Foundation of Cloud IoT Edge ML

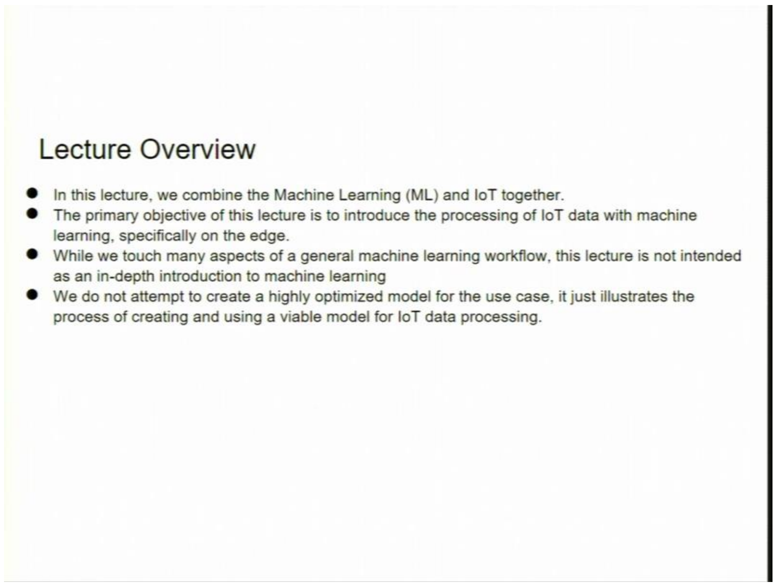
Professor Rajiv Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna

Lecture 8

ML based Predictive Maintenance at IoT Edge

I am Dr. Rajeev Mishra from IIT, Patna. The topic of today's lecture is ML based predictive maintenance at the IOT edge.

(Refer Slide Time: 0:26)



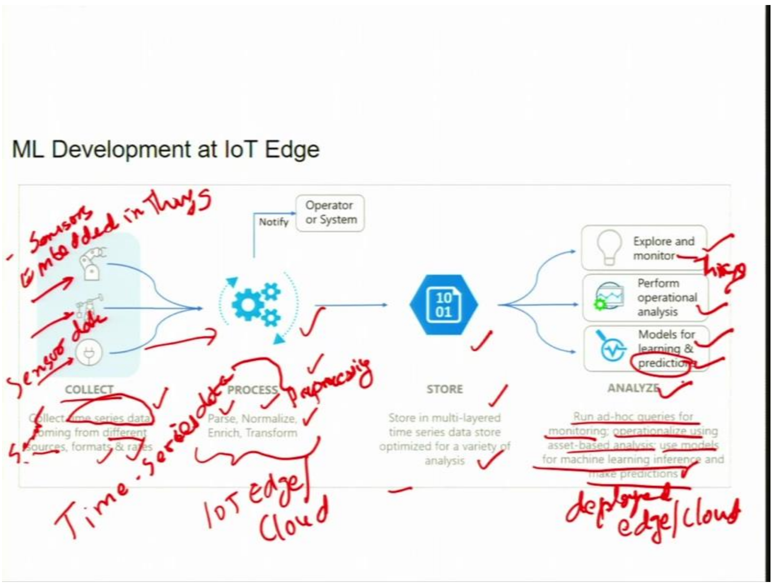
Lecture Overview

- In this lecture, we combine the Machine Learning (ML) and IoT together.
- The primary objective of this lecture is to introduce the processing of IoT data with machine learning, specifically on the edge.
- While we touch many aspects of a general machine learning workflow, this lecture is not intended as an in-depth introduction to machine learning
- We do not attempt to create a highly optimized model for the use case, it just illustrates the process of creating and using a viable model for IoT data processing.

In this lecture we are going to cover the following details listed over here as overview. So, in this overview we will now briefly talk about the machine learning and IOT, how they goes together. Then the prime objective of this lecture is to introduce the processing of an IOT sensor data with the machine learning and that specifically at the edge. While we touch many aspects of machine learning workflow in this lecture but this lecture is not intended as an in-depth understanding or introduction to the machine learning.

So we do not attempt to create highly optimized model for the use cases we use here in our lecture. So, it is just an illustration to represent how the machine learning is used for creating the predictive maintenance process for IOT and at IOT edge. That is the process of creating on using the viable model from machine learning for IOT data processing.

(Refer Slide Time: 1:39)


ML Development at IoT Edge

Notify
Operator or System
COLLECT
Collect time series data coming from different sources, formats & rates
PROCESS
Parse, Normalize, Enrich, Transform
STORE
Store in multi-layered time series data store optimized for a variety of analysis
ANALYZE
Explore and monitor
Perform operational analysis
Models for learning & predictions
Run ad-hoc queries for monitoring; operationalize using asset-based analysis; use models for machine learning inference and make predictions
Sensors embedded in things
Sensor data
Time-series data
preprocessing
IoT edge/ Cloud
things
deployed edge/cloud

So let us get started with looking at the machine learning development at an IOT edge workflow. So, that starts with the data collection. So, the sensors used to send the data of that particular environment which they are embedded into. That is, they will collect a time series data which is coming from various sources and formats and the rate. So, this is an example where these sensors are embedded into the things.

So, these sensors will collect the sensor data. Sensor data is collected and this particular data is in a particular form and that is called the time series data which is coming from different sources that is from different sensor sources. And often having a different formats and the different rate in which these sensors are sending the data.

So, this is called the collection and this collection has to now communicate through the telemetry or through data ingestion mechanism to the process and this particular process will may run either at the IOT edge or at the cloud. The next stage after this processing is doing the pre-processing of this incoming data and that is to parse, normalize, enrich and transform.

So, this particular process is a pre-processing of this sensor data which is also sometimes called as a time series data. So, sensor or a time series data which sensors use to send will put under the pre-processing stage, that is called the process here which is shown over here using various tools and techniques which are available at the cloud.

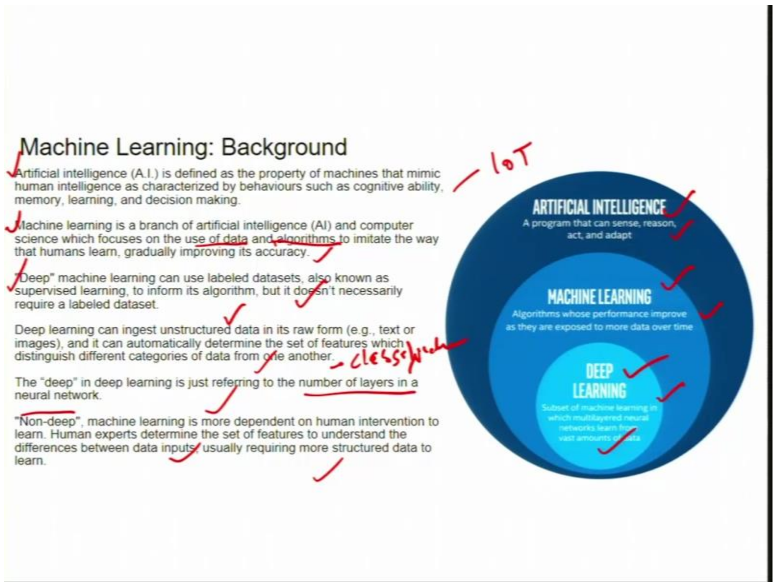
The next stage after this pre-processing is the storage. So, this particular storage will be either in the cloud or at the edge to a little size. So, the cloud has enormous level of storage and it uses various multi-layered time series data store which is optimized for variety of analysis.

So, there are many possible data stores which are being offered as a service by different cloud providers and the application developer has to choose one of these particular platform.

So, once the data is stored then this particular data will then be explored and put on the monitoring of these things first of all; second is, it may perform an operational analysis, that means using the dashboard you can see the progress how the machine is working. Third is using this particular data which is often put in the store, create the model using the machine learning algorithm and this is called the learning phase.

This particular learning model when they are deployed with and when they will see the new data incoming from the sensor they will be able to do the prediction. So, this is done in the analysis phase. So, analysis is just to run ad hoc queries for monitoring the environment or operationalize using asset based analysis or use the machine learning algorithm to train the model and this trained model often put or deployed in the cloud or at the edge to make the predictions. And these particular predictions are often used to make this business intelligence or the business logic.

(Refer Slide Time: 6:49)



Machine Learning: Background

Artificial intelligence (A.I.) is defined as the property of machines that mimic human intelligence as characterized by behaviours such as cognitive ability, memory, learning, and decision making. – IoT

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

"Deep" machine learning can use labeled datasets, also known as supervised learning, to inform its algorithm, but it doesn't necessarily require a labeled dataset.

Deep learning can ingest unstructured data in its raw form (e.g., text or images), and it can automatically determine the set of features which distinguish different categories of data from one another. – classification

The "deep" in deep learning is just referring to the number of layers in a neural network.

"Non-deep", machine learning is more dependent on human intervention to learn. Human experts determine the set of features to understand the differences between data inputs, usually requiring more structured data to learn.

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Now let us go ahead with discussing a little background of the machine learning. Nevertheless, we are not going to discuss everything about the machine learning; only a little background which is good enough to understand this particular topic. So, the machine learning with IOT reference we will talk about. So, starting with some little definitions. So, as you see that this broad area is often called as an artificial intelligence.

Artificial intelligence is defined as the property of a machines that mimic the human intelligence behavior as characterized by the cognitive ability, memory, learning and decision making. So, that means artificial intelligence will make this computer or the machine mimic the intelligence like human intelligence and will be specialized in the topics like learning, forecasting, predictions, decision making and other cognitive ability.

So, therefore we can understand this artificial intelligence is a program that can sense reason, act and adopt and very well often used in the IOT parlance. Now this particular artificial intelligence is supported by the machine learning. So, machine learning is a branch of artificial intelligence and a computer science which focuses on the use of the data. So, here we will refer to the IOT data.

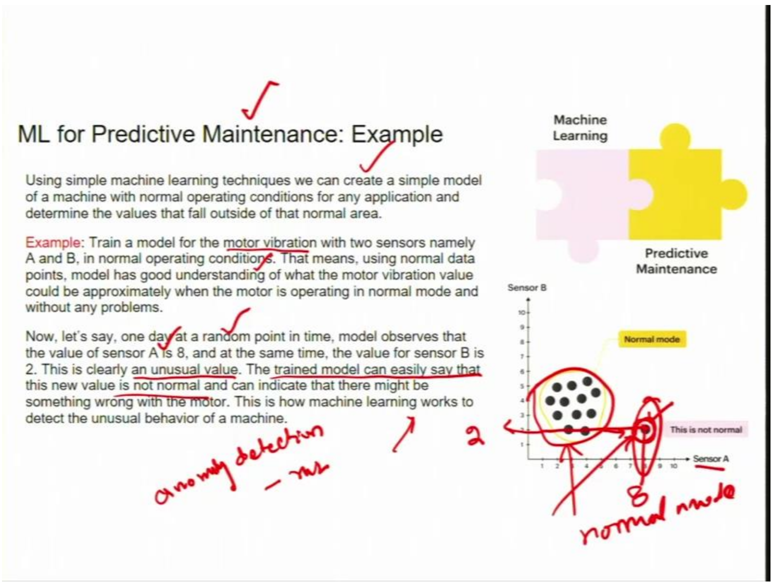
So, focuses on the use of data and the machine learning algorithm to imitate the way humans learn through the examples and gradually improves the accuracy of learning for making the predictions. So, therefore the machine learning algorithms whose performance improves as they are exposed to more data over the time.

So, this particular aspect since these IOT sensors are generating lot of data; as the time goes on, more data comes into, therefore it is a good opportunity to train the machine learning model with a good accuracy at all times. Now the next part that is the deep learning. So, deep learning uses the labelled data set also known as supervised learning is to inform its algorithm but it does not necessarily require the label data set sometimes.

So, deep learning can ingest unstructured data in its raw form and it automatically determines the set of features which distinguish different categories of the data from one and another. That is called classification. So, the deep in the deep learning is just referring to the number of layers in the neural network. So, non-deep machine learning is more dependent on human intervention and human experts determine the set of features to understand the differences between the data inputs.

So, the deep learning is a subset of a machine learning in which multi-layered neural network learn from vast amount of data. So, as you know that the humans also uses neurons and neural network here in deep learning it uses artificial neural networks for learning. So, once it sees the data, it learns through these artificial neural network. So, as we collect vast amount of data, therefore learning this model will bring more accuracy to the predictions.

(Refer Slide Time: 10:58)



ML for Predictive Maintenance: Example

Using simple machine learning techniques we can create a simple model of a machine with normal operating conditions for any application and determine the values that fall outside of that normal area.

Example: Train a model for the motor vibration with two sensors namely A and B, in normal operating conditions. That means, using normal data points, model has good understanding of what the motor vibration value could be approximately when the motor is operating in normal mode and without any problems.

Now, let's say, one day at a random point in time, model observes that the value of sensor A is 8, and at the same time, the value for sensor B is 2. This is clearly an unusual value. The trained model can easily say that this new value is not normal and can indicate that there might be something wrong with the motor. This is how machine learning works to detect the unusual behavior of a machine.

anomaly detection

– ML

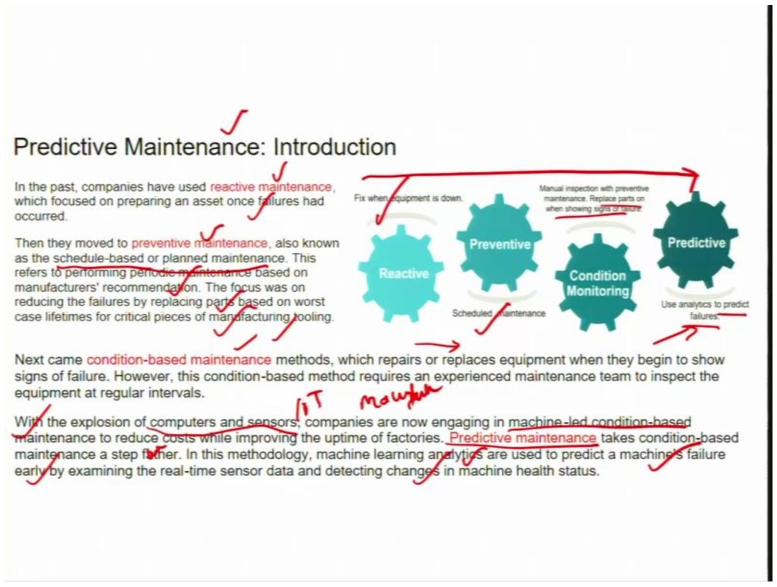
Now let us see the use of machine learning for predictive maintenance. So, predictive maintenance using simple machine learning algorithm we can create a simple model of a machine with a normal operating conditions for any application and determine the values that falls outside that normal area. So, for example if you can train a model for a motor which is capturing the vibration with two sensors, namely A and B in the normal working conditions.

That means that using normal data points the machine has a good understanding what the motor vibration value would be approximately when the motor is operating in a normal mode. Normal mode falls into this particular area. Now let us say one day at a random point in the time the model observes that the value of sensor A is 8 here.

That means it is away from the normal and the value of the sensor B is still 2. So, you can see that this particular value of a sensor B is 2 whereas for sensor A, this value becomes 8. So, this particular point which is drawn over here is not the normal operation. So, this is clearly an unusual value. So, the trained model can easily say that this new value is not normal and this is called an anomaly detection.

Often, it is very easy to do this anomaly detection with the help of machine learning and this is also one of the way you can do this predictive maintenance or you can predict when this abnormal behavior will be now coming and showing into that particular operation of the motor. So, this is how the machine learning works to detect an unusual behavior of a machine.

(Refer Slide Time: 13:16)



Predictive Maintenance: Introduction

In the past, companies have used reactive maintenance, which focused on preparing an asset once failures had occurred.

Then they moved to preventive maintenance, also known as the schedule-based or planned maintenance. This refers to performing periodic maintenance based on manufacturers' recommendation. The focus was on reducing the failures by replacing parts based on worst case lifetimes for critical pieces of manufacturing tooling.

Fix when equipment is down.
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Reactive
Preventive
Condition Monitoring
Predictive
Scheduled maintenance
Use analytics to predict failures.

Next came condition-based maintenance methods, which repairs or replaces equipment when they begin to show signs of failure. However, this condition-based method requires an experienced maintenance team to inspect the equipment at regular intervals.

IT [illegible]

With the explosion of computers and sensors, companies are now engaging in machine-led condition-based maintenance to reduce costs while improving the uptime of factories. Predictive maintenance takes condition-based maintenance a step further. In this methodology, machine learning analytics are used to predict a machine's failure early by examining the real-time sensor data and detecting changes in machine health status.

So, let us go ahead in talking about more details about the predictive maintenance. So, predictive maintenance has now completed its journey starting with the reactive maintenance. So, let us introduce the predictive maintenance with the full pipeline with the full growth to this stage which is called the predictive maintenance stage which is the most advanced predictive maintenance for industry 4.0.

So, in the past the companies used the reactive maintenance. So, reactive maintenance when the equipment is down this is focusing on preparing the assets once the failures have occurred. Now then, this will stop functioning of the machine in the factory environment in the reactive maintenance than they thought of improving this downtime than using the preventive maintenance.

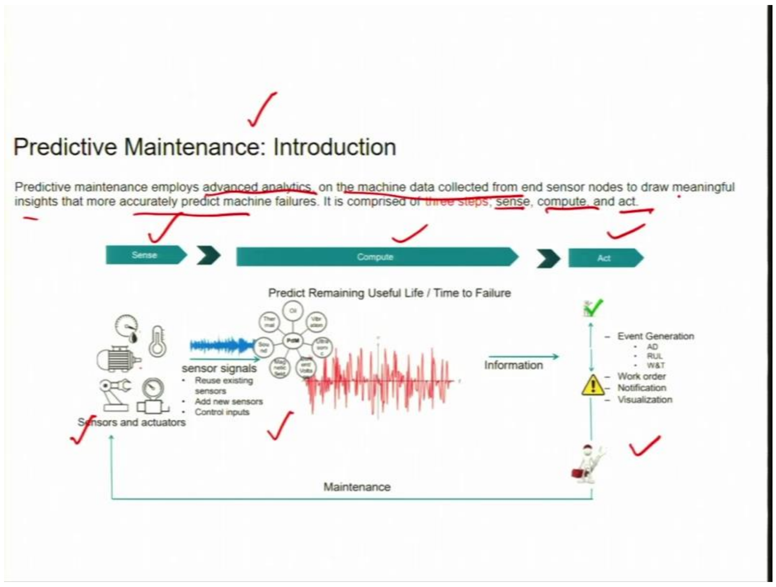
Preventive maintenance is also called the schedule based or a planned maintenance. So, schedule based or a planned maintenance refers to performing the periodic maintenance based on the manufacturer's recommendation and the focus was on reducing the failures by replacing the parts based on the worst case lifetime for critical pieces of manufacturing tooling. So, this preventive maintenance is better than reactive in terms of improving the downtime of the machine but still it is not sufficient to reduce the downtime to a zero.

Now then comes the condition based monitoring or a maintenance. So, here manually inspect with preventive maintenance and replace the parts when showing the sign of the failure. Now with this explosion of computers and sensors now these companies and IOT, these companies or the industries or manufacturing companies are now engaged in machine learning led the condition based maintenance to reduce this cost, while improving the uptime of this particular factory and there comes the predictive maintenance.

So, predictive maintenance takes the condition based maintenance to a further level and in this methodology machine learning analytics is used to predict the machines' failure early by examining the real time sensor data and detecting the changes in the machine health status. So, it often uses the sensor data which is nothing but a time series data and will predict when the machine is going to fail.

So, that is why this is called the predictive maintenance it will do the analytics to find out or to predict the failures of the machine and therefore before it fails the maintenance is done and therefore there is a very little downtime and now the companies are improving with the technologies like IOT.

(Refer Slide Time: 16:26)

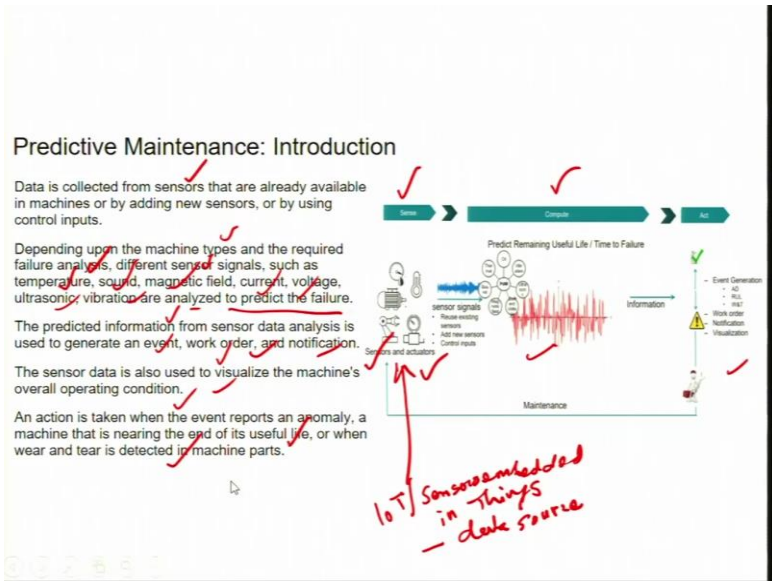


Predictive Maintenance: Introduction

Predictive maintenance employs advanced analytics, on the machine data collected from end sensor nodes to draw meaningful insights that more accurately predict machine failures. It is comprised of three steps; sense, compute, and act.

So, let us introduce this predictive maintenance in more details. So, predictive maintenance implies the advanced analytics on the machine data which is collected from the end sensor nodes to draw the meaningful insights and more accurate predict the machine failures beforehand. It comprises of three steps - one is to sense, the second is to do the computation and third is to perform an act – sense, compute and act. Let us understand these three different things in more details.

(Refer Slide Time: 17:04)



Predictive Maintenance: Introduction

Data is collected from sensors that are already available in machines or by adding new sensors, or by using control inputs.

Depending upon the machine types and the required failure analysis, different sensor signals, such as temperature, sound, magnetic field, current, voltage, ultrasonic, vibration are analyzed to predict the failure.

The predicted information from sensor data analysis is used to generate an event, work order, and notification.

The sensor data is also used to visualize the machine's overall operating condition.

An action is taken when the event reports an anomaly, a machine that is nearing the end of its useful life, or when wear and tear is detected in machine parts.

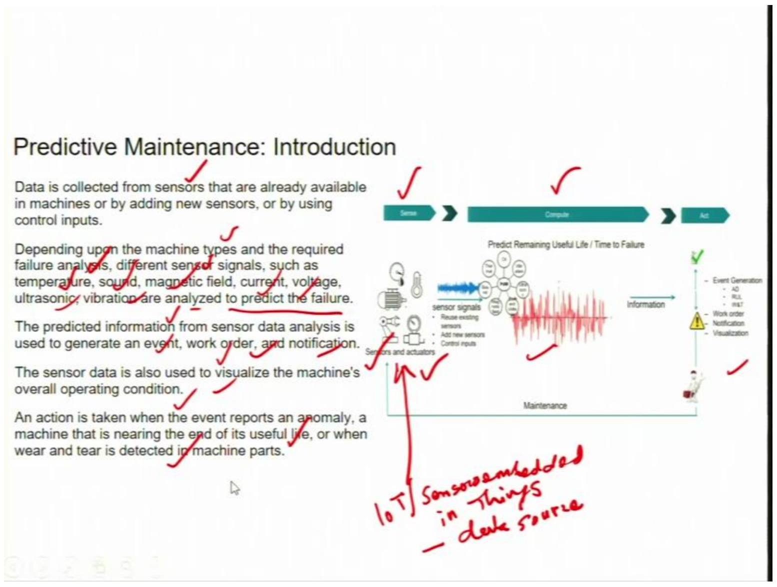
So, here the data from the sensors like here the sensor are embedded into the things. So, we are talking about IOT, that is the sensors embedded in the things and thus they become the data source. So, these particular sensors will generate the data. So, depending upon the machine types and required the failure analysis different sensors are used.

So, different sensor signals such as the temperature sensor or a sound, magnetic field, current, voltage, ultrasonic, vibration, these sensor data are now analyzed to predict the failures. So, you can see here these sensor often send the data and then these are often used to do this computation and predict the time to fail.

So, the predicted information from the sensor data analysis is used to generate an event work order and notification, so the sensor data is also used to visualize the machines overall operating conditions. So, that is an action. So, action is taken when the event reports an anomaly, a machine that is nearing the end of its useful life or when wear and tear is detected in the machine part.

Therefore, what we have told you here about the introduction of a preventive maintenance is that the sensors will monitor the machines, its parameters such as temperature, sound, magnetic field, current, voltage, ultrasonic, vibrations etc, and send this particular data for performing the computations. So, this particular sensor data is often used to visualize the machines' overall operating conditions. So, the action is taken when the event reports an anomaly a machine is then nearing the end of its useful life.

(Refer Slide Time: 19:24)



Predictive Maintenance: Introduction

Data is collected from sensors that are already available in machines or by adding new sensors, or by using control inputs.

Depending upon the machine types and the required failure analysis, different sensor signals, such as temperature, sound, magnetic field, current, voltage, ultrasonic, vibration are analyzed to predict the failure.

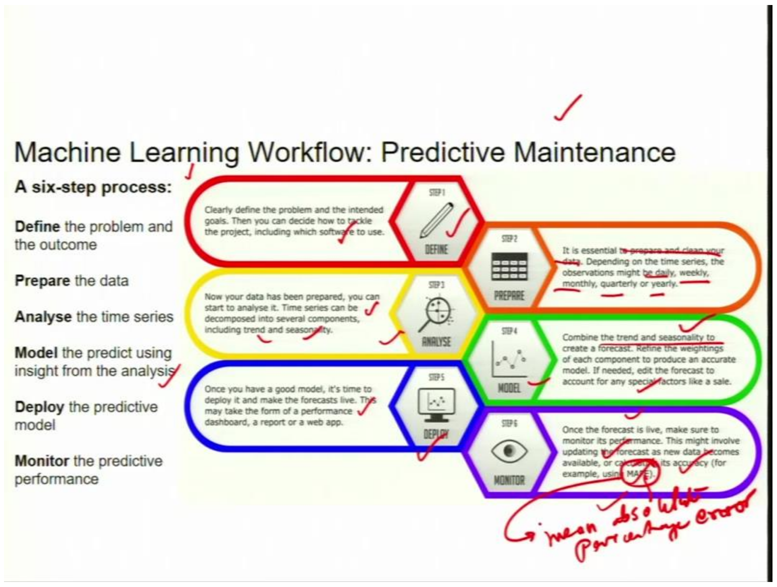
The predicted information from sensor data analysis is used to generate an event, work order, and notification.

The sensor data is also used to visualize the machine's overall operating condition.

An action is taken when the event reports an anomaly, a machine that is nearing the end of its useful life, or when wear and tear is detected in machine parts.

Predictive maintenance problems. So, let us see what are the issues which are addressed in the predictive maintenance. Will this equipment fail in a given period of time? What is the remaining useful life or a time of failure of a machine? How to quantify the wear and tear of expandable components? So, this is a subset of remaining useful life and focuses on the shorter living subsystems. For detecting anomalies in equipment behavior with further analysis it can improve the failure classification and also optimize the equipment setting.

(Refer Slide Time: 20:09)



Machine Learning Workflow: Predictive Maintenance

A six-step process:

Define the problem and the outcome

Prepare the data

Analyse the time series

Model the predict using insight from the analysis

Deploy the predictive model

Monitor the predictive performance

STEP 1 DEFINE: Clearly define the problem and the intended goals. Then you can decide how to tackle the project, including which software to use.

STEP 2 PREPARE: It is essential to prepare and clean your data. Depending on the time series, the observations might be daily, weekly, monthly, quarterly or yearly.

STEP 3 ANALYSE: Now your data has been prepared, you can start to analyse it. Time series can be decomposed into several components, including trend and seasonality.

STEP 4 MODEL: Combine the trend and seasonality to create a forecast. Refine the weightings of each component to produce an accurate model. If needed, edit the forecast to account for any special factors like a sale.

STEP 5 DEPLOY: Once you have a good model, it's time to deploy it and make the forecasts live. This may take the form of a performance dashboard, a report or a web app.

STEP 6 MONITOR: Once the forecast is live, make sure to monitor its performance. This might involve updating the forecast as new data becomes available, or calculating its accuracy (for example, using MAPE).

So, let us see the machine learning workflow in the predictive maintenance. There are six steps is we have already some of them we have told. The first is to define is that clearly define the problem and intended goals, then you decide how to tackle this particular problem.

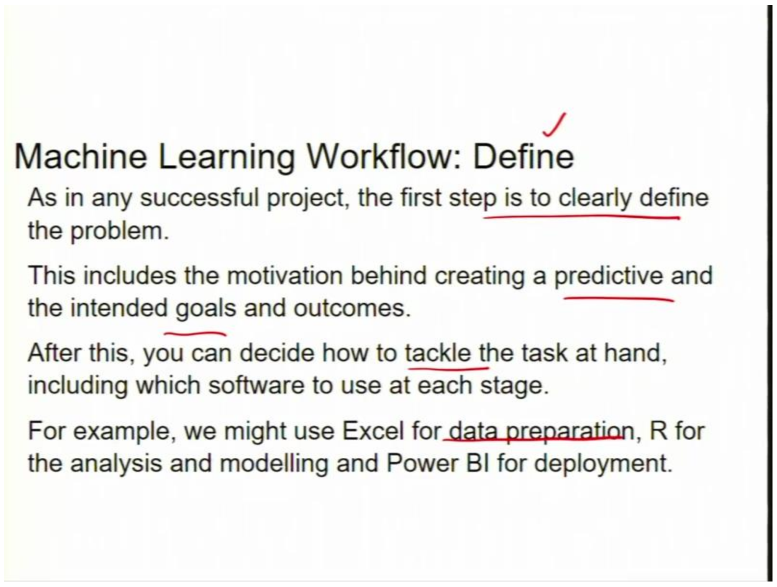
Second step is to prepare the essential thing here in the predictive maintenance under preparation is to prepare and clean your data, that is pre-processing.

Depending upon the time series, the observations may be a daily, weekly, monthly, quarterly or yearly which required to be classified and prepared in this time series data. Third is to analyze that is now your data has been prepared, you can start doing the analysis. Time series data can be decomposed into the several component including the trend and seasonality. Fourth is the model. So, model is to predict using the insight from the analysis.

So, therefore the step four of this generating or training the model is to combine the trend and seasonality to create a forecast. Refine the weightings of each component to produce an accurate model; if needed, edit the forecast to account for any special factors like sale and so on. Fifth factor is about deploy. So, once you have a good model, it is time to deploy it and make the forecast live.

This may take the form of a performance dashboard report or a web. Now finally the stage of the preventive predictive maintenance is called monitor. So, once the forecast is live, make sure that monitor its performance. This might involve updating the forecast as the new data comes and available and calculates its accuracy. For example MAPE, that is mean absolute percentage error.

(Refer Slide Time: 22:32)



Machine Learning Workflow: Define

As in any successful project, the first step is to clearly define the problem.

This includes the motivation behind creating a predictive and the intended goals and outcomes.

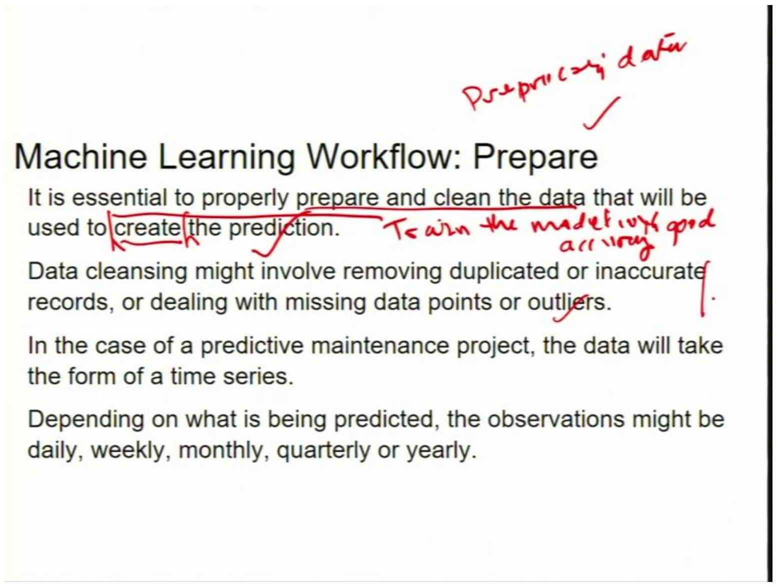
After this, you can decide how to tackle the task at hand, including which software to use at each stage.

For example, we might use Excel for data preparation, R for the analysis and modelling and Power BI for deployment.

So, with this let us go ahead talking about the machine learning workflow and we will see about the different aspects. So, as we have already stated that in the machine learning workflow, the important or the first step is to clearly define the problem what are you going to solve with the help of machine learning prediction models.

So, this will create the predictive and the intended goals and the outcomes; after that you can decide how to tackle the task at hand and then you can use the different tools for data preparation and then use this for analysis and modeling for the deployment purpose.

(Refer Slide Time: 23:19)



Preprocessing data ✓

Machine Learning Workflow: Prepare

It is essential to properly prepare and clean the data that will be used to create the prediction. To train the model with good accuracy

Data cleansing might involve removing duplicated or inaccurate records, or dealing with missing data points or outliers.

In the case of a predictive maintenance project, the data will take the form of a time series.

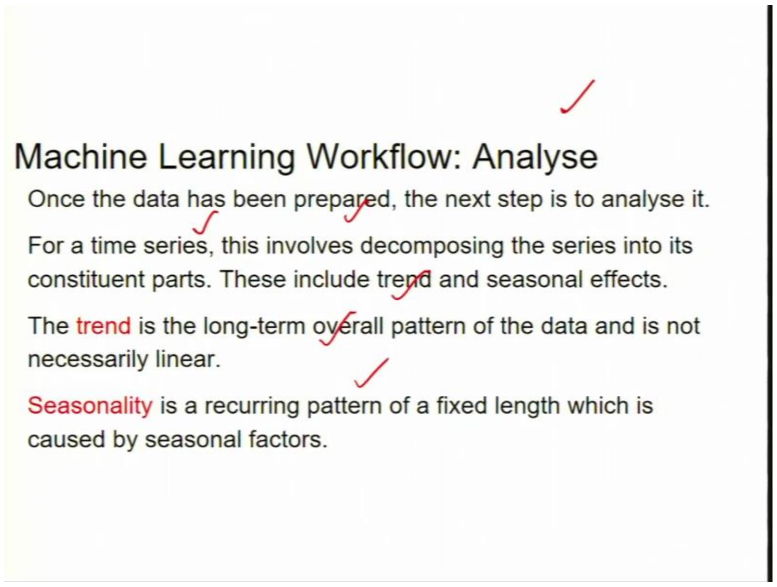
Depending on what is being predicted, the observations might be daily, weekly, monthly, quarterly or yearly.

Now in the prepare stage, that is the data preparation or a pre-processing the data, it is essentially to properly prepare and clean the data that will be used to train the model with good accuracy and then use it for making the predictions. So, this is very essential step of cleaning or preparing the data set so that the machine learning model can be trained with a good accuracy and it will be put for its predictions.

The next step is here into the data preparation is to data cleansing might involve removing some of the duplicate or noise or inaccurate and dealing with the missing data or sometimes the outliers, outline data to make this data… making it work with training the model accurately. So, in case of predictive maintenance the data will be in the form of a time series data, so all this preparation has to happen with the time series data.

Now depending upon what is being predicted, the observations might be categorized into daily, weekly, monthly, quarterly and yearly intervals.

(Refer Slide Time: 25:00)



Machine Learning Workflow: Analyse

Once the data has been prepared, the next step is to analyse it.

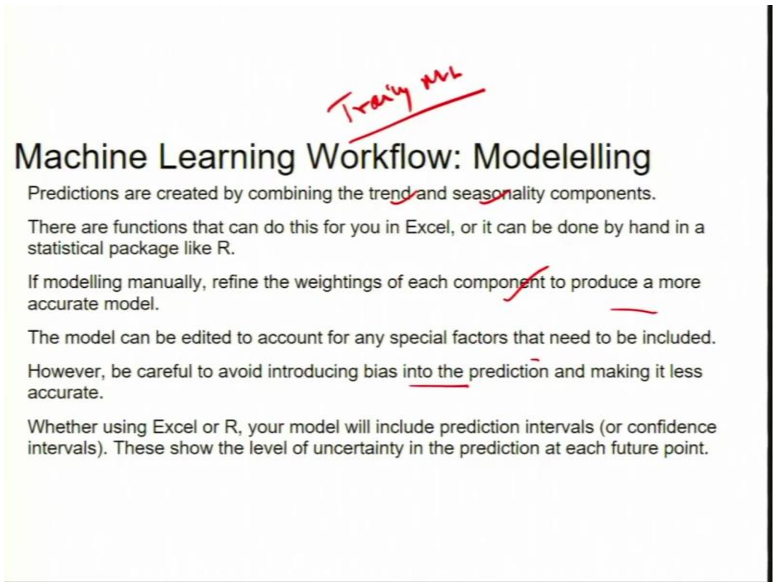
For a time series, this involves decomposing the series into its constituent parts. These include trend and seasonal effects.

The trend is the long-term overall pattern of the data and is not necessarily linear.

Seasonality is a recurring pattern of a fixed length which is caused by seasonal factors.

Now machine learning workflow in the analysis part once the data has been prepared the next is to analyze. Now, here for time series data this involves decomposing the series into the constituent part and they include the trend and seasonal effects. So, trend is a long-term overall pattern of the data and is not necessarily linear. Seasonality is a recurring pattern of a fixed length which is caused by seasonal factors. So, these two things are very important in the analysis.

(Refer Slide Time: 25:34)



Traing ML

Machine Learning Workflow: Modelelling

Predictions are created by combining the trend and seasonality components.

There are functions that can do this for you in Excel, or it can be done by hand in a statistical package like R.

If modelling manually, refine the weightings of each component to produce a more accurate model.

The model can be edited to account for any special factors that need to be included.

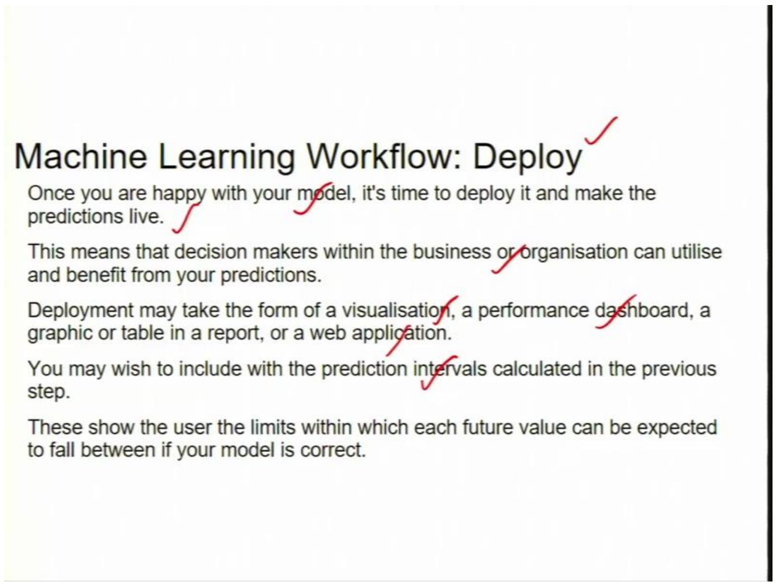
However, be careful to avoid introducing bias into the prediction and making it less accurate.

Whether using Excel or R, your model will include prediction intervals (or confidence intervals). These show the level of uncertainty in the prediction at each future point.

Modeling or the training the machine learning model is called modeling. Now to make the predictions when the model is created by combining trend and seasonality components you are having various softwares and these particular softwares will now manually refine the models to have this more accurate model.

So, different performance matrix we will see about that how to judge the accuracy of the model. So, models can be edited to account for any special factors, however careful introducing the biases into the predictions make it less accurate. So, we have to now be careful while deciding about the model.

(Refer Slide Time: 26:26)



Machine Learning Workflow: Deploy

Once you are happy with your model, it's time to deploy it and make the predictions live.

This means that decision makers within the business or organisation can utilise and benefit from your predictions.

Deployment may take the form of a visualisation, a performance dashboard, a graphic or table in a report, or a web application.

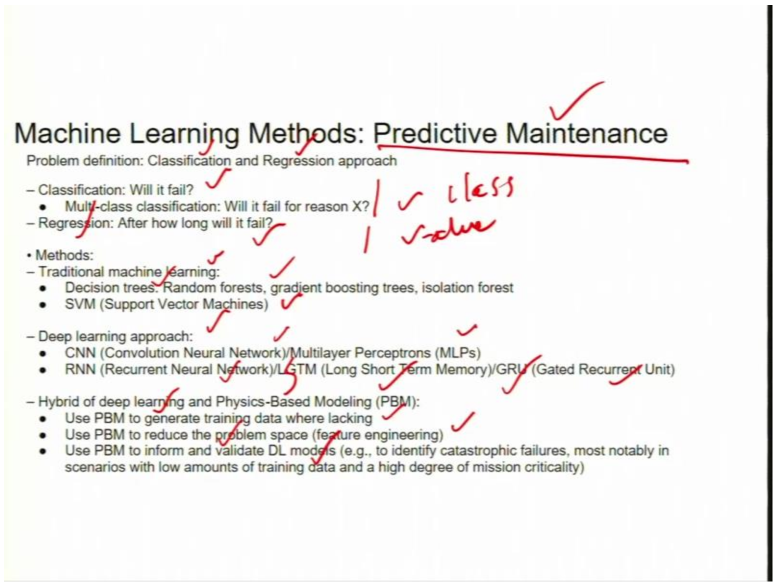
You may wish to include with the prediction intervals calculated in the previous step.

These show the user the limits within which each future value can be expected to fall between if your model is correct.

Now the next stage is called deploy the machine learning. Once the model is working right and acceptable accuracy, then it can be deployed and make it live for making the predictions in the real time. This means that the decision makers with the business logic can utilize and benefit out of these particular predictions.

Sometimes a visualization may also be helpful and it will be done through the performance dashboards or graphic or a table or a report or a web application. So, you may also include the prediction intervals as we have stated in the previous slide and expected this particular limits is expected in the future values to fall within between if the model is correct.

(Refer Slide Time: 27:25)



Machine Learning Methods: Predictive Maintenance

Problem definition: Classification and Regression approach

– Classification: Will it fail?
  - Multi-class classification: Will it fail for reason X? | ✓ class
– Regression: After how long will it fail? | ✓ value

• Methods:
– Traditional machine learning:
  - Decision trees: Random forests, gradient boosting trees, isolation forest
  - SVM (Support Vector Machines)

– Deep learning approach:
  - CNN (Convolution Neural Network)/Multilayer Perceptrons (MLPs)
  - RNN (Recurrent Neural Network)/LSTM (Long Short Term Memory)/GRU (Gated Recurrent Unit)

– Hybrid of deep learning and Physics-Based Modeling (PBM):
  - Use PBM to generate training data where lacking
  - Use PBM to reduce the problem space (feature engineering)
  - Use PBM to inform and validate DL models (e.g., to identify catastrophic failures, most notably in scenarios with low amounts of training data and a high degree of mission criticality)

So, monitor is the next step in the machine learning models. Once the prediction goes live, it is important to monitor its performance and therefore various accuracies and error mechanisms are there to monitor the model performance. So, the popular measures are mean absolute percentage error and the mean absolute deviation. We are going to look into these aspects more clearly.

So, a data scientist, they are specializing in this kind of technology, so using error they will be able to monitor the performance of the model. So, therefore machine learning methods for predictive maintenance from the machine learning point of view, they are going to solve two different problems - one is called classification, the other is called regression problem to solve this predictive maintenance. So, classification means will it fail or not?

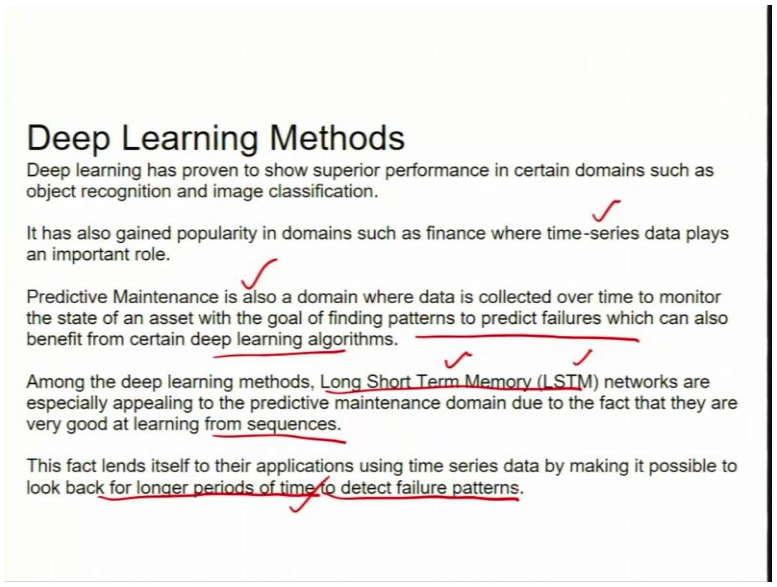
So, it will classify into two classes. Sometimes multi-class classification also, will it fail for a particular region X? This is all classification where yes or no or multiple classes are there, that is called a classification in the predictive maintenance. Regression aspects or a regression problem in the predictive maintenance is about after how long that machine or the failure will happen, so this will be a predicted value, not the class.

So, classification has to predict about the class where the regression has to predict about a value. Now to do this classification and regression related problems in predictive maintenance, you require different traditional machine learning approaches such as decision tree, random forest, gradient boosting trees, isolated forest, support vector machines are some of the traditional machine learning algorithms.

Similarly for deep learning, different approaches are there, that is called convolutional neural network - CNN or multi-layer perceptrons – MLPs, recurrent neural networks and LSTM - long short term memory and GRUs, that is gated recurrent unit. Now there are hybrid of deep learning and a physics based modeling.

So, you can use the physics based modeling to generate the training data set where the data is lacking. Similarly, you can also physics based modeling to reduce the problem space, that is you can use for feature engineering, you can use physics based modeling to inform and validate different deep learning models and so on. So, we have now covered the entire predictive maintenance from the machine learning aspects.

(Refer Slide Time: 30:25)



Deep Learning Methods

Deep learning has proven to show superior performance in certain domains such as object recognition and image classification.

It has also gained popularity in domains such as finance where time-series data plays an important role.

Predictive Maintenance is also a domain where data is collected over time to monitor the state of an asset with the goal of finding patterns to predict failures which can also benefit from certain deep learning algorithms.

Among the deep learning methods, Long Short Term Memory (LSTM) networks are especially appealing to the predictive maintenance domain due to the fact that they are very good at learning from sequences.

This fact lends itself to their applications using time series data by making it possible to look back for longer periods of time to detect failure patterns.

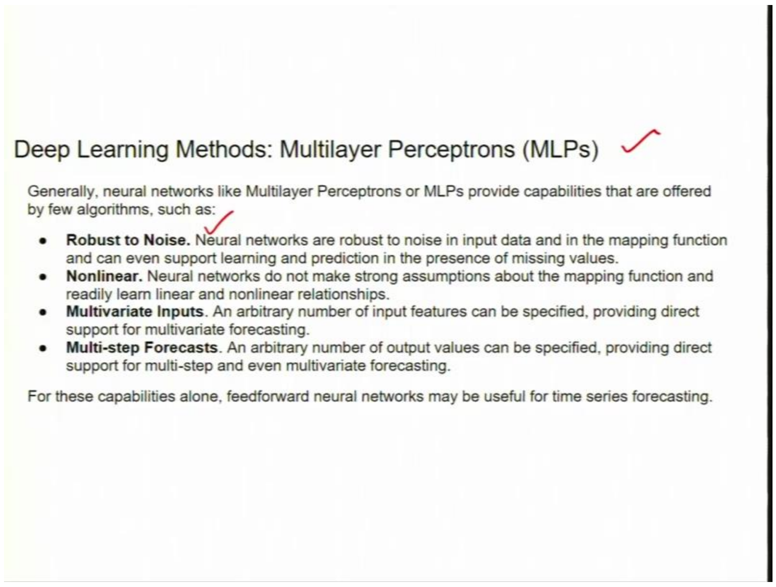
So, let us go ahead and understanding so called deep learning methods which is very much needed in the predictive maintenance. So, deep learning has proven to show superior performance in certain domains such as object recognition and image classification, nevertheless using time series data also it has shown a very improved models and often very popular in the time series data analysis also.

Now, predictive maintenance is also a domain where data is collected over the time to monitor the state of the assets with the goal of finding pattern to predict the failures. For this, you know that the deep learning algorithms are very powerful and capable to function this kind of problems in the predictive maintenance.

So, among the deep learning the popular models to do this time series predictions are long short term memory, that is LSTM networks or the deep learning models which especially appeal the to the predictive maintenance due to the fact they are very good in learning from the sequences and time series data is nothing but a sequence of data over the time.

So, this fact lends itself to their application using the time series data by making it possible to look back for a longer period of time to detect the failures. So, therefore this LSTM allows you to monitor over a longer period of time to detect the failure patterns and therefore by detecting the anomalies over the period of observations.

(Refer Slide Time: 32:18)



Deep Learning Methods: Multilayer Perceptrons (MLPs) ✓

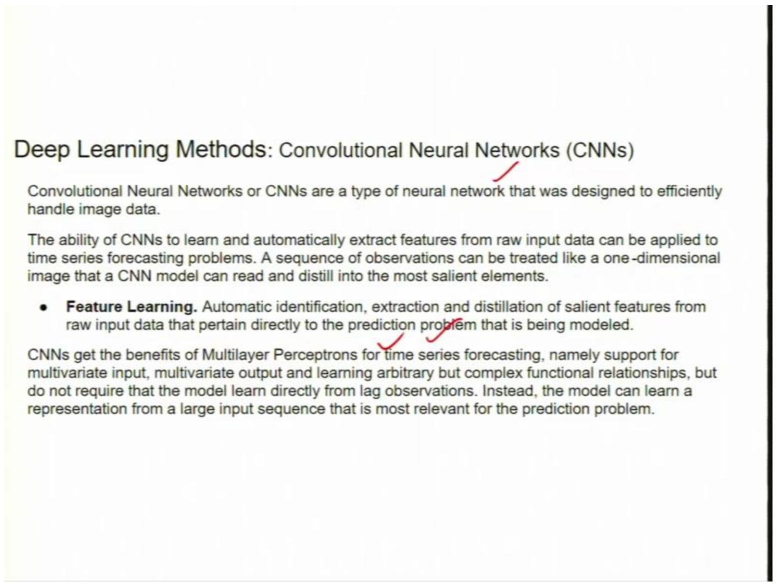
Generally, neural networks like Multilayer Perceptrons or MLPs provide capabilities that are offered by few algorithms, such as: ✓

- **Robust to Noise.** Neural networks are robust to noise in input data and in the mapping function and can even support learning and prediction in the presence of missing values.
- **Nonlinear.** Neural networks do not make strong assumptions about the mapping function and readily learn linear and nonlinear relationships.
- **Multivariate Inputs**. An arbitrary number of input features can be specified, providing direct support for multivariate forecasting.
- **Multi-step Forecasts**. An arbitrary number of output values can be specified, providing direct support for multi-step and even multivariate forecasting.

For these capabilities alone, feedforward neural networks may be useful for time series forecasting.

Now the deep learning method another is called multi-layer perceptrons. So, there the neural networks like multi-layer perceptrons provide the capabilities which are offered by the few algorithms such as robust to the noise, nonlinear, multi-variate inputs, multi-step forecast.

(Refer Slide Time: 32:38)



Deep Learning Methods: Convolutional Neural Networks (CNNs)

Convolutional Neural Networks or CNNs are a type of neural network that was designed to efficiently handle image data.

The ability of CNNs to learn and automatically extract features from raw input data can be applied to time series forecasting problems. A sequence of observations can be treated like a one-dimensional image that a CNN model can read and distill into the most salient elements.

- **Feature Learning.** Automatic identification, extraction and distillation of salient features from raw input data that pertain directly to the prediction problem that is being modeled.

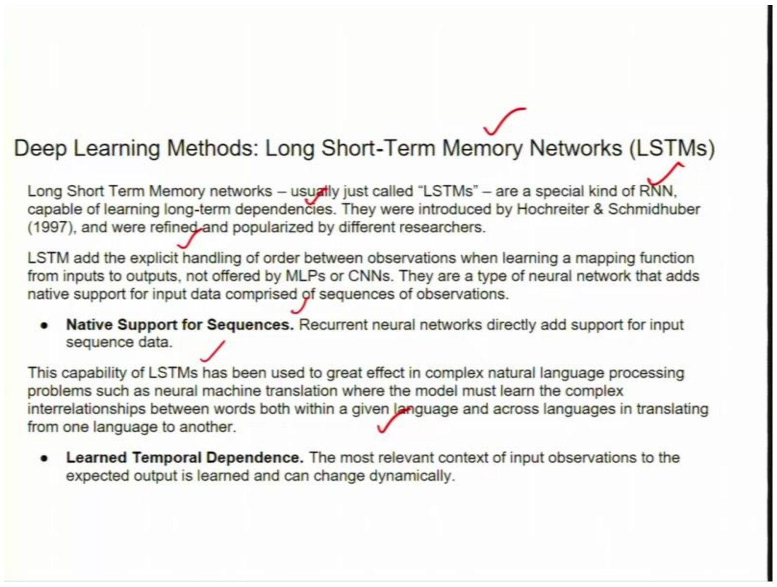
CNNs get the benefits of Multilayer Perceptrons for time series forecasting, namely support for multivariate input, multivariate output and learning arbitrary but complex functional relationships, but do not require that the model learn directly from lag observations. Instead, the model can learn a representation from a large input sequence that is most relevant for the prediction problem.

Now the next model into the deep learning is called convolutional neural networks, CNNs are a type of neural networks, they are designed efficiently to handle the image data. The ability of CNN to learn automatically, extract the feature from the raw input data, that is in the form of images can be applied to the time series forecasting problems as well.

So, that means a series of observation can be treated as one-dimensional image that CNN model can read and distill the most salient elements. So, the feature learning is an automatic identification, extraction, distillation of salient features from the raw input data that pertain directly to the prediction problems that is being modeled.

So, CNN gets benefit out of multi-layer perceptrons for the time series forecasting, namely the support for multi-variate input, multi-variate output and learning arbitrary but complex function relations but do not require the model learn directly from the lag observations. The model can learn the representation from the large input and sequence that is most relevant in the production problem.

(Refer Slide Time: 33:50)



Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by different researchers.

LSTM add the explicit handling of order between observations when learning a mapping function from inputs to outputs, not offered by MLPs or CNNs. They are a type of neural network that adds native support for input data comprised of sequences of observations.

- **Native Support for Sequences.** Recurrent neural networks directly add support for input sequence data.

This capability of LSTMs has been used to great effect in complex natural language processing problems such as neural machine translation where the model must learn the complex interrelationships between words both within a given language and across languages in translating from one language to another.

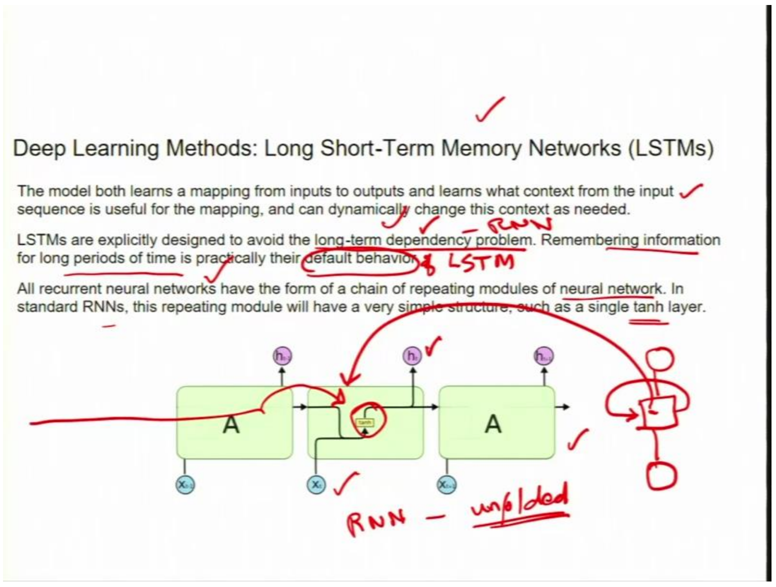
- **Learned Temporal Dependence.** The most relevant context of input observations to the expected output is learned and can change dynamically.

So, now let us see the next deep learning model, that is long short-term memory network, that is LSTM networks. So, LSTM networks usually called LSTMs are a special kind of recurrent neural networks RNNs that is capable of long-term dependencies. So, they were introduced by Hochreiter and Schmidhuber in 1997 and they were refined and popularized by different researchers.

LSTM add explicit handling of the order between observations when learning a mapping function from the input to the output not offered by multi-layer perceptrons and CNNs. They are the type of neural networks that add native support for input comprised of sequence of observations.

So, native support for the sequences is the advantage of this recurrent neural network and these capabilities of LSTM can be used to a great effect in the complex problems such as neural networks where the models learn the complex interrelationship problems. So, learned temporal dependence, so the most relevant context of input observation to the expected output is learned and can change dynamically.

(Refer Slide Time: 35:20)


Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

The model both learns a mapping from inputs to outputs and learns what context from the input sequence is useful for the mapping, and can dynamically change this context as needed.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior.

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

Now deep learning methods, that is long short-term memory, LSTM, the method both learned a mapping from inputs to the outputs and learned what context from the input sequence is useful for the mapping and can dynamically change the context as needed. So, in this particular slide we are now going in details of explaining the architecture or intricacies of long short-term networks, that is LSTM networks.

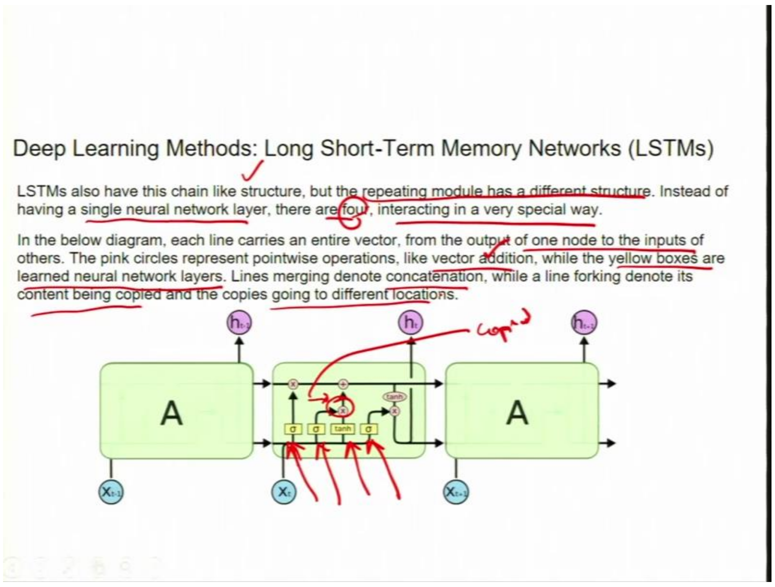
So, LSTM networks are designed to avoid the long-term dependency problem which happened in the recurrent neural network. So, that means it is capable to sustain or to work with a long-term dependency problems which was the drawback of recurrent neural networks, that is remembering the information for a long period of time is practically the default behavior of LSTM.

Now recurrent neural networks have the form of a chain of repeating the modules of a neural network. In the standard neural network, recurrent neural network, this repeating modules will have very simple structure such as a single tanh layer. So, this is recurrent neural network which is an unfolded form, rather you can think of this is the input and this is the neurons and then it will feedback. So, this is unfolded way of showing this.

So, this structure of a neuron is shown over here. So, it will be using the previous input, the output of a previous cell as the input. So, that is also called feedback. So, this particular previous input or output of the previous cell is also used here in calculating and also the input together applied the tanh and this particular tanh will become the output of this particular cell and also it will give the input to the to the next cell and so on.

So, you can see that in this way it will be having the short-term dependency because it knows what has happened in the previous time step but it is not able to memorize the long-term dependencies over several period of instances. So, that is the drawback of RNN.

(Refer Slide Time: 38:27)



Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

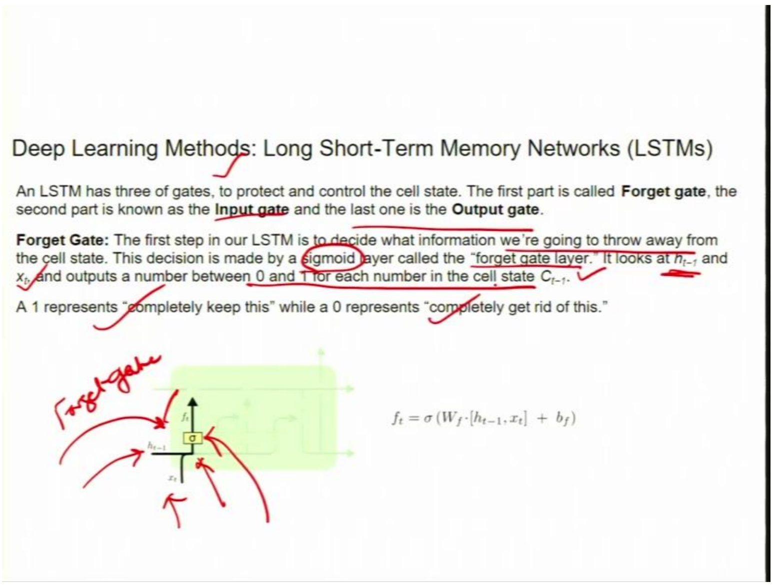
In the below diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

So, the LSTM also will have this kind of chain like structure but repeating the module has a different structure. So, this particular repeating modules will have instead of the single neural network layer, it has four interacting in a very special manner. So, you can see here instead of only one tanh it has a population it has a different four different interacting - this is first, this is second, this is third and this is four.

So, there are four different interacting neurons here in a particular cell, this is called a cell of an LSTM network. So, in the below diagram so each line carries the entire vector from the output of one node to the input of the other. So, this you can see that this is the input line to the cell, this is input to the cell, keeps on coming in a form of a streams or a flow and then you can see here, there are four different vector operations are being performed.

So, these pink circles represents the point operation like this is the vector addition, and this is the vector multiplication. While the yellow box here which is shown, while the yellow boxes are learn neural network layers. So, these are the learn neural network layers, their functionalities we will explain later. So, these lines are merging, they denote concatenation, while the line forking denote its content being copied here, is copied and the copies goes to different locations.

(Refer Slide Time: 40:55)



Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

An LSTM has three of gates, to protect and control the cell state. The first part is called **Forget gate**, the second part is known as the **Input gate** and the last one is the **Output gate**.

Forget Gate: The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}.

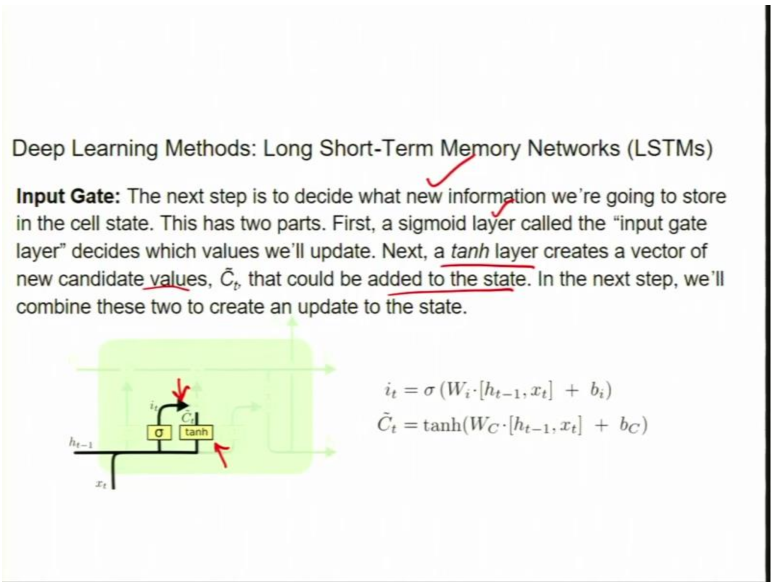
A 1 represents "completely keep this" while a 0 represents "completely get rid of this."

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

So, LSTM cell has three gates to protect and control the cell state. So, this you can see over here. The first part is called the forget gate, this is called the forget. The second part is called the input gate and the last one is called the output gate. So, forget gate is the first step of LSTM is to decide what information you are going to throw away from the cell, this decision is made by this sigmoid layer. The sigmoid layer is also called the forget gate layer.

So, it looks at the previous cell output that is HT minus 1 in the previous time and XT means the current input and outputs a number between 0 and 1 for each number in the cell state CT, CTI. So, 1 means completely keep this and 0 means completely get rid of this. So, that means if 0 is enabled then it will completely forget gate.

(Refer Slide Time: 42:13)



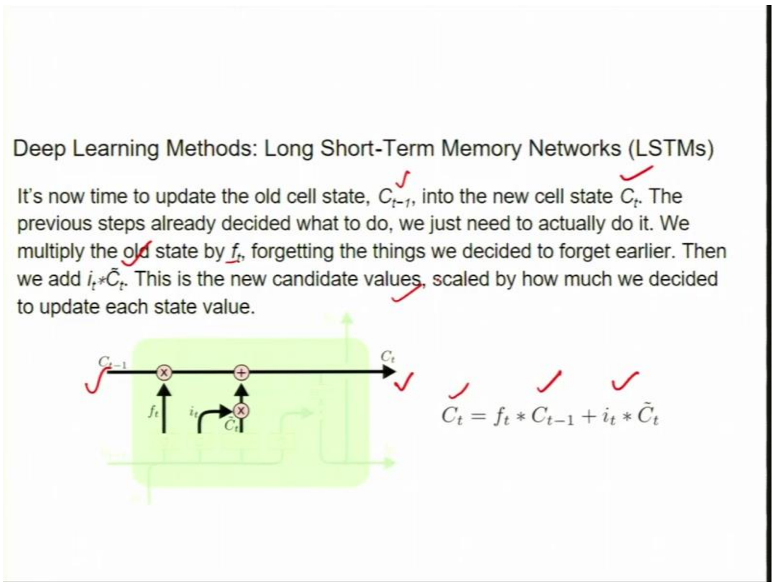
Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

Input Gate: The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a *tanh* layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. In the next step, we'll combine these two to create an update to the state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

So, input gate is the next step is to decide what new information you are going to store in the cell state. This has two parts. First is that sigmoid layer called the input gate layer, this decide which values will be updated. The next is tanh layer - creates a vector of a new candidate values, C hat T, this could be added to the state. In the next state we will combine these two to create an update to the state.

(Refer Slide Time: 42:46)



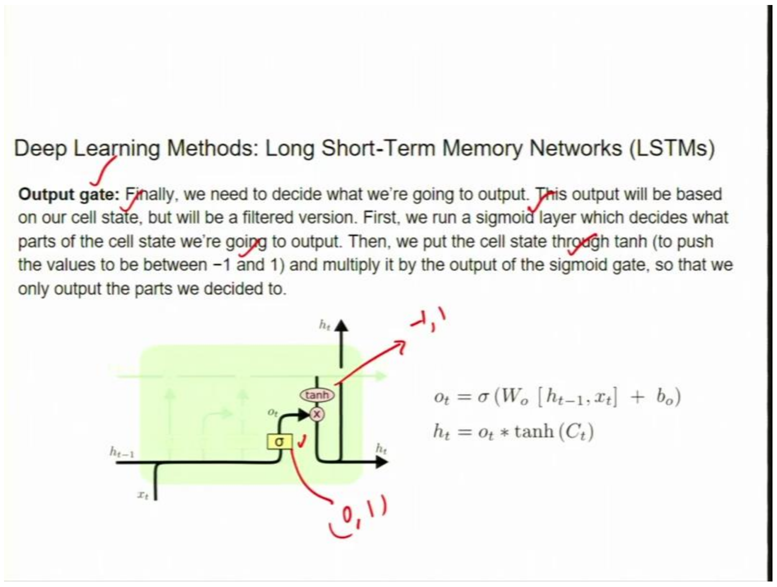
Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it. We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$. This is the new candidate values, scaled by how much we decided to update each state value.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Deep learning models long short-term memory networks LSTM networks. It is now time to update the old cell that is C of TI minus 1 into the new cell state that is CT. The previous steps already being decided what to do. So, we need to actually do in this the multiply the old state by FT, that is forgetting the things we decide to forget earlier, then we add IT into C hat T, this is the new candidate values scaled up by how much we decide to update each state values.

(Refer Slide Time: 43:35)



Deep Learning Methods: Long Short-Term Memory Networks (LSTMs)

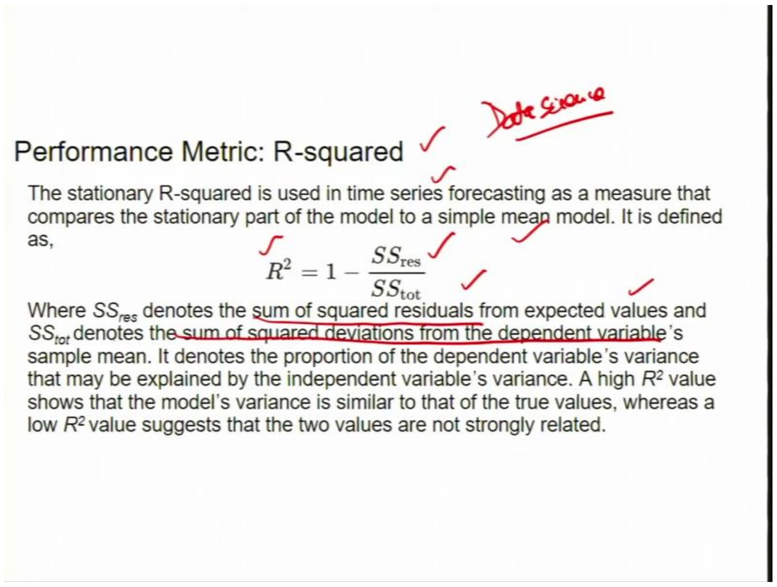
Output gate: Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Now the output gate of LSTM. Finally, we need to decide what we are going to output out of the cell. So, this output will be based on our cell state but will be filtered version. First, we run the sigmoid layer which decide what parts of the cell state we are going to output, then we put the cell state through this tanh to push the values between minus 1 and 1.

So, tanh now push the values between minus 1 to 1 whereas sigmoid is between 0 and 1. And to multiply the output of the sigmoid gate so that the only output parts we decided will be used up.

(Refer Slide Time: 44:26)



Performance Metric: R-squared

Data Science

The stationary R-squared is used in time series forecasting as a measure that compares the stationary part of the model to a simple mean model. It is defined as,

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

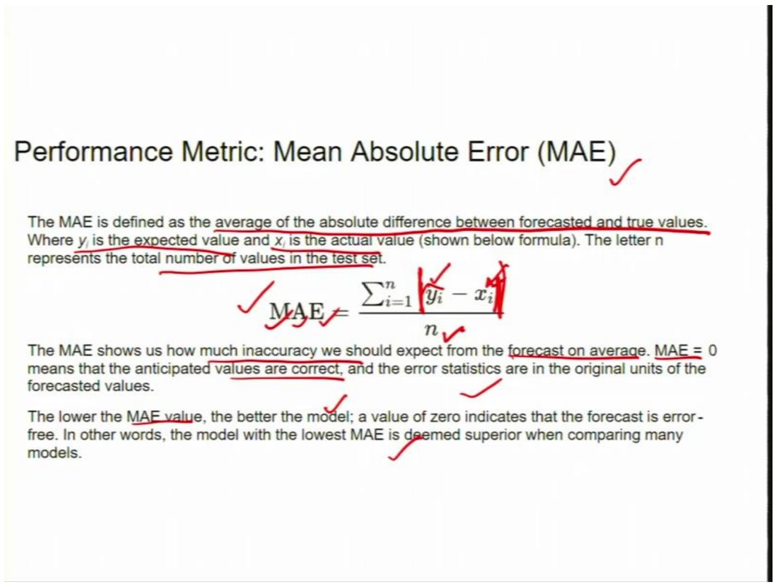
Where SS_{res} denotes the sum of squared residuals from expected values and SS_{tot} denotes the sum of squared deviations from the dependent variable's sample mean. It denotes the proportion of the dependent variable's variance that may be explained by the independent variable's variance. A high R^2 value shows that the model's variance is similar to that of the true values, whereas a low R^2 value suggests that the two values are not strongly related.

Now let us see the model performance. So, we have seen the LSTM and we have also covered RNN but now we will see that once the model is trained with the time series data using the deep learning model that is LSTM now how are we going to check the performance? That is the part of a data science.

So, there are various performance metric and this is very useful to understand whether your model is good enough to be deployed or still require more data for training. So, the stationary R squared is used in the time series forecasting as a measure that compares the stationary part of the model to a simple mean model.

So, this R squared is defined as 1 minus stationary, that is the sum of squared residuals from expected values and SS total, that is the sum of squared deviations from the dependent variables sample mean. So, it denotes the proportion of dependent variables variance a high R squared value shows the models variance is similar to that two values, whereas the low R square value suggests that two values are not too strong related.

(Refer Slide Time: 46:10)



Performance Metric: Mean Absolute Error (MAE)

The MAE is defined as the average of the absolute difference between forecasted and true values. Where y_i is the expected value and x_i is the actual value (shown below formula). The letter n represents the total number of values in the test set.

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

The MAE shows us how much inaccuracy we should expect from the forecast on average. MAE = 0 means that the anticipated values are correct, and the error statistics are in the original units of the forecasted values.

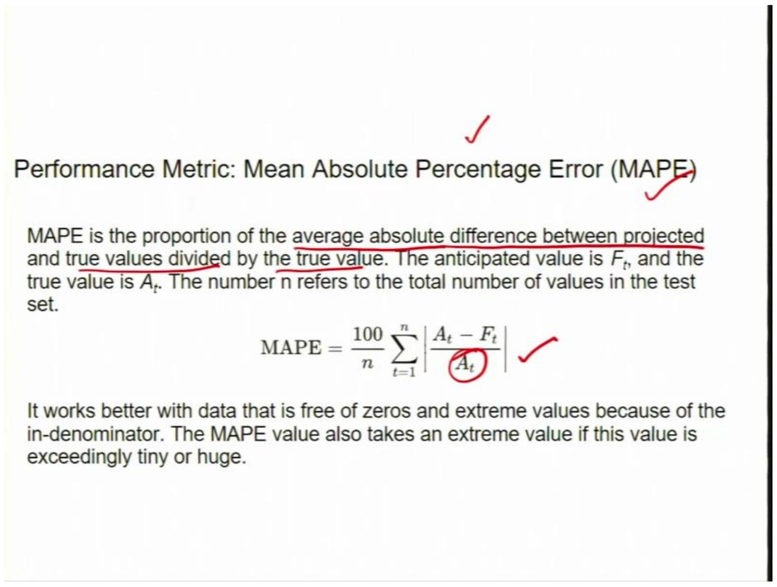
The lower the MAE value, the better the model; a value of zero indicates that the forecast is error-free. In other words, the model with the lowest MAE is deemed superior when comparing many models.

Next performance metric is called mean absolute error, that is called MAE. MAE is defined as the average of the absolute difference between the forecasted and the true values. So, where Yi is the expected value, so this is the predicted value you can say and Xi is the actual value. Now the letter n shows the total number of values in the test set and this is called and you have to take the absolute of it and this is the mean absolute error.

So, the error in the prediction with the actual values its absolute is averaged over that particular number of test set that becomes mean absolute error. Mean absolute error shows how much in accuracy we expect from the forecast or the average. If mean absolute error is zero means that anticipated values are very correct and the error statistics are in the original units.

The lower MAE values, the better the model is and the value zero indicates is error free. So, the model with the lower MAE is superior when compared to many other models. So, this is also a very good performance metric to see the performance of that particular model which is trained how that is all working.

(Refer Slide Time: 47:36)



Performance Metric: Mean Absolute Percentage Error (MAPE)

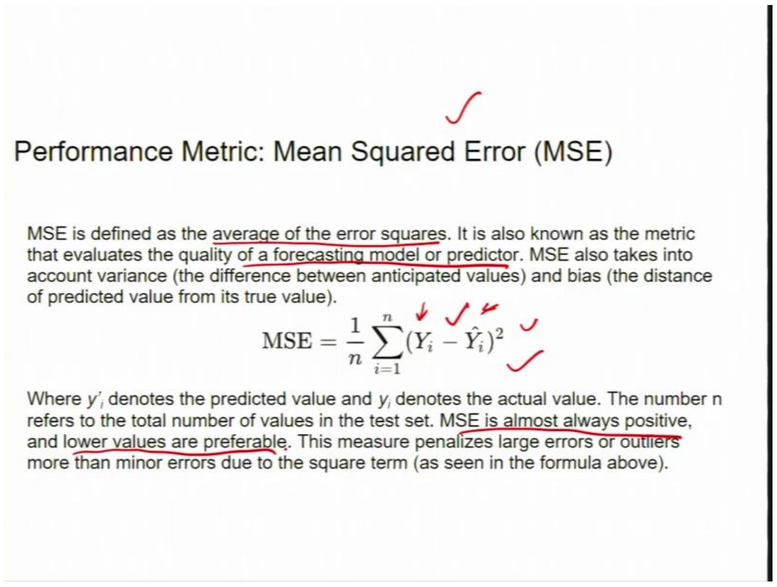
MAPE is the proportion of the average absolute difference between projected and true values divided by the true value. The anticipated value is F_t, and the true value is A_t. The number n refers to the total number of values in the test set.

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

It works better with data that is free of zeros and extreme values because of the in-denominator. The MAPE value also takes an extreme value if this value is exceedingly tiny or huge.

Now the next performance metric is called mean absolute percentage error MAPE. So, here MAPE is the proportion of the average absolute difference between the projected and the true value and is divided by the true values. So, here you can see that there is a difference and its percentage is taken called mean absolute percentage error.

(Refer Slide Time: 47:59)



Performance Metric: Mean Squared Error (MSE)

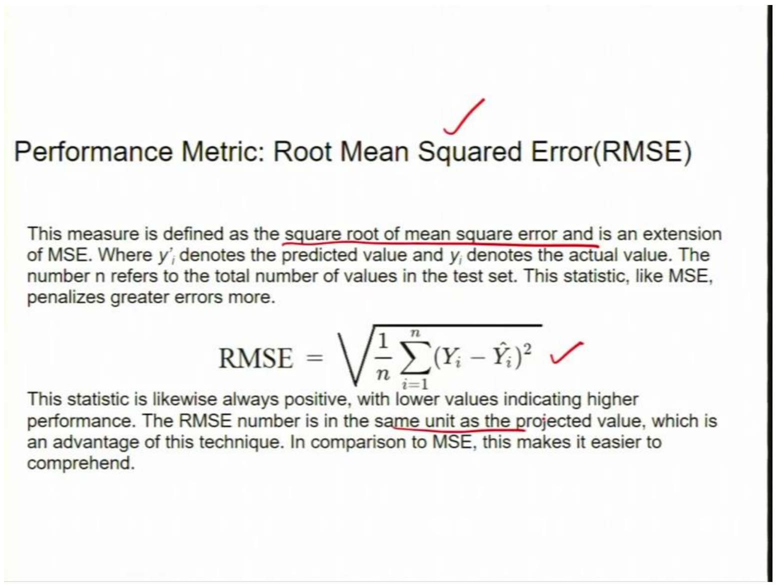
MSE is defined as the average of the error squares. It is also known as the metric that evaluates the quality of a forecasting model or predictor. MSE also takes into account variance (the difference between anticipated values) and bias (the distance of predicted value from its true value).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where y'_i denotes the predicted value and y_i denotes the actual value. The number n refers to the total number of values in the test set. MSE is almost always positive, and lower values are preferable. This measure penalizes large errors or outliers more than minor errors due to the square term (as seen in the formula above).

Now the next very important performance metric is called mean squared error MSE is an average of error squares. So, here you can see that this is the predicted value and this is the absolute value and the squared and its mean is taken up, evaluates the quality of forecasting model or predictor. So, MSE takes into account the variance and bias. So, this MSE is almost always positive and lower values are preferred and the measure, this measure penalizes the large errors or outliers more than the minor errors.

(Refer Slide Time: 48:40)



Performance Metric: Root Mean Squared Error(RMSE)

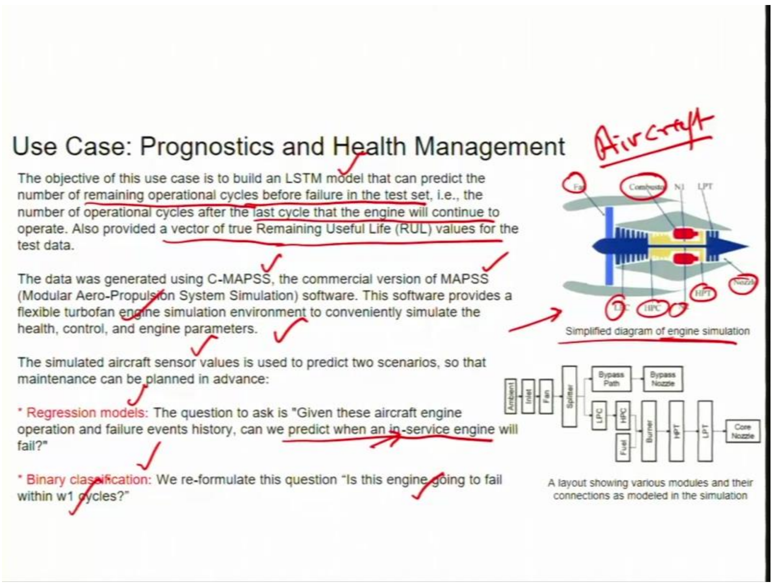
This measure is defined as the square root of mean square error and is an extension of MSE. Where y'_i denotes the predicted value and y_i denotes the actual value. The number n refers to the total number of values in the test set. This statistic, like MSE, penalizes greater errors more.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}$$

This statistic is likewise always positive, with lower values indicating higher performance. The RMSE number is in the same unit as the projected value, which is an advantage of this technique. In comparison to MSE, this makes it easier to comprehend.

Another performance metric is called root mean squared error or MSE. This measure is defined as the square root of mean squared error and in extension of MSE. So, here this if you compare this RMSE number is in the same unit as the projected value which is an advantage of this particular method. We will see all these later.

(Refer Slide Time: 46:08)



Use Case: Prognostics and Health Management

Aircraft

The objective of this use case is to build an LSTM model that can predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

The data was generated using C-MAPSS, the commercial version of MAPSS (Modular Aero-Propulsion System Simulation) software. This software provides a flexible turbofan engine simulation environment to conveniently simulate the health, control, and engine parameters.

The simulated aircraft sensor values is used to predict two scenarios, so that maintenance can be planned in advance:

* Regression models: The question to ask is "Given these aircraft engine operation and failure events history, can we predict when an in-service engine will fail?"

* Binary classification: We re-formulate this question "Is this engine going to fail within w1 cycles?"

Simplified diagram of engine simulation

A layout showing various modules and their connections as modeled in the simulation

So, let us go ahead with the use case for predictive maintenance that is prognostics and health management. So, objective of this use case is to build LSTM model that can predict the number of remaining operational cycles before the failure from the test data set. That is the number of operational cycles after the last cycle that engine will continue to operate.

So, this is a simple aircraft engine and this is a simple diagram and we will see that the sensors are placed to monitor the fan combustion and there are different so many sensors are there here to monitor this particular working of this engine, to operate and also provided a vector of the true remaining useful life that is values from the test data set.

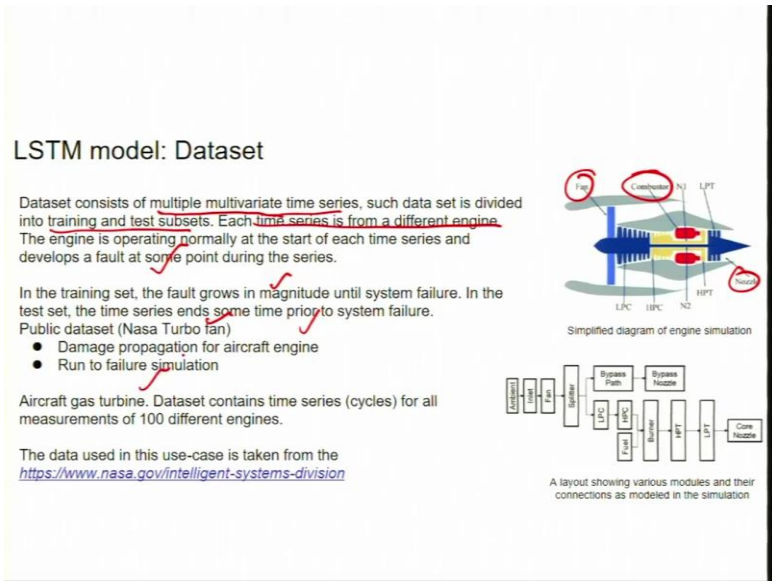
So, the data set which was generated is called CMAPS and its commercial version is called MAPS data set modular aero propulsion system simulation. So, this software provides the flexible turbofan engine simulation environment to conveniently simulate the health control and engine parameters.

So, by looking up this prognostics and health management and for the predictive maintenance using the remaining useful life you can also as well apply in any of the machines in the industry. So, therefore this particular use case is very important because it uses the IOT sensor data to monitor this predictive maintenance.

So, the simulated aircraft sensor values is used to predict two scenarios - one is the regression model, the other is called boundary classification. So, regression model will answer the questions like given these aircraft engine operation and the failure event history can we predict when an in-service engine will fail?

So, that means it requires the time how many cycles are still it has to now execute before that engine will fail. This is a regression model. Similarly the binary classification problem here in this domain will be that is the engine going to fail within w1 cycles or not? So, let us define more details into the data set.

(Refer Slide Time: 51:51)



LSTM model: Dataset

Dataset consists of multiple multivariate time series, such data set is divided into training and test subsets. Each time series is from a different engine. The engine is operating normally at the start of each time series and develops a fault at some point during the series.

In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure.
Public dataset (Nasa Turbo fan)

- Damage propagation for aircraft engine
- Run to failure simulation

Aircraft gas turbine. Dataset contains time series (cycles) for all measurements of 100 different engines.

The data used in this use-case is taken from the *https://www.nasa.gov/intelligent-systems-division*

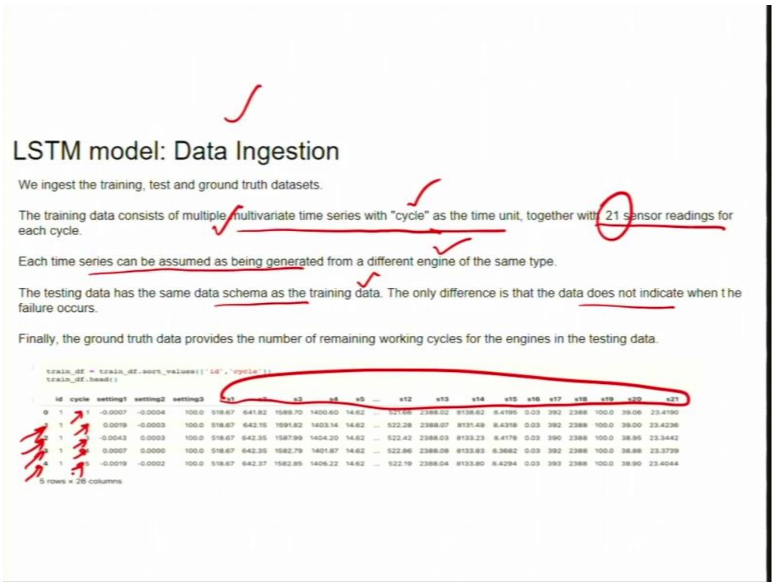
Simplified diagram of engine simulation

A layout showing various modules and their connections as modeled in the simulation

So, the data set consists of multiple multivariate time series data. Why? Because so many sensors are needed to monitor the fan, combustor and various component nozzle of aircraft engine. So, therefore such data set is divided into the training and test data set. Now each time series data is from different engine. So, engine is operating normally at the start of each time series data and develops a fault at some point during the time series all is captured into the data set.

Now in the training data set the fault grows in the magnitude until the system failure; in the test data set the time series ends sometime prior to the system failure. So, this particular data set is publicly available that is popularly known as the NASA turbofan data set. So, aircraft gas turbine and data set contains the time series cycle for all the measurements of 100 different engines.

(Refer Slide Time: 52:56)



LSTM model: Data Ingestion

We ingest the training, test and ground truth datasets.

The training data consists of multiple multivariate time series with "cycle" as the time unit, together with 21 sensor readings for each cycle.

Each time series can be assumed as being generated from a different engine of the same type.

The testing data has the same data schema as the training data. The only difference is that the data does not indicate when t he failure occurs.

Finally, the ground truth data provides the number of remaining working cycles for the engines in the testing data.

```
train_df = train_df.sort_values(['id','cycle'])
train_df.head()
```

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044

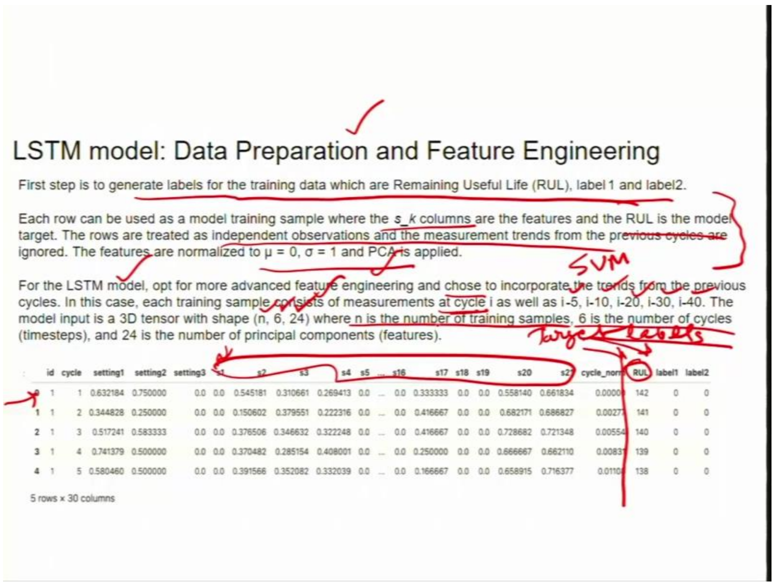
5 rows × 26 columns

So, let us use this LSTM model for data ingestion. So, we ingest the training test and the ground data set. So, the training model consists of multiple multivariate time series. So, here in this particular use case we see that the multivariate time series with cycles as the time unit. So, cycles we are considered as the time unit together there are 21 sensor readings for each cycle.

So, each time series data can be assumed being generated from different engine of the same type. So, the testing data has the same data set as the training data set. So, the only difference is that data does not include when the failure occurs and finally this particular data set is shown over here.

So, you can see that these are a different sensors S1, S2 and so on there are 21 sensors which are giving these readings at and these sensors sensor nodes are having these IDs and the cycle is also the time unit, we are now cycle 1, 2, 3, 4 time unit and different settings is also shown over here in the data set.

(Refer Slide Time: 54:12)



LSTM model: Data Preparation and Feature Engineering

First step is to generate labels for the training data which are Remaining Useful Life (RUL), label1 and label2.

Each row can be used as a model training sample where the *s_k* columns are the features and the RUL is the model target. The rows are treated as independent observations and the measurement trends from the previous cycles are ignored. The features are normalized to $\mu = 0$, $\sigma = 1$ and PCA is applied.

For the LSTM model, opt for more advanced feature engineering and chose to incorporate the trends from the previous cycles. In this case, each training sample consists of measurements at cycle i as well as i-5, i-10, i-20, i-30, i-40. The model input is a 3D tensor with shape (n, 6, 24) where n is the number of training samples, 6 is the number of cycles (timesteps), and 24 is the number of principal components (features).

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s16	s17	s18	s19	s20	s21	cycle_norm	RUL	label1	label2
0	1	1	0.632184	0.750000	0.0	0.0	0.545181	0.310661	0.269413	0.0	...	0.0	0.333333	0.0	0.0	0.558140	0.661834	0.0000	142	0	0
1	1	2	0.344828	0.250000	0.0	0.0	0.150602	0.379551	0.222316	0.0	...	0.0	0.416667	0.0	0.0	0.682171	0.686827	0.00277	141	0	0
2	1	3	0.517241	0.583333	0.0	0.0	0.376506	0.346632	0.322248	0.0	...	0.0	0.416667	0.0	0.0	0.728682	0.721348	0.00554	140	0	0
3	1	4	0.741379	0.500000	0.0	0.0	0.370482	0.285154	0.408001	0.0	...	0.0	0.250000	0.0	0.0	0.666667	0.662110	0.00831	139	0	0
4	1	5	0.580460	0.500000	0.0	0.0	0.391566	0.352082	0.332039	0.0	...	0.0	0.166667	0.0	0.0	0.658915	0.716377	0.0110	138	0	0

5 rows × 30 columns

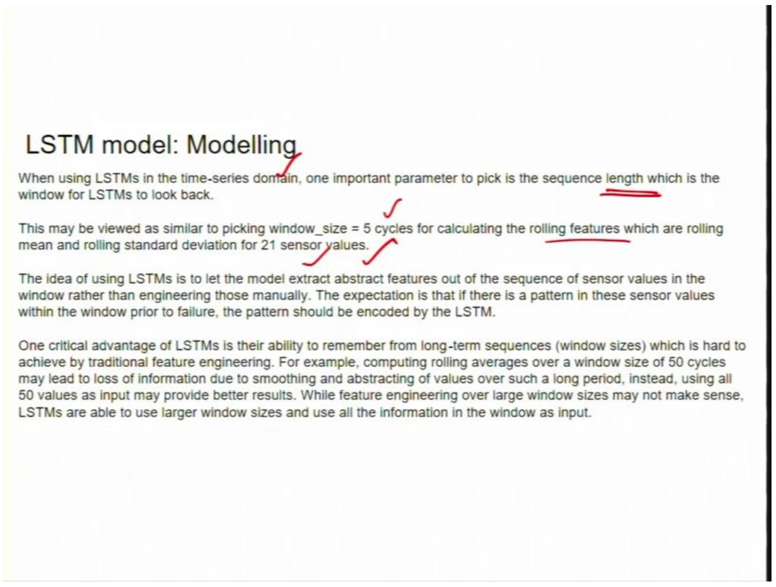
So, once the data set is available then the next part is to prepare the data and do the feature engineering. So, we have to generate the label for the training data set which are remaining useful like label 1 and label 2, that is RUL label 1 and label 2. So, this is what is called the labeling. So, this is added while data is to be prepared. Now each row can be used as the model training sample where SK columns are the features.

So, you can see these are the features and the remaining useful life is the model target. So, these are the target labels. So, the rows are treated as the independent observations and the measurement trends from the previous cycles are ignored. So, these particular features are normalized using principal component analysis.

So, for LSTM model we have to opt for more advanced feature engineering because this particular earlier feature engineering is good enough for simpler model because in the machine learning you have to do the feature engineering and like support vector machine can be used for that data set.

Now if you want to use LSTM model and you have to opt for advanced feature engineering you have to choose the appropriate trend from the previous cycles. So, in this case the training sample consists of the measurements at cycle I, that is i minus 5, i minus 10, 20, 30 and 40 cycles. So, the model input is 3D tensor with a shape n s 6, 24, and so on, n is the number of training sample, 6 is the number of cycles, 24 is the number of principal component features and so on.

(Refer Slide Time: 56:29)



LSTM model: Modelling

When using LSTMs in the time-series domain, one important parameter to pick is the sequence length which is the window for LSTMs to look back.

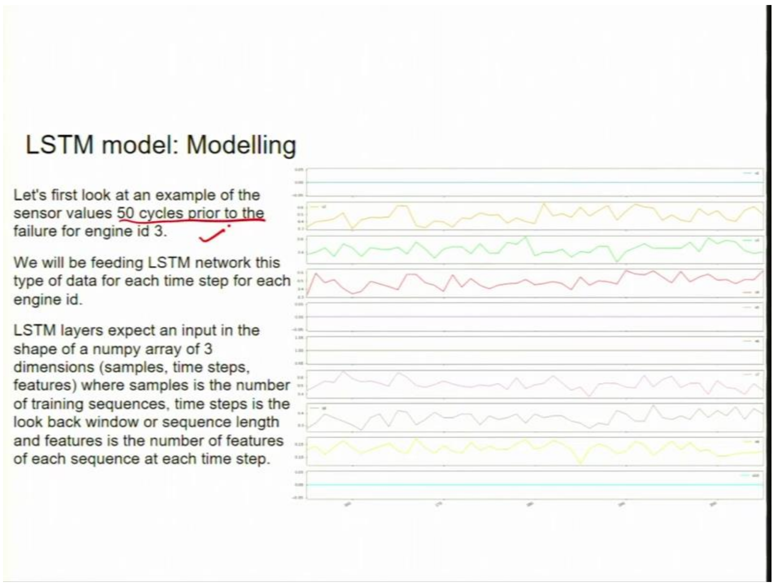
This may be viewed as similar to picking window_size = 5 cycles for calculating the rolling features which are rolling mean and rolling standard deviation for 21 sensor values.

The idea of using LSTMs is to let the model extract abstract features out of the sequence of sensor values in the window rather than engineering those manually. The expectation is that if there is a pattern in these sensor values within the window prior to failure, the pattern should be encoded by the LSTM.

One critical advantage of LSTMs is their ability to remember from long-term sequences (window sizes) which is hard to achieve by traditional feature engineering. For example, computing rolling averages over a window size of 50 cycles may lead to loss of information due to smoothing and abstracting of values over such a long period, instead, using all 50 values as input may provide better results. While feature engineering over large window sizes may not make sense, LSTMs are able to use larger window sizes and use all the information in the window as input.

So, let us see the model. So, LSTM in the time series domain, one important parameter here is the sequence length. So, this may be viewed similar to the picking the window length, that is of 5 cycles earlier and calculate the rolling features which are rolling means and rolling standard deviation for 21 sensor values. The idea of using LSTM is to let the model extract the abstract features out of the sequence of sensor data.

(Refer Slide Time: 57:05)



LSTM model: Modelling

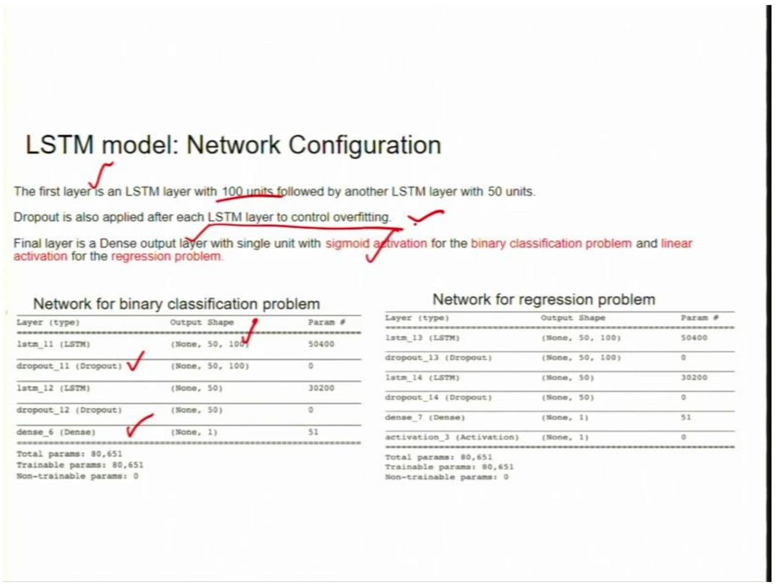
Let's first look at an example of the sensor values 50 cycles prior to the failure for engine id 3.

We will be feeding LSTM network this type of data for each time step for each engine id.

LSTM layers expect an input in the shape of a numpy array of 3 dimensions (samples, time steps, features) where samples is the number of training sequences, time steps is the look back window or sequence length and features is the number of features of each sequence at each time step.

So, here you can see that an example of a sensor values of a 50 cycles prior to the failure of the engine is shown over here and while feeding this LSTM network, this type of data for each time step and the engine id is shown.

(Refer Slide Time: 57:23)



LSTM model: Network Configuration

The first layer is an LSTM layer with 100 units followed by another LSTM layer with 50 units.

Dropout is also applied after each LSTM layer to control overfitting.

Final layer is a Dense output layer with single unit with sigmoid activation for the binary classification problem and linear activation for the regression problem.

Network for binary classification problem

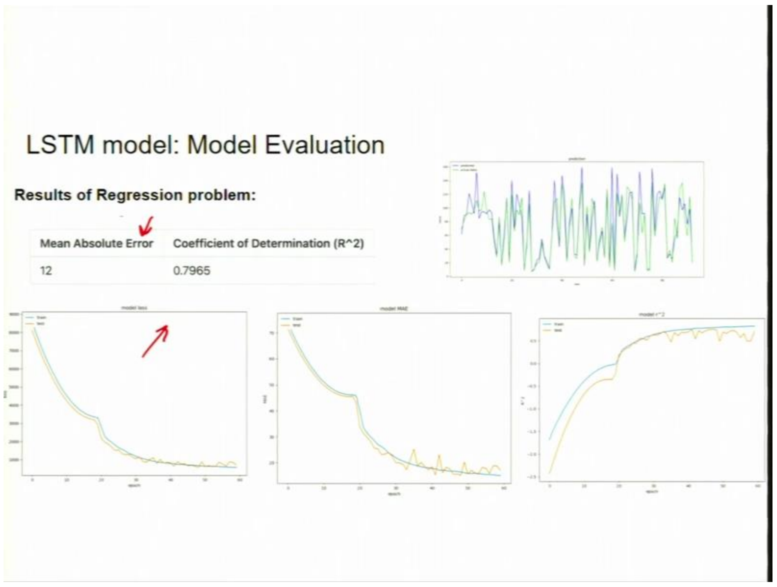
```
Layer (type)                 Output Shape              Param #
===============================================================
lstm_11 (LSTM)               (None, 50, 100)           50400
_______________________________________________________________
dropout_11 (Dropout)         (None, 50, 100)           0
_______________________________________________________________
lstm_12 (LSTM)               (None, 50)                30200
_______________________________________________________________
dropout_12 (Dropout)         (None, 50)                0
_______________________________________________________________
dense_6 (Dense)              (None, 1)                 51
===============================================================
Total params: 80,651
Trainable params: 80,651
Non-trainable params: 0
```

Network for regression problem

```
Layer (type)                 Output Shape              Param #
===============================================================
lstm_13 (LSTM)               (None, 50, 100)           50400
_______________________________________________________________
dropout_13 (Dropout)         (None, 50, 100)           0
_______________________________________________________________
lstm_14 (LSTM)               (None, 50)                30200
_______________________________________________________________
dropout_14 (Dropout)         (None, 50)                0
_______________________________________________________________
dense_7 (Dense)              (None, 1)                 51
_______________________________________________________________
activation_3 (Activation)    (None, 1)                 0
===============================================================
Total params: 80,651
Trainable params: 80,651
Non-trainable params: 0
```

So, let us see this LSTM network for doing this LSTM network modeling. What is the network configuration? LSTM network will consist of the first layer of 100 units and then it will be added a drop out layer to control the over fitting. Final layer is the dense output layer with a single unit with the activation sigmoid activation and a binary classifier and a linear activation and regression problem.

(Refer Slide Time: 57:54)



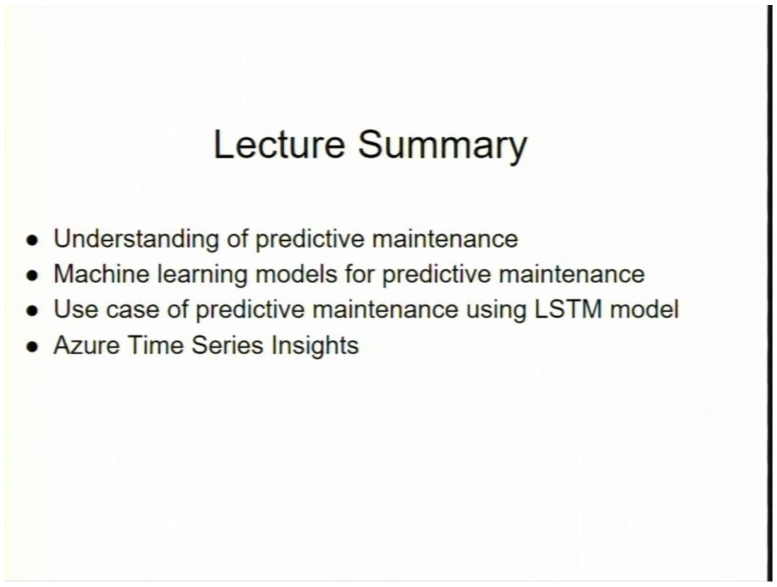
LSTM model: Model Evaluation

Results of Regression problem:

Mean Absolute Error	Coefficient of Determination (R^2)
12	0.7965

So, this is what we have now shown that if you evaluate this model what you will get is the different values of these errors, that is mean absolute error and the coefficient R squared error, all these errors are shown over here as the model.

(Refer Slide Time: 58:17)



Lecture Summary

- Understanding of predictive maintenance
- Machine learning models for predictive maintenance
- Use case of predictive maintenance using LSTM model
- Azure Time Series Insights

So, let us summarize the lecture. So, we have understood in this particular lecture about the predictive maintenance, the machine learning models for predictive maintenance, use case of predictive maintenance using LSTM model and also we have seen the time series analysis inside. Thank you.