACF & PACF



ACF PACF

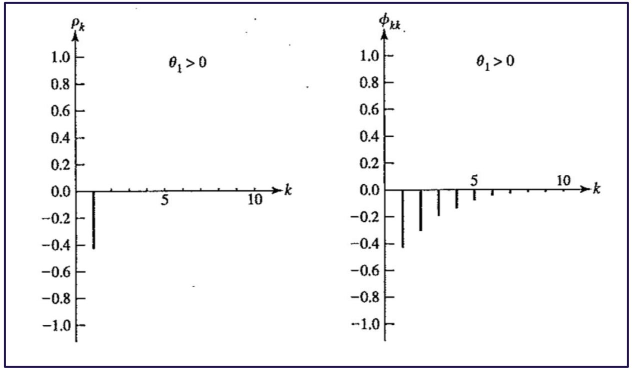
- ❖ $AR(2)$ process : ACF & PACF



ACF PACF

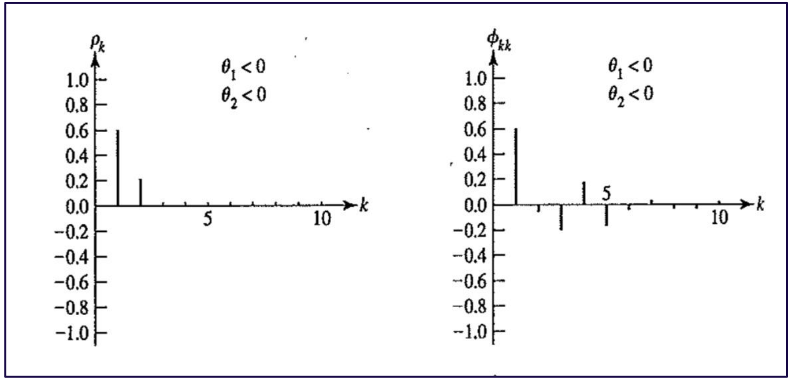
Sample ACF and PACF of MA(1) and MA(2)

❖ MA(1) process : ACF & PACF



ACF PACF

❖ MA(2) process : ACF & PACF



ACF PACF

Identifying the ARMA order via Extended sample autocorrelation function (ESACF)

		lag 1	lag 2	lag 3		
AR/MA	0	1	2	3	...	
0	$r_1^{(0)}$	$r_2^{(0)}$	$r_3^{(0)}$	$r_4^{(0)}$	...	ACF of original time series
1	$r_1^{(1)}$	$r_2^{(1)}$	$r_3^{(1)}$	$r_4^{(1)}$	...	ACF of residual occurring from the time series fitted by $AR(1)$
2	$r_1^{(2)}$	$r_2^{(2)}$	$r_3^{(2)}$	$r_4^{(2)}$	...	ACF of residual occurring from the time series fitted by $AR(2)$
3	$r_1^{(3)}$	$r_2^{(3)}$	$r_3^{(3)}$	$r_4^{(3)}$	...	ACF of residual occurring from the time series fitted by $AR(3)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	

General ESACF table for the ARMA(p,q)

$r_j^{(p)}$ represents the sample ACF of residual which occurs from the time series fitted by $AR(p)$ and j = 1,2,3,..., is the number of the ACF lag.

Identifying the ARMA order via Extended sample autocorrelation function (ESACF)

AR/MA	0	1	2	3	4	5	...
0	X	X	X	X	X	X	...
1	X	O	O	O	O	O	...
2	X	X	O	O	O	O	...
3	X	X	X	O	O	O	...
4	X	X	X	X	O	O	...
5	X	X	X	X	X	O	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

General ESACF table for the ARMA(1,1)

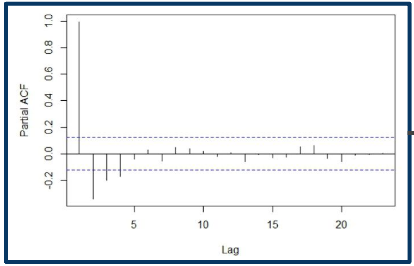
$$r_j^{(m)} = \begin{cases} \text{“X” ; } r_j^{(m)} \text{ within Mean} \pm 2 \times \text{S.D.} \\ \text{“O” ; otherwise} \end{cases}$$

Model Identification

- ❖ Box-Jenkins methodology
  - ➔ Box-Jenkins model identification (*p,d,q*)x(*P,D,Q*)

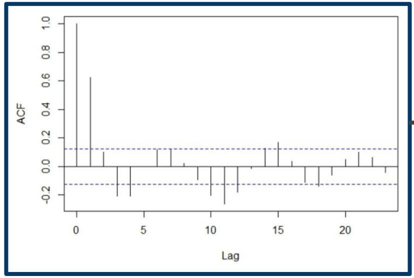
- ❖ Auto-arima method
  - ➔ The Akaike information criterion (AIC)

Sample PACF



Identify AR order

Sample ACF



Identify MA order

ESACF

AR/MA	1	2	3	4	5	6	...
0	X	X	X	X	X	X	...
1	X	O	O	O	O	O	...
2	X	X	O	O	O	O	...
3	X	X	X	O	O	O	...
4	X	X	X	X	O	O	...
5	X	X	X	X	X	O	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Identify ARMA order

Automatic ARIMA method

- ❖ Box-Jenkins methodology
- ➔ Box-Jenkins model identification (*p,d,q*)x(*P,D,Q*)

- ❖ Automatic ARIMA method
- ➔ The Akaike information criterion (AIC)

The Akaike information criterion or AIC is

$$\boxed{AIC = -2\ln(L) + 2(p+q+k)}$$

where p is the autoregressive order.

q is the moving average order.

$k = 1$, if there is an intercept or constant in the ARIMA model and 0 otherwise.

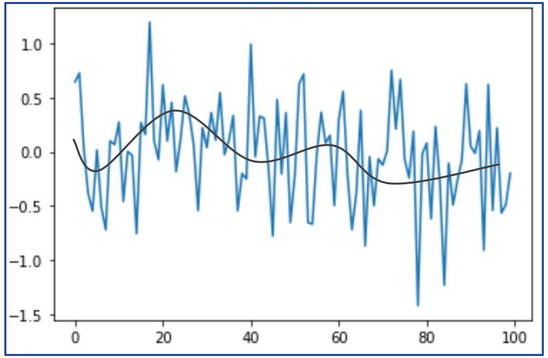
$\ln(L)$ is the maximized log likelihood function.

- ❖ Box-Jenkins methodology
- ➔ Box-Jenkins model identification $(p,d,q)x(P,D,Q)$

- ❖ Automatic ARIMA method
- ➔ The Akaike information criterion (AIC)

The best model is obtained by minimizing the AIC varying the AR order and the MA order from the grid search.
From the example:

		The AR(p) order					
		0	1	2	3	4	5
The MA(q) order	0	3588.66	3588.47	3589.88	3591.62	3592.18	3593.31
	1	3588.61	35.84.68	3586.26	3599.26	3590.17	3592.00
	2	3590.03	3586.26	3588.32	3590.25	3590.73	3594.10
	3	3591.88	3589.08	3583.76	3593.01	3589.64	3591.00
	4	3592.88	3590.16	3592.25	3594.10	3583.88	3586.88
	5	3594.05	3590.79	3594.07	3596.02	3586.78	3587.79

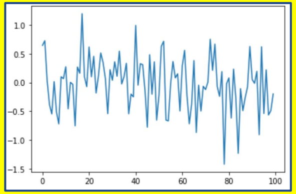


Model Identification

- ❖ Box-Jenkins methodology
- ➔ Box-Jenkins model identification (*p,d,q*)x(*P,D,Q*)

- ❖ Automatic ARIMA method
- ➔ The Akaike information criterion (AIC)

Forecasting time series



Linear process
(Constant variance)

- ❖ ARMA model
- ❖ ARIMA model
- ❖ SARIMA model

Nonlinear process
(Non-constant variance)

- ❖ ARCH model
- ❖ GARCH model

Literature survey

Time series forecasting using a hybrid ARIMA and Neural network model

G. Peter Zhang Neurocomputing Volume 50, January 2003, p. 159-175

The hybrid methodology

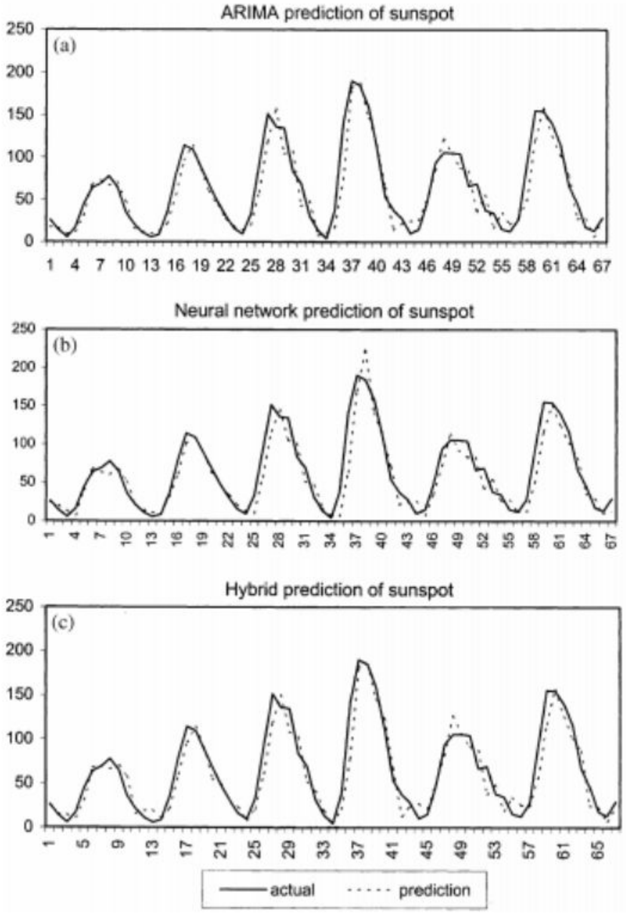
Time series data
$Y_t = L_t + N_t$

Linear structure
via ARIMA

1. Fit the best ARIMA model
2. Fit the residual using ANN

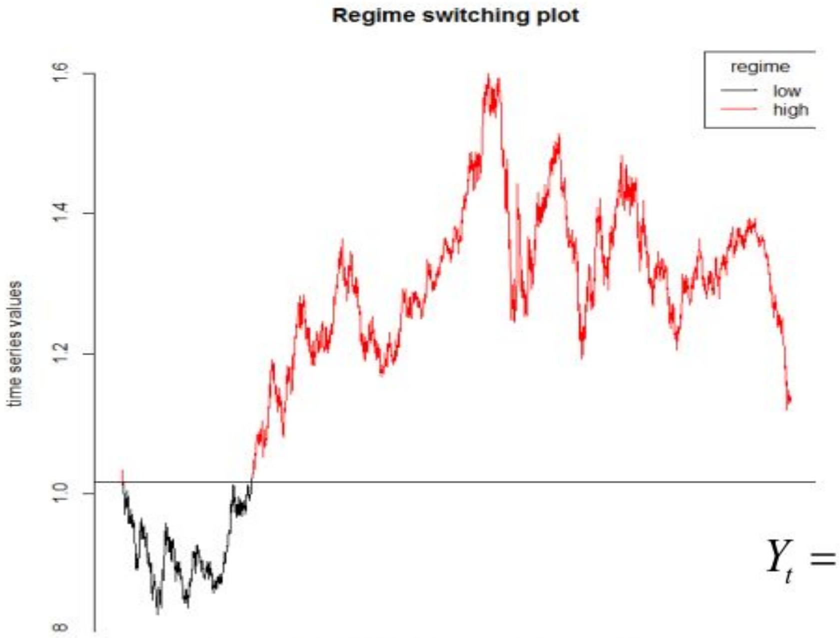
Nonlinear structure
via ANN

$$e_t = f(e_{t-1}, \ldots, e_{t-n}) + \varepsilon_t$$



Emrah Hanifi Firat, Mathematics and Statistics, 5(1): 2017, p. 33-55.

SETAR = Self-excited Threshold AutoRegressive model

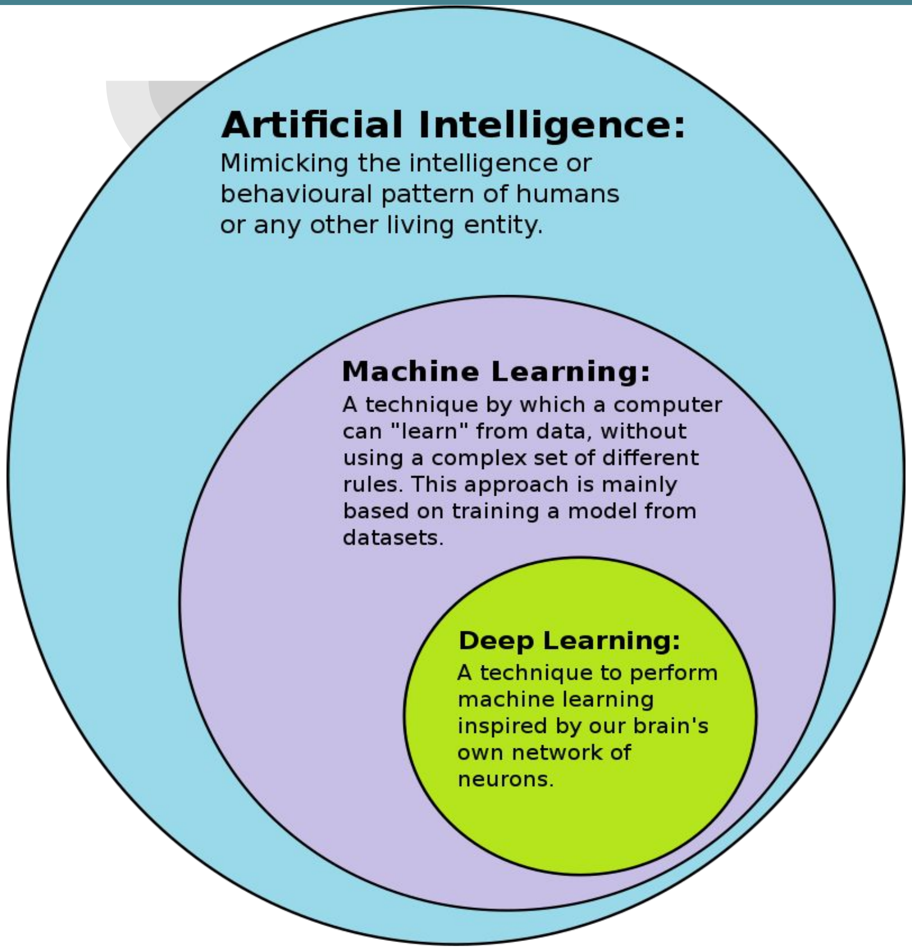


Graphic 7. EUR/USD, D1- Regime change graphics

$$Y_t = \beta_0^{(i)} + \beta_1^{(i)} Y_{t-1} + \ldots.. + \beta_{p_i}^{(i)} Y_{t-p} + \varepsilon_{it}$$

$$\tau_{i-1} < x_{t-d} < \tau_i\,, \quad i = 1, \ldots.., k \qquad (1)$$

Gentle introduction to deep learning



What is AI-ML-DL?

- AI = Artificial Intelligence:Create a computer system to mimic intelligence of humans or living things.
- ML = Machine Learning:Concentrate on technique or programming to "learn" from data by training a model from historical datasets.
- DL = Deep Learning:Specific technique in machine learning which is inspired by human's brain and networks of neurons.

Source: https://en.wikipedia.org/wiki/Deep_learning

What is deep learning and why will it be effective at learning?

Deep Learning = multiple layers from NN to progressively extract higher level features from raw input.

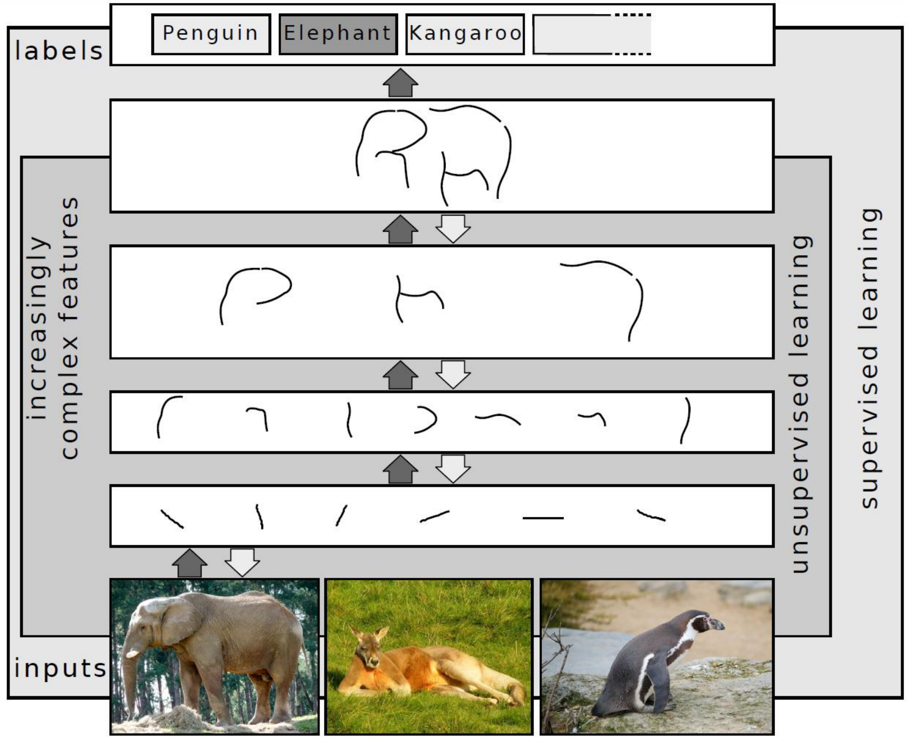


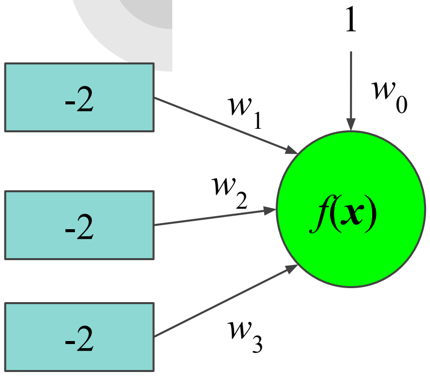
Image source: https://en.wikipedia.org/wiki/Deep_learning

NN (Neural network) had been proposed for at least 25 years before? Why it did not work then.

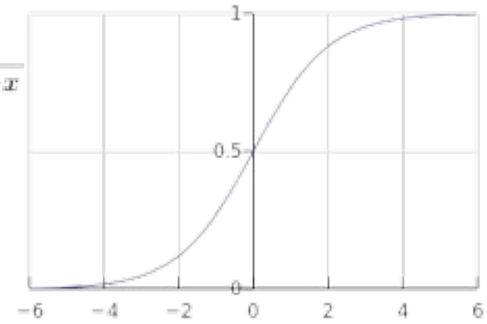
- Learning weights for NN could only be achieve for a single hidden layer.
- No effective method to train very deep neural network until now.

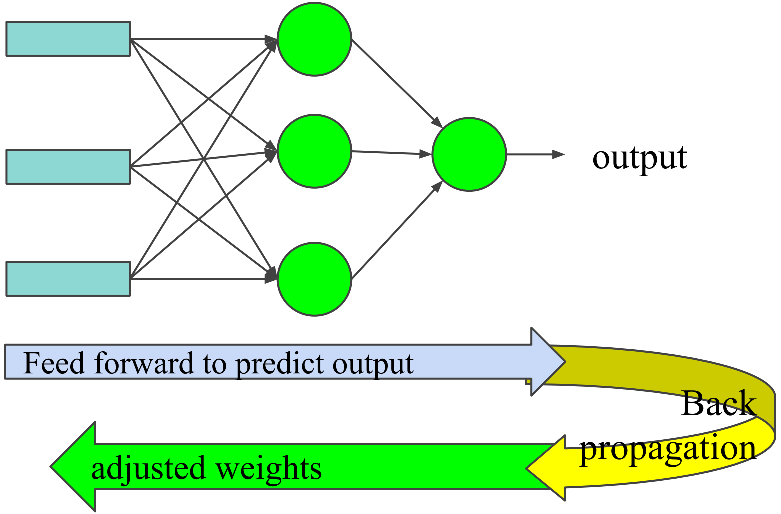
How does deep learning work? (Architecturally)

Basic one neural node



$$f(x) = \frac{1}{1+e^{-x}}$$



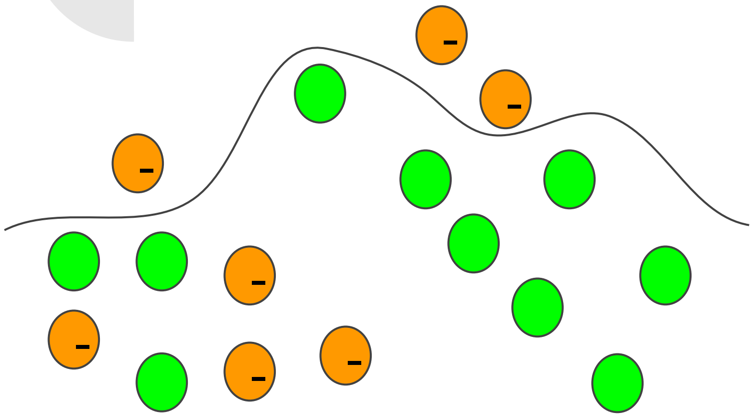


NN = neural network (network of neural nodes)

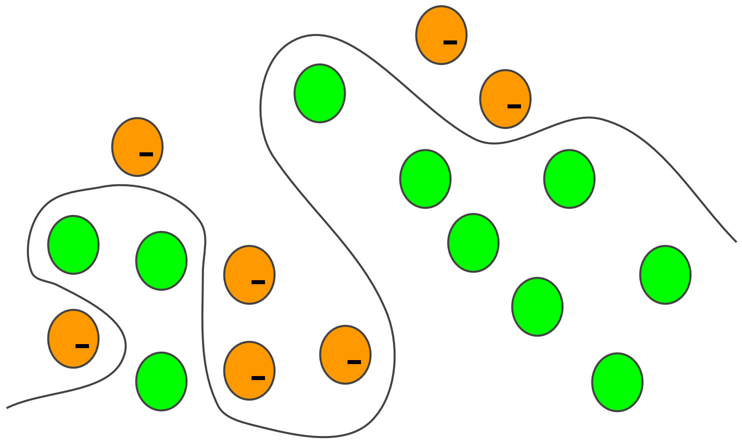
- Input layer:numeric input data for each column
- Hidden layer:layer of hidden nodes
- Output layer:output of NN

How does deep learning work? (Empirically)

Learning decision boundary



Find the best decision boundary to split two classes



Find the best decision boundary to split two classes

Feature detectors

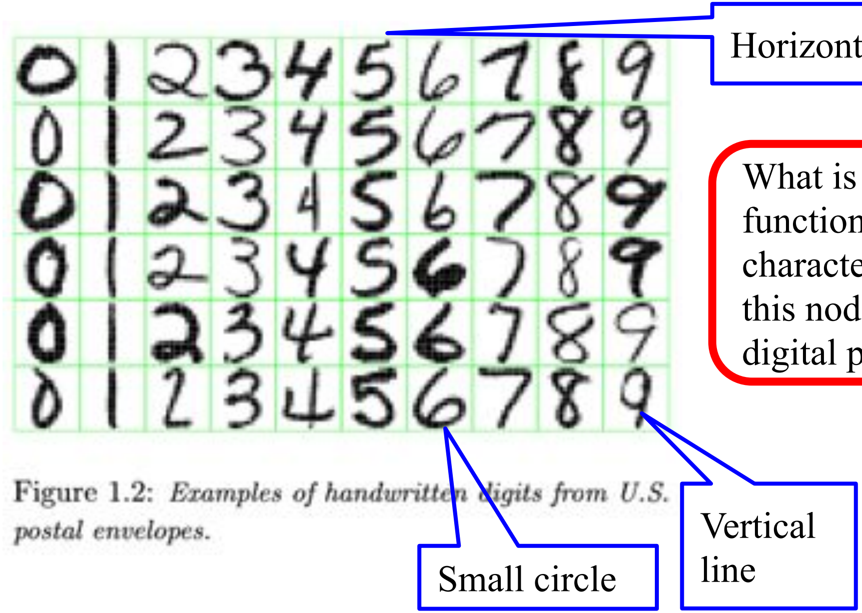
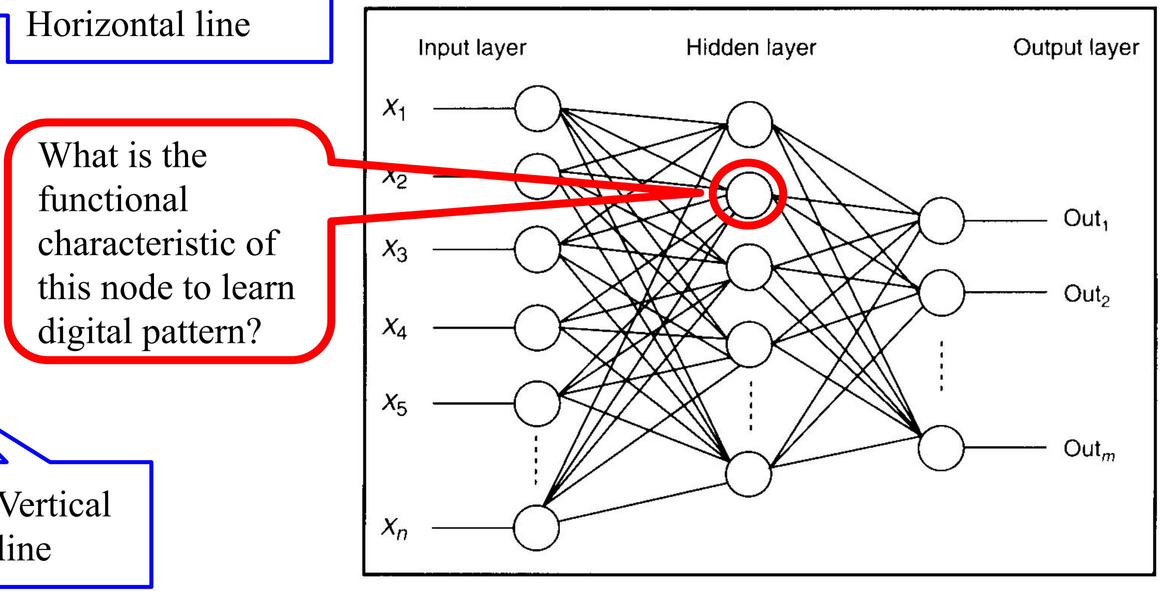


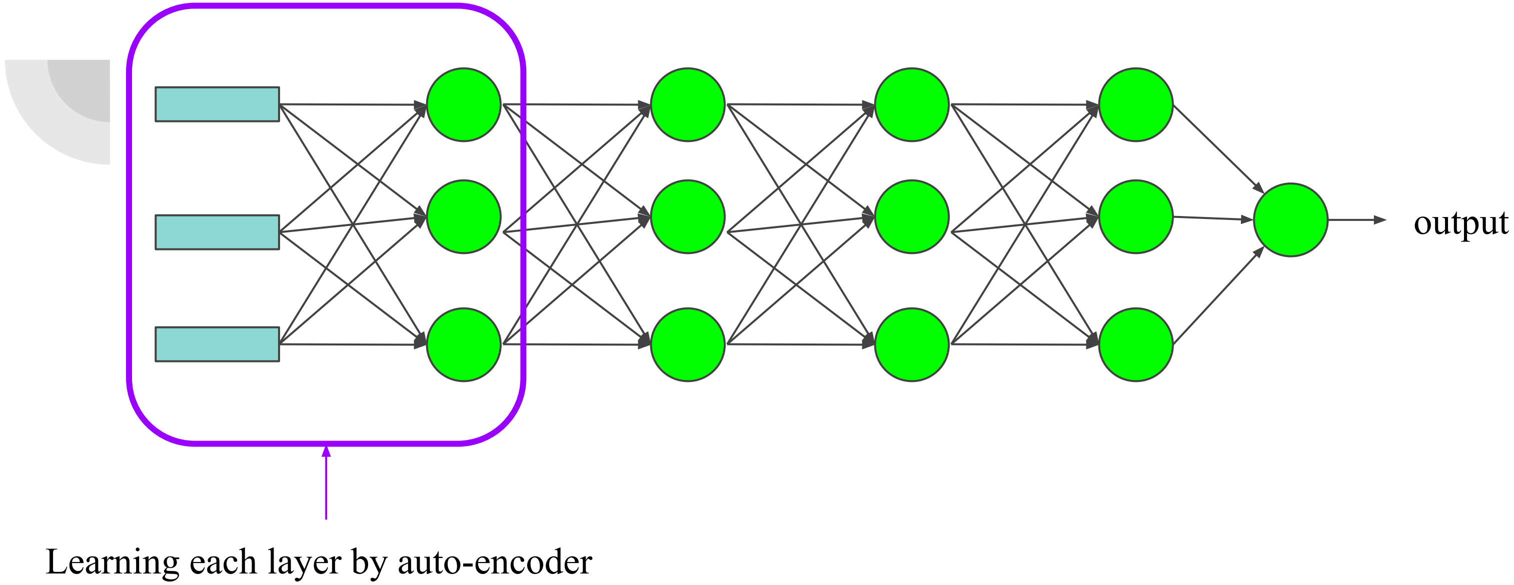
Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*



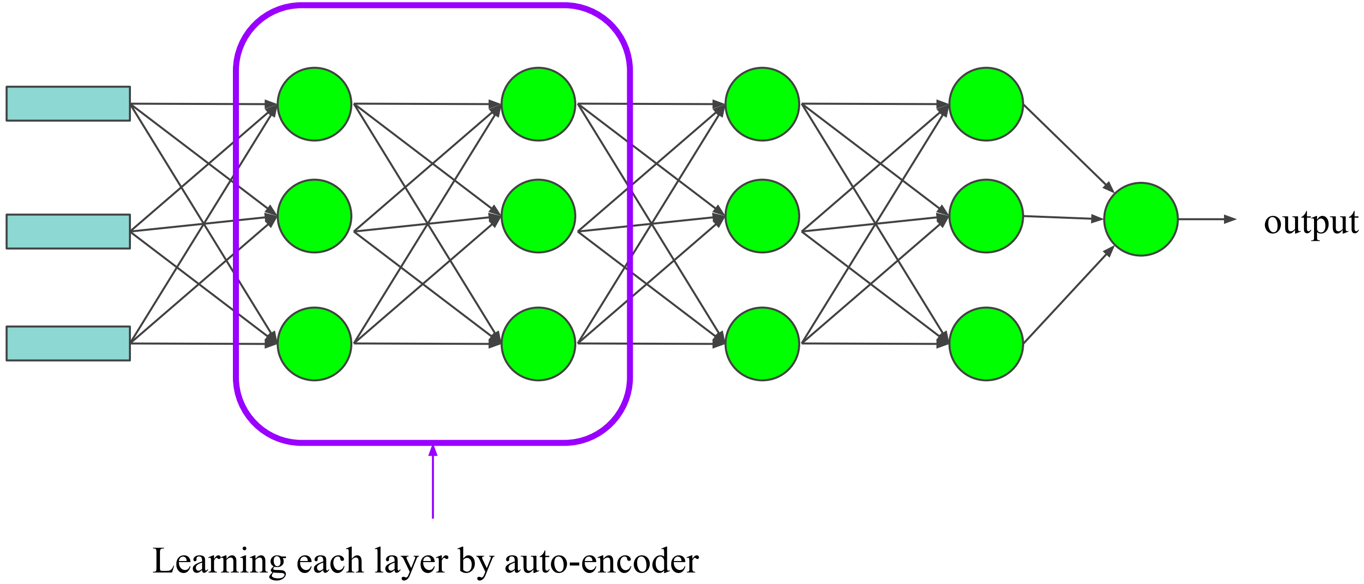
There are distinct characteristics of each digit

Each hidden node performs a recognition of some features.

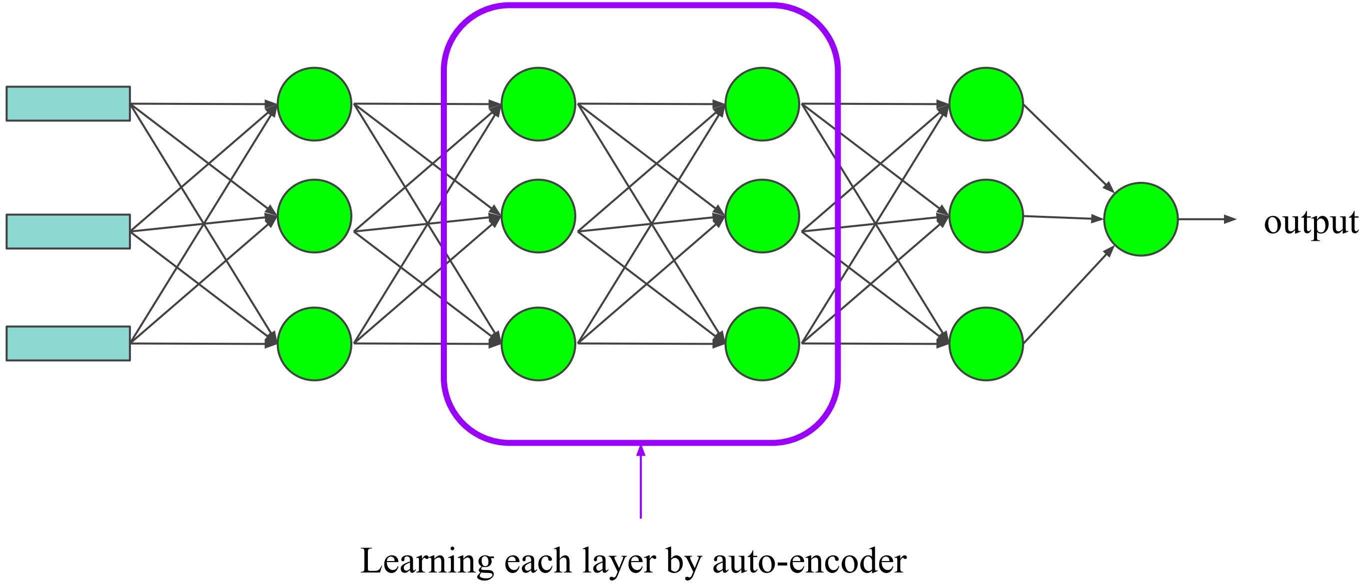
Multiple layers to learn features



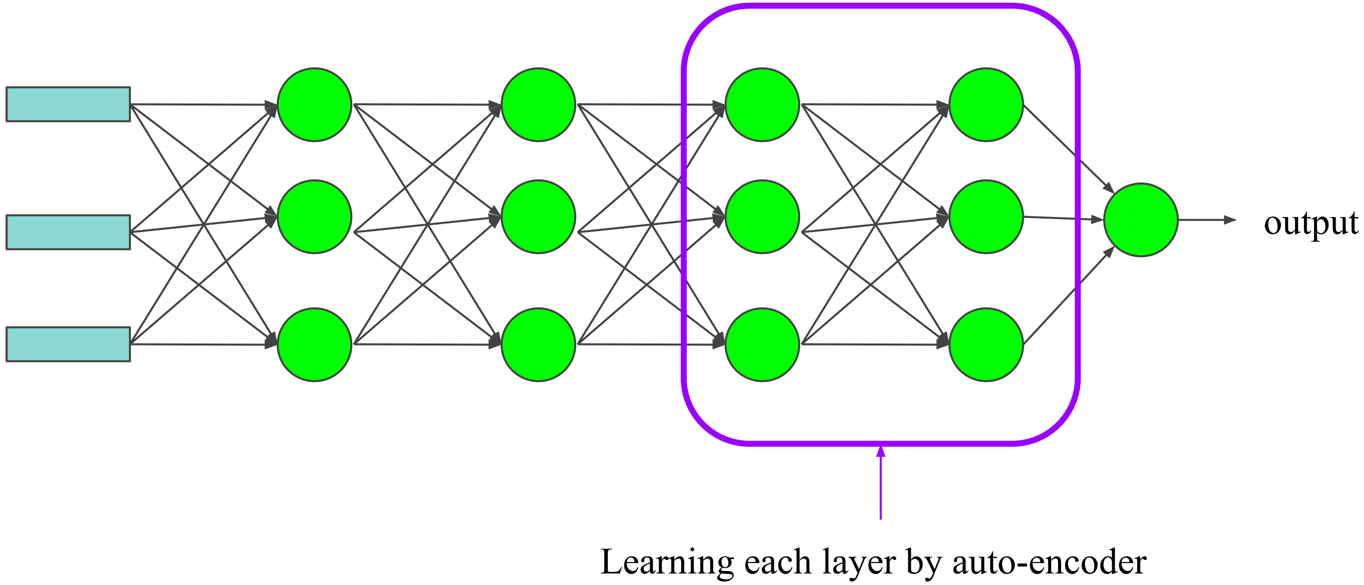
Multiple layers to learn features



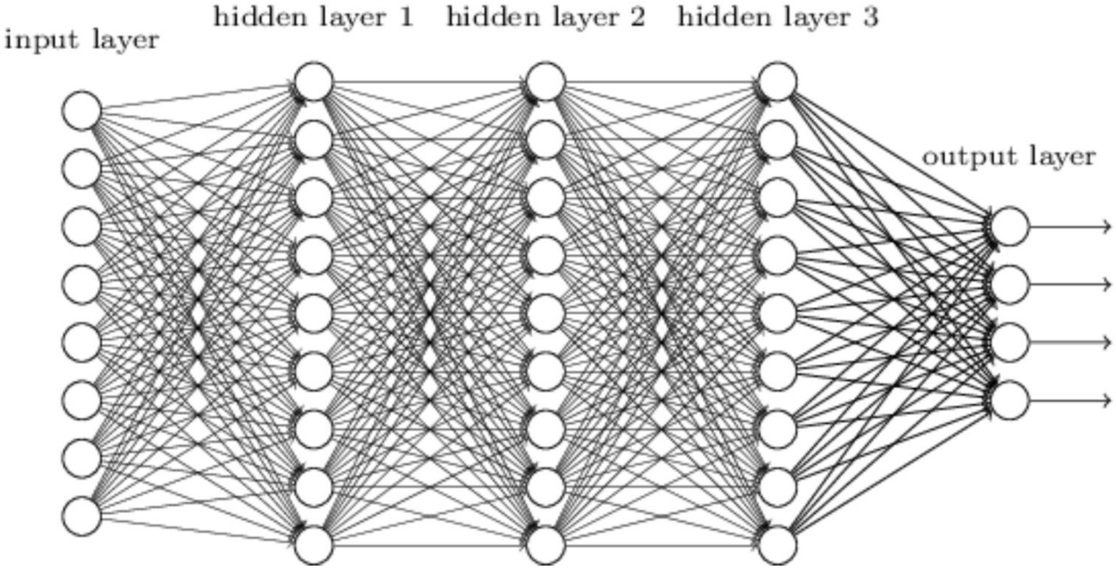
Multiple layers to learn features



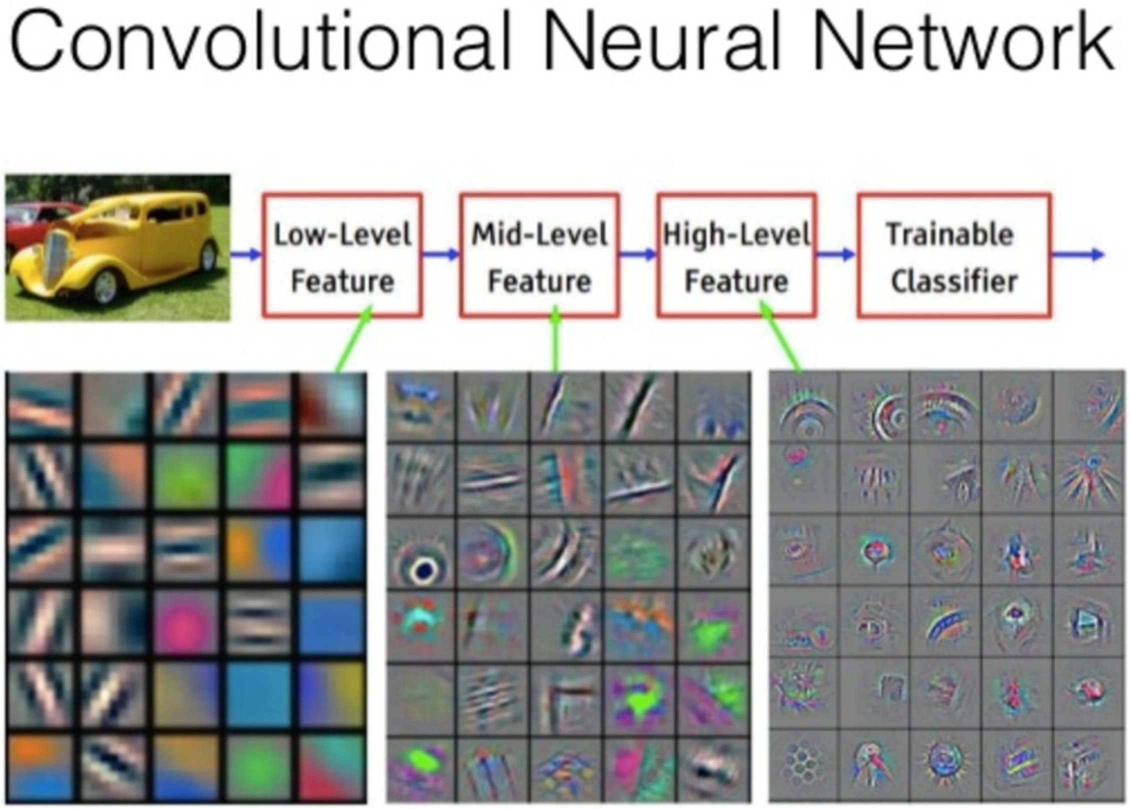
Multiple layers to learn features



Fully connected layer



https://www.cs.tau.ac.il/~dcor/Graphics/pdf.slides



Feature Visualization of Convnet trained on ImageNet from [Zeiler & Fergus 2013]

Convolutional concept

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

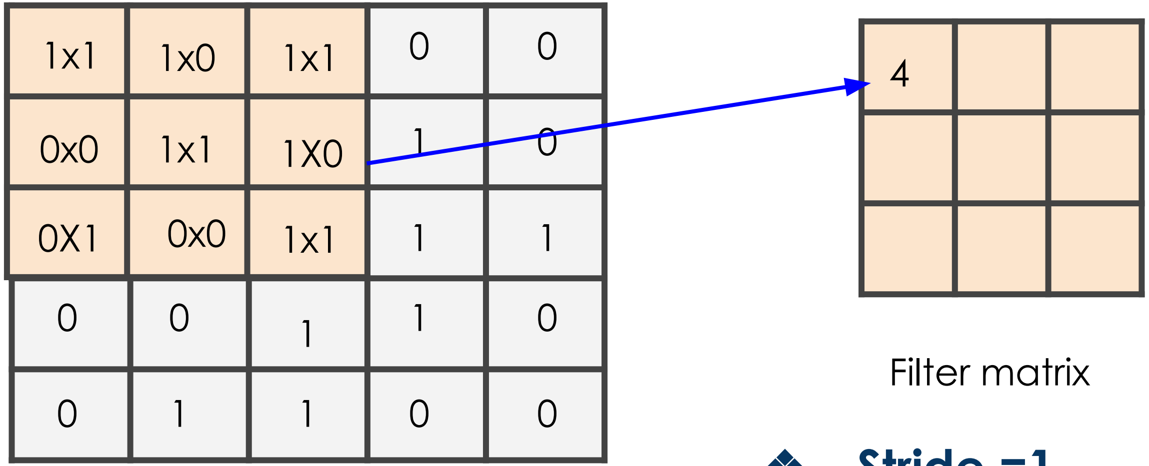
image

1	0	1
0	1	0
1	0	1

Filter matrix

- **Stride =1**

Convolutional concept

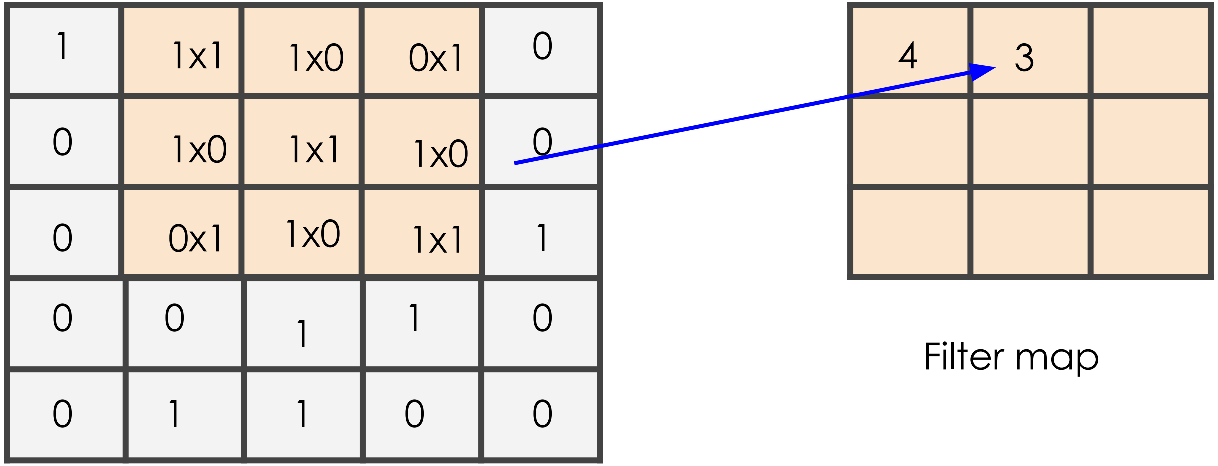


image

Filter matrix

- **Stride =1**

Convolutional concept



image

Filter map

Convolutional concept

1	1	1x1	0x0	0x1
0	1	1x0	1x1	0x0
0	0	1x1	1x0	1x1
0	0	1	1	0
0	1	1	0	0

image

4	3	4

Filter map

Convolutional concept

1	1	1	0	0
0x1	1x0	1x1	1	0
0x0	0x1	1x0	1	1
0x1	0x0	1x1	1	0
0	1	1	0	0

image

4	3	4
2		

Filter map

Convolutional concept

1	1	1	0	0
0	1x1	1x0	1x1	0
0	0x0	1x1	1x0	1
0	0x1	1x0	1x1	0
0	1	1	0	0

image

4	3	4
2	4	

Filter map

Convolutional concept

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

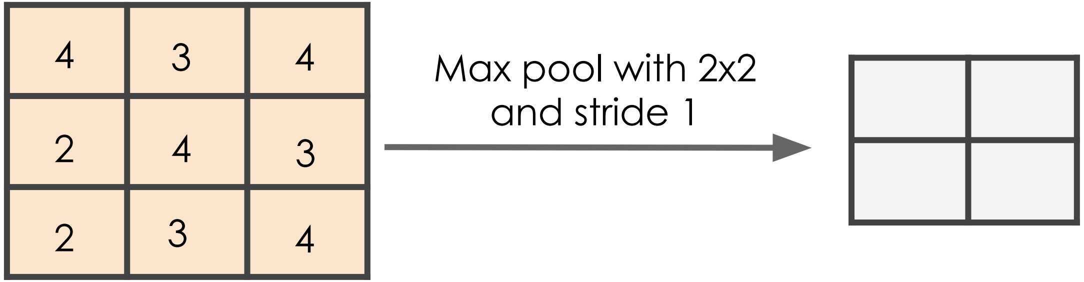
image

4	3	4
2	4	3
2	3	4

Filter map

Pooling layer

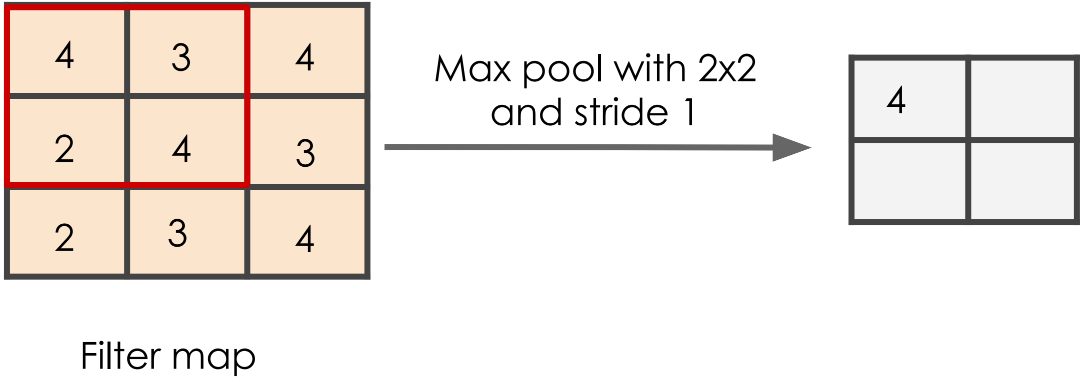
- Max pooling



Filter map

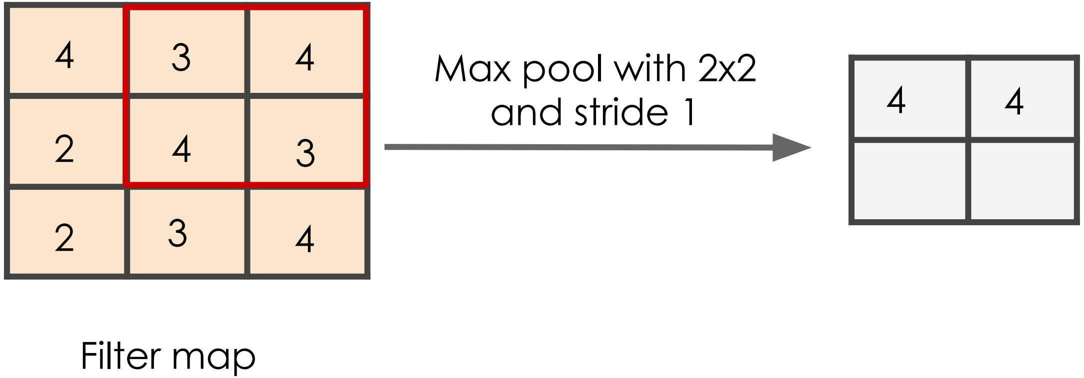
Pooling layer

- Max pooling



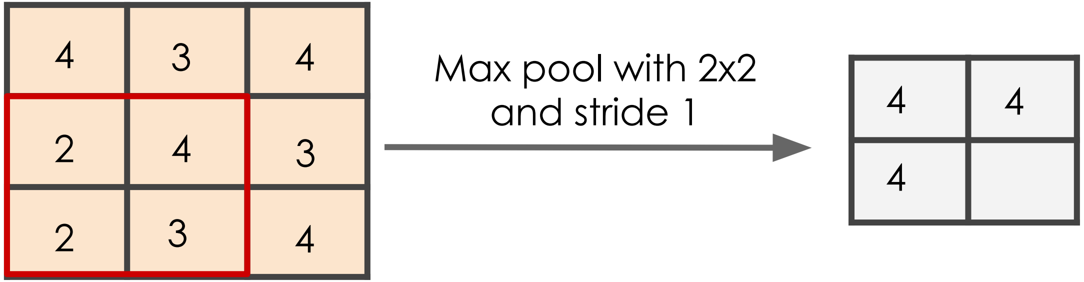
Pooling layer

- Max pooling



Pooling layer

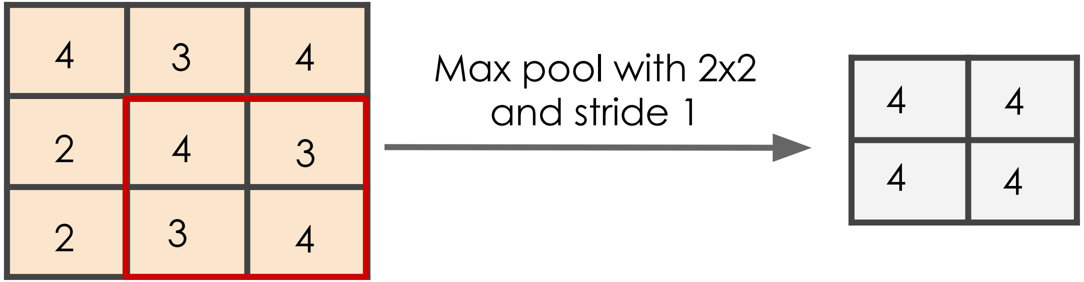
- Max pooling



Filter map

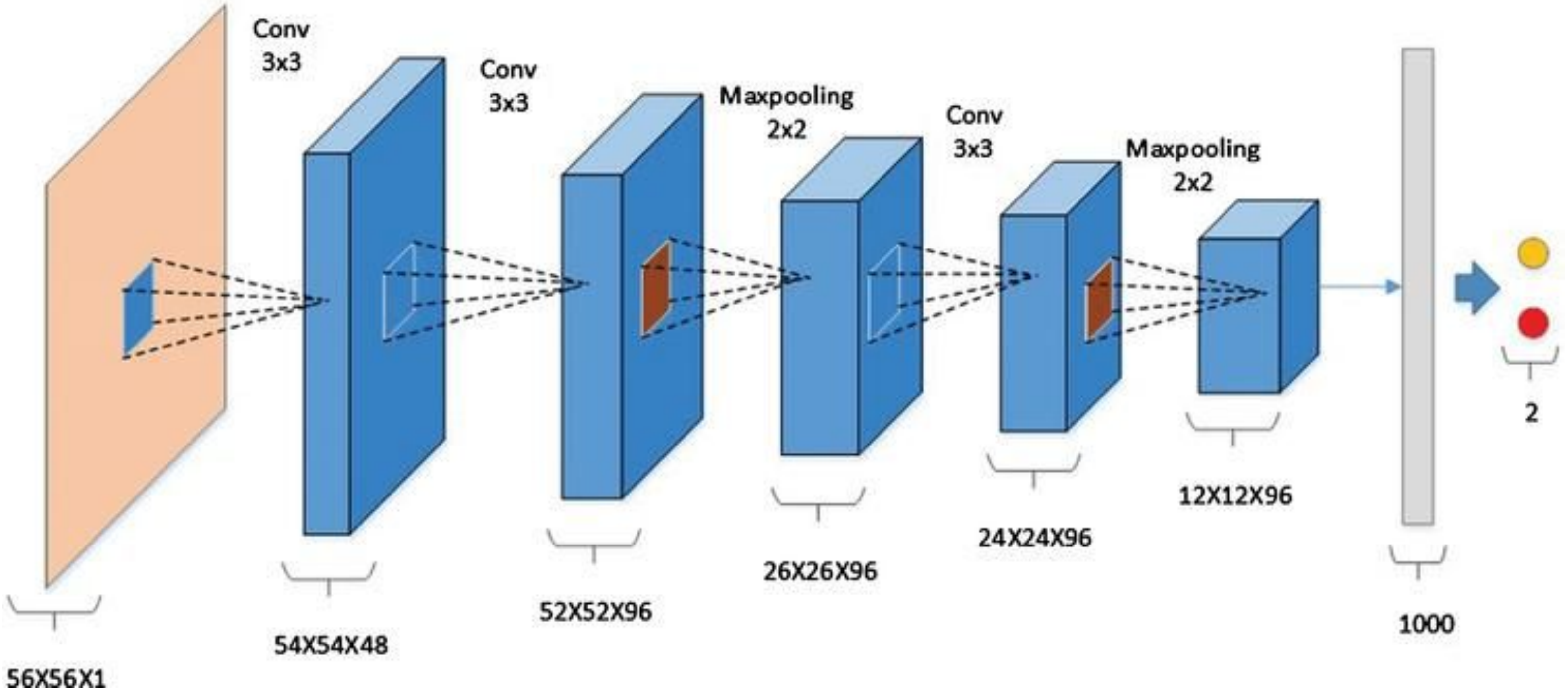
Pooling layer

- Max pooling



Filter map

Deep learning architecture



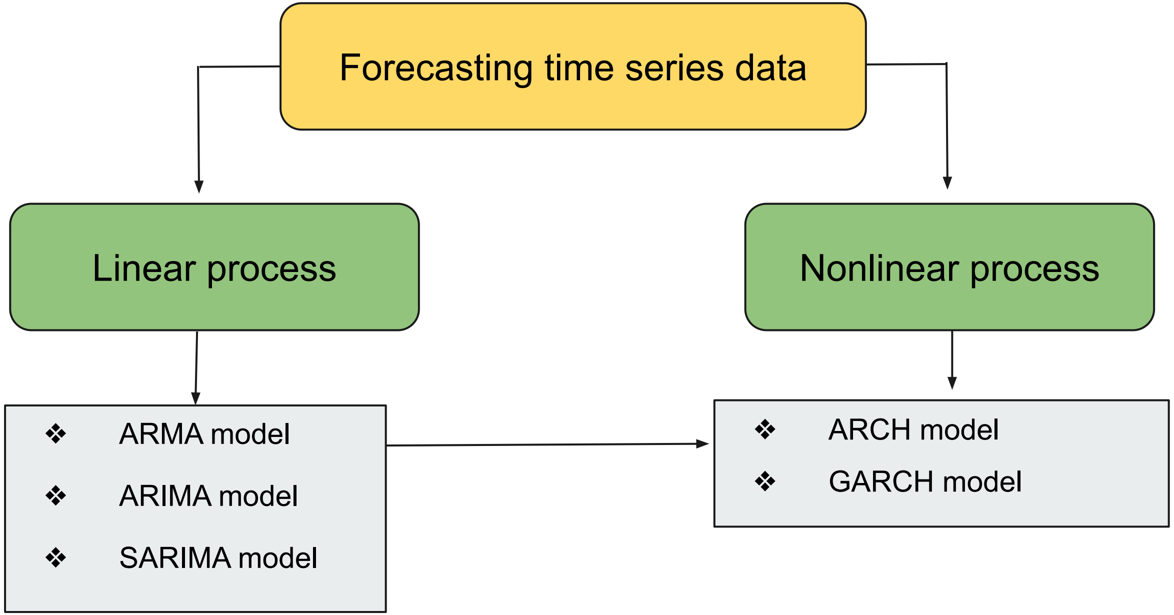
https://content.iospress.com/articles/journal-of-x-ray-science-and-technology/xst17302

Our methodology

Model Identification

- Box-Jenkins methodology
- Box-Jenkins model identification

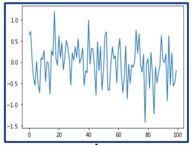
- Auto ARIMA method
- Selection criterion:AIC



Traditional process of building time series model

ARMA model

Time series data



Generate

- Sample ACF
- Sample PACF
- ESACF

Identify

ARMA(*p*,*q*)

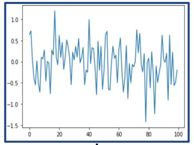
Fit coefficients

$$\phi(B)x_t = \theta(B)\epsilon_t$$

Forecast

ARIMA model

Time series data



Generate

- Identify differencing order (*d*)
- Sample ACF
- Sample PACF
- ESACF

Identify

ARIMA(*p*,*d*,*q*)

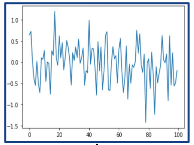
Fit coefficients

$$\phi(B) \bigtriangledown^d x_t = \theta(B)\epsilon_t$$

Forecast

SARIMA model

Time series data



Generate

- Identify seasonal period (*s*)
- Sample ACF
- Sample PACF
- ESACF

Identify

SARIMA(*p*,*d*,*q*)x(*P*,*D*,*Q*)

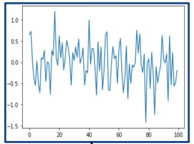
Fit coefficients

$$\Phi(B^s)\phi(B) \bigtriangledown^d \bigtriangledown^D x_t = \Theta(B^s)\theta(B)\epsilon_t$$

Forecast

ARCH model

Time series data



Fit

SARIMA(*p*,*d*,*q*)x(*P*,*D*,*Q*)

Generate

Residuals

Identify

ARCH(*q*)

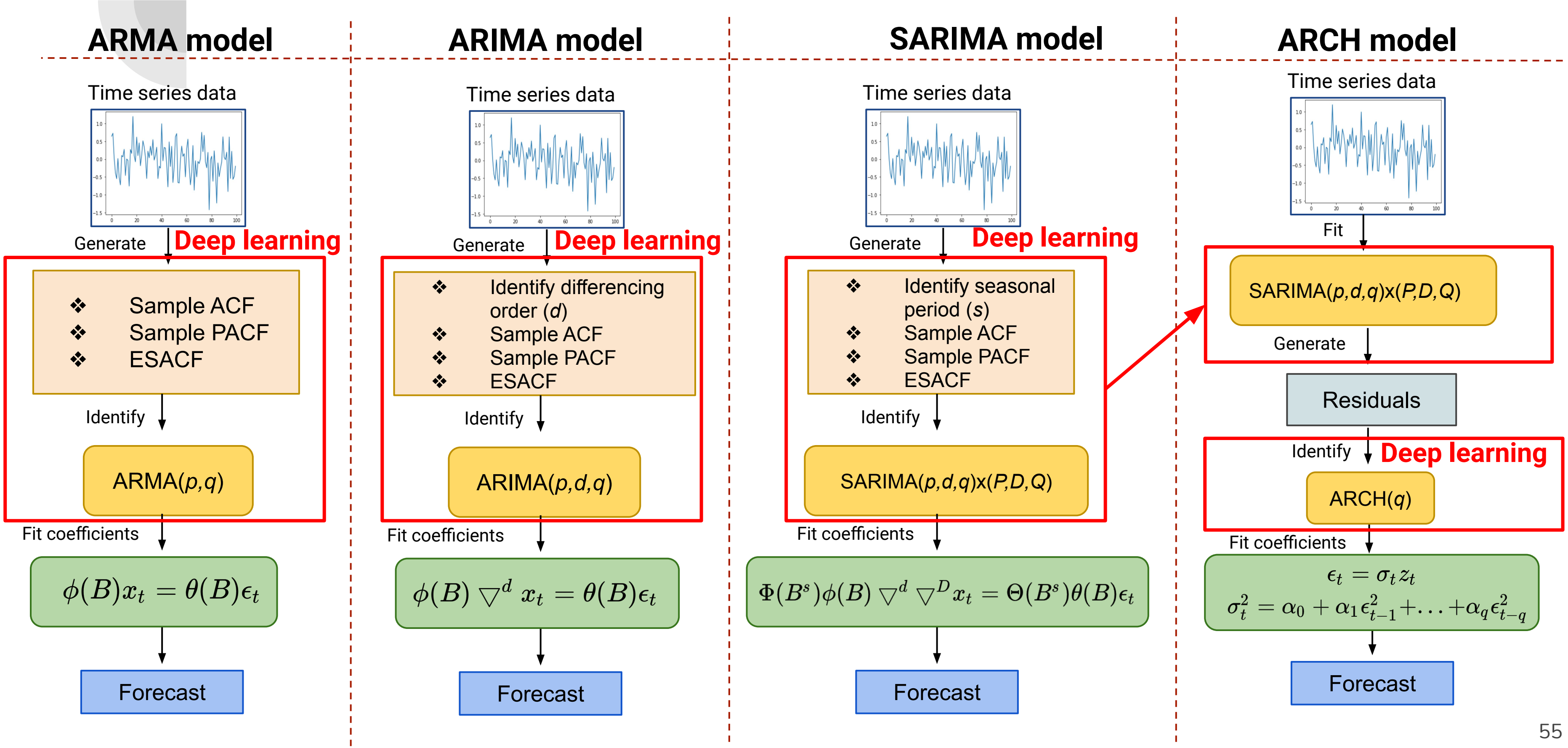
Fit coefficients

$$\epsilon_t = \sigma_t z_t$$

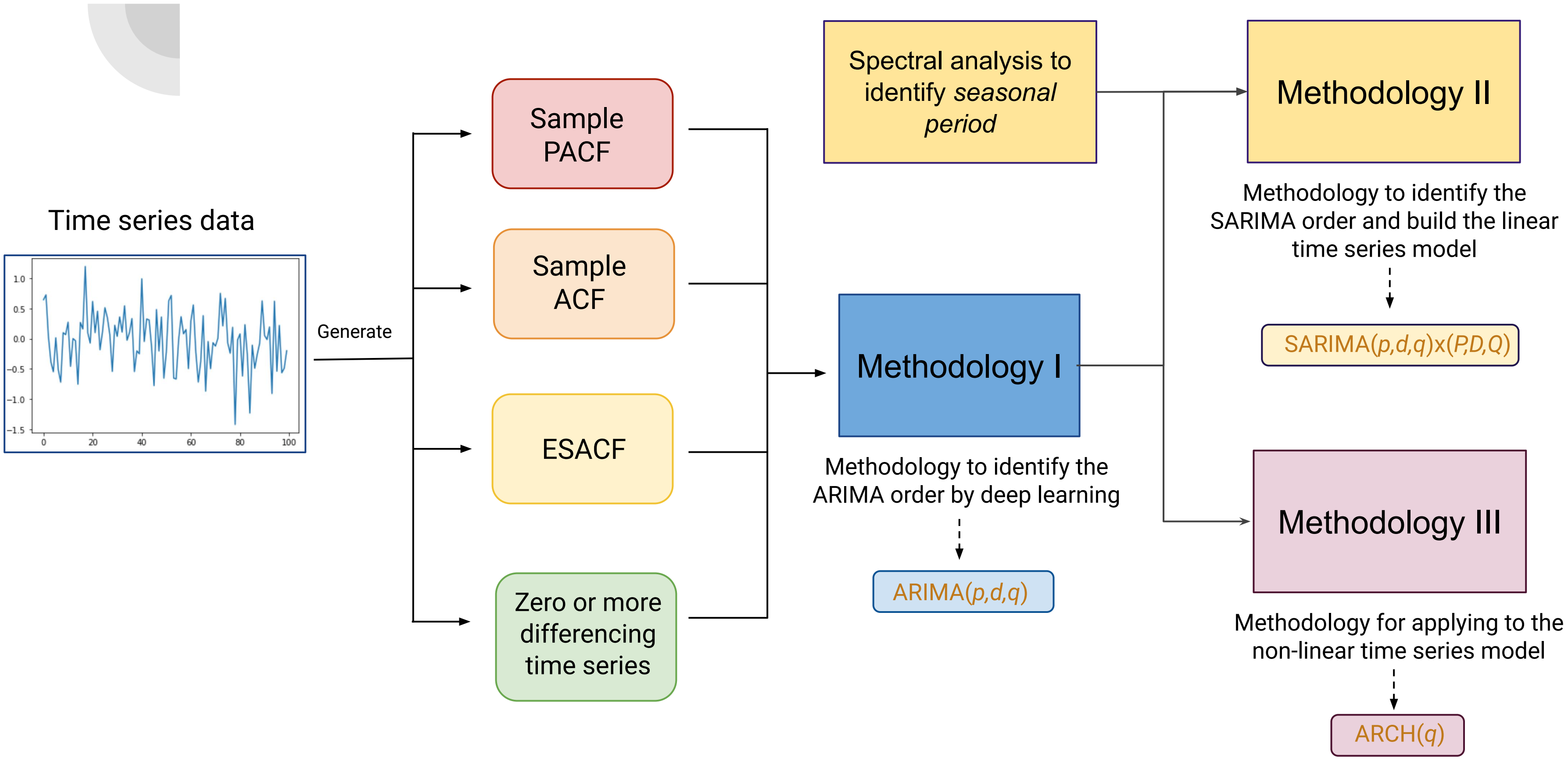
$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \ldots + \alpha_q \epsilon_{t-q}^2$$

Forecast

The new methodologies



Methodologies



White Noise Processes

A process $\{\varepsilon_t\}$ is called a white noise process if it is a sequences of uncorrelated random variables from a fixed distribution.

$$\mathrm{E}[\varepsilon_t] = \mu_\varepsilon$$

$$\mathrm{Var}(\varepsilon_t) = \sigma^2_\varepsilon$$

ACF

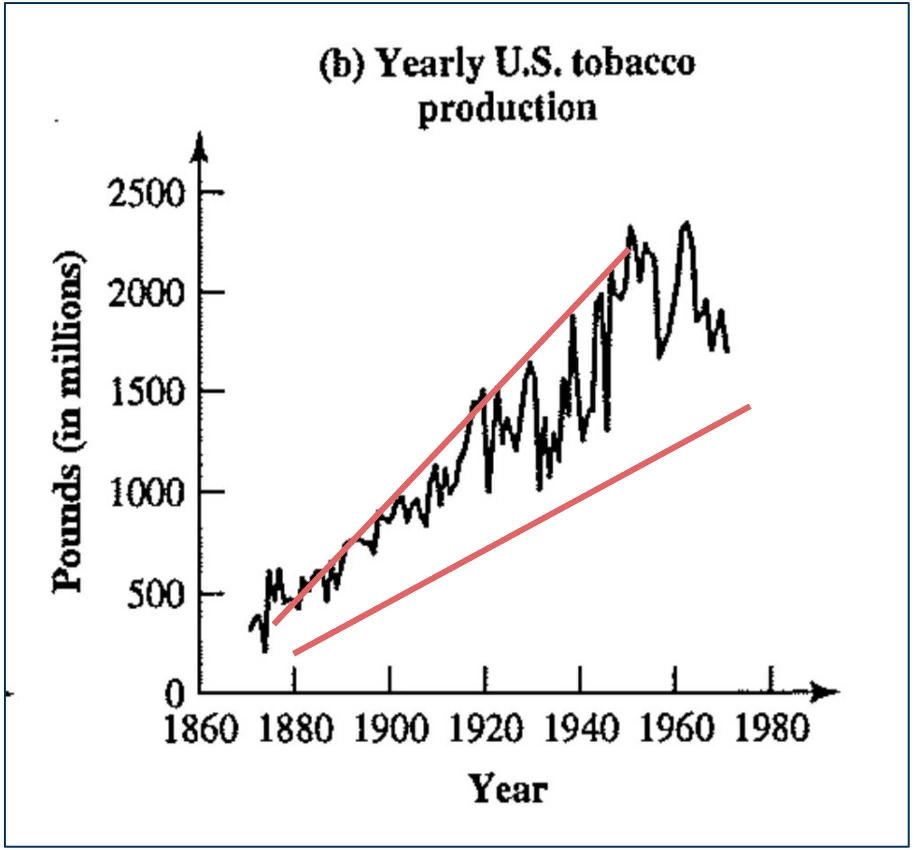
$$\rho_k = \begin{cases} 1 & , k = 0 \\ 0 & , k \neq 0 \end{cases}$$

PACF

$$\phi_{kk} = \begin{cases} 1 & , k = 0 \\ 0 & , k \neq 0 \end{cases}$$

Heteroskadasticity

Time series with variance change by time



Wei, W. W. (2006). Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*.

Autoregressive Conditional Heteroskedasticity (ARCH) model order (q)

Let ε_t be the error terms, σ_t be the standard deviation of the error terms and $z_t \sim N(0, 1)$ be a Gaussian white noise process

where $$\varepsilon_t = \sigma_t z_t$$

and $$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_q \varepsilon_{t-q}^2$$

$\alpha_0, \alpha_1, \ldots, \alpha_q$: coefficients of the ARCH model

Evaluation of forecasting model

Time series model measurements

Let x_i and $\hat{x}_i$ be the actual data and the forecasting data, respectively and N be the number of the forecasted data.

The mean absolute percentage error

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - \hat{x}_i|}{|x_i|}$$

The symmetric mean absolute percentage error

$$SMAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - \hat{x}_i|}{\frac{|x_i| + |\hat{x}_i|}{2}}$$

The root mean square error

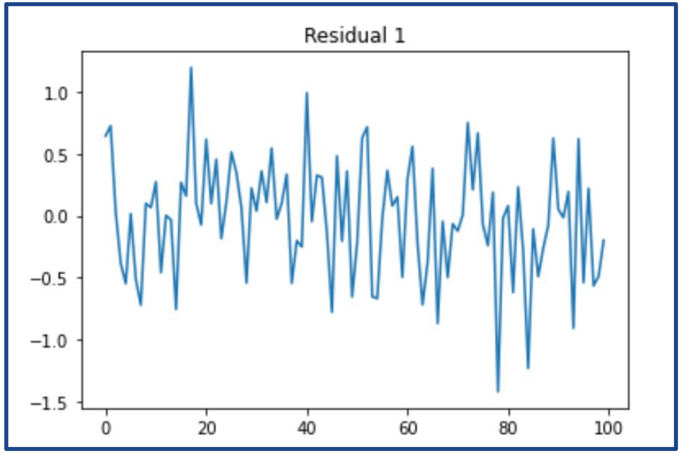
$$RMSE = \sqrt{\sum_{i=1}^{N}\frac{(x_i - \hat{x}_i)^2}{N}}$$

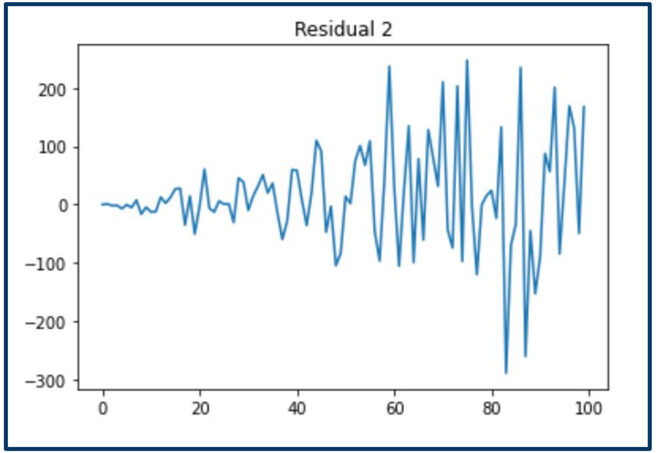
The mean absolute error

$$MAE = \sum_{i=1}^{N}\frac{|x_i - \hat{x}_i|}{N}$$

Residual test of forecasting model

Ljung-Box test of residuals from time series model





The Ljung–Box test can be defined as

- H_0: The data are independently distributed or the correlation in the population from the sample is 0.

- H_1: The data are not independently distributed; they show correlation which is not equal to 0 .

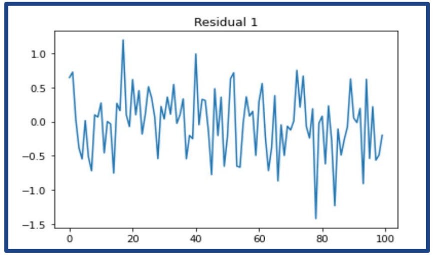
The Ljung–Box test for testing the residual of the fitted time series

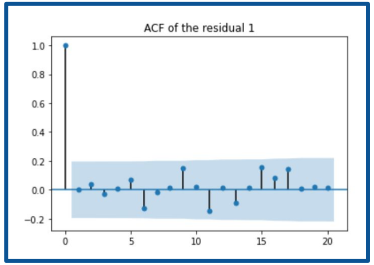
The test statistic of the Ljung-Box test is defined by Q(h) where h is the number of ACF lags being tested

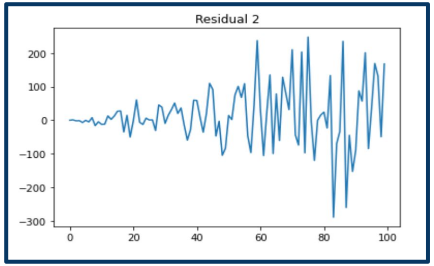
H_0 is rejected when $Q(h) > \chi^2_{1-\alpha,h}$

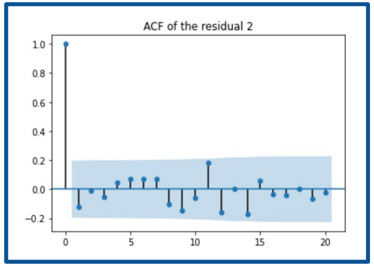
H_0 is rejected when p-value $< \alpha$

The example of using the Ljung-Box test



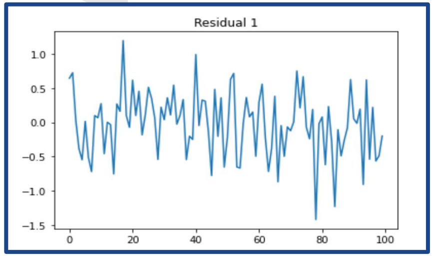




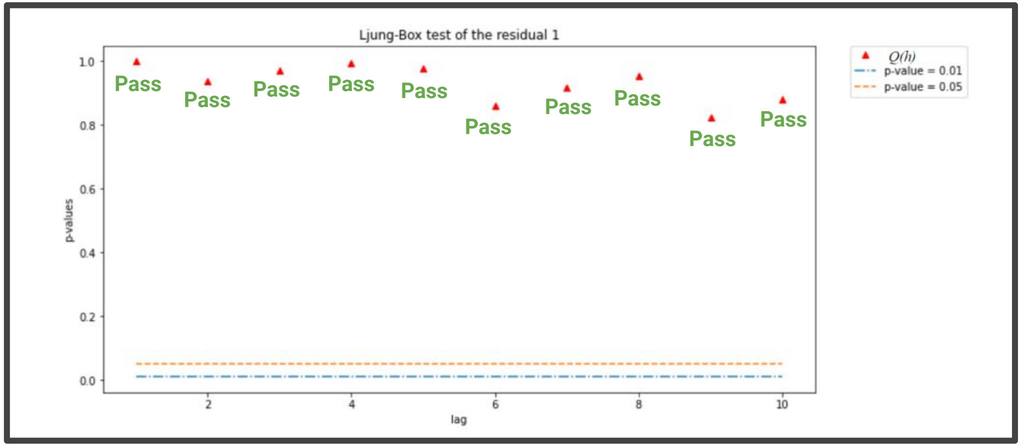


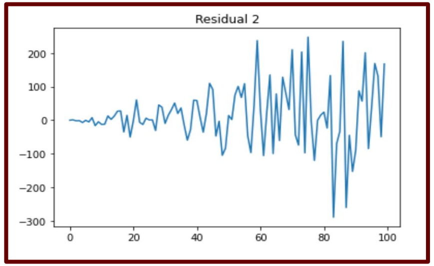
Ljung-Box test of residuals from time series model

The p-values plot of Ljung-Box test

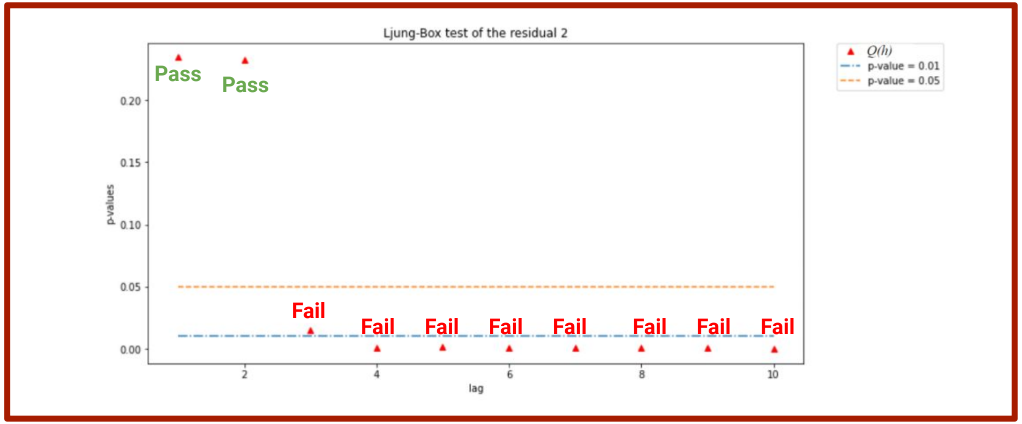


test





test



Evaluation of classifiers

The model measurements for a classification problem

The model measurements for the classification problems using consist of the Precision, the Recall and the F1-score defined by the confusion matrix as follows.

The confusion matrix

		Predicted class	
		Class = Yes	Class = No
Actual class	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Compute

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

METHODOLOGY I

METHODOLOGY TO IDENTIFY THE ARIMA ORDER by deep learning

❖ **The self-identification deep learning or the SID model**

- Constructing the SID model
- Experimental results of the SID model

❖ **The self-identification ResNet-ARIMA model or the SIRO model**

- Constructing the SIRO model
- Experimental results of the SIRO model

❖ **The ACF-PACF-ESACF convolutional neural network ARIMA order identification or the APEA model**

- Constructing the APEA model
- Experimental results of the APEA model

Self-Identification Deep Learning ARIMA

https://iopscience.iop.org/article/10.1088/1742-6596/1564/1/012004/meta

https://iopscience.iop.org/article/10.1088/1742-6596/1564/1/012004/meta

IOPscience Journals Books Publishing Support Login

Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Self-Identification Deep Learning ARIMA

Paisit Khanarsa[1], Arthorn Luangsodsai[1] and Krung Sinapiromsaran[1]

Published under licence by IOP Publishing Ltd

Journal of Physics: Conference Series, Volume 1564, International Conference on Mathematical Models & Computational Techniques in Science & Engineering 22-24 February 2020, London, UK

Citation Paisit Khanarsa *et al* 2020 *J. Phys.: Conf. Ser.* **1564** 012004

Article PDF

References

+ Article information

Abstract

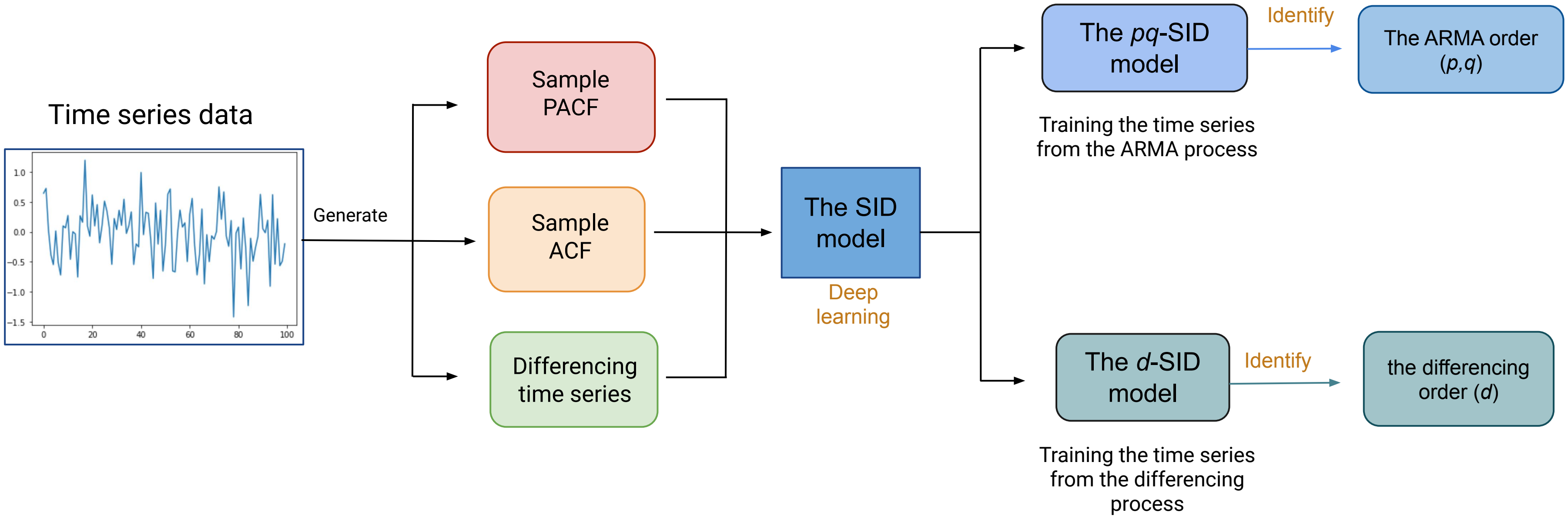
References

Abstract

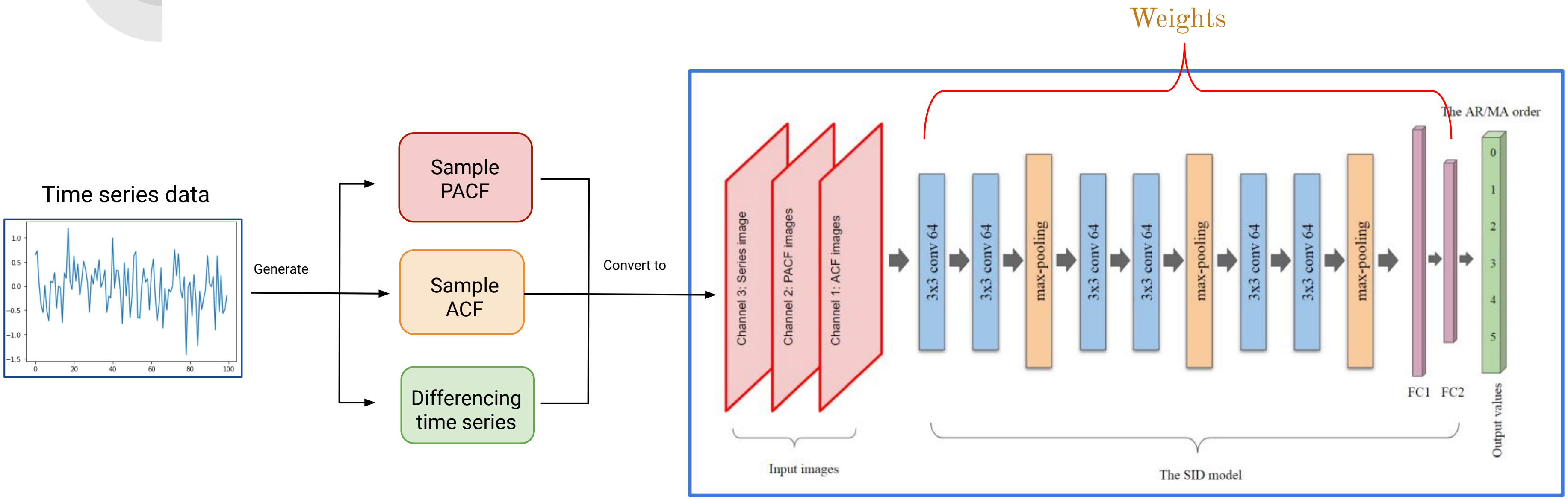
The aspiration to predict the future values as close to the actual values as possible leads to the invention of time series models, the autoregressive integrated moving average (ARIMA) model which requires appropriate parameters of model identification, the ARIMA order, prior to fit coefficients of the models using the Box-Jenkins method. Statisticians for a decade identified the order via the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF) which were very challenging for a human eye. To circumvent this issue, the recent model identification development uses a likelihood based-method that automatically generates orders and fits coefficients by varying the ARIMA order and pick the best one having the smallest Akaike information criterion (AIC) or Bayesian information criterion (BIC). The acquired ARIMA model may fail residual diagnostics. Consequently, this paper proposes the convolution neural network model, called the self-identification deep learning (SID) model, to automatically identify the ARIMA order via

PDF
Help

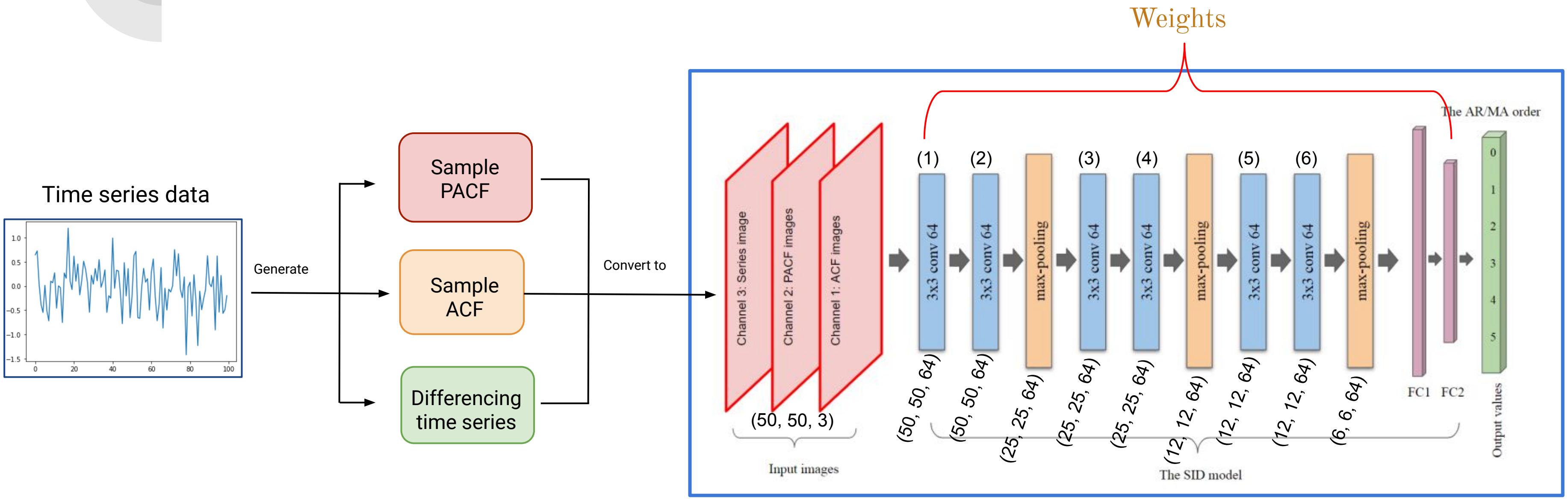
SID model process



pq-SID Architecture



Number of parameters of the pq-SID model



Number of parameters (Conv1) $= 64 \times (3 \times (3 \times 3) + 1) = 1792$

Number of parameters (Conv2-6) $= 64 \times (64 \times (3 \times 3) + 1) = 36{,}928$

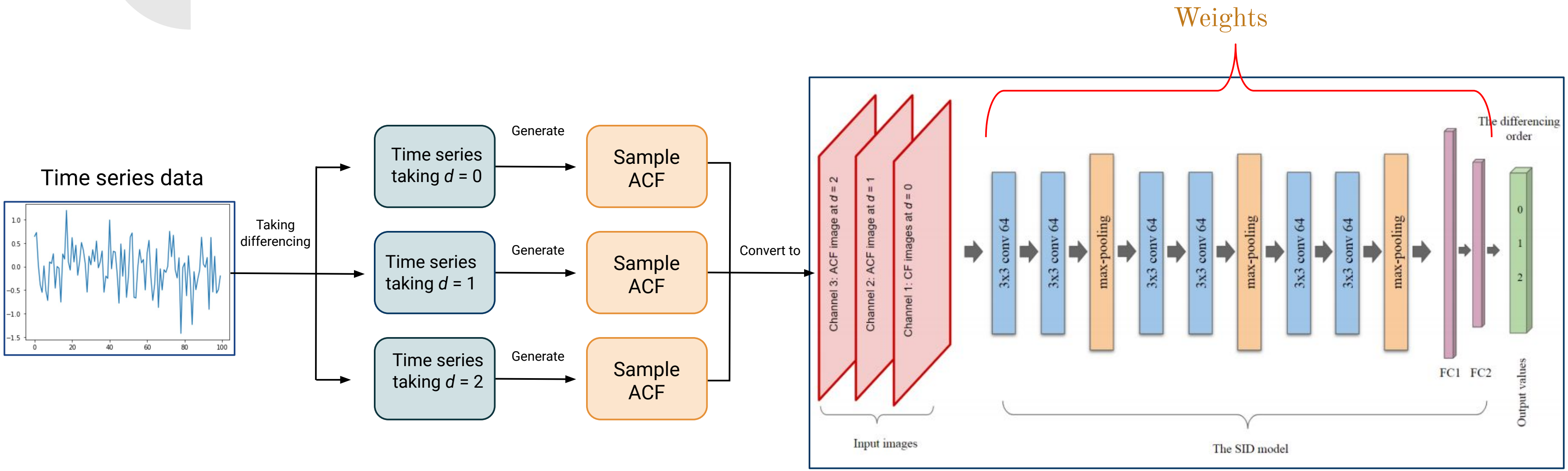
Number of parameters (FC1) $= 1024 \times (2304 + 1) = 2{,}360{,}320$

Number of parameters (FC2) $= 512 \times (1024 + 1) = 524{,}800$

Number of parameters (Outputs) $= 6 \times (512 + 1) = 3{,}078$

Number of parameters : 3,074,630

d-SID Architecture

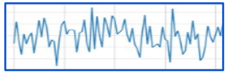


Number of parameters : 3,074,630

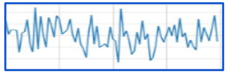
Synthesized time series data inputs for the SID Architecture

Label (p,q) | Random coefficients satisfying stationary and invertibility | **ARMA model** | Simulate time series data | **Time series data**

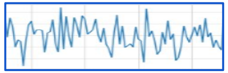
ARMA(0,0) → $x_t = \epsilon_t$ →



ARMA(1,0) → $x_t = \phi_1 x_{t-1} + \epsilon_t$ →

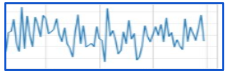


ARMA(2,0) → $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \epsilon_t$ →

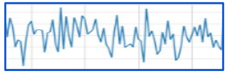


⋮

ARMA(0,1) → $x_t = \epsilon_t - \theta_1 \epsilon_{t-1}$ →

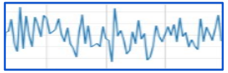


ARMA(0,2) → $x_t = \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2}$ →



⋮

ARMA(1,1) → $x_t = \phi_1 x_{t-1} + \epsilon_t - \theta_1 \epsilon_{t-1}$ →

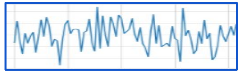


⋮

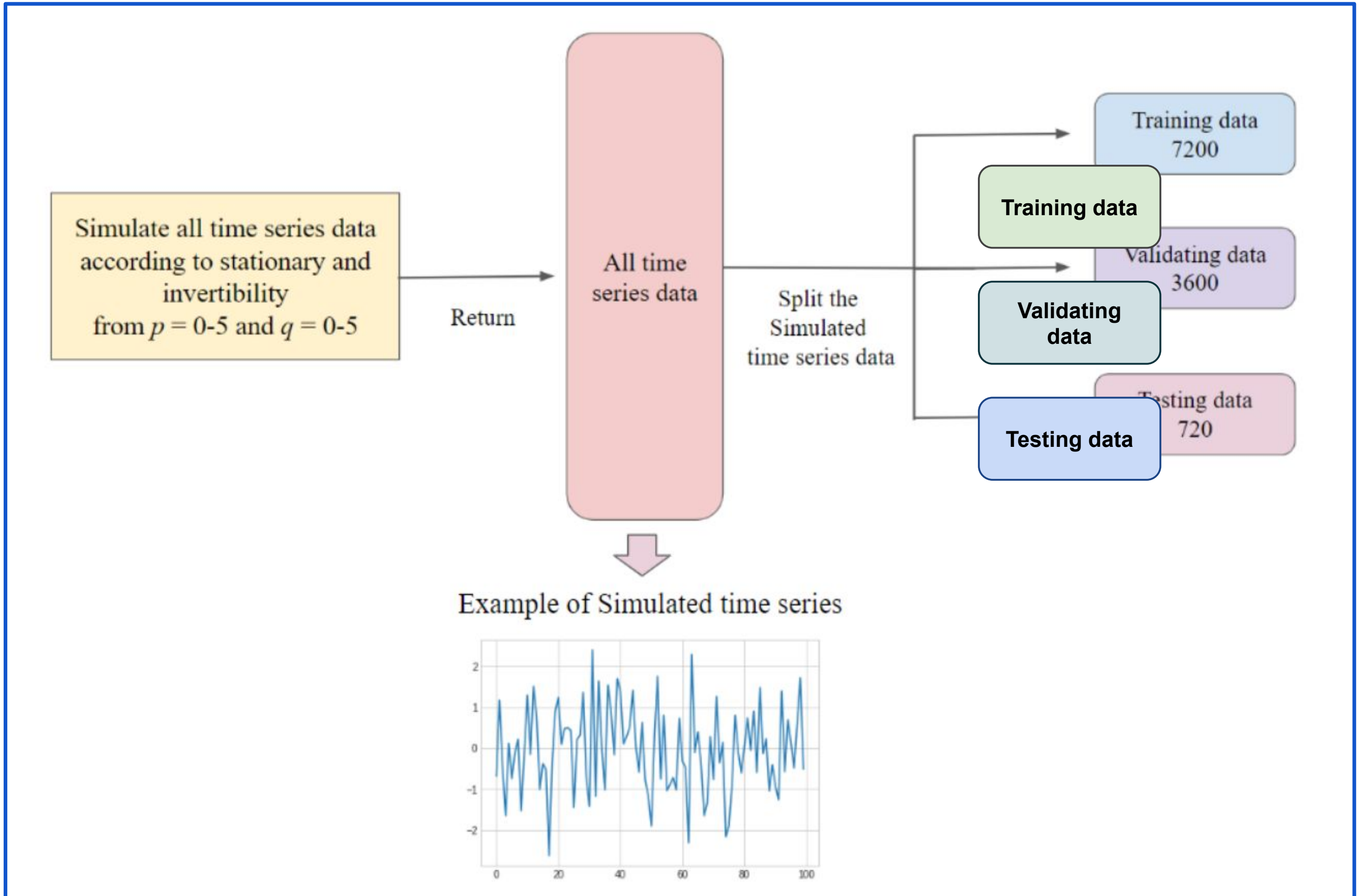
ARMA(5,5) →

$$x_t = \phi_1 x_{t-1} + \ldots + \phi_5 x_{t-5} + \epsilon_t - \theta_1 \epsilon_{t-1} - \ldots - \theta_5 \epsilon_{t-5}$$

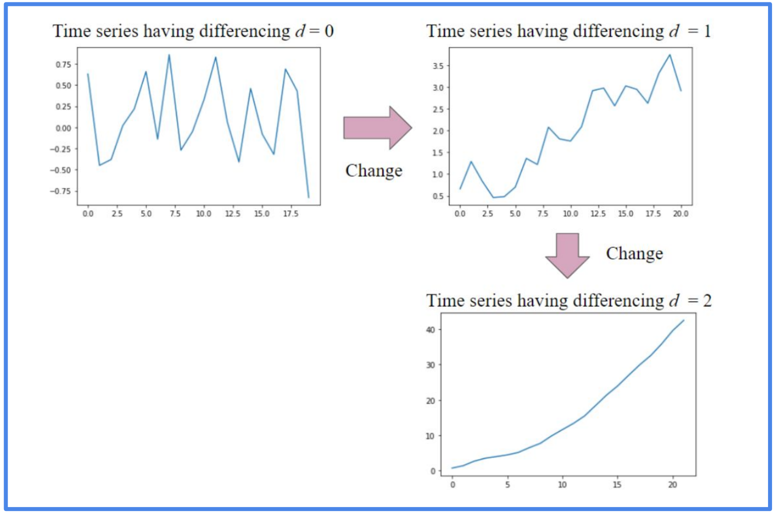
→



Simulation for pq-SID model



Simulation for d-SID model



The example of time series having differencing $d = 0, 1, 2$.

$$\boxed{y_{t+1} = x_t + y_t}$$

where x_t is the simulated data from the ARMA process.

y_t is the simulated data having differencing $d = 1$.

The initial value y_0 is random from the uniform distribution within range -1 to 1.

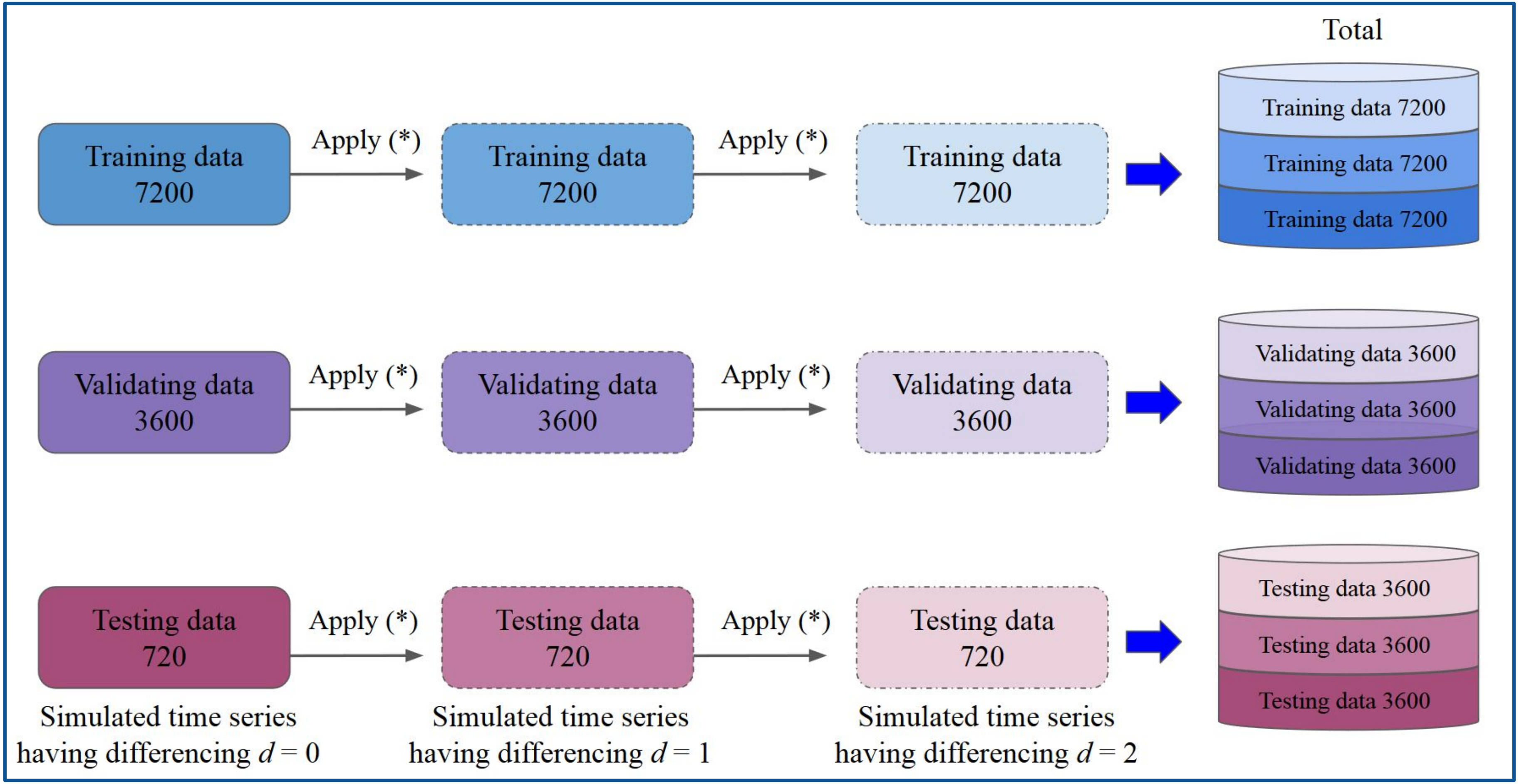
$$\boxed{z_{t+1} = y_t + z_t}$$

where y_t is the simulated data having differencing $d = 1$.

z_t is the simulated data having differencing $d = 2$.

The initial value z_0 is random from the uniform distribution within range -1 to 1.

Train-Validate-Test datasets



Visualizing result of pq-SID model

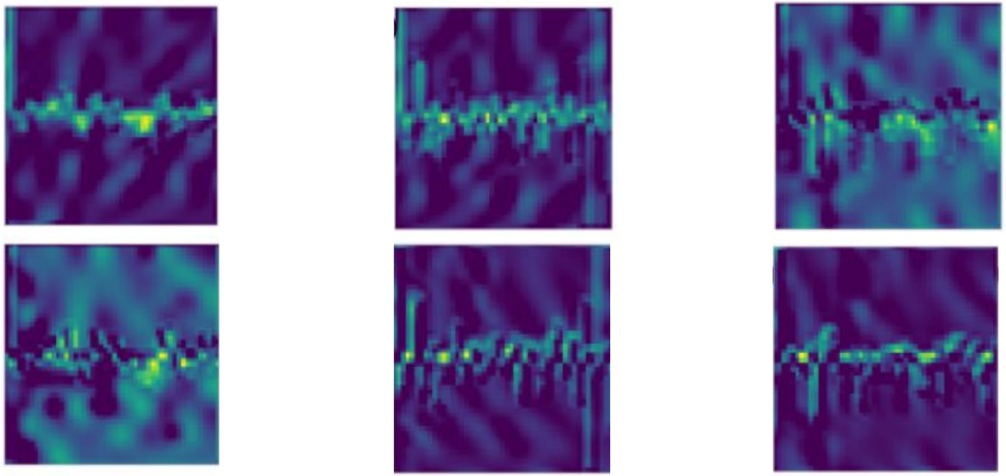
Visualization of the convolutional neural network filter

Minimum value  Maximum value

The color scales of “Viridis”

Conv 1

Conv 2



The filter matrices the AR order is 0

The filter matrices the AR order is 1

The filter matrices the AR order is 2

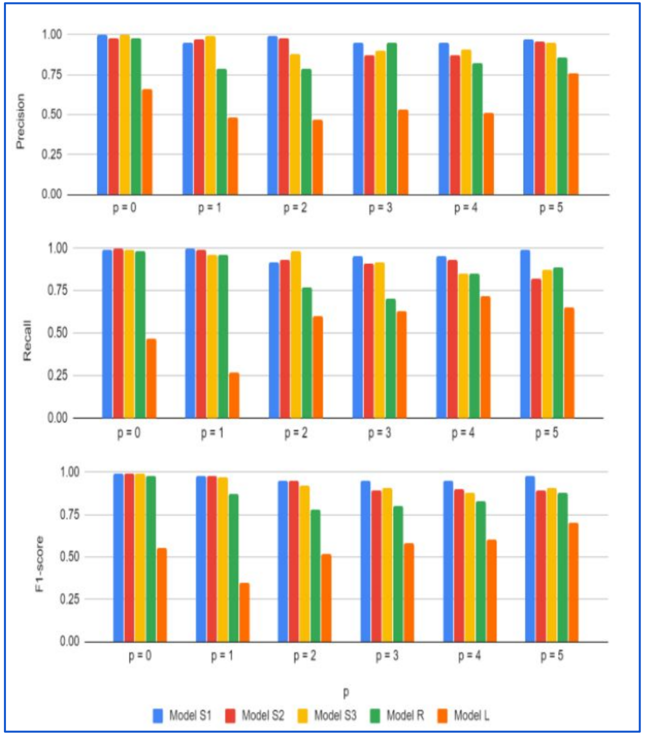
Description of different models

Description of each channel in the models for identifying the ARIMA order

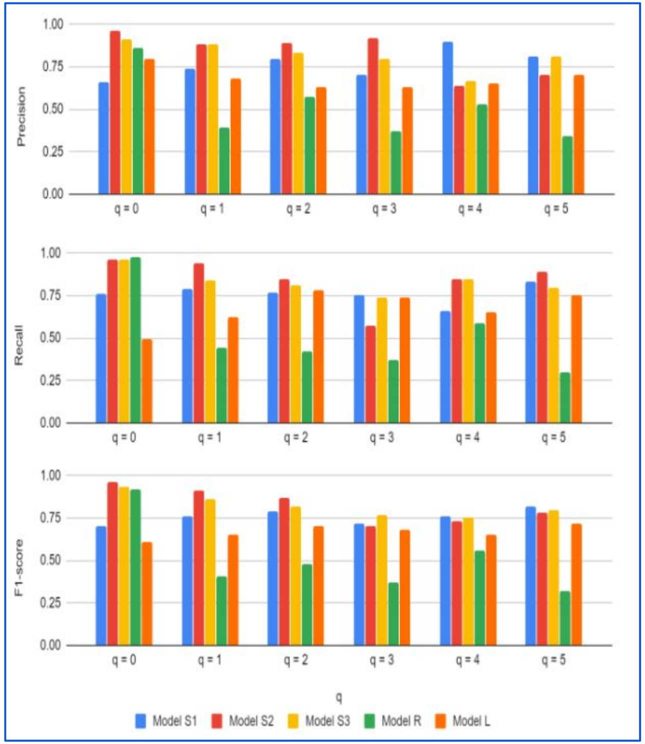
	Model	Case: identify p	Case: identify q	Case: identify d
		Channel(s)	Channel(s)	Channel(s)
Model S1	SID	1 channel: PACF images	1 channel: ACF images	3 channels: ACF images with $d = 0$, 1 and 2
Model S2	SID	2 channels: PACF and ACF images	2 channels: PACF and ACF images	None
Model S3	SID	3 channels: PACF, ACF and time series images	3 channels: PACF, ACF and time series images	None
Model R	ResNet50	Series	Series	None
Model L	Likelihood (AIC criteria)	Series	Series	Series

Identification result for p, d, q of the SID model

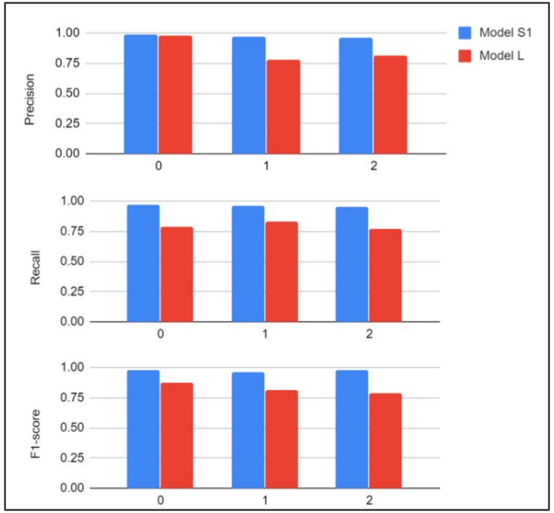
Identifying the p order



Identifying the q order

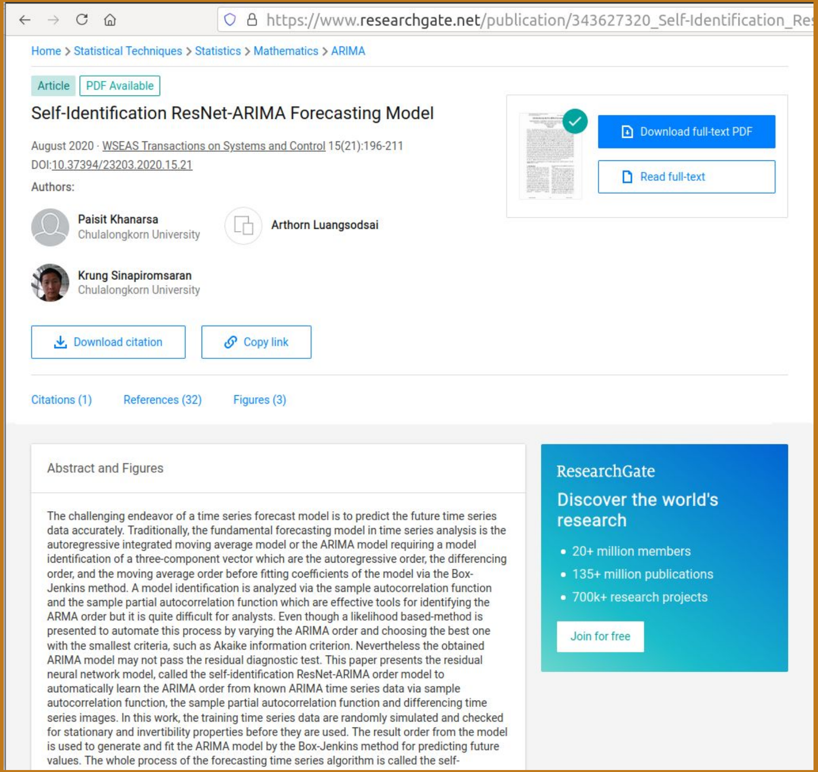


Identifying the d order



Self-Identification ResNet-ARIMA Forecasting Model

https://www.researchgate.net/publication/343627320_Self-Identification_ResNet-ARIMA_Forecasting_Model



https://www.researchgate.net/publication/343627320_Self-Identification_Res

Home > Statistical Techniques > Statistics > Mathematics > ARIMA

Article PDF Available

Self-Identification ResNet-ARIMA Forecasting Model

August 2020 · WSEAS Transactions on Systems and Control 15(21):196-211

DOI:10.37394/23203.2020.15.21

Authors:

Paisit Khanarsa
Chulalongkorn University

Arthorn Luangsodsai

Krung Sinapiromsaran
Chulalongkorn University

Download full-text PDF

Read full-text

Download citation

Copy link

Citations (1) References (32) Figures (3)

Abstract and Figures

The challenging endeavor of a time series forecast model is to predict the future time series data accurately. Traditionally, the fundamental forecasting model in time series analysis is the autoregressive integrated moving average model or the ARIMA model requiring a model identification of a three-component vector which are the autoregressive order, the differencing order, and the moving average order before fitting coefficients of the model via the Box-Jenkins method. A model identification is analyzed via the sample autocorrelation function and the sample partial autocorrelation function which are effective tools for identifying the ARMA order but it is quite difficult for analysts. Even though a likelihood based-method is presented to automate this process by varying the ARIMA order and choosing the best one with the smallest criteria, such as Akaike information criterion. Nevertheless the obtained ARIMA model may not pass the residual diagnostic test. This paper presents the residual neural network model, called the self-identification ResNet-ARIMA order model to automatically learn the ARIMA order from known ARIMA time series data via sample autocorrelation function, the sample partial autocorrelation function and differencing time series images. In this work, the training time series data are randomly simulated and checked for stationary and invertibility properties before they are used. The result order from the model is used to generate and fit the ARIMA model by the Box-Jenkins method for predicting future values. The whole process of the forecasting time series algorithm is called the self-

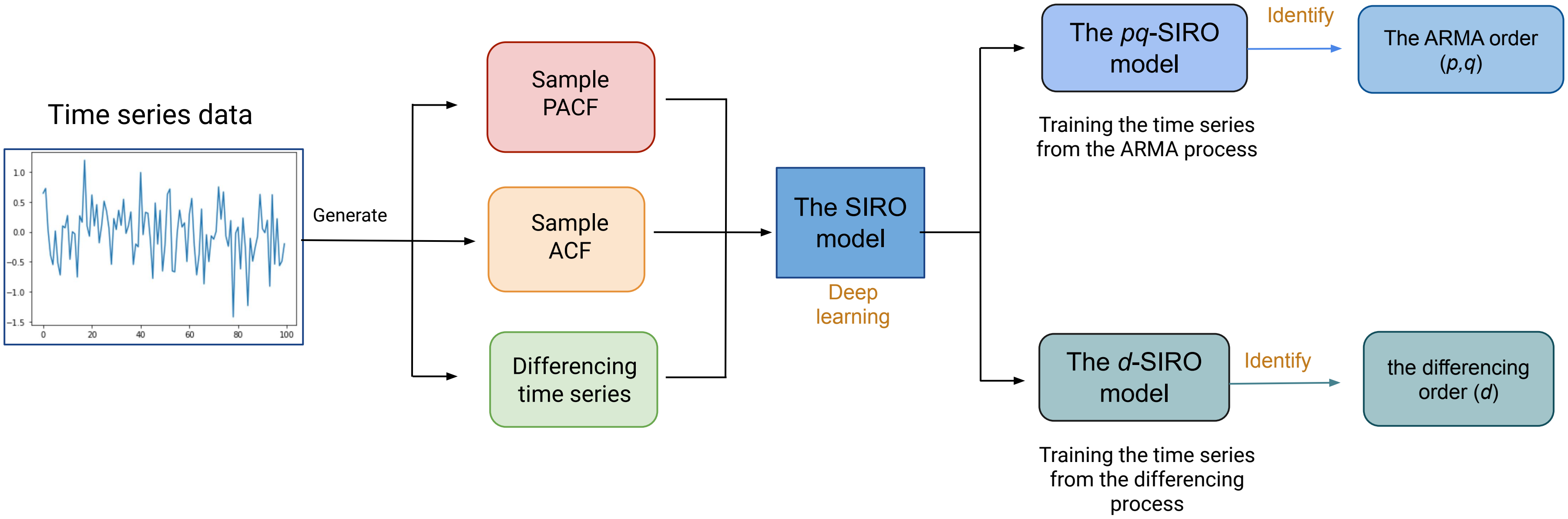
ResearchGate

Discover the world's research

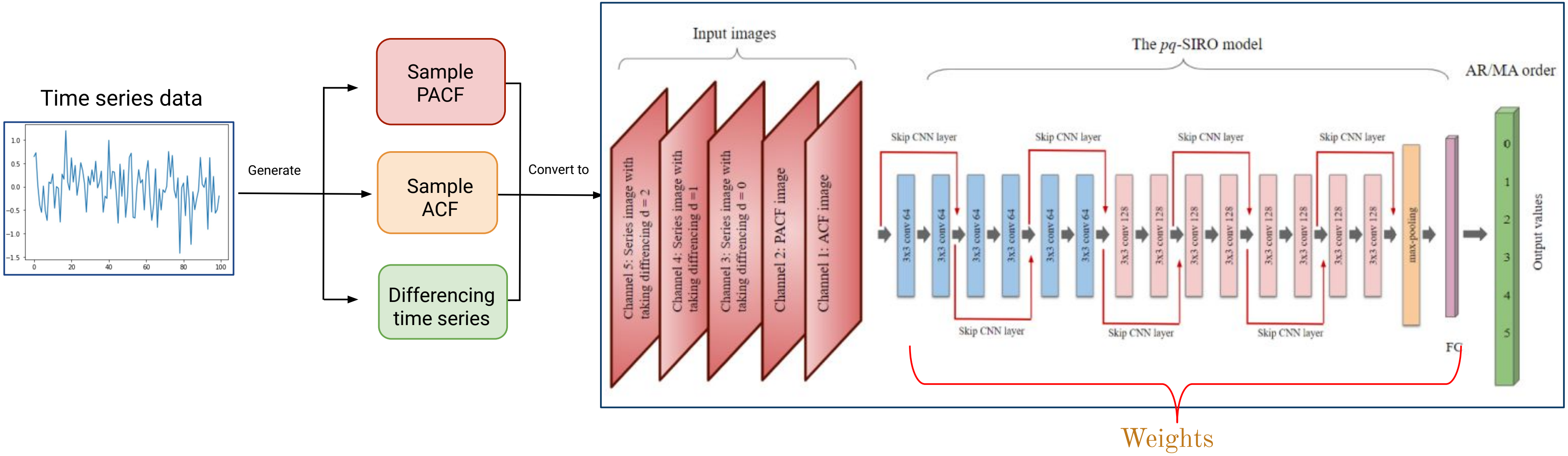
- 20+ million members
- 135+ million publications
- 700k+ research projects

Join for free

The process of the SIRO model

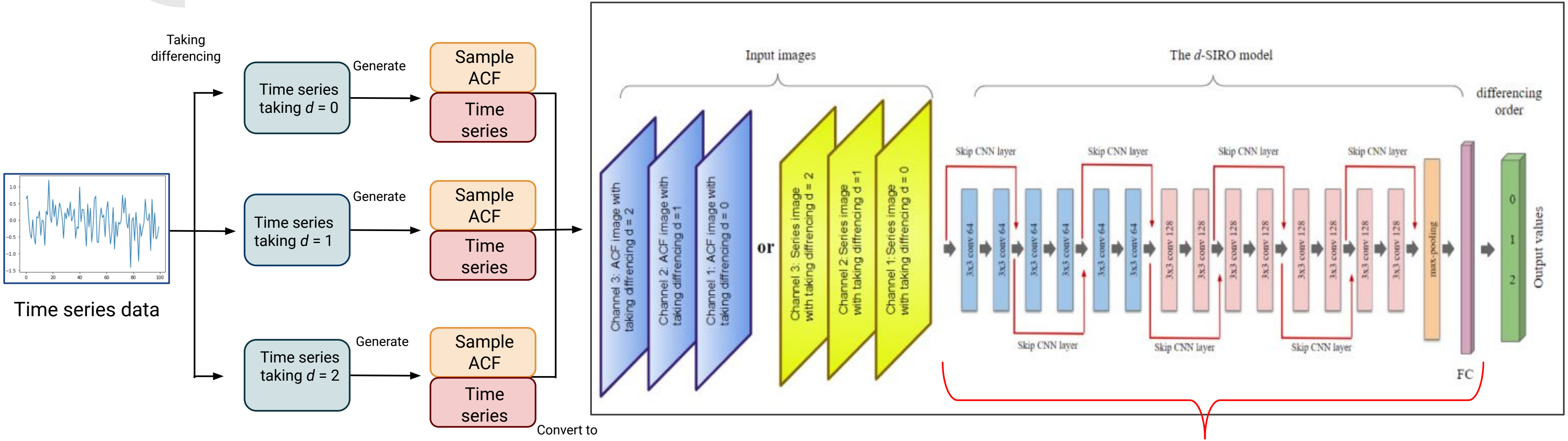


pq-SIRO architecture



Number of parameters : 3,809,030

The architecture the *d*-SIRO model



Weights

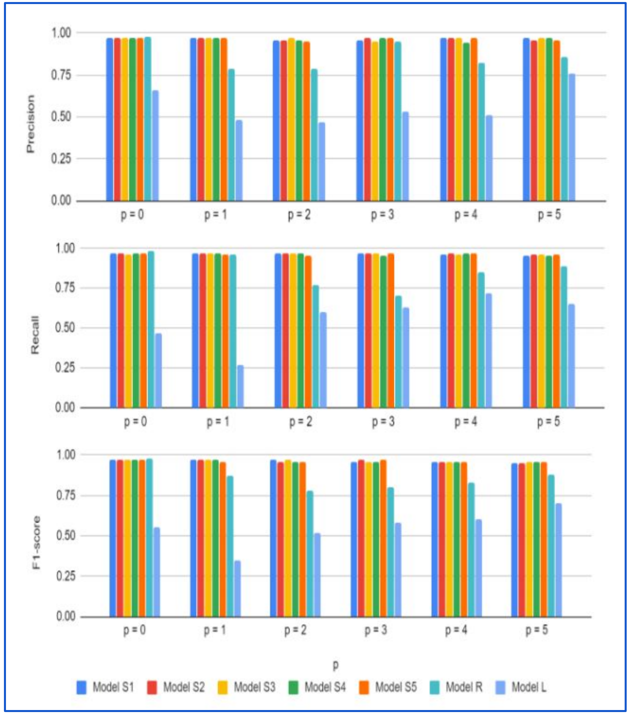
Number of parameters : 3,812,166

Description for selection of the best model

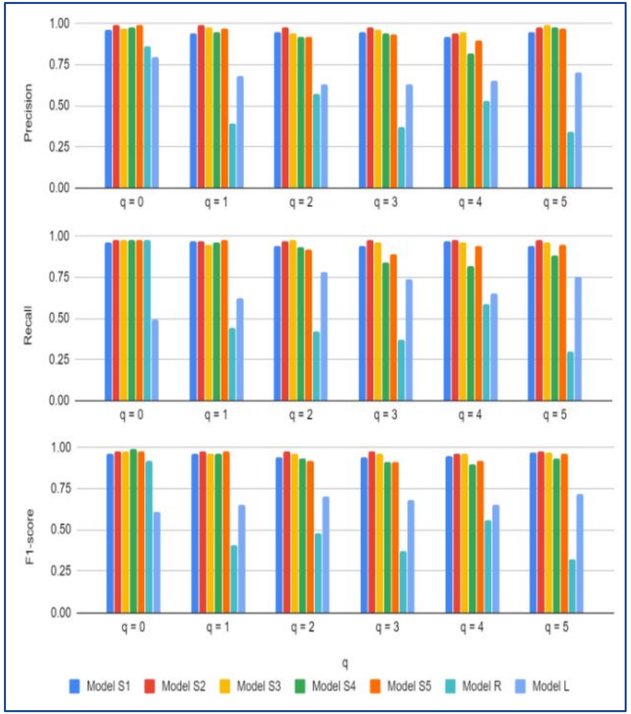
		Model S1	Model S2	Model S3	Model S4	Model S5	Model R	Model L
Model		SIRO	SIRO	SIRO	SIRO	SIRO	ResNet50	Likelihood (AIC criteria)
Case: identify p	Channel(s)	1 channel: PACF images	2 channels: PACF and ACF images	3 channels: PACF, ACF and time series images with d=0	4 channels: PACF, ACF and time series images with d=0,1	5 channels: PACF, ACF and time series images with d=0,1,2	Series	Series
Case: identify q	Channel(s)	1 channel: ACF images	2 channels: PACF and ACF images	3 channels: PACF, ACF and time series images with d=0	4 channels: PACF, ACF and time series images with d=0,1	5 channels: PACF, ACF and time series images with d=0,1,2	Series	Series
Case: identify d	Channel(s)	3 channels: ACF images with d =0,1,2	3 channels: time series images with d =0,1,2	None	None	None	None	Series

Result for identifying p, d, q from the SIRO model

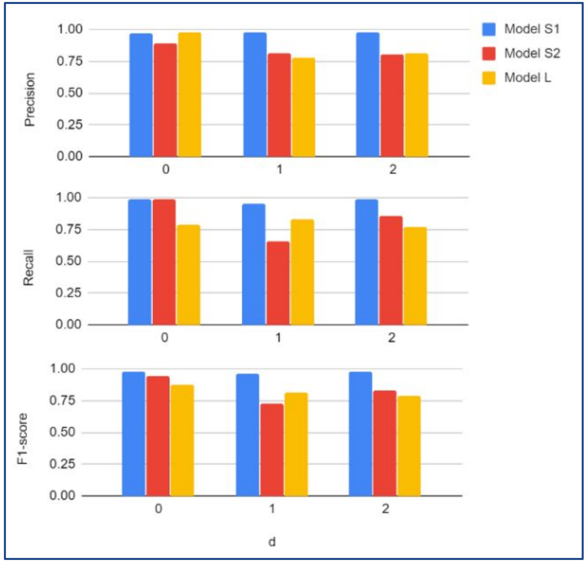
Identifying the p order



Identifying the q order

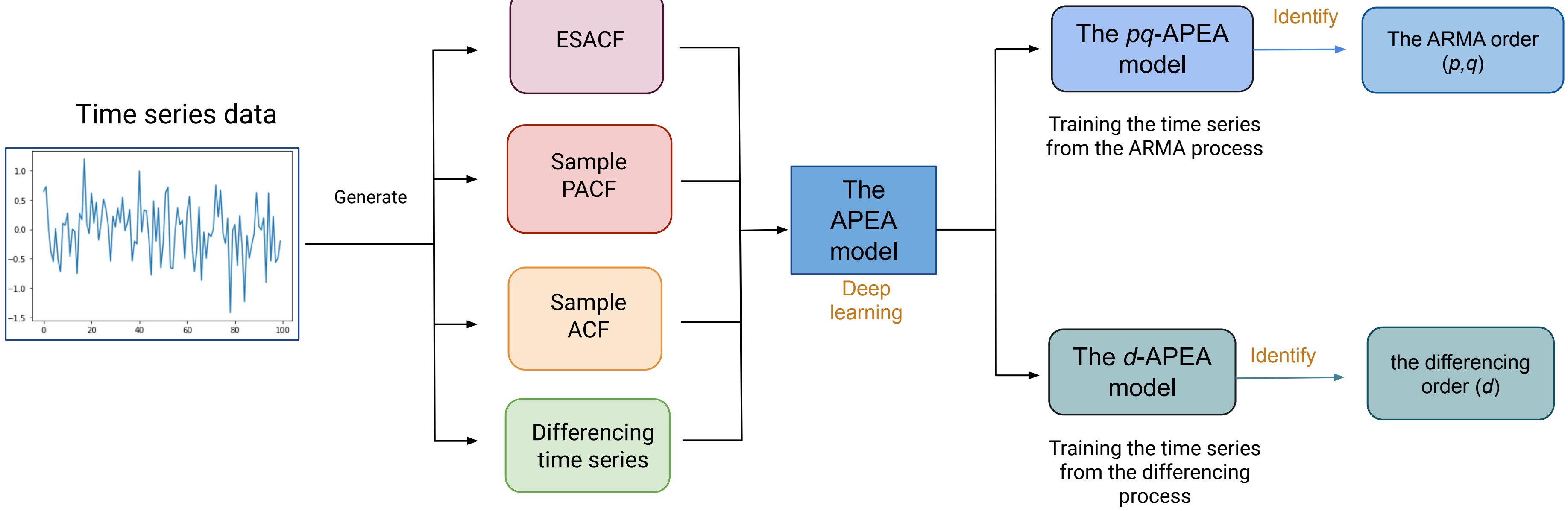


Identifying the d order

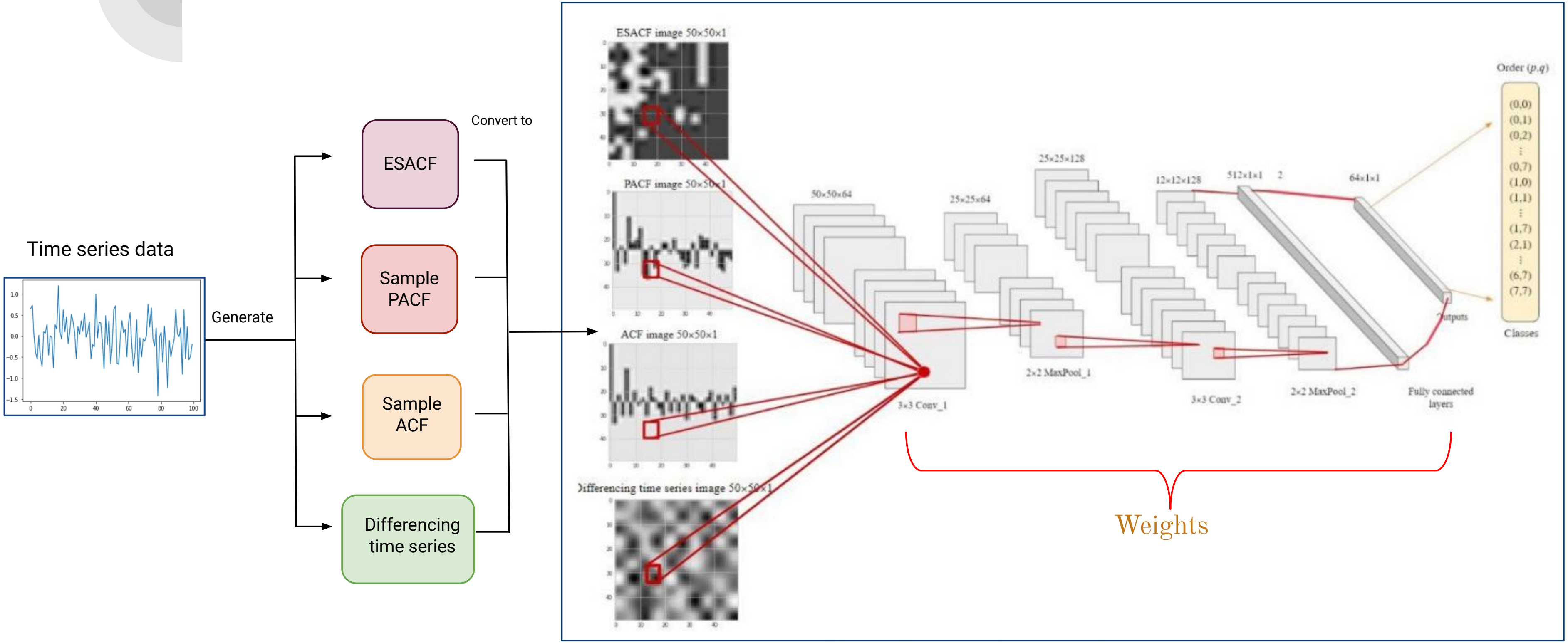


(3) The ACF-PACF-ESACF convolutional neural network ARIMA order identification or the APEA model

The process of the APEA model

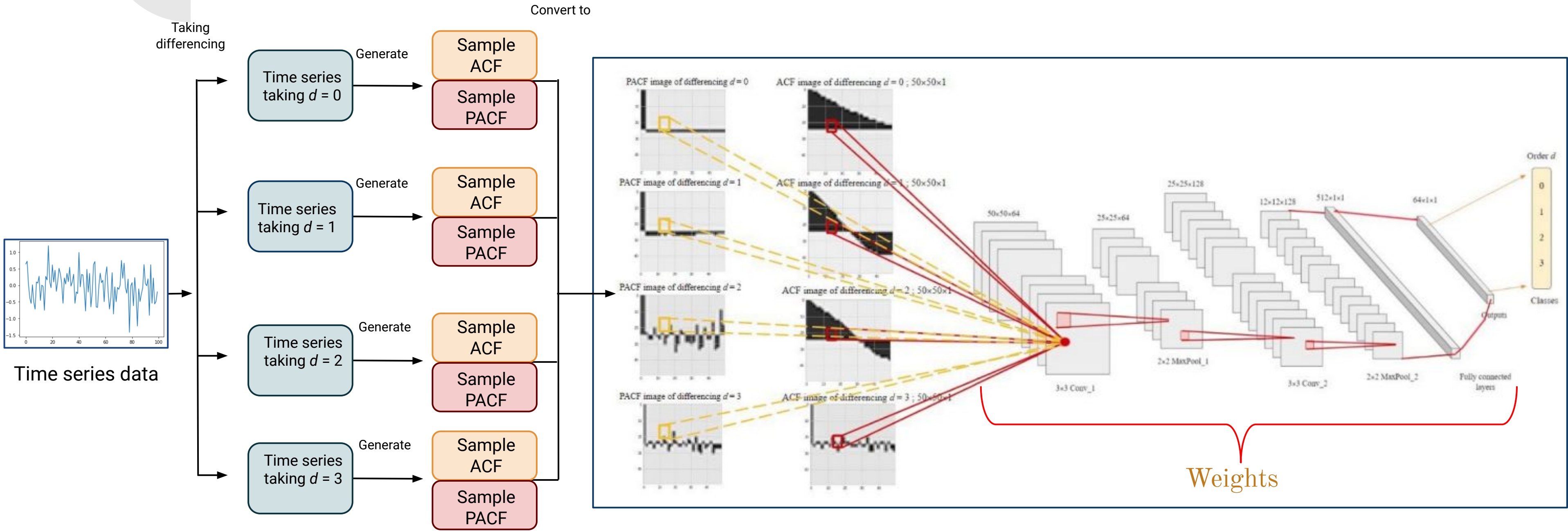


The architecture of the pq-APEA model



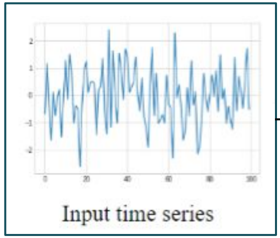
Number of parameters : 9,546,752

The architecture of the d-APEA model



Number of parameters : 9,518,276

The ESACF image

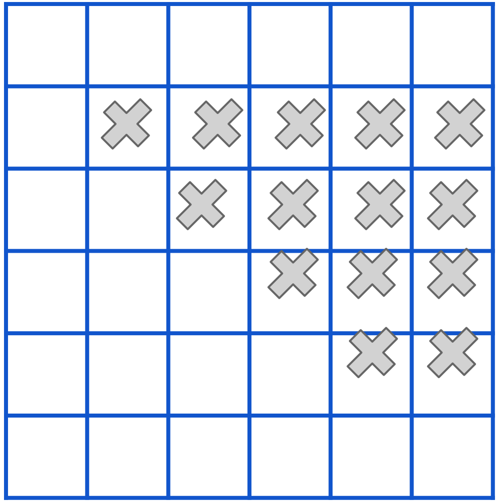


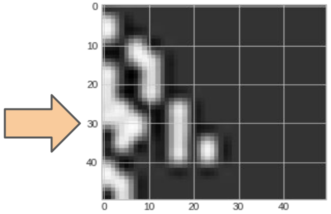
Calculate

ESACF table

AR/MA	1	2	3	4	5	6	...
0	X	X	X	X	X	X	...
1	X	O	O	O	O	O	...
2	X	X	O	O	O	O	...
3	X	X	X	O	O	O	...
4	X	X	X	X	O	O	...
5	X	X	X	X	X	O	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Generate





The example of generating the ESACF image

Visualization of the APEA model weights

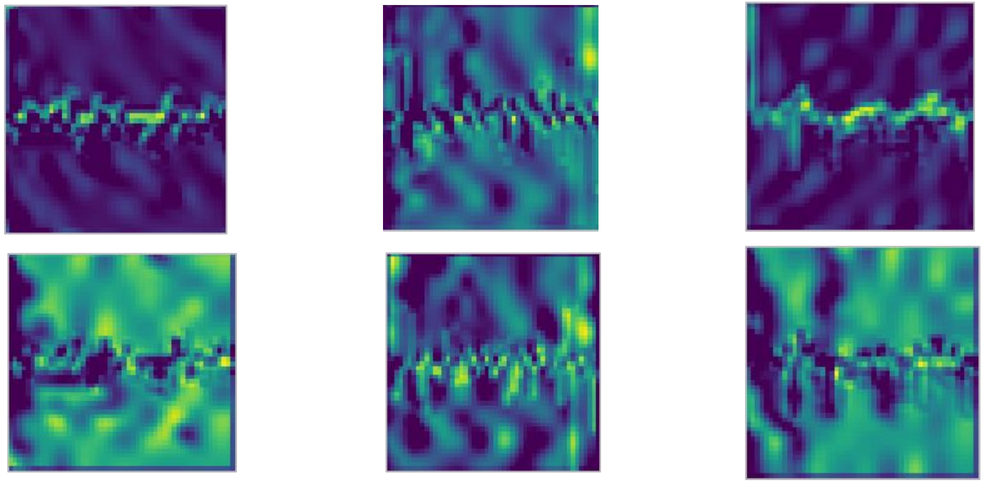
Minimum value

Maximum value

The color scales of “Viridis”

Conv 1

Conv 2



The filter matrices the AR order is 0

The filter matrices the AR order is 1

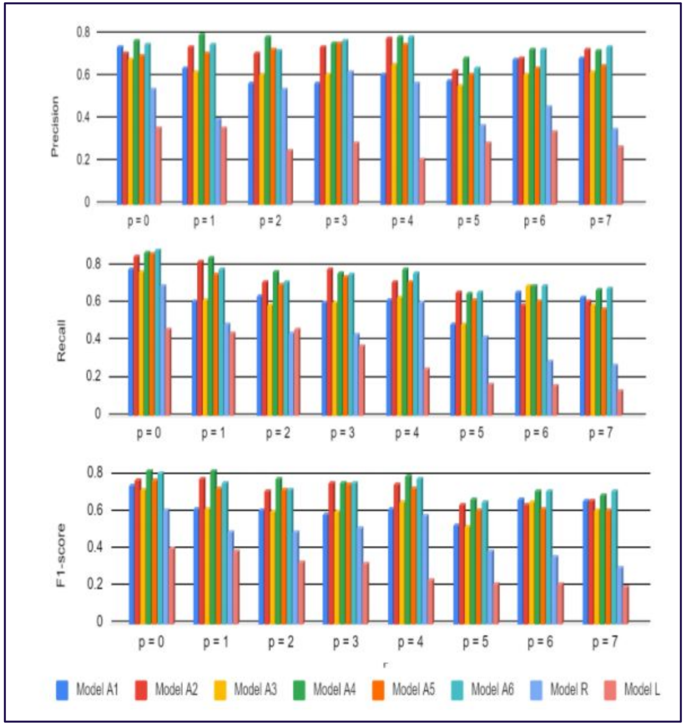
The filter matrices the AR order is 2

Description for different APEA models

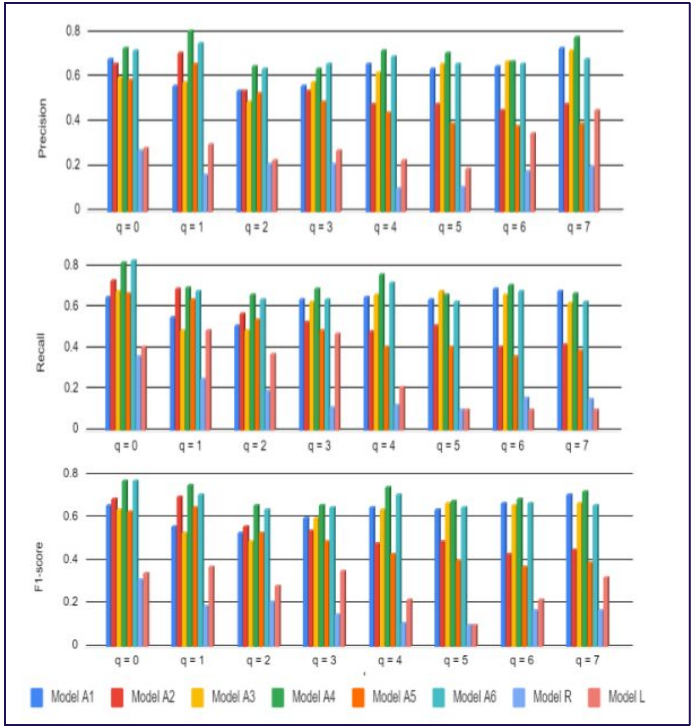
		Model A1	Model A2	Model A3	Model A4	Model A5	Model A5	Model R	Model L
Model		APEA	APEA	APEA	APEA	APEA	APEA	ResNet50	Likelihood (AIC criteria)
Case: identify p and q	Channel(s)	1 channel: ESACF images	2 channels: PACF and ACF images	2 channels: ESACF and differencing time series images	3 channels: ESACF, PACF and ACF images	3 channels: PACF, ACF and differencing time series images	4 channels: ESACF, PACF, ACF and differencing time series images	Series	Series
Case: identify d	Channel(s)	4 channels: ACF images with d =0,1,2,3	4 channels: PACF images with d =0,1,2,3	8 channels: PACF and ACF images with d =0,1,2,3	None	None	None	None	Series

Result of identifying p, d, q of the APEA model

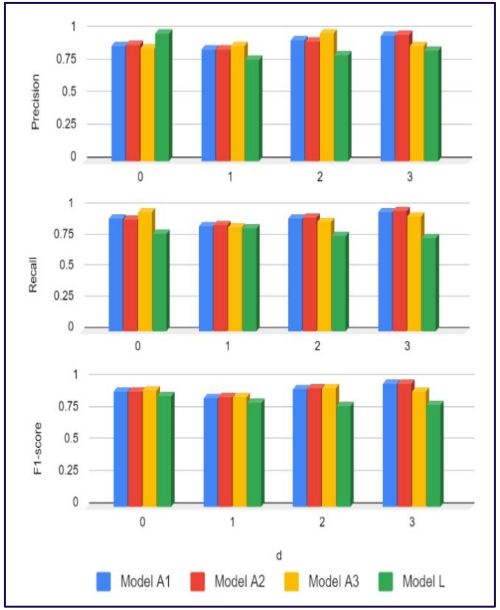
Identifying the p order



Identifying the q order



Identifying the d order



Methodology I

Build the self-identification deep learning model (SID)

- PACF, ACF, Time series images (w/o differencing)
- (p, d, q)-order

Build the self-identification ResNet-ARIMA model (SIRO)
* ResNet architecture

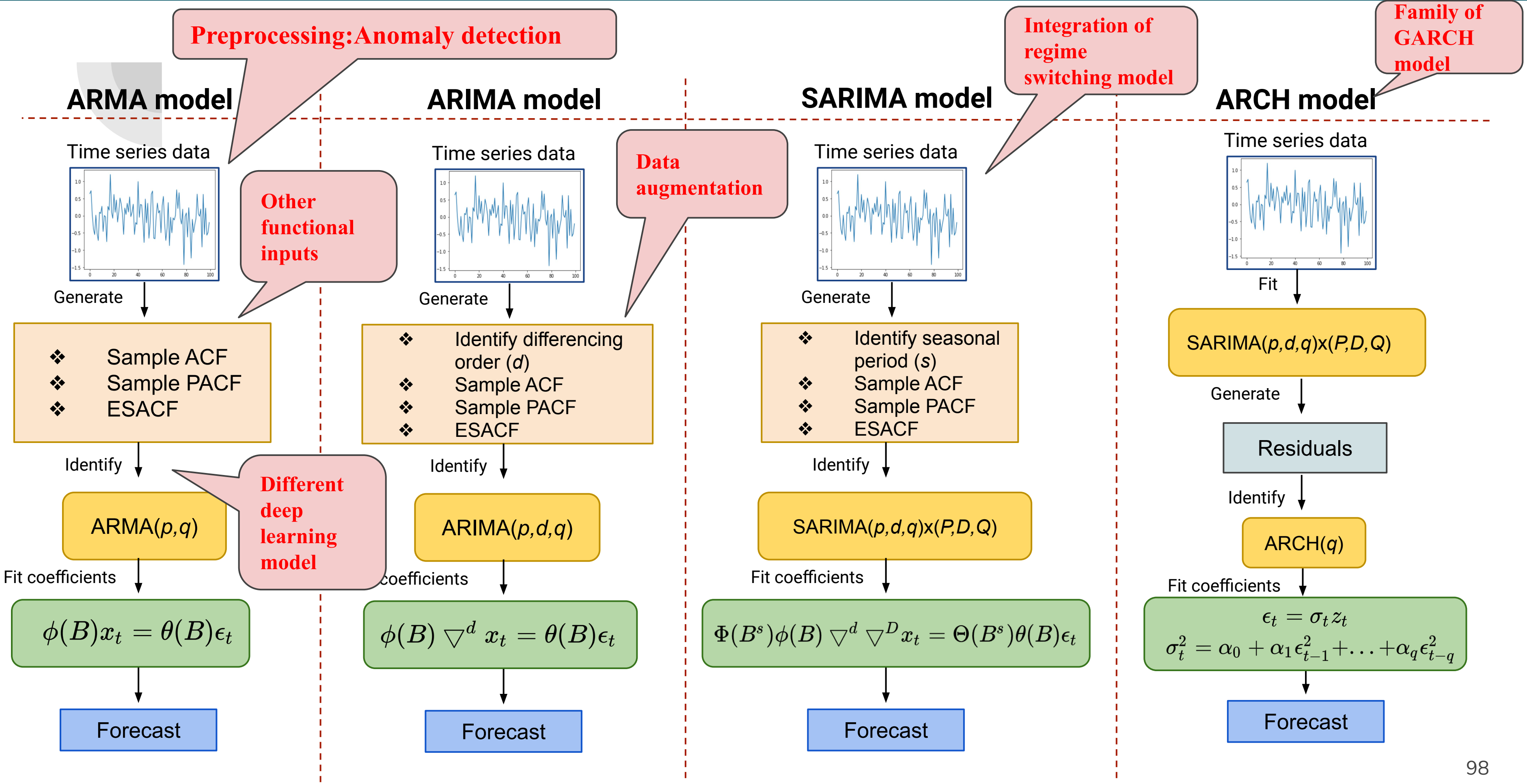
- PACF, ACF, Time series images (w/o differencing)
- (p, d, q)-order

Build the ACF-PACF-ESACF -convolutional neural network ARIMA order identification model (APEA)

- PACF, ACF, ESACF, Time series images (w/o differencing)
- (p, d, q)-order

Future works

Future work


Preprocessing:Anomaly detection
Integration of regime switching model
Family of GARCH model
ARMA model
ARIMA model
SARIMA model
ARCH model
Time series data
Time series data
Time series data
Time series data
Other functional inputs
Data augmentation
Generate
Generate
Generate
Fit
❖ Sample ACF
❖ Sample PACF
❖ ESACF
❖ Identify differencing order (d)
❖ Sample ACF
❖ Sample PACF
❖ ESACF
❖ Identify seasonal period (s)
❖ Sample ACF
❖ Sample PACF
❖ ESACF
SARIMA(p,d,q)x(P,D,Q)
Generate
Residuals
Identify
Identify
Identify
Identify
Different deep learning model
ARMA(p,q)
ARIMA(p,d,q)
SARIMA(p,d,q)x(P,D,Q)
ARCH(q)
Fit coefficients
coefficients
Fit coefficients
Fit coefficients
$\phi(B)x_t = \theta(B)\epsilon_t$
$\phi(B) \nabla^d x_t = \theta(B)\epsilon_t$
$\Phi(B^s)\phi(B) \nabla^d \nabla^D x_t = \Theta(B^s)\theta(B)\epsilon_t$
$\epsilon_t = \sigma_t z_t$
$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \ldots + \alpha_q \epsilon_{t-q}^2$
Forecast
Forecast
Forecast
Forecast

Publications

- Self-Identification Deep Learning Model, ***Journal of Physics: Conference Series***, Volume 1564(2020) 012004.

IOP Publishing

- Self-Identification ResNet-ARIMA Forecasting Model, ***WSEAS Transactions on Systems and Control***, ISSN / E-ISSN: 1991-8763 / 2224-2856, Volume 15, 2020, Art. #21, pp. 196-211.

WSEAS

WSEAS TRANSACTIONS ON SYSTEMS AND CONTROL

- Automatic SARIMA Order Identification Convolutional Neural Network, ***International Journal of Machine Learning and Computing***.

IJMLC ISSN 2010-3700

an Open Access Journal on Machine Learning and Computing

Questions?