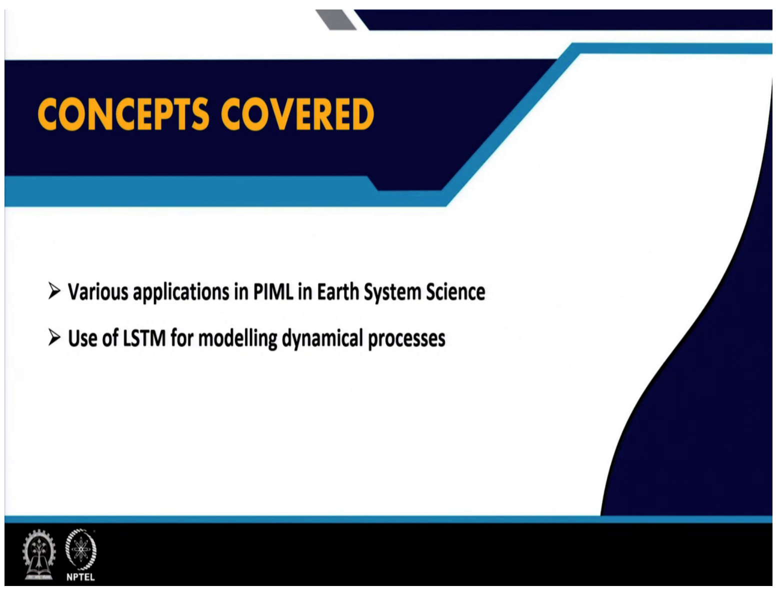
**Machine Learning for Earth System Sciences**
**Prof. Adway Mitra**
**Department of Computer Science and Engineering**
**Centre of Excellence in Artificial Intelligence**
**Indian Institute of Technology, Kharagpur**

**Module - 05**
**Machine Learning for Earth System Modelling**
**Lecture - 37**
**Physics-Inspired Machine Learning for Process Models - 2**

Hello, everyone. Welcome to lecture 37 of this course on Machine Learning for Earth System Science. We are currently in module 5, our last module where we are dealing with how machine learning can help in earth system models and the topic of today's lecture is Physics–Inspired Machine Learning for Process Models, Part 2.

(Refer Slide Time: 00:44)

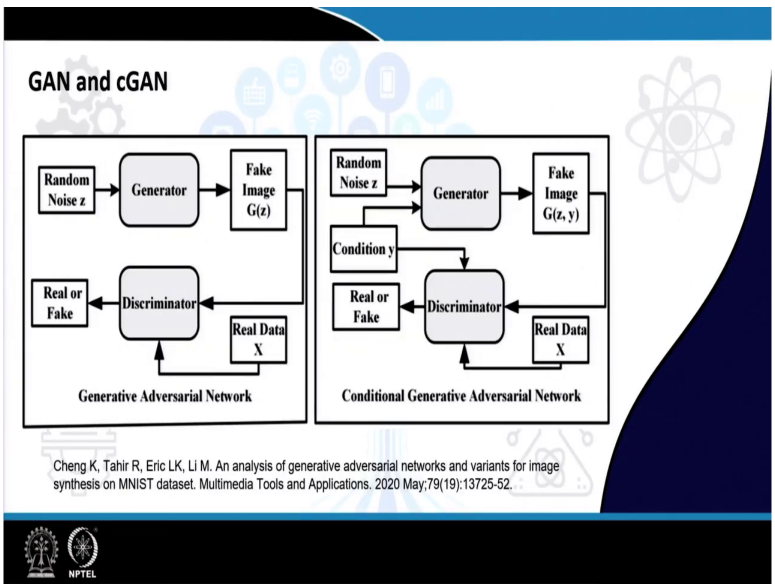



So, in the previous lecture, we had discussed how machine learning can be used for emulation of different process-based model especially like we considered the GLM the general lake model and we saw how machine learning can either emulate it or help to estimate its parameters in through a number of papers.

And, we also saw how the concept of physics inspired machine learning can help in this kind of emulation by adding a loss function that maintains the consistency of the values predicted by the ML algorithm with the physics of the system. In this lecture also, we will continue the same topic we will see further applications of PIML in different earth science system science application ah.

Additionally, like we will see how LSTM can be used for using various like dynamical processes especially those related to hydrology.

(Refer Slide Time: 01:42)




So, let us discuss about these generative adversarial networks. So, these are very important like this is a deep learning model or a series of models rather which are very important for the emulation because they can synthesize new data. We have in our one of our previous lectures we have briefly discussed about generative adversarial networks.

So, these are the models in greater detail. So, the a generative adversarial network has two components a generator and a discriminator. The generator, like in a normal GAN. So, like the aim is to produce some data like typically the images which should be indistinguishable from a given data set of images which has been used for training.

That is to say the model will synthesize new images by looking at the image a reasonable person will not be able to say whether it has like it was already there in the data set or it has been created by the image or by the model that is it should look very realistic. So, the model has two components: the generator and the discriminator. Now, this is called generative adversarial network.

This adversarial is important in because these two things the generator and the discriminator in a sense they are working as adversaries of each other. This the target of this generator is to produce fake images and the discriminator tries to classify every image as whether it is real or fake whether it came from the data set or it was manufactured by the generator.

And, like this like so, if the discriminator is able to identify that a an image generated by the generator is fake it did not come from the data set, then the generator needs to improve itself further. So, it will receive that feedback and try to improve its architecture by I mean by tuning its parameter values and so on, till it is able to produce such good fake images that the discriminator is ultimately fooled. It is no longer able to classify any given images whether it came from the like the real data set or it came from the fake data set.

Now, this architecture is useful for emulation because of two reasons: one is that because the output is basically an image so that is to say some like we have seen or we have discussed several times that like in these the outputs of these geophysics of these process models, they are often spatial maps of geophysical variables and such spatial maps are basically they can be treated as images.

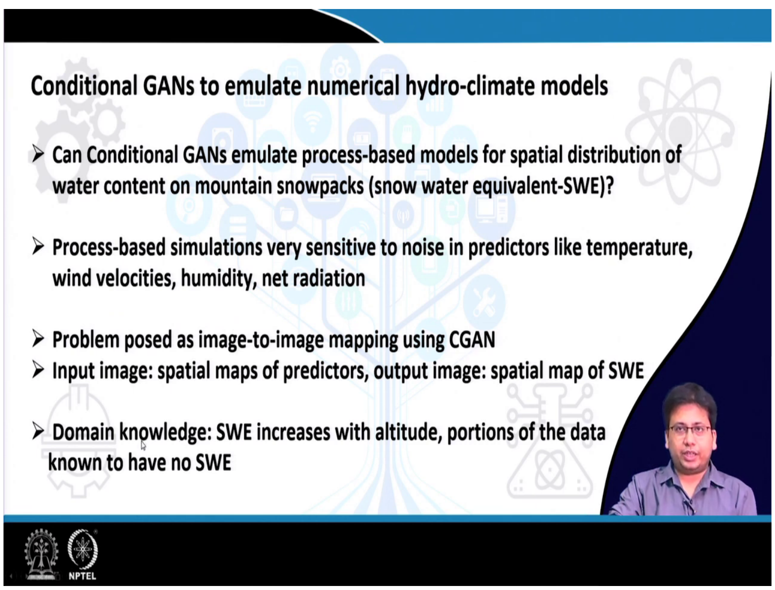
So, if we are trying to create new geophysical maps of variables, then this architecture which synthesizes new images is just suitable for doing so. Now, there is there are various extensions of this generative adversarial networks that are available in machine learning literature. One famous extension of this is the cGAN or the conditional GAN.

This conditional GAN it is same as the normal GAN except that the generator takes into its like as input a condition $y$ that is some kind of like some kind of input variable which somehow carries the information what kind of fake image it should generate. Now, what kind of information this is whether this is a binary or it is a real number or it is a whole image itself that it may vary from one application to another, but basically this condition is telling the generator what kind of fake images to generate.

As you can see the fake image in this case was just $G(z)$ that is a function of the initial random noise, but in this case it is a function of both $z$ and $y$ that is the condition determines what kind of fake image will be generated. And, now in the earth system science applications where our aim is to that is especially in case of emulation of geophysical models, typical process model what it does is it receives the spatial maps of some variables and produces a spatial map of another variable.

So, the spatial maps of those input variables they can be treated as the conditions. So, that way this conditional GAN can be very useful for this the whole idea of emulation of such process models.

(Refer Slide Time: 06:37)




So, like here we see an application of conditional GANs to emulate numerical hydro-climatic models. So, the target here is to like simulate the spatial distribution of water contents on mountain snow packs. So, this is sometimes also known as the snow water equivalent or SWE. So, like that is we have a mountain on which snow has accumulated on different heights and as we know the snow like it is frozen water.

So, the thing the idea is we want to estimate how much water is stored in different altitudes of the mountain in the form of frozen snows. So, that is what we want to find out we I mean not at one single location, but I want a spatial map of this snow water equivalent or which is often known as SWE.

So, there are some process-based models which can do it that is it takes into account the I mean the input map of where all the how much snow is accumulated; such maps can be obtained from remote sensing and it can also take into account predictor various other predictors like temperature, wind velocities, humidity, net radiation. All these things are useful in determining how much water is frozen in the form of snow.

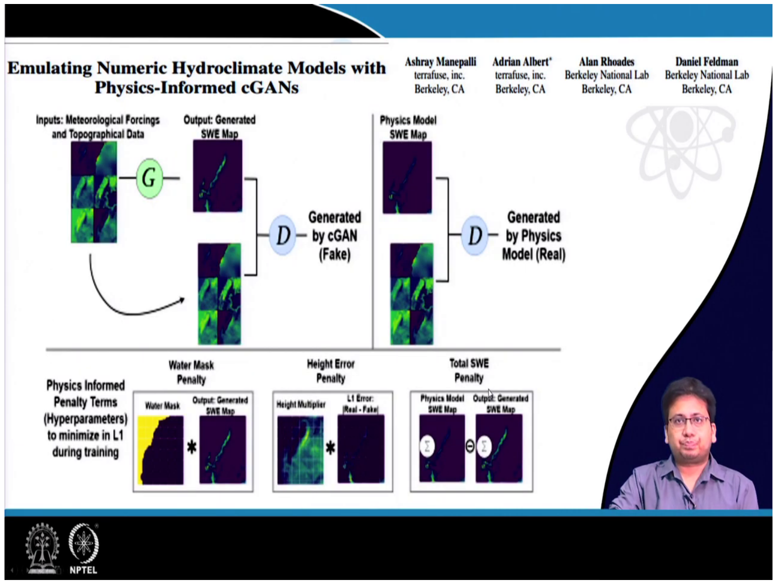
Now, these process based simulation simulators they exist, but they are found to be very sensitive to the noise in the different predictors. So, scientists have posed this problem as an image to image mapping problem using the conditional GAN as we as I just said there is an input image as the condition that is that helps the this GAN to determine what kind of image it will produce as the output.

So, the input is the image is the spatial map of the different predictors including the snow as the output image is a spatial map of snow water equivalent. The and like when we are designing this conditional GAN, we also have some domain knowledge which we can use to make the neural network architecture as physics inspired that is the physics constraints can be added as laws functions as discussed several times in the previous lecture.

So, some of the domain knowledge useful in this case are first of all the snow water equivalent it increases with altitude that should be quite clear and also the there might be portions of the data which are known to have no SWE that is they it may have no snow at all. So, that is when we are running this emulator one important input to provide is like at certain locations we are we are sure that the value of the SWE the target variable SWE will be 0. It cannot be anything else. So, that is like a hard constrain.

So, that can help the model to do the translation task that is it should know that certain values of the input variables must be mapped to zero.

(Refer Slide Time: 09:50)




So, this is what the whole thing looks like. So, this is the input variable or the map of the different input variables as well as the topographical data that is altitude and so on. So, these like these small things which you are seeing this small panels each of them are the spatial maps of a particular variable. So, these are provided as conditioned to conditions to the generator. This G stands for the generator of the cGAN and the output of it is the generated SWE map.

So, like as you can see this is a spatial map of the SWE. So, you can see many pixels having a dark value which indicate no water is there and then there are like this longest structure of green or light blue values that indicates that these are the places where water is stored in the form of snows. And, then these are some of the actual maps of this SWE content which might have been obtained from by running the process-based simulator.

So, the target of the discriminator is to like compare this map with the maps which obtained from the process-based simulation to find whether like it can distinguish or not. So, when the process converges, then the generator will be able to produce such SWE maps which will be indistinguishable from the maps that are like from the data set of the maps which have been provided by the like provided for training purposes by generating from the simulation.

This is not to say that the generated output will be exactly equal or identical to any of the maps in the data set, that is not the idea. The idea is that the statistical properties of this map various statistical property, any statistical property that you can think of it will be indistinguishable from the same properties if we calculate them from the data set, ok.

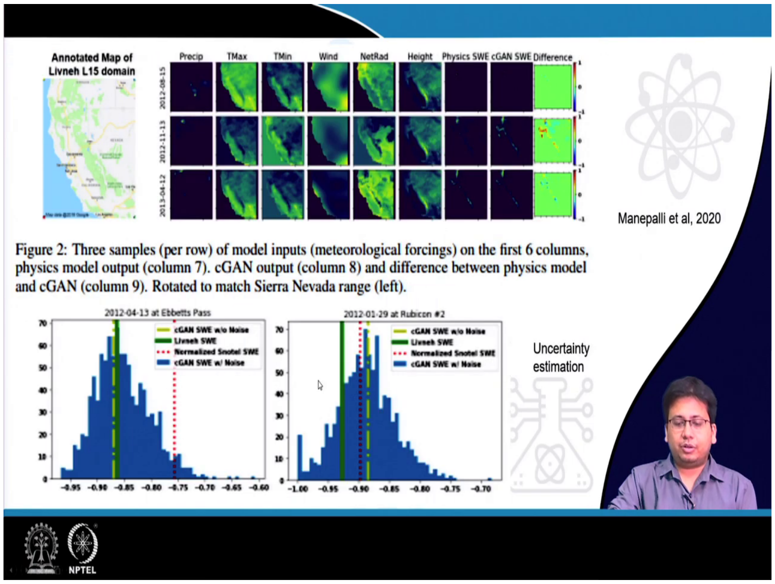
So, like the discriminator hence we will compare this the physics model SWE map with the generated like with the generated SWE maps and see if it can distinguish in any way and if it is not then we can say that the training is complete the and this generator model is now able to perfectly emulate the physics based model.

So, as I said earlier there are some penalties to be added there is the water mask penalty; that means, that there are some places which about which we are sure that it certainly does not contain any SWE maybe those places where there has been no snowfall at all or maybe those places where like which are on plain ground and so on that is. So, those places the like we can define a water mask that is it is hard coated that the SWE content in those regions are zero.

So, if the model ends up predicting some nonzero value of SWE in such places it should incur some penalty that is the water mass penalty. And, similarly the height error penalty as we already discussed the SWE increases in altitude. So, like remember that one of the input variables is the topographical map including the height of the different locations.

So, if it is found that like a higher location is containing less SWE than a lower location then in that case that will be considered as a height error penalty. So, like the total penalty comes not only from the usual reconstruction penalty that is usual in the in case of when we are training again that is in the original GAN model itself it is there for training, but in additional to that we also have these physics based penalties coming from the water mass and the height error.

(Refer Slide Time: 13:54)



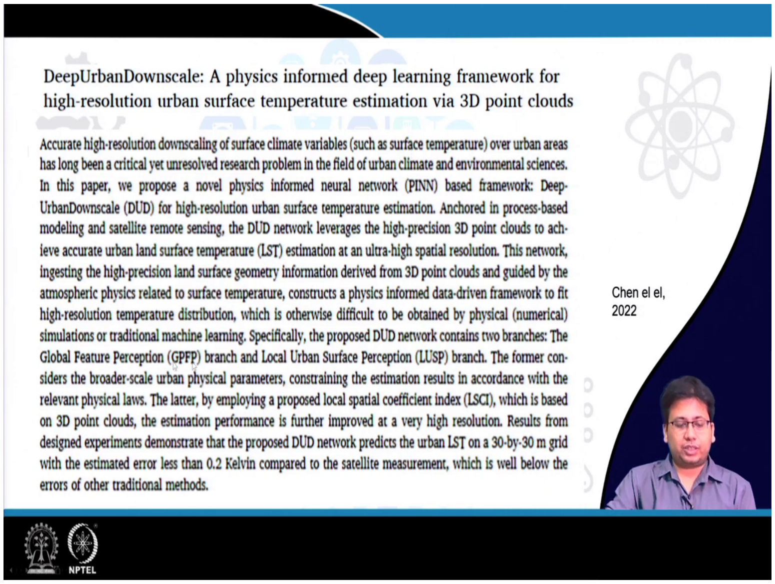

And, as a result like we are able to generate such images from the cGAN that are identical to the outputs of the physics based simulation given that the inputs are the same. So, here if you see this is the precipitation map. They are the maximum temperature map, minimum temperature map, wind speed and wind speed this net radiation and the height maps.

So, these are maps of these variables are provided as inputs to the physics based SWE simulator and these are the maps the maps of SWE which we get like this. Similarly, when all of these things are provided as the input as the condition to the cGAN these are the maps produced by it and as you can see they are almost identical to the physics SWE.

So, that we can like say that like given the same input conditions our cGAN is able to recreate or like predict ahead what the physics based SWE and simulator will predict without actually running the simulator. In other words, our cGAN has learnt to emulate the physics based SWE perfectly not only that it is also able to give some uncertainty estimates of the different variables that it is predicting.

So, this is like a plot of the uncertainty estimate. So, this is the green the yellow column is basically the mean values of the quantities as predicted by the SWE at different altitudes and locations, but. So, this is the mean value, but it can also give an uncertainty estimate so, like this is like in something like a probability distribution. So, that is what is plotted in these maps unlike the physics based SWE which may not be able to do such an estimate.

(Refer Slide Time: 15:53)




Now, we come to another application where we want to emulate another physics based model. So, here the instead of SWE our target variable is urban surface temperature. So, accurate high-resolution down scaling of surface climate variables such as surface temperature over urban areas has long been a critical yet unresolved research problem in the field of urban climate and environmental sciences.

In this paper, we propose a novel physics informed neural network based architecture deep urban down scale or DUD for high-resolution urban surface temperature estimation. Anchored in process-based modelling and satellite remote sensing the DUD network leverages the high-precision 3D point clouds to achieve accurate urban land surface temperature or LST estimation at an ultra high spatial resolution.

That is, it will receive various predictor variables as inputs at high resolution from remote sensing and so on using that using those predictors, it will create an ultra high resolution map for urban land surface temperature. This network ingesting the high precision land surface geometry information derived from the 3D point clouds and guided by atmospheric physics related to surface temperature constructs a physics informed data driven framework to fit a high-resolution temperature distribution which is otherwise difficult to be obtained by physical or numerical simulations for traditional machine learning.

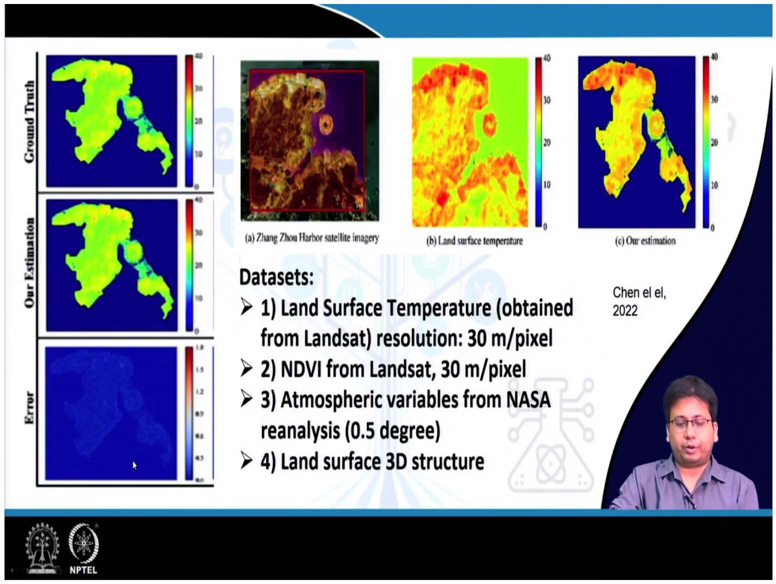
Specifically, the proposed DUD network contains two branches the global feature perception branch and the local urban surface perception branch. The former considers the broader-scale urban physical parameters, constraining the estimation results in accordance with the relevant physical laws. The latter, by employing a proposed local spatial coefficient index or which they are calling as LSCI which is based on 3D point clouds the estimation performance is further improved at a very high resolution.

So, like there are two components – one is the like the like if you consider the urban region as a whole it will have certain properties because of that is it is you can say it is overall properties, but then there are there might even within the urban region there might be some specific locations which may have like very high temperature because of some local process maybe there is some factory which like involves like which emits a lot of gases and so on.

So, in such places the temperature might be very high compared to say another part of the city which has some park or some lake. So, like these local factors are brought in here. So, it has it is these 3D point clouds to identify such local place impacts and that is done by this local urban surface perception map.

The results from the designed experiment demonstrate that the proposed DUD network predicts an urban LST on a 30-by-30 meter grid with the estimated error less than 0.2 kelvin compared to the satellite measurement which is well below the error of other traditional models.

(Refer Slide Time: 19:26)




So, this is what it looks like. So, this let us say if you consider the this is a satellite imagery of a particular region in China and this is the corresponding the land surface temperature which has been estimated based on satellite imagery. So, we have already discussed in the module 4 how the satellite imagery might be used to generate very high-resolution maps of certain variables. So, that is with or without machine learning that can be done.

We have seen in module 4 that machine using machine learning can improve the improve it, but whatever way let us say that based on satellite they have got this very high resolution map of the land surface temperature over this city. Now, this is the map which they are obtaining in another way based on the this process emulation model called the DUD.

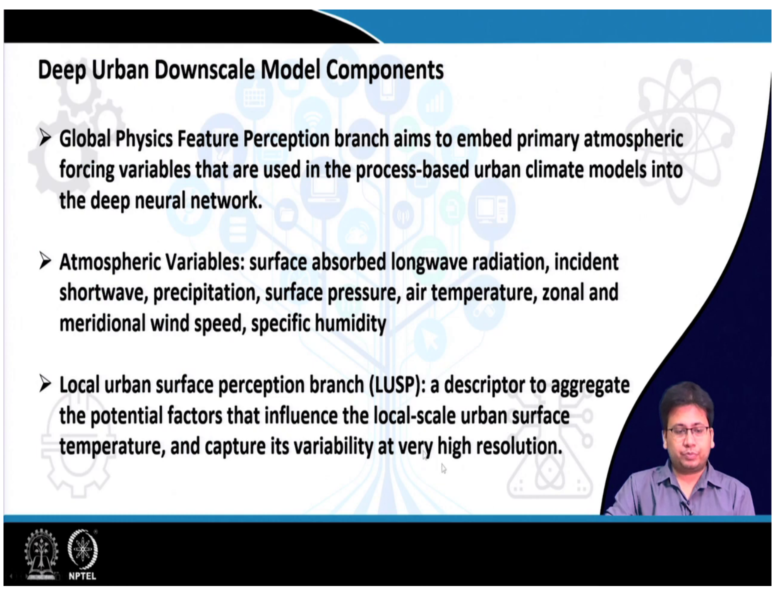
So, like here they are showing that in the like what like the temperature distribution in the different parts of the city and as you can see that they are like there are these very small local patches which are very hot like this. So, these kinds of local patches they have been able to get because of this local urban surface perception.

So, this is the ground truth and of that land surface temperature this is let us say this is this is what they have estimated and this is the difference which you can see is negligible. So, the data sets in like is of course, the land surface temperature which is obtained from Landsat satellite at a resolution of 30 meter per pixel. This is the ground truth data.

And, now for the predictors they need the NDVI from the Landsat where the same resolution we have already discussed what this what NDVI is the and why basically is the vegetation index. So, we can like understand intuitively that low NDVI may mean higher temperature and vice versa. And, apart from that, we have various atmospheric or meteorological variables which are obtained from NASA reanalysis at a much coarser resolution at 0.5 degree.

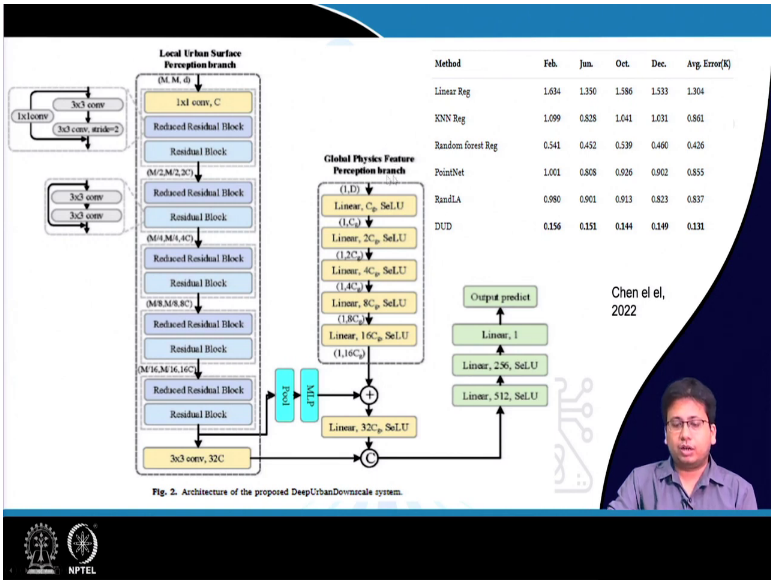
So, these are the like these are the variables which will like impact the overall temperature of the region and this will be input to the this global feature perception, while the second one the NDVI and so on this will be more for the localized effects, and then we also have the land surface 3D structure, the 3D point cloud which they mentioned in the abstract that is also an input.

(Refer Slide Time: 22:02)




So, based on the global feature perception map the branch as already mentioned.

(Refer Slide Time: 22:09)




So, this is basically the what the neural network looks like. So, like as you can see these are the inputs the input images of the different variables which we just mentioned the NDVI and the high resolution NDVI as well as the low resolution meteorological variables. So, you can see it is mostly a convolutional neural network with residual blocks also. So, this is the local component and this is the global component. The global component like this is like relatively simpler I mean all like straight forward convolution sequence of convolutional layers with the SeLU activation function.

And, so, what happens here is that is the what is produced by the global physics perception branch and what is produced by the local urban surface perception branch these two are like these two are merged together again through a like that is first they are they are merged and then treated with another round of convolution. And, then followed by several these fully connected operations, like these are all linear I mean the function here is linear along with SeLU to get the final prediction which is the very high resolution map.

So, the global physics feature perception branch it aims to embed the primary atmospheric forcing variables that are used in the process based models. So, like if you are like they so, the benchmark physics based process model which they are using that receives the all the atmospheric variables at a low resolution. Those are those variables are provided as the inputs to this GPFP branch.

And, so, the atmospheric variables in this case are surface absorbed long waves radiation, the incident shortwave, precipitation, surface pressure, air temperature, zonal and meridional wind as well as specific humidity. So, all of these variables they have a bearing on the land surface temperature as you can understand.

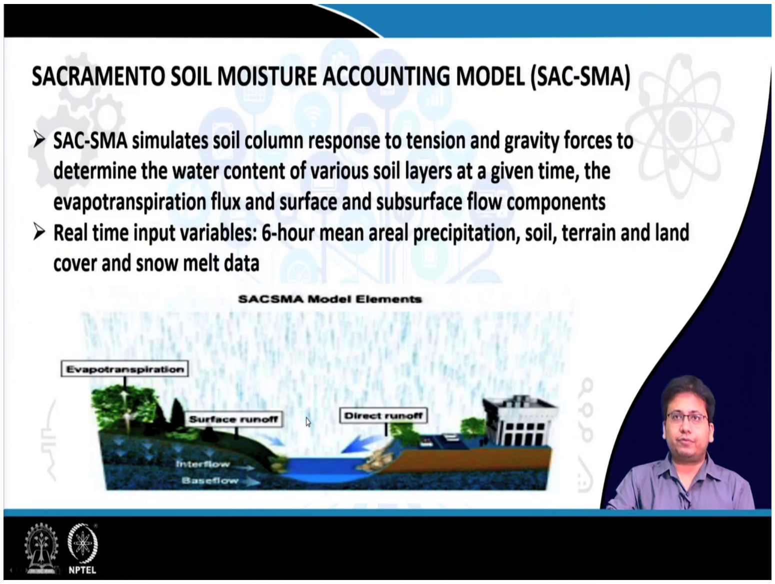
And, then the there is the more detailed the local urban surface perception branch it is basically a descriptor to aggregate the potential factors that can influence the local scale urban surface temperature and also capture its variability the spatial variability at very low resolution at very high resolution. So, that is like that is the task of this block which is evidently more sophisticated with like various with so many layers followed by convolution residuals operations and so on.

So, these are some of the results which they have got compared to the. So, linear regression, KNN, random forest etcetera these as you can understand are conventional machine learning models which do which neither take these kinds of like which do not which are not based on these kind of physics based architecture and then it is also compared against these physics based models like RandLA and point net etcetera.

But, compared to that the proposed architecture as you can see the RMSE of the predicted temperature at the different locations at very high resolution is very significantly lesser than what is incurred in by any of the other models, ok. So, that is how they so, this is like unlike the other physics informed neural networks where they were building a physics based laws function, in this case as you can see there is no separate physics based laws function. Rather the architecture itself is physics base.

The I mean in the physics is incorporated in the architecture by itself by having a separate global component and a local component and in the local component again it receives the NDVI map and the local 3D point cloud. So, those have are like treated in suitable ways by this kind of residual blocks and so on.

(Refer Slide Time: 26:24)




So, here it is a case of the physics informed architecture. Now, let us come to some other applications in hydrology. So, like there are lots of models in hydrology especially for the for calculating the runoff. So, like we know that like when there is like all the rain the rivers and so on, they create the water to flow from the land to the sea or to lakes or whatever. So, that is basically the runoff from the surface and which can be either from the surface or like it can be directly also due to like due to rainfall and so on.

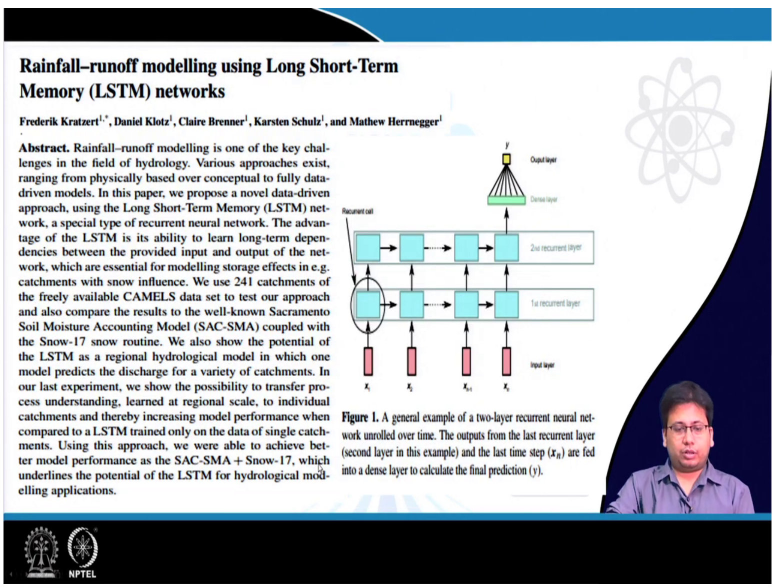
So, the like it can be this kind of runoff can either be direct discharge into the like into the water body like say for example, the river flowing into the water bodies or even sewage water from the from a city or from a factory etcetera flowing into the flowing into the target water body or it can be a surface runoff where like we know that whenever there are trees then like there is this evapotranspiration happening as a result of which water is stored in the ground.

And, that ground water can like flow into the like from the ground into the water body. So, that is the that can be called as surface runoff. So, this calculating the amount of runoff is like one of the most important problems of hydrology and especially how the runoff varies as a function of time. So, this is very like especially if you are considering the runoff into a large river or a sea etcetera it is very important to determine the currents and so on and as well as other properties.

So, what these there are models to simulate this they are one of the most widely used hydrological models for runoff is the SAC-SMA model sacramento soil moisture accounting model. So, it simulates the soil column response to tension and gravity forces to determine the water content of various soil layers at a given time, the evapotranspiration flux and surface and subsurface flow components.

The real time input variables to the SAC-SMA model is a 6-hour mean aerial precipitation, soil, terrain and land cover and snow melt data. So, these are all provided as inputs at as real time. So, as you can see 6 hourly based on that the this model calculates what is the like in real time what is the amount of runoff.

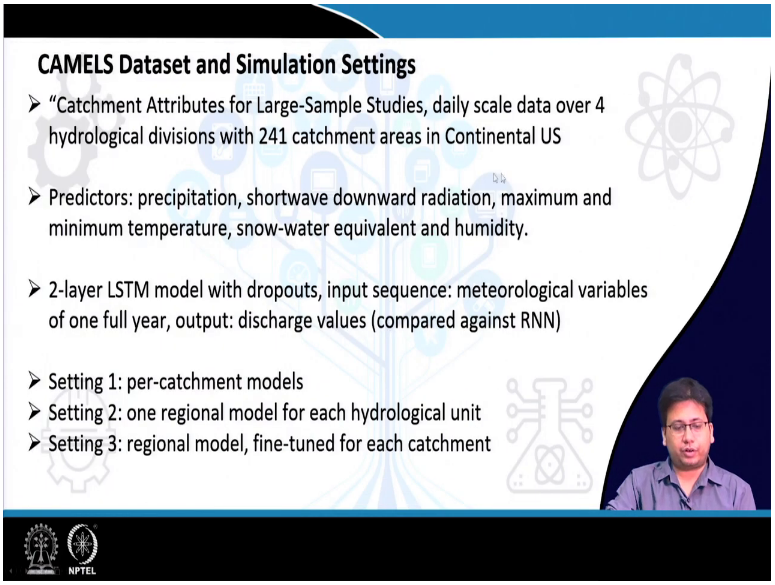
(Refer Slide Time: 29:07)




So, the here the aim is to emulate this kind of SAC-SMA model using like models like using basically machine learning. So, as it is a dynamical process basically a time series of the land of the runoff needs to be produced, so, they are using LSTM. So, this is basically an like it is a two-layer recurrent neural network which is as you can understand it has been unrolled over time.

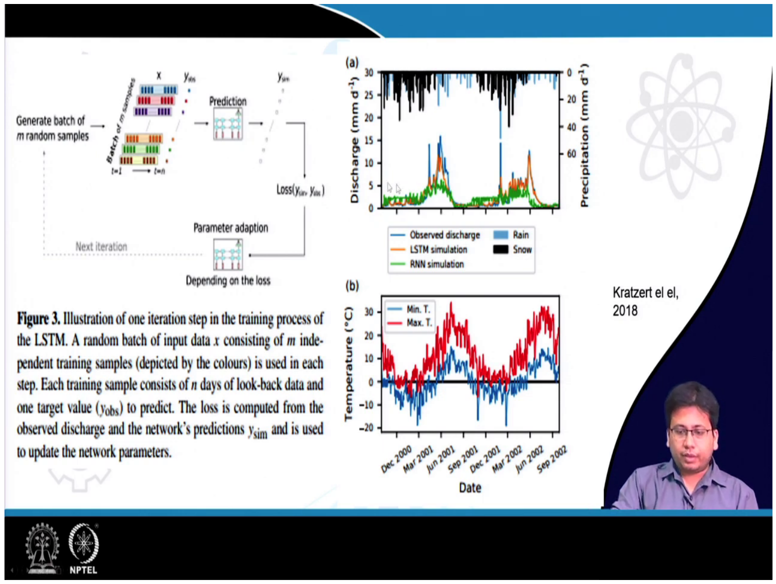
So, the outputs of the last recurrent layer and the last time step are fed into the dense layer for the final prediction the final. So, the input are the those atmosphere these all these real time input variables which are also provided to the SAC-SMA and the output is the amount of runoff.

(Refer Slide Time: 29:54)

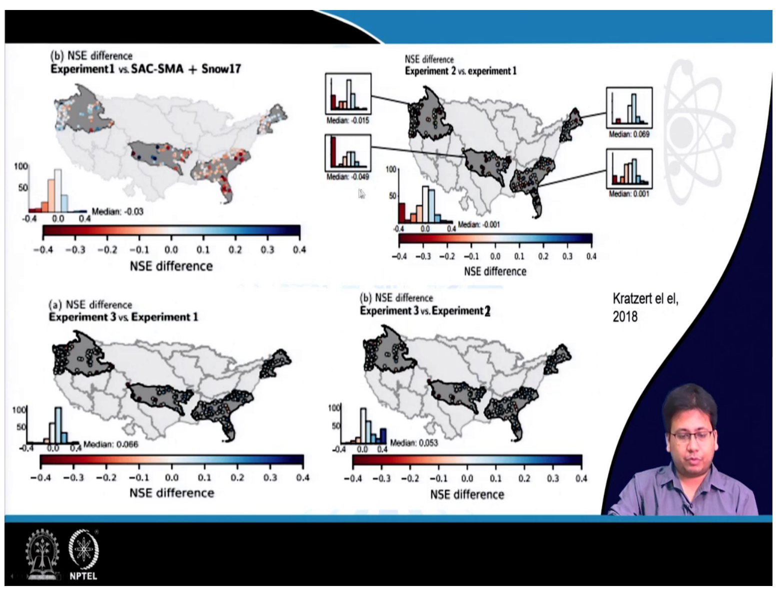



So, in this case in the paper that we are discussing here like the catchment attributes for large-sample studies has been like it is available at a daily scale over 4 hydrological divisions in the continental USA with 241 catchment areas.

(Refer Slide Time: 30:17)




(Refer Slide Time: 30:13)




So, like these are some of the four the so, these are the four hydrological units as you can see in four different parts of us and each of these has several catchment areas. So, like these dots basically. So, there are 241 such catchment areas where they have done their study. The predictors like are the same predictors as provided to the SAC-SMA model. So, and the model here is a two layer LSTM model, they have compared the LSTM and the simple RNN the LSTM model has dropouts also.

So, the input sequence will be these meteorological variables for one full year and the output will be the discharge values. So, they have like. So, as far as the model is concerned there is no innovation here it is like a simple like LSTM just with two layers the. So, there are they have done this experiment in two settings. One is for each catchment.

So, there are 241 catchment areas one is that you can build the separate model for each catchment in the second one setting they have like one regional model for each hydrological unit. So, there are four hydrological unit. So, each one has a different model, but all the catchment areas within a hydrological unit will have the same model.

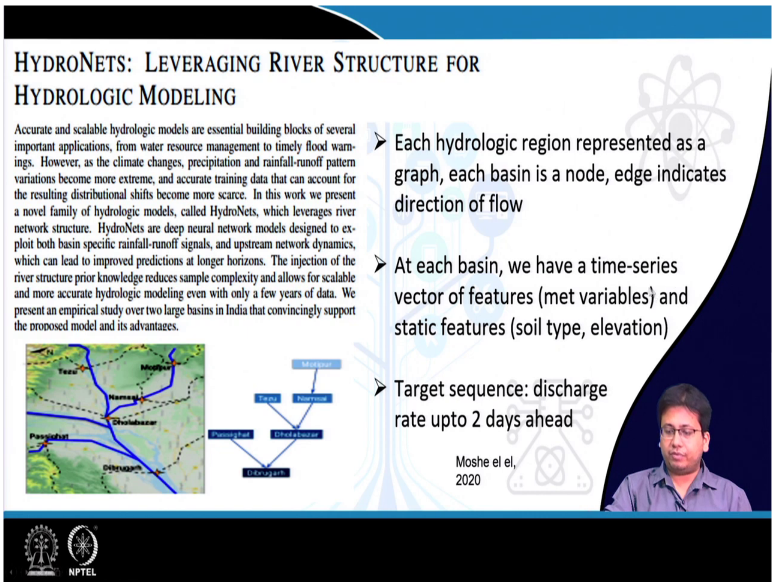
The third is to have a regional model alright as in the setting 2, but for each catchment it will be fine tuned that is the certain parts of the LSTM will be relearned for the each catchment. So, like this is the broad framework where the and this is what the simulation results looks like. So, they are simulating the.

So, the inputs include the precipitation which is one of the input variables. So, that is plotted here like on the top and on the bottom they have like plotted the discharge rate obtained from the different sources. One is the observed discharge the blue the next is the RNN the LSTM simulation which is in orange and the green is the RNN simulation. So, as you can see the orange and the blue tend to overlap meaning that the LSTM is able to like map the observed discharge more closely than RNN.

So, see the similar thing they can do for temperature also. So, these are the results of the different settings as we have shown. So, in the first one the like the proposed model is compared against the SAC-SMA process based model and like a small improvement is observed. Not a great improvement, but a small improvement especially in different places which means that on an average this the proposed LSTM based model is just about as good as the process based model while definitely it takes significantly less time and then they have repeated the experiments in the different settings mentioned.

And, they find that the third setting where they I mean both the second setting and the third setting are good, but the third setting is the best where they build separate models for each hydrological unit, but then within for each catchment area within the hydrological unit they fine tune the model.

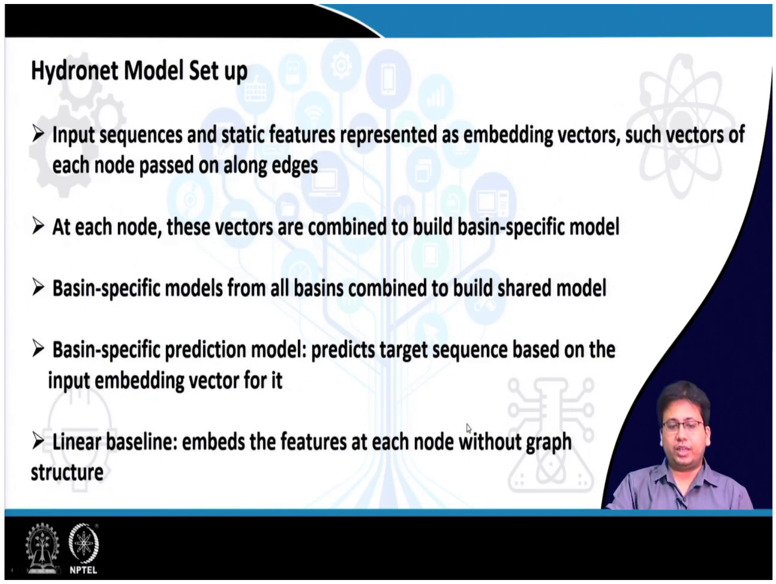
(Refer Slide Time: 33:29)




And, then there is one final paper will just briefly came through this is hydronets where they try to leverage the river structure for hydrological modelling. So, they have they consider two river basins the Ganga basin and the Brahmaputra basin in India. So, they have several like the basin is divided into several hydrological region. Each it is represented basically the basin is represented as a graph sorry, I mean the hydrological region is represented as a graph and each basin is a node and the edge indicates the direction of the flow.

And, in each basin we have a time series of various meteorological variables which are it is input as well as certain static features like the soil type. The target is to like at every basin they want to predict the discharge rate up to two days in advance.

(Refer Slide Time: 34:22)

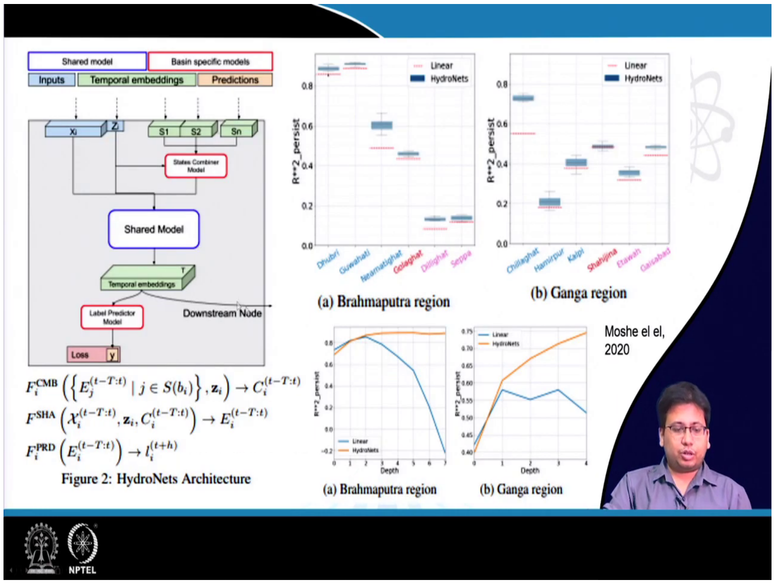



So, the curious thing is that they have all these graph structure and as well as the inputs they have somehow represented as embedding vectors using some kind of graph embedding techniques that are common in machine learning. So, at each node, these vectors are combined to build a basin specific model because a node is corresponds to a basin and the basin specific models from all the basins are combined to build a shared model of the whole region.

Now, the basin specific prediction model is used to predict the target sequence based on the input embedding vector and it is compared to a linear baseline which just embeds the features of every. So, at every node they have these input features the met variables and so on.

So, the baseline is at every node you predict like you train a model to predict the discharge at that basin based on its local parameters, but without taking the overall structure of the like of the catchment area as its input.

(Refer Slide Time: 35:26)




So, this is the model which they have developed as is a as mentioned there is a shared model as well as a like a specific model. So, they have this that is their hydronet model and they have showed that the results they get this way are superior to the linear baseline where they predict the discharge rate at each location separately.

(Refer Slide Time: 35:42)

So, these are the different references for the four papers that we discussed today. So, that brings us to the end of this lecture. So, in the next lecture, we will see some further applications of machine learning in earth system models.

So, till then, bye.