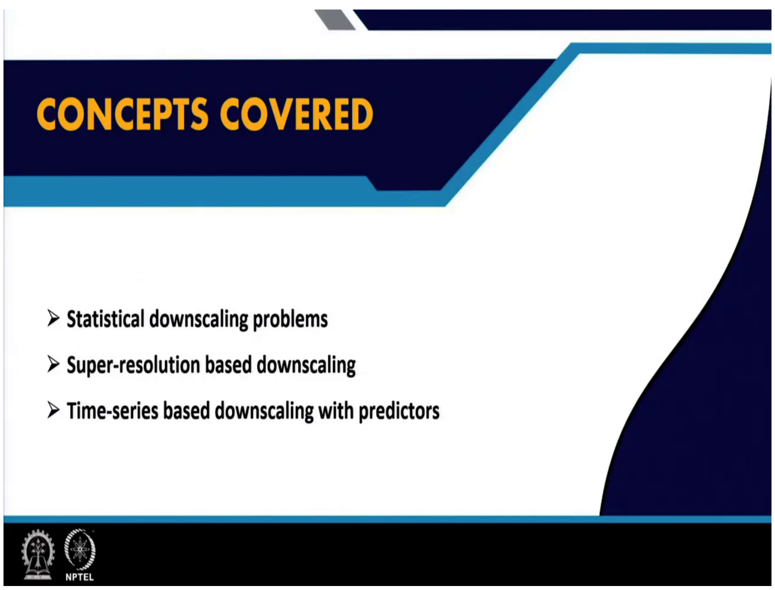
Machine Learning for Earth System Sciences

Prof. Adway Mitra
Department of Computer Science and Engineering
Centre of Excellence in Artificial Intelligence
Indian Institute of Technology, Kharagpur

Module - 03
Machine Learning for Discovering New Insights
Lecture - 17
Statistical Downscaling of Rainfall with Machine Learning

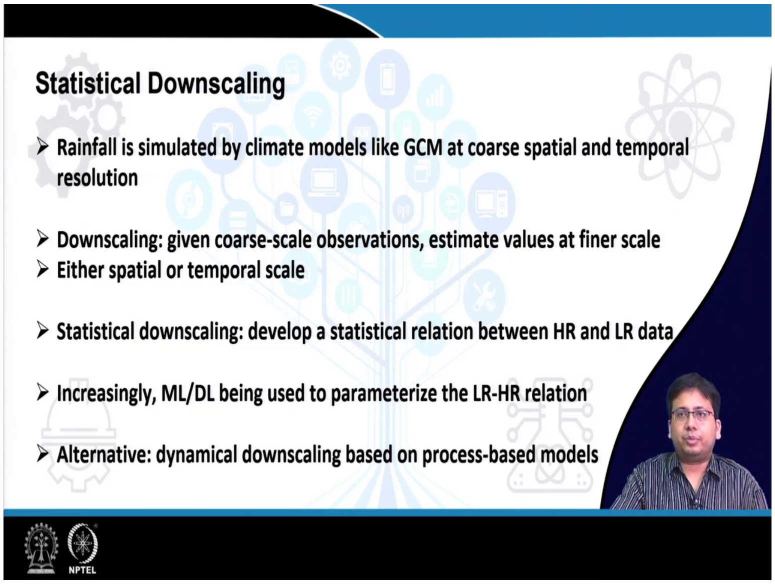
Hello everyone, welcome to lecture 17 of this course on Machine Learning for Earth System Science. We are still in module 3, which is about Machine Learning for Discovering New Insights. So, in today's lecture, we will discuss the topic of statistical downscaling of rainfall using machine learning.

(Refer Slide Time: 00:44)



So, the concepts we will discuss today are statistical downscaling problems, super resolution based downscaling and time series based downscalings in the presence of predictors or covariates.

(Refer Slide Time: 00:57)



So, let us start with what is downscaling problem. So, like in earlier parts of this course, we have been discussing about these models which simulate various spatiotemporal processes including rainfall. So, these simulations from the models they can take place at various resolutions, they are usually not very high resolution. Like for example, the GCMs, the global climate models also known as the general circulation models, they used to earlier carry out the simulations at 2.5^o a like grid size, which is basically about around 250 kilometers.

Now, of course, it has improved to 1^o or even $1/2^o$ so but that is like we can say is a reasonably coarse spatial dimension. Similarly, temporally also these simulations are often not that much fine grain that is or like while we have observations now in the order of seconds, if and a like a or at least hours, but in many of the simulations we still have daily scale or even monthly scale resolutions.

So, the down scaling problem is given coarse scale observations, how can we estimate the values at finer scales? By that I mean either spatial or temporal scales. So, that is the downscaling problem. Now, the there are broadly two alternative approaches to downscaling, one is statistical downscaling the other is dynamical downscaling. Now, dynamical downscaling is based on process based models. So, where we already have a process based model, which is carrying out the simulation at the coarse resolution.

Now, we use those model simulations to as inputs to another model at a finer resolution which will again a simulate the process at a finer scale or at a localized scale to give the high-resolution simulations. So, that is the approach of dynamical downscaling. The alternative is statistical downscaling where you develop a statistical relation between the HR and the LR data.

And these statistical relations like so these can be using probability distributions or regression or something like that, and increasingly machine learning or deep learning techniques are being used to parameterize the low resolution to high resolution relation. So basically, we want to define some kind of a mapping from the low resolution data to the high resolution data.

So, this mapping can be done by various machine learning algorithms. So, and this mapping can be either linear or non-linear, it can be either deterministic or probabilistic and so on and so forth.

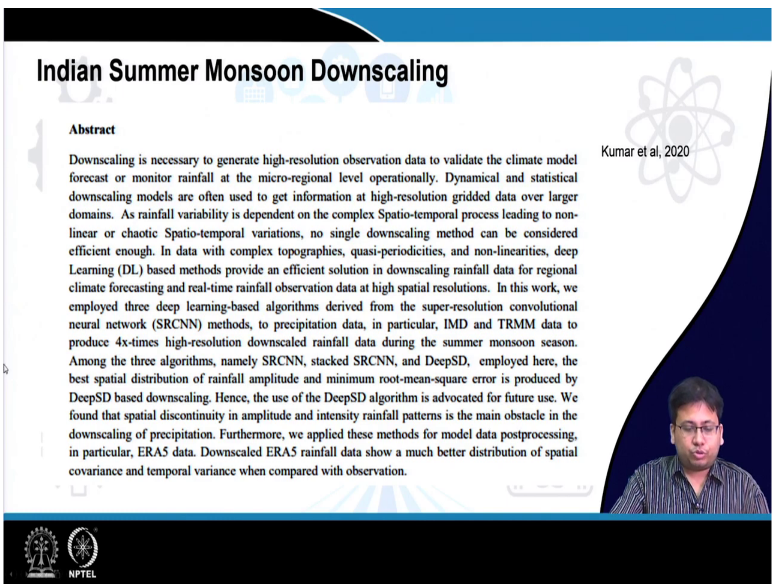
(Refer Slide Time: 03:54)

So, in today's lecture we will discuss a three case studies in which these kind of statistical downscaling has been achieved with the help of machine learning approaches. Now, the first case study is about Indian summer monsoon downscaling. So, in the previous lecture we already discussed about Indian summer monsoon and how it can be predicted and so on.

Now, like if you remember, we had talked about predicting the Indian monsoon either the ISMR value that is at a an all India scale or at the scale of homogeneous regions. Now, what if I want to predict the Indian rainfall at like much localized, say at city scale or village scale or something like that? So, that you know; obviously, that will that kind of prediction will be much more high resolution while our like existing technology is allowing us to make predictions only at very low resolutions, that is like we are making one prediction for a large region.

So, the task of downscaling is the input will be the predictions at higher resolution and the output will be the measurements, the corresponding measurements at much higher resolutions. So, this is the paper which came out of Indian institute of tropical meteorology in Pune a couple of years back. So, as it says the deep learning-based downscaling of summer monsoon rainfall data over the Indian region.

(Refer Slide Time: 05:27)



So, here is the abstract of this paper. So, it says downscaling is necessary to generate high-resolution observation data to validate the climate model forecast or monitor rainfall at micro-regional level operationally. Dynamical and statistical downscaling models are often used to get information at high-resolution gridded data over larger domains.

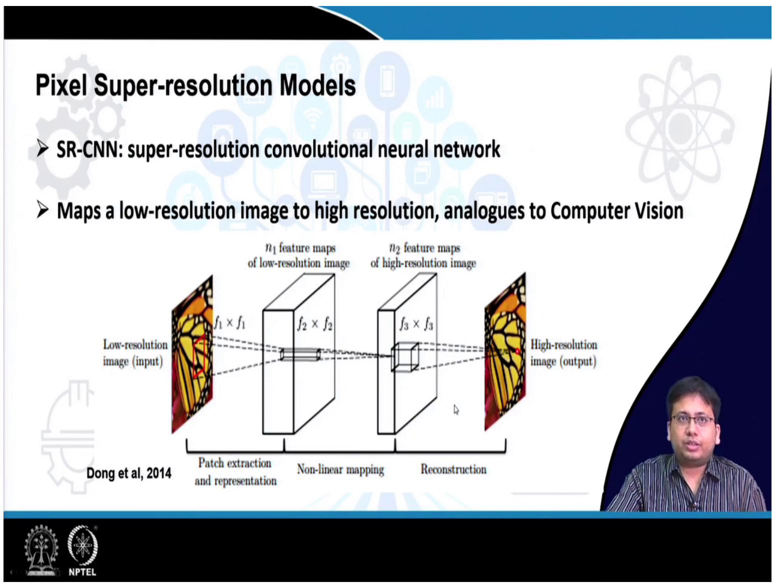
As rainfall variability is dependent on the complex spatio-temporal process leading to non-linear or chaotic spatio-temporal variations, no single downscaling method can be considered to be efficient enough. In data with complex topographies, quasi-periodicities, and nonlinearities, deep learning based methods provide an efficient solution in downscaling rainfall data for regional climate forecasting and real-time rainfall observation data at high spatial resolution.

In this work, we employed three deep learning based algorithms derived from the super-resolution convolutional neural network which is also known as SRCNN, to precipitation data, in particular, IMD and TRMM data to produce 4 times high resolution downscaled rainfall data during the summer monsoon period.

So, this IMD has the India meteorological department, it has its own like gridded data set which is at a scale of about 1^o and; that means, a roughly around 100 kilometers and TRMM it is a like a rainfall measurement satellite.

So, this satellite provides much higher resolution data up to $1/4^o$. So, from 1^o to $1/4^o$, so it is like a 4 time high resolution like downscaling of rainfall. So, they have compared different algorithms for this purpose.

(Refer Slide Time: 07:26)



The basic, the basic deep learning model which they plan to use here is the SR-CNN, the super resolution convolution neural network. Now, this SRCNN model this was earlier used the in the domain of computer vision way back in 2014. That was a time when like a deep learning was making huge progress in the domain of computer vision.

So, the what the SRCNN basically does it is simply that it defines a non-linear mapping from low resolution input to high resolution output. So, as you can see this, so we have as an input, we have a low resolution image, let us say it is $f1Xf1$ size. Now, there is like it is we define something like a convolutional neural network. So, these are like a set of convolutional layers and there might be multiple channels. So, the convolution operation is carried out as usual so like a patch of the image is selected and it is represented in the CNN, by the some the convolution operations.

So, similarly like we take the whole image we extract its different patches and for each patch, we develop a some kind of a feature map using the process of convolution. Now, once we have that, we do another round of the same operation. So, let us say this like this one creates $n1$feature maps of the low resolution image. So, like so this image is at a particular resolution, let us say $200X200$.

So, the like in the like here we make a representation of that image at the same resolution. Next, we come like we come out with another set of convolution layers. This time the like there is a mapping from the this lower dimensions, say to the higher dimension say $200X200$, to say $800X800$. So, the patch representations which we had already obtained, the feature maps for the patches which we had already obtained those are mapped to a to like the higher resolution image or the image with higher resolution.

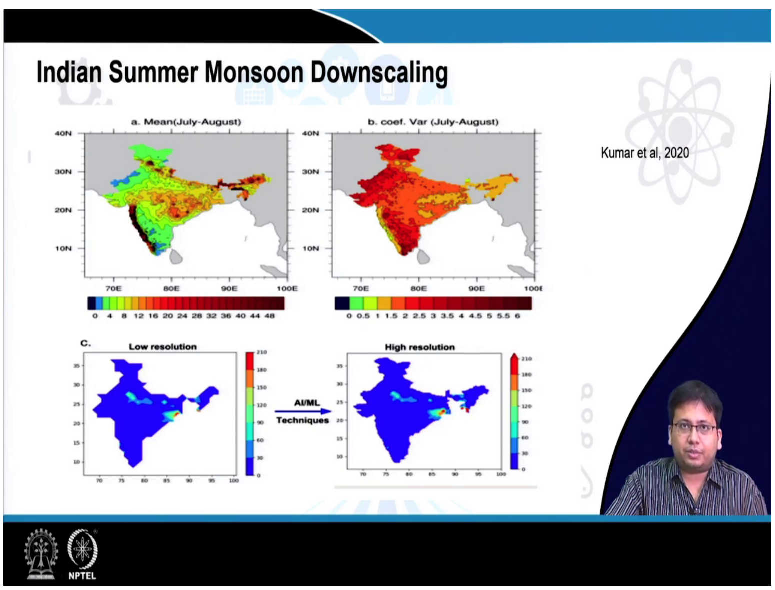
So, it is shown patches and there this mapping is once again represented through the convolution operation. And finally, the output of all these convolutions is the same image as the input, but at a higher resolution at the I mean at the same resolution, at the which as the second set of filters. So, that might be about $800X800$. So, the what is essentially happening is that we are taking the patches, image patches. So, within the patches there will be multiple pixels.

So, we are somehow carrying out some kind of a non-linear operations among the among those pixels and we are making that patch into a much higher resolution patch. That is whatever was the number of pixels enclosed in this patch we are replace, like we are creating a much larger number of pixels within the same patch, but that is done in a way that maintains the visual consistency of this patch.

So, and this operation is repeated for all so for all patches or for multiple overlapping patches in the image, these patches can be of different sizes also. So, that ultimately what we get is a like the same version as the input, I mean the same image of the input I (Refer Time: 11:15) at much larger much higher resolution. That is the information that was represented say with 1 pixel, now is the same represented that is now represented by let us say 4 pixels or even 64 X 4, 16, pixels.

So, that is a like and the information is divided into all those pixels so that we can see a like a much much smoother image and hence it is much more visually appealing also. So, this is the super resolution method which is known as SR-CNN.

(Refer Slide Time: 11:52)



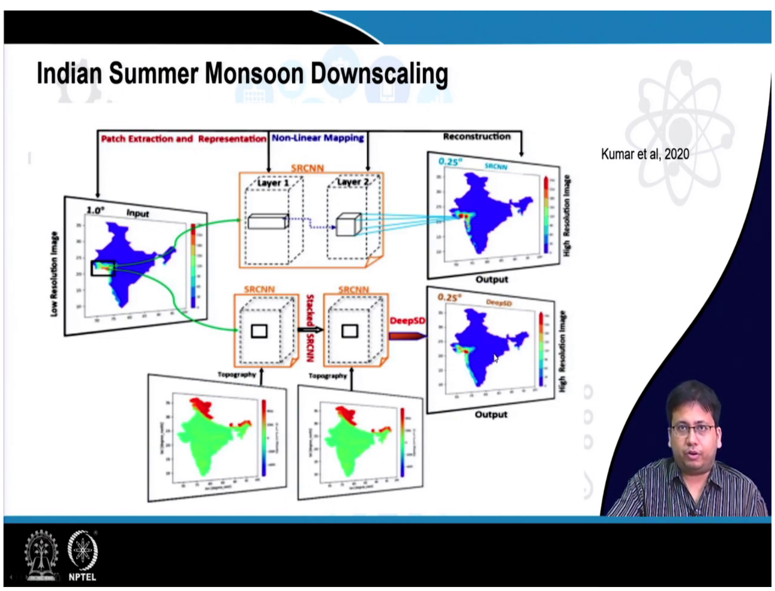
Now, what we do is we apply the same approach in the domain which we are interested in namely the rainfall downscaling. So, here we do not have an image instead we have a spatio-temporal map like this, the spatiotemporal map of rainfall. So, if you let us say if you focus on one particular day the rainfall map. So, here we can see the amount of rainfall is color coded, where this dark blue means little or no rainfall, this light blue means slightly higher rainfall, green means even more rainfall red means very high rainfall and so on.

So, like so this is what the map looks like. So, the like we can treat this as some kind of a an image where the pixels are the different grid locations. And the values stored in each pixel or the values of each pixel is basically the amount of rainfall in that particular grid point. And then we apply so this so called image is provided as input to this SRCNN kind of model and as a result what we get is a more high resolution map containing like this the say of the same information.

But except that like as I said what was represented earlier by a one grid may now be represented by four grids or four grid points or even 16 grid points. So, that is so like as a result the data that is the amount of rainfall which was let us say that one particular $1^o X 1^o$ region in the original image was indicated to have received about 100 millimeters of rainfall.

Now, that $1^o X 1^o$ region we will divide it into 16 regions, each of which is $1/4^o X 1/4^o$ and that 100 centimeter of rainfall we will distribute it among those $1/4^o X 1/4^o$ locations. And as a result it will the whole thing will look much more like coherent and visually, like it will make more sense it will look lot smoother and a lot sharper. So, that is what is known as the this statistical downscaling of rainfall in the spatial domain.

(Refer Slide Time: 14:21)



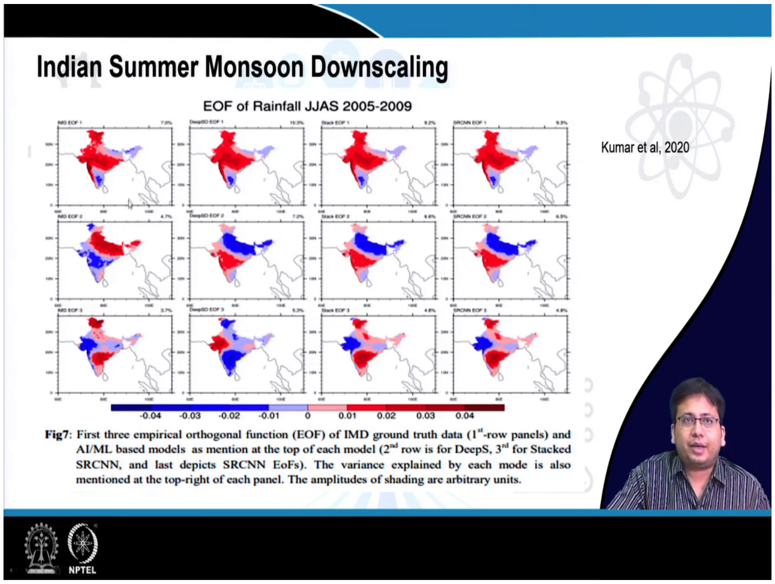
So, the it; so SRCNN is one way of doing it. So, as you can see this is the input is the 1^omap and the low resolution 1^o map and the output is the high resolution $1/4^o$ map where like in the SR-CNN, we have the first the layer one. That is basically, this set of convolution operations, then we have the layer two which is this set of convolution operations. And finally, we get the output at the desired target resolution.

Now, apart from SR-CNN, there is another approach known as the DeepSD, which is also roughly similar to the SRCNN approach only it is known as the stacked SRCNN where like there are more convolution layers, like this stacked one over another. And it is also known that this SR-CNN, sorry this DeepSD model this works better if it is provided with some auxiliary information such as the that elevation map of the region.

So, we know that rainfall has something to do with elevation, like the especially the concepts like topography and orography, they like they influence the rainfall, by orography I mean the presence of mountains etcetera. So, we know that when there is a when rain bearing winds are flowing and it is obstructed by some mountain range, as happens in the foothills of the Himalayan regions or in the Western Ghats then at the foot of the those mountains, there the rain bearing winds cause a lot of rainfall.

So, like in other regions also like the presence of high terrain, that may modulate the rainfall in various ways. So, there this elevation this can or the topography it can act as an interesting or important covariate for the rainfalls. And this DeepSD takes this into account to make the prediction of basically to create the down scaled version of the rainfall map.

(Refer Slide Time: 16:32)



So, and if we actually, like the like we if we want to evaluate these kinds of these down scaled rainfalls. So, the climate scientists use that the do the evaluation in different ways. So, first of all we do have the ground truth, the $1/4^{o}X1/4^{o}$ rainfall we have from the TRMM satellite. So, like we can like held hold out some test period in which we can compare the output of the SRCNN with the TRMM imagery for those testing period and see like at every grid we can do the comparison and see if the predicted rainfall values match the rainfall values as sensed by the TRMM satellite.

And there are like other ways of checking the spatio temporal like consistency and so on. So, there are like in the domain of earth sciences, one famous technique for spatio temporal data is the EOF, which is basically the empirical orthogonal function which is basically the principal component analysis done in the spatio-temporal domain. So, this EOF is a very important concept, so just like in normal machine learning we have principal components, means that we have like high dimensional data.

Now, we carry out some linear combination of those dimensions to come up with new dimensions and we then we can project the data along those new dimensions. So, in the spatio-temporal domain or rather at least in the spatial domain, we can do the similar principle component analysis. So, suppose we the original data is something like a spatio spatial map of a variable. Now, I can or decompose that spatial map into like certain modes and every spatial map then is a linear combination of those particular modes.

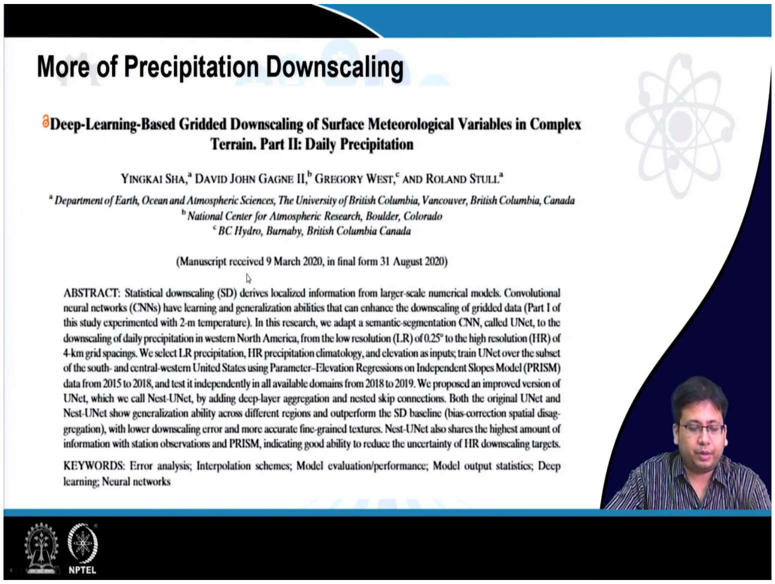
So, like here you can see that, what you are seeing here on the different; on the different rows, these are like what is the different EOFs. So, this one is the first EOF called the EOF1, we can consider it something like as the first principal component. So, its shows that these red regions they are having a slight positive anomaly of rainfall, while this South Eastern region it is having a slightly low anomaly of rainfall.

So, this is the usual the most prominent mode of rainfall distribution over India in the monsoon, because we know that like the much of the Indian landmass receives a rainfall, I mean receives more rainfall than usual during this period, but with the exception of the South Eastern region. Then this is the second mode which shows that strong rainfall over the North Indian the Indo Gangetic plains, while there is less rainfall over the Central Indian region.

Then this is the third principal component which shows that there is a strong rainfall in the South Central India while there is a less rainfall in the Western parts. So, any the claim is that any days rainfall map, whatever it is can be expressed as some kind of a linear composition of these different modes.

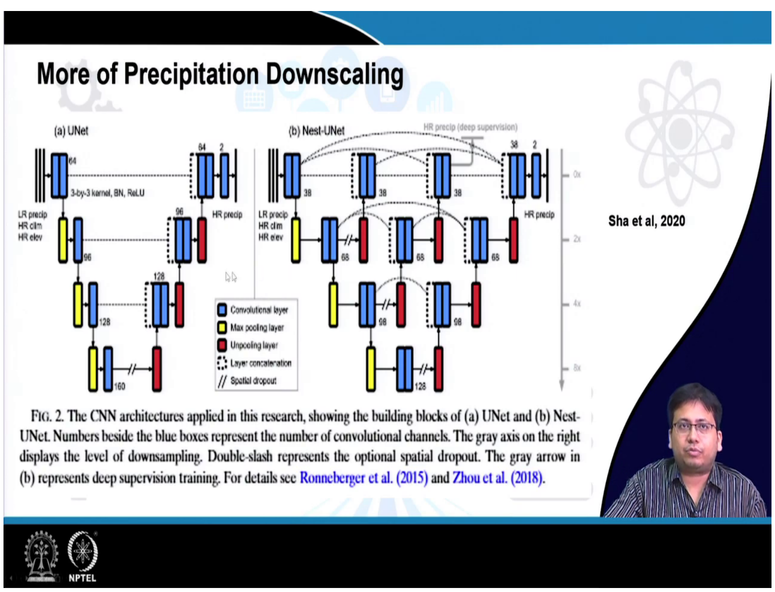
So, what they have done here as for evaluation is that, they have like calculated these modes for like either for the ground truth or from the down scale data obtained from the using SRCNN or using DeepSD etcetera and they have shown the similarities and differences between the different modes.

(Refer Slide Time: 20:32)



So, there along these lines, there are more work on the downscaling of precipitation. So, this is another paper deep learning based gridded down scaling of surface meteorological variables in complex terrain. So, they have considered the gridded downscaling problem in for different variables rainfall is only one of them.

(Refer Slide Time: 20:55)



So, these people have considered another like deep learning based model, this is known as UNet. So, UNet is a model which is used frequently for image segmentation. So, just like SRCNN is used, was originally used in the domain of computer vision for like scaling up a low resolution image to a high resolution image, this UNet was used for segmenting an image.

That is like you will have a full image to begin with and it will decompose or it will like a partition that image into different parts, like this part is corresponding to the sky, that part is corresponding to tree, that part is corresponding to sea etcetera. So, what the UNet does is basically it ok, first of all why it is called an UNet? So, as you can see, like if you like if you look at these layers it is like is arranged in the form of an U.

So what exactly it does is it first follows a sequence of convolution and pooling steps like this. So, like there are convolution and then again convolution followed by pooling, then again convolution again pooling again convolution again pooling and so on and as we do this progressively the image becomes smaller and smaller. So, we know that pooling, the operation that makes the image smaller, because it is like say if you are considering $3X3$ pooling.

So, like we take 9 pixels and remember, only the largest the value among them, so 9 pixels are basically replaced by 1 pixel. So, that way the at every pooling step the image becomes smaller and in each convolution step also the image becomes slightly smaller. And so by the sequence of operations the full image is progressively made into a very small code by throwing away a lot of the extra information.

And because in the max pooling step what we are doing is, we are throwing away the smaller pixel values and retaining only the locally larger ones. And in the convolution we are just like mixing every pixel value with its neighborhoods so that we get a highly convolved and reduced image at the end of this operation. After that we do the upscaling, that is we restore the image back into its original dimension through a series of, like a we can say deconvolution and unpooling step.

So, deconvolution can still be done by CNNs only, now this there is an un-pooling layer which basically does the opposite of pooling. And finally, we come back to the like desired resolution which we want, which might be the target resolution that is the input might be might have been the low resolution data and the output will be the high resolution. And like the an important feature of this UNet is these skip connections. So, you are seeing these dashed lines so these are the skipped skip connections.

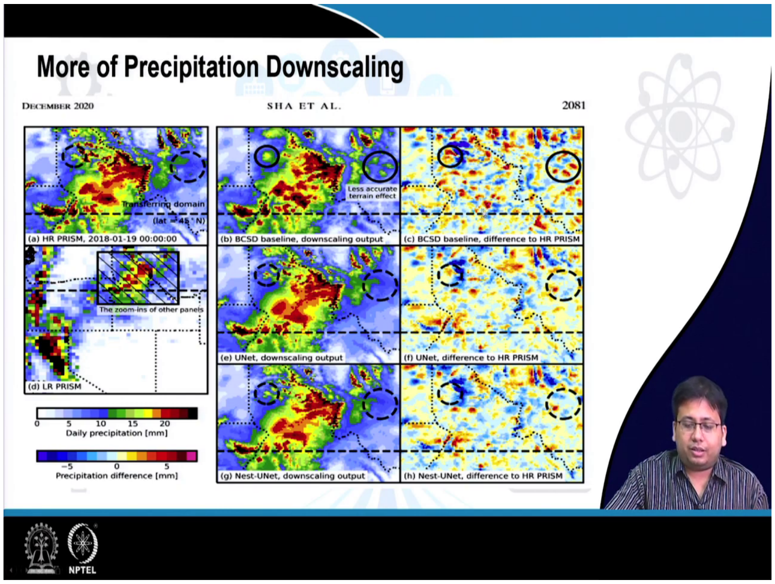
So, that is see for when we are considering this layer the input to this layer is coming not only from the previous layer, but also from this layer, that is it is getting receiving its input from two steps. So, this helps it to maintain the relation. So, like you might be wondering in the pooling steps, I have thrown away so many so much of this information. So, then how can unpooling happen? That is in pooling, they like I had $3X3$ pixels, I choose chose whichever is the highest pixel value among them. I preserved only that and threw away the earlier ones.

Now, if I want to do the unpooling, then if since I have already thrown away the remaining 8 values then how will I get them back? The answer is the answer comes from the fact that there are these skipped connections like this. So, whenever like you did the pooling operation or rather before doing the pooling operation, whatever was the input I did not completely throw it away. I remember it and in the unpooling process I will again like use that to guide the unpooling process.

So, that these skip connections they help in this kind of the restoration or the reconstruction operation. However, it is the reconstruction, the aim of reconstruction is not to get back the original image that would of course, be pointless. But to get back a higher resolution image which like; which will have some necessary like some of the detail, but not the not maybe not all the details. And; however, the like in this case the input is of one resolution and the output is of a higher resolution.

So, the these convolution and deconvolution steps ensure that like that is taken into account. And so, similarly similar to UNet there is also something known as a nested UNet, where this same process of like. So, if you see the it is like a collection of lots of small UNets like this. So, this is an U, then this is another U, this is another U etcetera so and the whole thing itself is a large U.

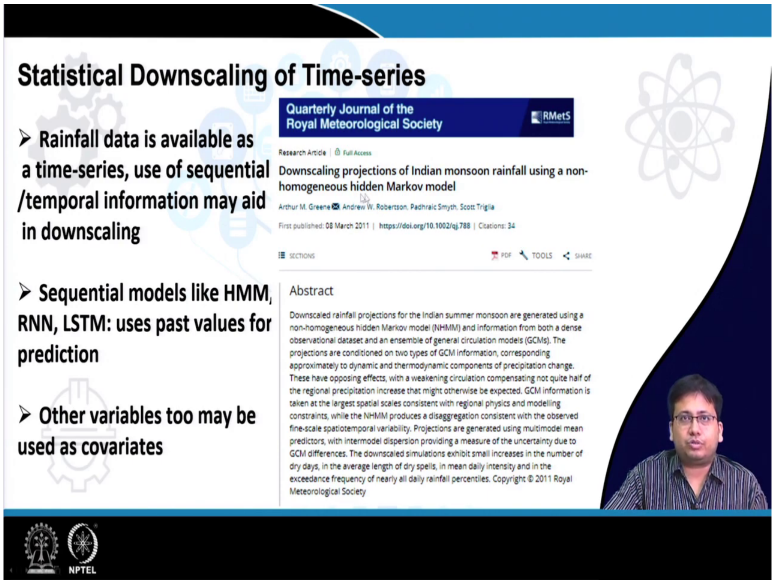
(Refer Slide Time: 26:33)



So, this is the more sophisticated version of the UNet called the nested UNet and then again it has been this also this approach of downscaling has using UNet has also been validated in different parts of the world, also on simulation models. Like in the previous case we like the high resolution data was the from the trim satellite and the low resolution data was the IMDs ground based records.

Similarly, in this case also they have the low-resolution model simulations and they may also have the high resolution model simulations or obtained from some kind of remote sensing satellite imagery. And then they like once again they calculate the differences between the down scaling output and the desired and the actual measurements, which they have obtained from some prism satellite. And then they show like they identify like pixel by pixel they carry out the identification.

(Refer Slide Time: 27:32)

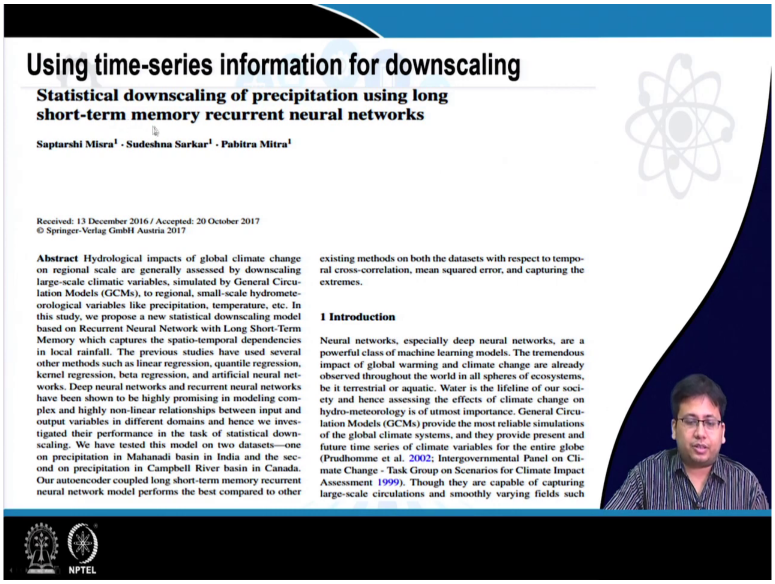


Now, the rainfall data it is likes in both of these approaches we are considering every day's rainfall map as or like as something as a static quantity as a. However, rainfall is not a static quantity, is it is usually present in the form of a time series. That is we do not deal with individual rainfall maps, but instead we have sequence of these rainfall maps.

So, when we are downscaling, we might be useful to use this kind of sequential information. That is suppose I have already generated yesterday's high resolution map that might help me to generate today's high resolution map also.

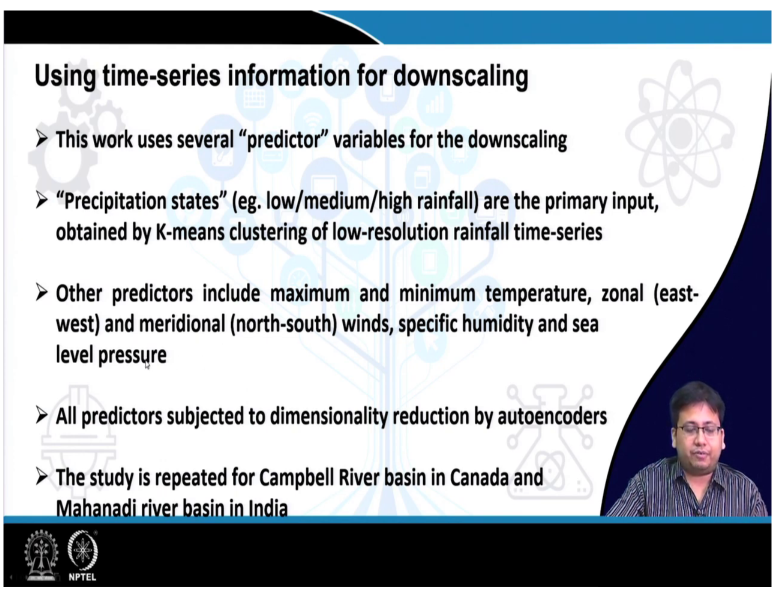
So, the sequential models like hidden Markov models, recurrent neural networks, LSTM etcetera these use the past values for the prediction purposes and other variables two may be used as covariates. So, like some of the earliest works of this rainfall downscaling, I mean statistical downscaling used to be done using things like hidden Markov model.

(Refer Slide Time: 28:42)



But now nowadays in recent years of course, we have moved from hidden Markov models to say recurrent neural networks etcetera for sequential data modeling. So, this is one paper which appeared from IIT, Kharagpur in 2017. So, here the aim is to do the statistical downscaling of precipitation using the sequential a properties as well as the presence of other covariates.

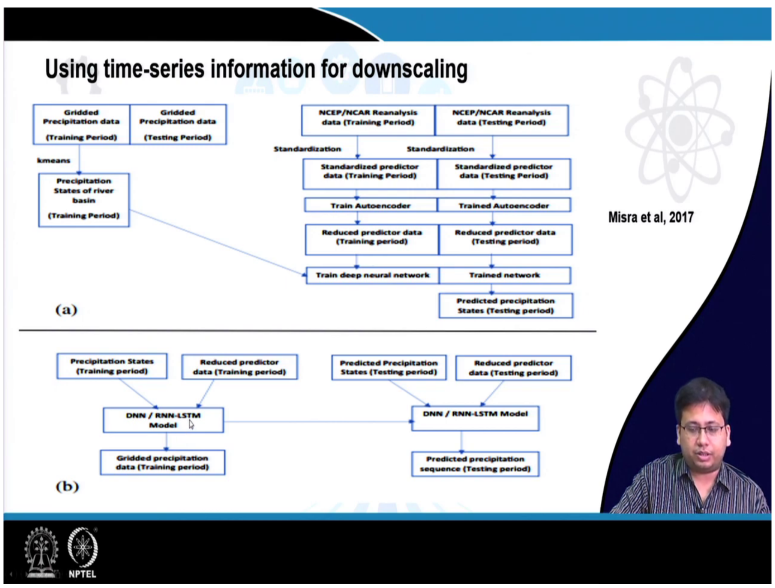
(Refer Slide Time: 29:07)



So, like the basic idea is as follows. So, this work uses several predictor variables for downscaling. It defines precipitation states, example low medium or high like low medium or high rainfall as their primary inputs. And these precipitation states are obtained by K-means clustering of the low resolution rainfall series. So, these precipitation states are used as some kind of predictors, while other predictors can include maximum and minimum precipitate temperature, zonal and meridional winds, specific humidity and sea level pressure.

And so, all these predictors they are subjected to dimensionality reduction with the help of auto encoders, earlier also we have discussed that auto encoders are used for non-linear dimensionality reduction. And this is study is carried out in two regions, one is in Canada and the other is in India.

(Refer Slide Time: 30:02)



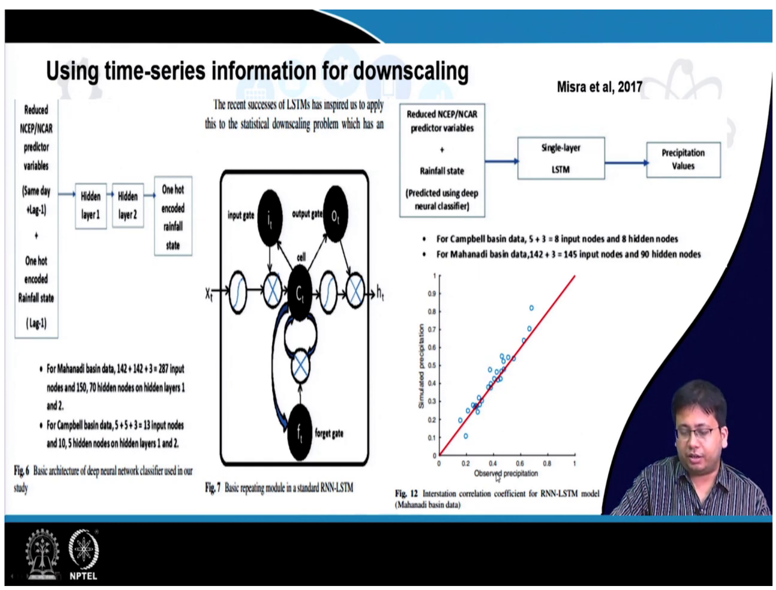
So, this is the workflow. So, the gridded precipitation data is already is first of all, it is the gridded precipitation data is divided like is reduced to this like rainfall states or precipitation states through the K-means clustering. And then this; so that gives us the these discrete states which are the input. Then we have the reanalysis data which is basically the covariates which I talked about. So, they are present in the form of a time series.

So, first of all they are standardized using the auto encoder which I mentioned, the for this also includes the training of the auto encoder the estimation of its parameters and so on, which is done during the training phase. So, once that auto encoder, the trained auto encoder is like a reduces the dimensionality of the data, then it is used to train the deep neural networks.

So, the deep neural network is something like an RNN or LSTM etcetera, which is used for the time series prediction purposes. So, the what it does is, it receives the precipitation states as inputs, like which is like already there which has already been calculated based on the low resolution time series. And then it also receives the reduced predictor data that is the other covariates. So, and then it of course, has its the previous values as its in its memory.

So, then the RNN or LSTM that predicts the gridded precipitation data and then like in the same thing once the like and in the training phase like we of course, know which what is the high resolution precipitation data. So, like this the output of it is like is calibrated with the desired output which enables us to train this deep this RNA or LSTM model.

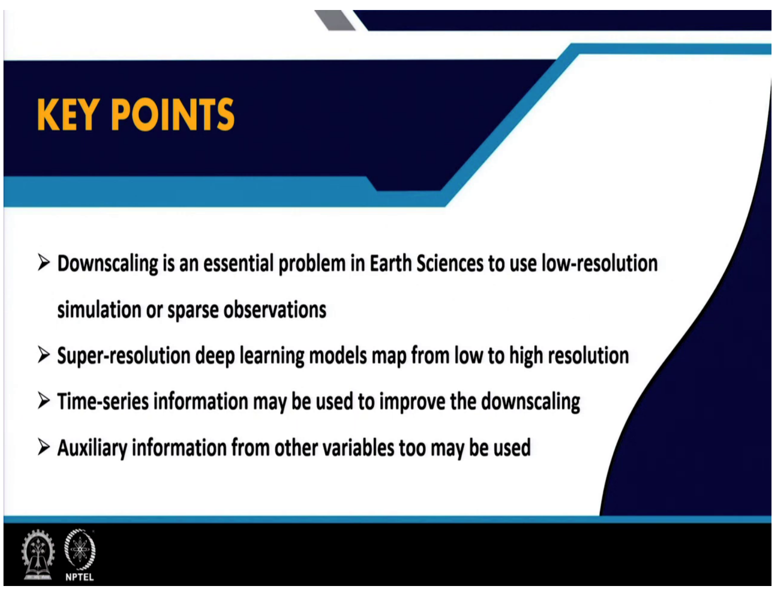
(Refer Slide Time: 32:08)



And so, this is what the this is basically the structure of the of the model. And in fact, they use the one hot encoding of the rainfall state, which I just mentioned and this way they are able to get very high quality of the high resolution data. That is the, so like here you can see the they have compared the observed precipitation against the simulated precipitation, by simulated I mean whatever is generated by the LSTM and you can see almost perfect map.

(Refer Slide Time: 32:38)

(Refer Slide Time: 32:41)



So, these are the references which we discussed. So, the key point here is that the downscaling is an essential problem in earth sciences to use low-resolution simulation or sparse observations. Now, these super resolution deep learning models they can be used to map from low to high resolution.

The another option is to use the time series information which can improve the downscaling and besides other variables apart from the variable, we are trying to down scale the other variables can also be used as predictors or covariates and that also might help in the downscaling purpose.

So, with that we come to the end of this lecture. In the next lectures we will discuss other applications in which machine learning is used to gain new insights in the earth sciences. So, till then goodbye.