**1)Explain ACF and PACF plots**

A correlogram (also called Auto Correlation Function ACF Plot or Autocorrelation plot) is a visual way to show serial correlation in data that changes over time (i.e. time series data). Serial correlation (also called autocorrelation) is where an error at one point in time travels to a subsequent point in time.

The PACF plot is a plot of the partial correlation coefficients between the series and lags of itself.

**2)Arima and Sarima, which to use when?**

**ARIMA** is an acronym for “autoregressive integrated moving average.” It's a model used in statistics and econometrics to measure events that happen over a period of time. The model is used to understand past data or predict future data in a series.

ARIMA models are applied in some cases where data show evidence of [non-stationarity](https://en.wikipedia.org/wiki/Stationary_process) in the sense of mean (but not variance/[autocovariance](https://en.wikipedia.org/wiki/Autocovariance)), where an initial differencing step (corresponding to the ["integrated"](https://en.wikipedia.org/wiki/Order_of_integration) part of the model) can be applied one or more times to eliminate the non-stationarity of the mean function (i.e., the trend).

**Seasonal Autoregressive Integrated Moving Average,** SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA.

It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

**3)How do you check the stationarity of the time series?**

The simplest way to check for stationarity is to split your total timeseries into 2, 4, or 10 (say N) sections (the more the better), and compute the mean and variance within each section. If there is an obvious trend in either the mean or variance over the N sections, then your series is not stationary.

**4)Hypothesis testing?**

The Hypothesis Testing is a statistical test used to determine whether the hypothesis assumed for the sample of data stands true for the entire population or not. Simply, the hypothesis is an assumption which is tested to determine the relationship between two data sets.

**The types of hypotheses testing:**

* Simple Hypothesis.
* Complex Hypothesis.
* Working or Research Hypothesis.
* Null Hypothesis.
* Alternative Hypothesis.
* Logical Hypothesis.
* Statistical Hypothesis.

**5)Data normalization and data standardization? And which one is prone to outliers?**

Normalization typically means rescales the values into a range of [0,1]. Standardization typically means rescales data to have a mean of 0 and a standard deviation of 1 (unit variance).

Normalizing the data is sensitive to outliers, so if there are outliers in the data set it is a bad practice. Standardization creates a new data not bounded (unlike normalization).

**6)How back propagation works?**

Back-propagation is just a way of propagating the total loss back into the neural network to know how much of the loss every node is responsible for, and subsequently updating the weights in such a way that minimizes the loss by giving the nodes with higher error rates lower weights and vice versa.

Backpropagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time.

**7)What is Vanishing and Exploding gradients?**

Vanishing: As the backpropagation algorithm advances downwards (or backward) from the output layer towards the input layer, the gradients often get smaller and smaller and approach zero which eventually leaves the weights of the initial or lower layers nearly unchanged. As a result, the gradient descent never converges to the optimum. This is known as the **vanishing gradients** problem.

Exploding: On the contrary, in some cases, the gradients keep on getting larger and larger as the backpropagation algorithm progresses. This, in turn, causes very large weight updates and causes the gradient descent to diverge. This is known as the ***exploding gradients*** problem.

**8)How do you overcome vanishing and Exploding gradients?**

Vanishing Gradients: The simplest solution is to use other activation functions, such as ReLU, which doesn't cause a small derivative. Residual networks are another solution, as they provide residual connections straight to earlier layers.

Exploding Gradients: A common solution to exploding gradients is to change the error derivative before propagating it backward through the network and using it to update the weights. By rescaling the error derivative, the updates to the weights will also be rescaled, dramatically decreasing the likelihood of an overflow or underflow.

**9)How do you initialise weights to NN? And explain Adagrad and Adam intializers?**

Neural network models are fit using an optimization algorithm called stochastic gradient descent that incrementally changes the network weights to minimize a loss function, hopefully resulting in a set of weights for the mode that is capable of making useful predictions.

This optimization algorithm requires a starting point in the space of possible weight values from which to begin the optimization process. Weight initialization is a procedure to set the weights of a neural network to small random values that define the starting point for the optimization (learning or training) of the neural network model.

Each time, a neural network is initialized with a different set of weights, resulting in a different starting point for the optimization process, and potentially resulting in a different final set of weights with different performance characteristics.

We cannot initialize all weights to the value 0.0 as the optimization algorithm results in some asymmetry in the error gradient to begin searching effectively.

**10)He-init and xavier Initialization differences?**

The main difference for machine learning practitioners is the following:

* He initialization works better for layers with ReLu activation.
* Xavier initialization works better for layers with sigmoid activation.

**11)Why CNN for images?**

CNNs are used for image classification and recognition because of its high accuracy. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed.

**12)How back propagation works in max pooling layer of cnn**?

For the backward in a max pool layer, we pass of the gradient, we start with a zero matrix and fill the max index of this matrix with the gradient from above. On the other hand, if we tread it as an average pool layer, we need to fill each cell with the value of the gradient from above.

**13)How do you train object detection model and deploy it?**

**Train Object Detection Model:**

1. Collect your datasets
2. Annotate the custom images using ‘labelImg’
3. Split them into train-test sets
4. Generate a TFRecord for the train-test split
5. Setup a config file
6. Train the actual model
7. Export the graph from the newly trained model
8. Bring in the frozen\_inference\_graph to classify in real-time

**14)How do you check the accuracy of ocr output?**

Measuring OCR accuracy is done by taking the output of an OCR run for an image and comparing it to the original version of the same text. You can then either count how many characters were detected correctly (character level accuracy), or count how many words were recognized correctly (word level accuracy).

**15)Brief on Transformers architecture and Attention models.**   
Transformers are a type of neural network architecture that have been gaining popularity. Transformers were recently used by OpenAI in their language [models](https://blog.openai.com/better-language-models/), and also used recently by DeepMind for [AlphaStar](https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/) — their program to defeat a top professional Starcraft player.

Transformers were developed to solve the problem of [**sequence transduction**](https://arxiv.org/abs/1211.3711)**,**or **neural machine translation.**That means any task that transforms an input sequence to an output sequence. This includes speech recognition, text-to-speech transformation, etc.

Attention models, or attention mechanisms, are input processing techniques for neural networks that allows the network to focus on specific aspects of a complex input, one at a time until the entire dataset is categorized. Attention models require continuous reinforcement or backpopagation training to be effective.

**16)Decorators and Iterators in python**

The python generators give an easy way of creating iterators. These generators instead of returning the function from the return statement use the **"yield”** keyword. These are the generator version of the list comprehensions.

If the function contains at least one “yield” statement, it becomes a generator function. Both the **yield** and **return**will return some value from the function.

we can implement decorators’ concept in two ways: Class decorators. Function decorators. Usually, a decorator is any callable object that is used to modify the function (or) the class.

**Iterators** are used mostly to iterate or convert other objects to an iterator using iter() function. Generators are mostly used in loops to generate an iterator by returning all the values in the loop without affecting the iteration of the loop. Iterator uses iter() and next() functions.

**17)Why can’t we use traditional machine learning algorithms for Time series?**

Time series forecasting is an important area of machine learning. It is important because there are so many prediction problems that involve a time component. However, while the time component adds additional information, it also makes time series problems more difficult to handle compared to many other prediction tasks. Time series data, as the name indicates, differ from other types of data in the sense that the temporal aspect is important. On a positive note, this gives us additional information that can be used when building our machine learning model — that not only the input features contain useful information, but also the changes in input/output over time.

Comparing the performance of all methods, it was found that the machine learning methods were all out-performed by simple classical methods, where **ETS and ARIMA models** performed the best overall. This finding confirms the results from previous similar studies and competitions.

**18)Why Vector Auto regression over LSTM’s?**

Vector autoregression (VAR) is a statistical model used to capture the relationship between multiple quantities as they change over time. VAR is a type of [stochastic process](https://en.wikipedia.org/wiki/Stochastic_process) model. VAR models generalize the single-variable (univariate) [autoregressive model](https://en.wikipedia.org/wiki/Autoregressive_model) by allowing for multivariate [time series](https://en.wikipedia.org/wiki/Time_series). VAR models are often used in [economics](https://en.wikipedia.org/wiki/Economics) and the [natural sciences](https://en.wikipedia.org/wiki/Natural_science).

VAR MODELING

With ARIMA we are using the past values of every variable to make the predictions for the future. When we have multiple time series at our disposal, we can also extract information from their relationships, in this way VAR is a multivariate generalization of ARIMA because it understands and uses the relationship between several inputs. This is useful for describing the dynamic behavior of the data and also provides better forecasting results.

To correctly develop a VAR model, the same classical assumptions encountered when fitting an ARIMA, have to be satisfied. We need to grant stationarity and leverage autocorrelation behaviors. These prerequisites enable us to develop a stable model. All our time series are stationary in mean and show a daily and weekly pattern.

# COMBINE VAR AND LSTM

Now our scope is to use our fitted VAR to improve the training of our neural network. The VAR has learned the internal behavior of our multivariate data source adjusting the insane values, correcting the anomalous trends, and reconstructing properly the NaNs.

Our strategy involves applying a two-step training procedure. We start feeding our LSTM autoencoder, using the fitted values produced by VAR, for multi-step ahead forecasts of all the series at our disposal (multivariate output). Then we conclude the training with the raw data, in our case they are the same data we used before to fit the VAR. With our neural network, we can also combine external data sources, for example, the weather conditions or some time attributes like weekdays, hours, and months that we cyclically encode.

We hope that our neural network can learn from two different but similar data sources and perform better on our test data. When performing multiple-step training we have to take care of the [**Catastrophic Forgetting**](https://arxiv.org/pdf/1312.6211.pdf)problem. Catastrophic forgetting is a problem faced by many models and algorithms. When trained on one task, then trained on a second task, many machine learning models “forget” how to perform the first task. This is widely believed to be a serious problem for neural networks.

To avoid this tedious problem, the structure of the entire network has to be properly tuned to provide a benefit in performance terms. From these observations, we preserve a final part of our previous training as validation.

Technically speaking the network is very simple. It’s constituted by a seq2seq LSTM autoencoder which predicts the available sensors N steps ahead in the future. The training procedure is carried out using [**keras-hypetune**](https://github.com/cerlymarco/keras-hypetune)**.**This framework provides **hyperparameter optimization** of the neural network structures in a very intuitive way. This is done for all three training involved (the fit on VAR fitted values, the fine-tuning fit with the raw data and the standard fit directly on the raw data)