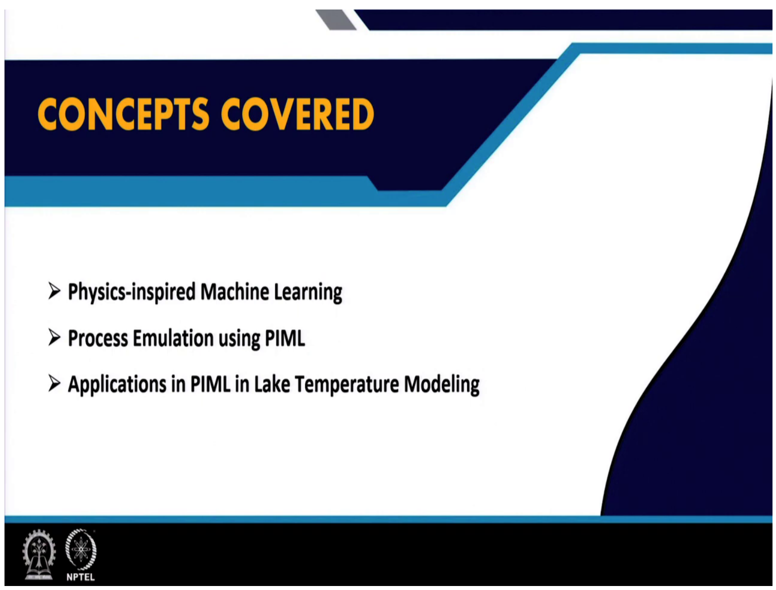
Machine Learning for Earth System Sciences
Prof. Adway Mitra
Department of Computer Science and Engineering
Centre of Excellence in Artificial Intelligence
Indian Institute of Technology, Kharagpur

Module - 05
Machine Learning for Earth System Modelling
Lecture - 36
Physics-Inspired Machine Learning for Process Models - 1

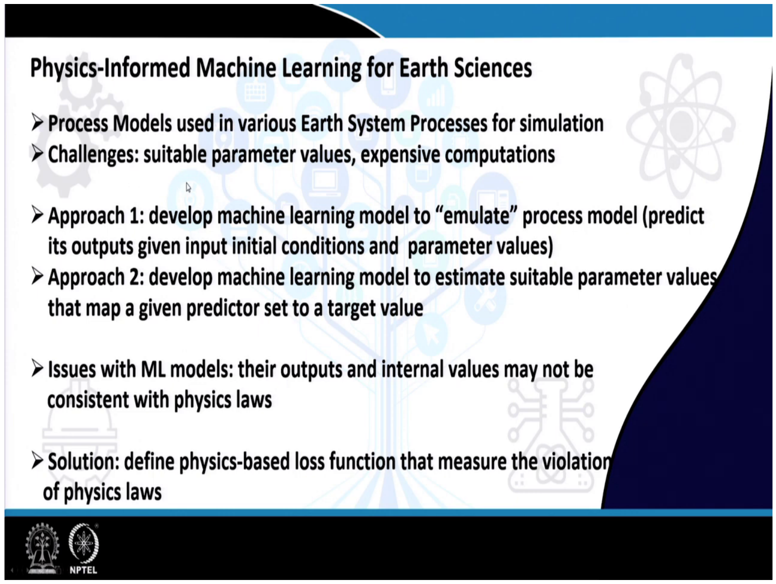
Hello everyone, welcome to lecture 36 of this course on Machine Learning for Earth System Science. Like we are currently in module 5 the last module where we are dealing with how machine learning can be used for earth system modeling and the topic of this lecture is Physics-Based machine Learning for Process Models.

(Refer Slide Time: 00:45)



So, the different concepts which we are going to cover here are physics-inspired machine learning and how we can emulate process-based models using such a using this PIML. And we will also see one specific application of PIML in lake temperature modeling.

(Refer Slide Time: 01:03)



So, first of all what is this PIML all about? So, we have like in the last lecture we were considered process-based models versus statistical models. Now these process-based models are used in various earth system processes for simulation. Now they these suffer from multiple challenges like suitable parameter values are often not known, the computations are very expensive require a lot of time and a lot of computing power and so on.

So, now, in what ways can machine learning alleviate such problems? So, one approach is to develop such models which can emulate these process models. That is to say you will you do not really need to run these process models the key here is that most of these models are deterministic that is if you set certain initial conditions like its determine like a that is there can only be one particular output or I mean or final conditions that can arise. The question is what that will be?

So, this can be considered as something like a machine learning problem like which is to like map the initial conditions to the final conditions. So, like we can aim to train our machine learning model which will simply be able to predict what the model will still what it will be able to predict the predictions by the model by the process-based models.

That that is if you give the initial conditions then your machine learning model will be able to predict what will be the values of the final condition which the process-based model will come up with. So, that is called emulation of these process-based models.

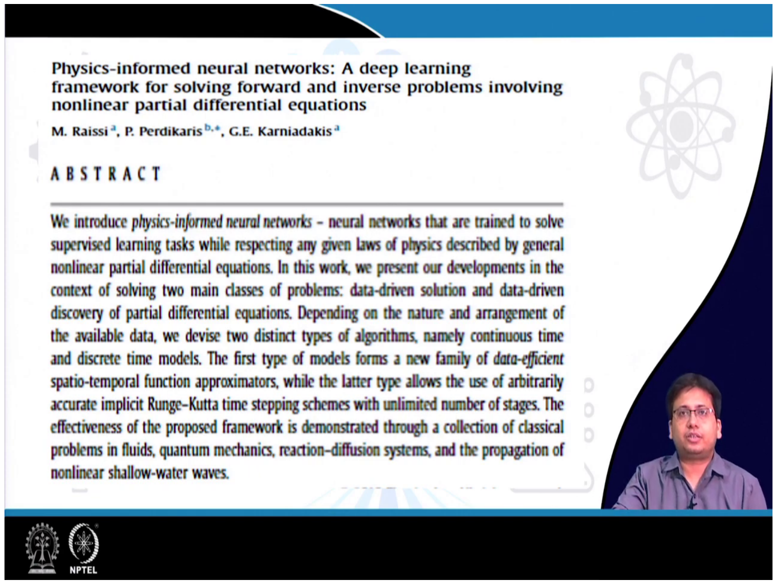
And a second approach is to develop a machine learning model to estimate suitable parameter values that map a given predictor set to a target value. That is like a more principled way of estimating the parameters. Instead of some experts specifying the parameter values and using that same parameter values all the time like this approach is here you like let us say we already know the initial conditions as well as final conditions.

How we know? That is from the observations. Then we will see whether the model is actually able to map those initial conditions to the final conditions. And then and if so for what values or what settings of the parameters? So, that is the like the way of data driven parameter estimations.

Now these kinds of models or machine learning based models which we are talking about one like one obstacle of using machine learning for such problems is that the they their outputs and also the different internal values which they generate they may not be consistent with the laws of physics.

By internal values, I mean suppose we have a machine learning model which consists of different like variable which deals with various different variables of the process. So, like if you are simulating over a particular time sequence at different points of times the values of those variables will be estimated and so on. Now those estimates may not be consistent with the laws of physics. So, the solution to this is to define a physics-based loss function that measures the violation of the physics loss.

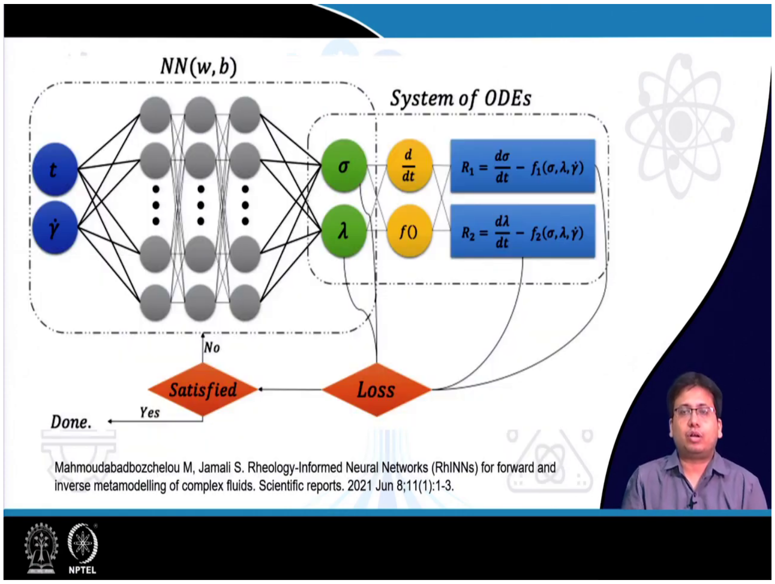
(Refer Slide Time: 04:43)



In recent times there is this paradigm called physics-inform neural networks which has like emerge and this is a famous paper which uses physics these physics-informed neural network. That is a deep learning framework for solving forward and inverse problems involving non-linear partial differential equations.

So, most of the process-based models are actually based on differential equations that is I already mentioned governing equations in earlier model. So, earlier lecture often those governing equations are actually differential equations.

(Refer Slide Time: 05:13)



So, what this paper talks about its like using a deep learning framework for solving these kinds of differential equations. And to solve like one like we can say one technology which becomes very relevant in solving these differential equation problems is that of automatic differentiation.

Now this is what are physics-informed neural network looks like. So, let us say that just focus on these two things d's. Let us say that there are two variables σ and this λ and both of these are dependent on time and there is a particular like very parameter called γ. You can treat it as something like a model parameter.

So, let us say that these are the governing equations of the model of the process-based model. So, that is

$$d\sigma/dt \ = \ f_1(\sigma, \lambda, \gamma)$$

$$d\lambda/dt \ = \ f_2(\sigma, \lambda, \gamma)$$

So, these are the two model equation which specify the process-based models. So, the process-based model will of course, start with some initial values of all these quantities and at every given point it will calculate these derivatives and then update the two variables and then based on that and then for following time step again it will do this and so on.

So, like but these the time steps that are being considered here are of course, infinitesimal. So, if I have to even if I have to simulate for a particular let us say 5 seconds or something like that to for that simulation it will have to carry out so many of these variable updates. Because at every step it is the time is advancing by only an infinitesimal step.

So, the so, that is going to take of course, a very a lot of time and this is still a very simple model. So, the what in case of neural network what we will aim to do is like we will specify the time and we will specify the model parameter. The model really the neural network really needs to predict the values of σ and λ at that time for that for that value of the parameter, ok.

So, the neural network is of course, the like universal function approximator. So, f_1, f_2 all these functions it will it should be able to approximate provided it is trained with a lot of data. But whatever values it estimates that is σ and λ, can we really guarantee that it will be; it will be following these two constraints that is can we will can we say that $d\sigma/dt$ will actually be equal to this function and $d\lambda/dt$ will be equal to this function? That in general we cannot say.

So, what we will do is, once we get the values of σ and λ being produced by the neural network, we will actually calculate $d\sigma/dt$ and $d\lambda/dt$ from the generated values using automatic differentiation. And we will also actually compute the these functions. So, based on that we will actually calculate that these two losses R_1 and R_2. So, as you can see R_1 is how much the will like estimated value of σ that is for the estimated value of σ are these two equal or not.

If they are not equal then what is the error? Similarly for λ also are these two things equal or not? If they are not equal then how much is the error? So, like if σ and λ are different the predicted σ and λ are different from what they should be then of course, these errors will be large. So, accordingly we need to like update the values of σ and λ. And how will such update happens? The update will of course, happen if we change the different weights of the neural network.

So, they are like we will define this kind of a loss function which like which will help to update the weights of the neural network. So, the in this case the loss function will include the this physics loss the that is which is measuring how much by how much the predicted values are differing from what they should be.

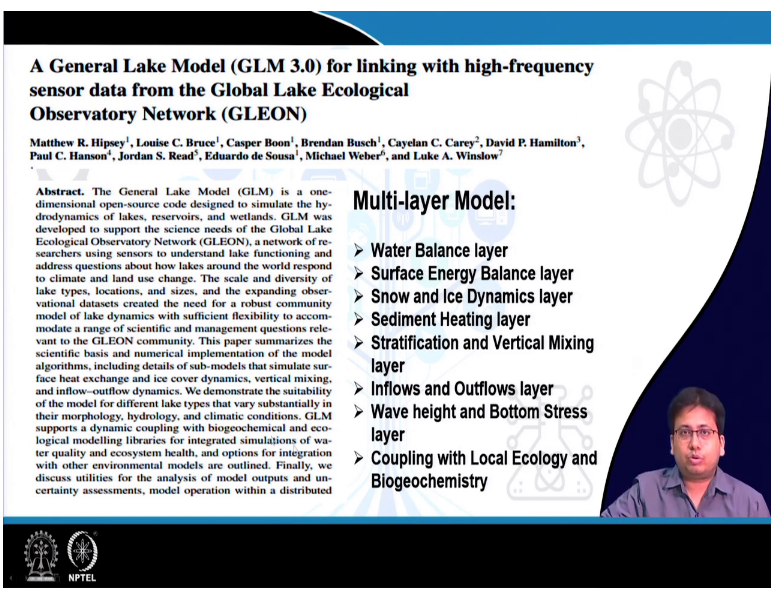
So, like there are now two different kinds of loss functions. One is the usual loss where we are simply comparing the simulated values of σ and I mean the predicted value of σ and λ with the like what the process-based model would have given. Like if we had actually solved these differential equations and till the time t and obtained the values of σ and λ.

So, let us call those as σ^ and λ^. So, one a one part of the loss function is coming by comparing σ with σ^ and λ with λ^. The other part of the loss is coming from the physics-based loss. Which is basically calculating by how much these two like whether these two equations have been satisfied or not? And if not, then what is the amount of error?

So, like we can say the prediction loss as well as the physics loss. So, these two things. So, like the like as you can see this is the loss function. So, the loss function is getting its input from two things. One is from the σ and λ themselves that is the prediction loss and the other is from these R_1 and R_2 that is the physics loss.

So, these two things are combined and the to get a new loss function. And based on the and it is this combined loss function that we try to minimize through back propagation and so on. By which we update the different weights of the model till we have been able to get the optimal set of weights. So, that is how the this PINN or physics-inform neural network works.

(Refer Slide Time: 11:46)



So, in these lecture and also in the following lecture we will see several applications of the of this kind of idea where we will see the loss function of a machine learning model being used to like capture some kind of physical constraint like this. So, in today's lecture we will consider one particular application of lake temperature modeling.

So, suppose there is a lake, I want to estimate the different temperature I mean the temperatures at different heights of that or depths within that lake. Of course, I cannot measure everywhere. So, I have only a few in-situ measurements at the different depths and different locations. I want to get that temperature profile of the entire lake.

Now, the first thing is there is a process-based model like because as I said there are only in-situ measurements. So, we do not have the full data. In the absence of the full data, we have to depend on what is simulated by a process-based model which the experts have developed over the years.

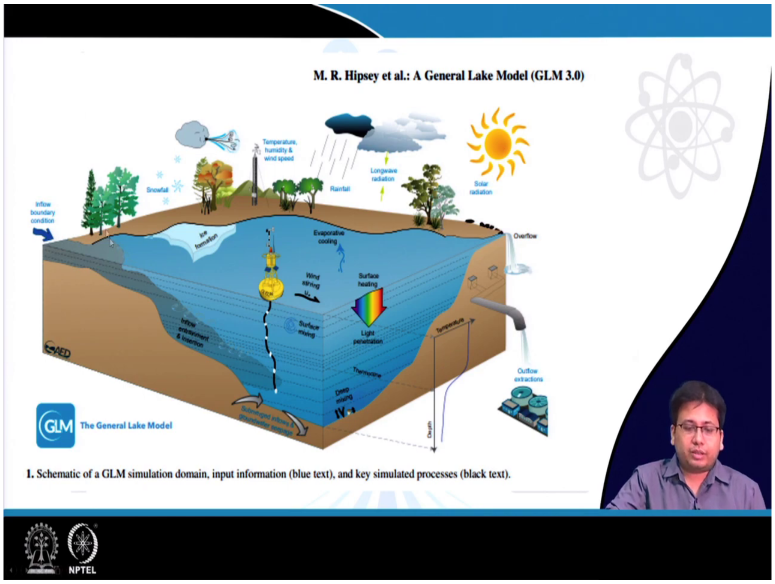
So, this is what the process-based model which we are talking about its a general lake model also known as GLM. It is used to it is like it is basically developed using some like in one for one particular lake like which is provided with lots of sensors at different locations and so on. But the aim is to apply it for any lake.

So, this general lake model it is a like its a very highly complex elaborate model with multiple layers. So, these are the different layers which are mentioned the water balance, the surface energy balance, the snow and ice dynamics and so on. Of course, this is not this is not relevant to many lakes especially those in the tropical regions.

But in the extra tropical regions where ice forms on the in the lakes this is an important thing. Apart from that sediment heating, stratification and vertical mixing, inflows and outflows, the wave height and the bottom stress then and as well as the lake will have its own like organisms living organisms living in it the then the plants and the which have grown all over.

So, like so, there is some kind of bio geochemistry also involved. So, the interactions of the lakes water with this ecology and biogeochemistry these are also taken care of by some model.

(Refer Slide Time: 14:19)

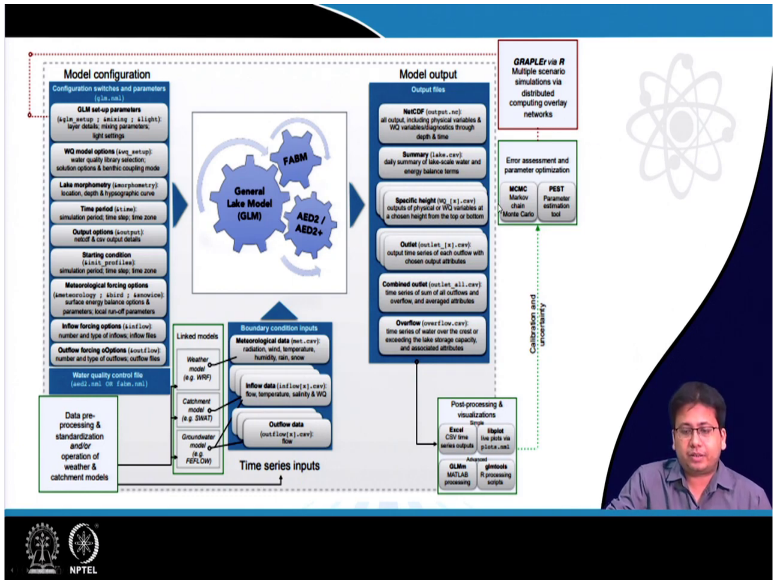


So, typically this is a schematic diagram of the lake model. So, as you can see like this is a lake like. So, first of all there are some external factors like the rainfall, there is radiation of I mean there is solar radiation incoming solar radiation and then outgoing longwave radiation part of it might be blocked by the clouds etcetera.

And these like the solar radiation and that is how much solar energy it is receiving from the sun I mean incoming and how much of it is outgoing? So, these also have a bearing on the temperature of the lake. Then the lake might overflow there might be it might be flowing out to different rivers or canals which may be connected with it.

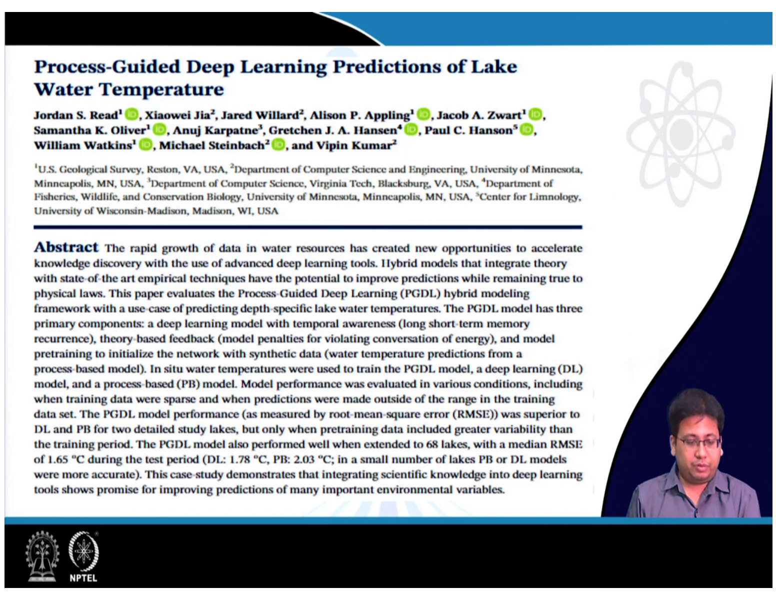
And then there is a depth profile at different depths the different properties might change, then there are these like its possible that there may be snowfall in one like in some places around the lake. As a result of which ice may form on the lake. Then as the wind blows waves can be created in the lake and according to the heights of the waves the temperature profile might also change. There can be shear stress as the water is flowing and as a result that might also have some bearing on the temperature and so on.

(Refer Slide Time: 15:47)



So, like they have developed this kind of a very elaborate models like they the model has so many different modules. Each of these modules they pass on they have their first of all they have their own parameters and they like they exchange information between the different modules and so on. So, like as you can understand this is a very highly complex and elaborate model.

(Refer Slide Time: 16:11)



Now, the aim is and simulating it will also require like a lot of time and a lot of observations. So, like and lots of parameters. So, this has been perfected for maybe one lake for which we have a lot of observations like a meticulous deployment of sensors of various kinds. Now the; however, this model is considered supposed to be a general lake model.

So, that it should be applicable to any lake even if we do not have so all these observations. So, the aim here is to see if machine learning can emulate this kind of the general lake model. So, that is what we will see in the three papers three very related papers which we will discuss one after the another now.

So, the first paper process guided deep learning predictions of lake water temperature. the rapid growth of data in water resources has created new opportunities to accelerate knowledge discovery with the use of advanced deep learning tools.

Hybrid models that integrate theory with state-of-the-art empirical techniques have the potential to improve predictions while remaining true to the physical laws. This paper evaluates a physics guided deep learning hybrid modeling framework with a use case of predicting depth specific water lake water temperature. The PGDL model has three primary components.

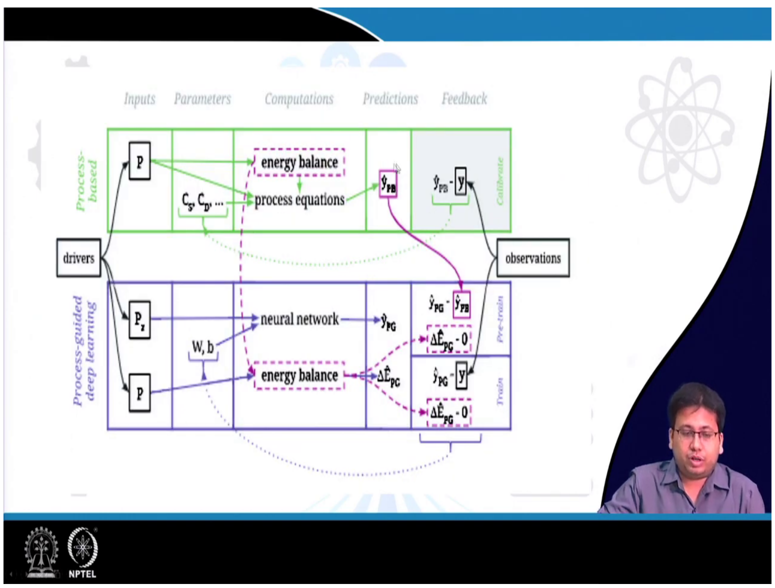
A deep learning model with temporal awareness that is a long short-term memory, a theory-based feedback which includes model penalties for violating conservation of energy and model pretraining to initialize the network with synthetic data that is water temperature predictions from a process-based model. The in-situ water temperatures were used to train the PGDL model a deep learning model and a process-based models.

Model performance was evaluated in various conditions, including when training data was sparse and when predictions were made outside of the range in the training data set. So, these are various like ablation studies which is necessary to do in these kinds of things. Like when we are training machine learning model to emulate the this kind of a process-based model it is important that we it should not require too much of training data. Because we have already said that these models are expensive to run.

So, if you need to too much of data for training the model then; that means, they are just to train the model we will have to run the process-based model lots of times maybe lakhs or millions of times. Now if that you are doing anyway then why then what is even the idea of having the machine learning model, right? So, the model should be able to work quite well even by with a small amount of data that might be obtained by running the process-based model only a few times.

The PGDL model performance was superior to the DL and PB for two detailed study lakes, but only when pre trained data included greater variability than the training period. The PGDL model also performed well when extended to 68 lakes, with a median RMSE of 1.65 degree Celsius.

(Refer Slide Time: 19:50)



So, like as I said earlier the model the aim is to develop the model for one lake and then use it for multiple lakes. So, this is a like rough scheme of the model the machine learning model being used in this case is a simply is a LSTM. So, as far as that is concerned there is no great innovation in this paper. Innovation lies in the way it is applied.

So, there are these drivers or the covariates which are used in both cases. So, first like on one hand there is the process-based model. The GLM with the general lake model which we just discussed and parallel to that is the process guided deep learning model which is aiming to emulate it.

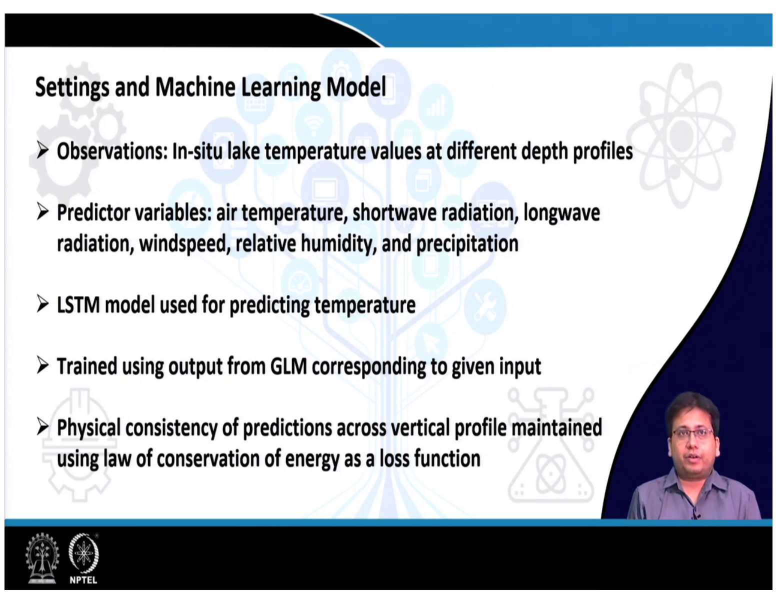
So, this process-based model is of course, based on the another concept of energy balance and water balance and so on for that it has its own process equation. So, it like from certain initial conditions it will produce its predictions and then while we are training the neural network it will receive the drivers as well as the in initial conditions and the it will know that the what its ideal outputs will be as that is the predictions by this process-based models.

So, accordingly the neural network will be trained. but there, but while it is being trained it is like just as it happened in this case its not only important to make the predictions properly, but it is also necessary to make sure that the predictions are satisfying the different laws of physics.

So, that is why the energy balance its like which is of course, a very important concept in this in the process-based models. We will have to see whether the like whatever values is being predicted by the neural network whether they are maintaining the energy balance or not?

So, we will define a loss function accordingly which basically sees how much of discrepancy has arisen based on what the neural network predicted. The discrepancy in terms of energy balance and then the aim is to of course, to minimize this discrepancy. So, that acts as a some kind of a feedback to the neural network and then the neural network again like updates its weights and so on.

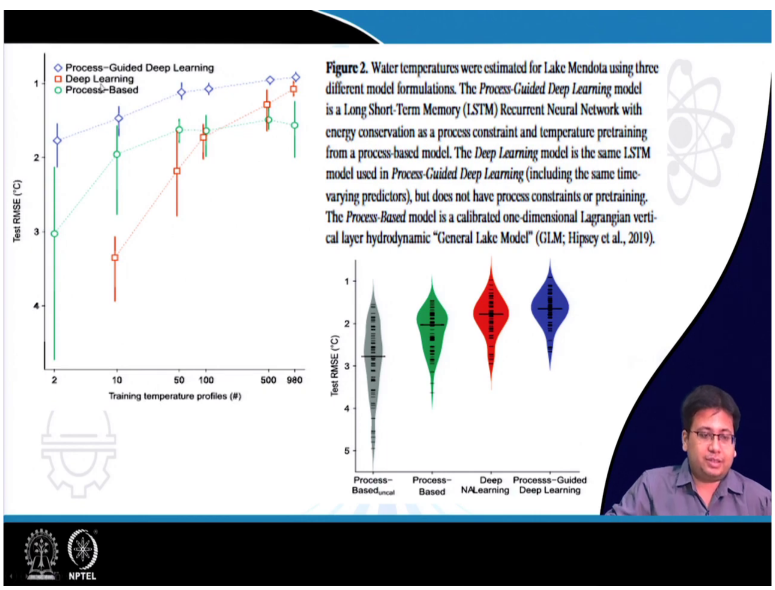
(Refer Slide Time: 22:11)



So, here the settings are as follows. So, the observations are the in-situ lake measurements at different depth depths and the predictor or the covariates which we considered here at the air temperature, short wave radiation, long wave radiation, windspeed, relative humidity precipitation, etcetera. So, in this model like it already this model already shows how all these covariates can like influence the lake water temperature at the different depth. So, they are all provided as covariates.

Now, the LSTM model is used for the predicting of temperatures the it because LSTM is used because LSTM is a sequential model and it is necessary for the predictions to be temporally to be to be temporally consistent. And that the train it is trained using output from the GLM corresponding to the given input. And the physical consistency of the predictions are maintained across the vertical profile using the laws of conservation of energy using the like as a loss function, ok.

(Refer Slide Time: 23:16)



And so, like then comes the validation part. So, like the RMSC that has been obtained like at different temperature profiles that is I mean to say at different depths of the lake this will this kind of thing has to be studied.

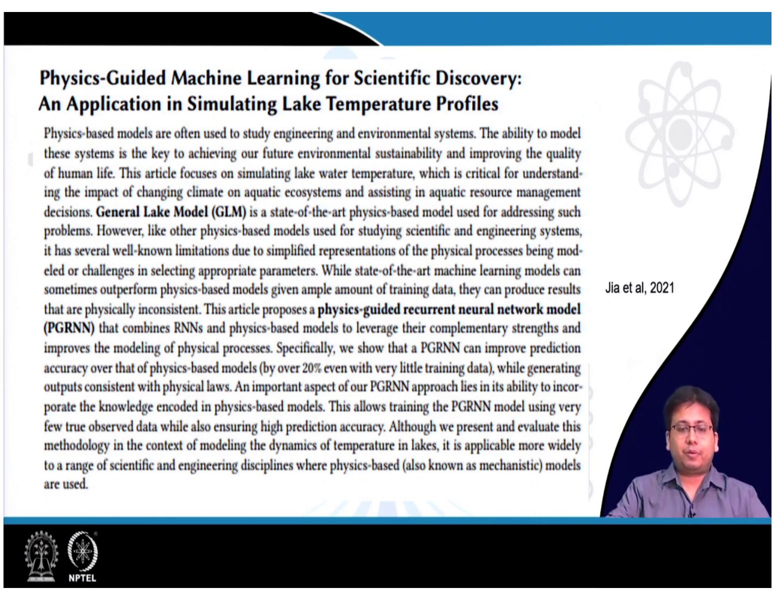
So, they have considered, they have compared the process-based models in green, the red is just the deep learning if you do not consider this energy balance loss function just if you train a neural network to produce the produce to predict the output from the input. So, that is the blue deep learning and red is and sorry, that is red the deep learning and this is a blue which is the proposed model the process guided deep learning.

So, as you can see in many cases the RMSE is like they have that is they have calculated the error of prediction for the different models. So, like as you can see that in many cases like in many cases the. So, initially the error is high using the proposed model, but then as we receive more and more temperature profiles that is the number of.

So, training the so, here on the x axis they are like basically comparing the training data and it shows that as we have sufficient training data like this us the proposed model is able to achieve almost equal RMSE as the as obtained from the process-based models and so on.

And similarly the here we have also like another similar study has been done for a particular lake called lake Mendota and they are also they have come across similar results.

(Refer Slide Time: 25:11)



Based on this study we come on to like an another very related paper here physics guided machine learning for scientific discovery and application in simulating lake temperature profiles.

Now physics-based models are often used to study engineering and environmental problems. The ability to model these systems is the key to achieving our future environmental sustainability and improving the quality of human life.

This article focuses on simulating lake water temperature, which is critical for understanding the impact of changing climate on aquatic ecosystems and assisting in aquatic resource management decisions. General lake model is a state-of-the-art physics-based model used for addressing such problems.

However, like other physics-based models used for studying scientific and engineering problems it has several well-known limitations due to simplified representation of the physics processes being modeled or challenges in selecting appropriate parameters.

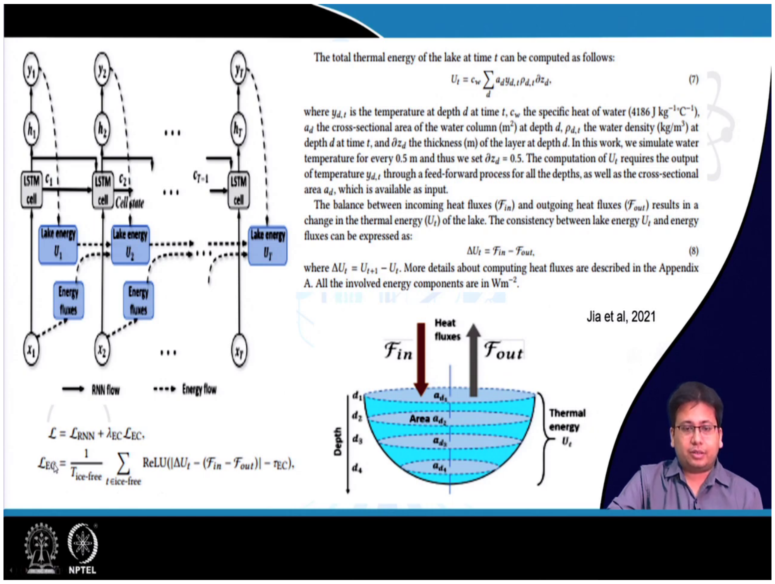
While state-of-the-art machine learning models can sometimes outperform physics-based models given ample amounts of training data, they can produce results which are physically inconsistent. This article proposes a physics-guided recurrent neural network model that combines RNNs and physics-based models to leverage their complementary strength and improves the modeling of the physical process.

Specifically, we show that a PGRNN can improve prediction accuracy over that of physics-based laws by over 20 percent even with very little training data, while generating outputs consistent with physical laws. So, that way this paper seems to be an improvement over this paper.

So, we here we see that actually the proposed model the which is marked in this blue it requires significant amount of training data to require reach good performance, but here they are saying that even with 20 percent of the even with small training data it is actually able to get a 20 percent improvement over the process-based model. While it also generating outputs which are consistent with the physical laws.

And important aspect of our PGRNN approach lies in its ability to incorporate the knowledge encoded in the physics-based models. This allows training the PGRNN model using very few true observed data while also ensuring high prediction accuracy. Although we present and evaluate this methodology in the context of modeling of dynamics of temperatures in lakes, it is applicable more widely to a range of scientific and engineering disciplines where physics-based models are used.

(Refer Slide Time: 27:58)



So, this is like so, like this is specifically the setting. So, the like this is a lake and at their different depths there are the temperature. So, like the main governing equation of this model here is this thing that is the energy balance. So, the lake has a thermal energy U_t.

So, and then there is a heat flux there is incoming heat and there is outgoing heat F_{in} and F_{out}. And the change of this total thermal energy of the lake is the like the difference between the incoming and outgoing waves at any given time. Now the whenever there is a change in the total thermal energy of the lake that is reflected at the different heights in different ways.

So, let us say that at like we have some initial conditions of that or we have some initial estimates of the total of the lake total thermal energy in the lake, then we like as heat that is we have observations of incoming and outgoing heat. We see how this total energy changes and as a result of it how the temperature also changes at the different depths.

So, this is the like the model. So, as you can see there is the LSTM cells which like using the storing the hidden variable. The hidden variable in this case is the cells the each one the in the cell state etcetera. Like you can consider these two to indicate the past values of the total thermal energy and so on. So, I am sorry I mean not the total thermal energy, but the temperature profile at any given point.

So, that is the temperature profile at any given point is passed out to the next time point and the observations or the inputs at every time point are these x_1, x_2 which are like which is basically the heat flux. And the what the this model does? It at it is that at every point based on the incoming heat flux it tries to estimate the lake energy which is again passed on to the next step and.

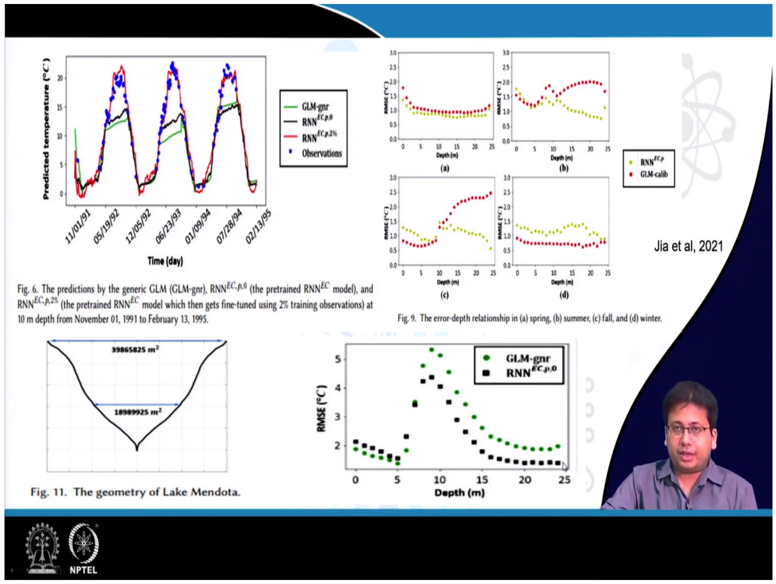
So, there is a separate variable here like U_1, U_2 etcetera that is that tell that maintains the time sequence of the total energy of the lake and it is updated using these energy fluxes like which is measured at every time point as inputs. And accordingly, the cell states or the hidden states of the LSTM they maintain the temperature profile and like at the end of the this process the there is some kind of a decoder model which like provides us the temperature profile of the lake based on the hidden values of the LSTM.

So, when we are training this kind of an LSTM the there is as you can see there is a dual loss function L_{RNN} is of course, the usual error of prediction. I mean the predicted values how different the predicted values are of the actual values. By predicted values I mean the values of the temperature at the different depths.

And additionally, there is this the energy conservation loss that is to say how much of energy. That is as far as law of conservation of energy is concerned how much error has there been? So, as you can see from time like the total lake energy at every time step that is being estimated by the model and the input fluxes are also being provided at every time step.

So, ideally the this change of lake energy should be equal to the heat flux, but has that actually happened or is there an error. So, that is what is measured by this part of the loss function. So, this is the physics-based loss and this is the prediction loss.

(Refer Slide Time: 32:12)



So, based on that again the model is trained so, as to minimize this loss function and this is what the simulations look like. So, as you can see over a over several days or over several months like if you plot the time series of the temperature of the lake at a particular depth.

So, like we can see that what is shown in blue is actually what the observations are like. Obviously, its a something like a periodic because we know that the lakes just like temperature is periodic similarly the lakes temperature will also have to be periodic at the different depths.

So, they have compared the GLM in green that is the general lake model which we discussed earlier and then simple like a simple RNN with in black and the in red they have plotted the physics based RNN which has about 2 percent of the training observations.

The initial one is the pretrained RNN model. That is this is like this is the RNN model which is like which is generated like which is trained just for one particular lake, but for the target lake it has not received any observations. So, that is a pretrained model and then for like for this particular lake which is being considered if it is just retrained once on this particular lake. That is to say some of its different parameters are re-estimated once using the observations.

Then we see like near-perfect performance as you can see that this red curve here follows almost the blue dots almost perfectly. So, that we can that is with only 2 percent of the observations it is able to fit so well. And so like also like they have calculated the error at different depths and as we can see that like at.

So, this green is what you get from the proposed model and red is what we get from the calibrated lake model general lake model which you saw. And we see that like in at certain depths for example, in this case at certain depths the proposed model gives lower RMSC than what even the calibrated GL process-based model could do.

In other situations so, they have measured it separately in different seasons and its show it is seen that in some seasons the proposed model performs as well as the general lake models. In some other situations it actually outperforms especially in summer and fall, ok. And similarly like that is if you consider across all the seasons then also, we see that the proposed RNN this actually performs better at the in perform like in estimating the temperature at different depths.

(Refer Slide Time: 35:18)

So, based on this the same topic we have one more paper here the aim is to estimate the parameters the physical parameters. So, like here the aim is to find the parameter settings which allows the process models to replicate the observed values this is not only for lakes, but can also be done for reverse systems also.

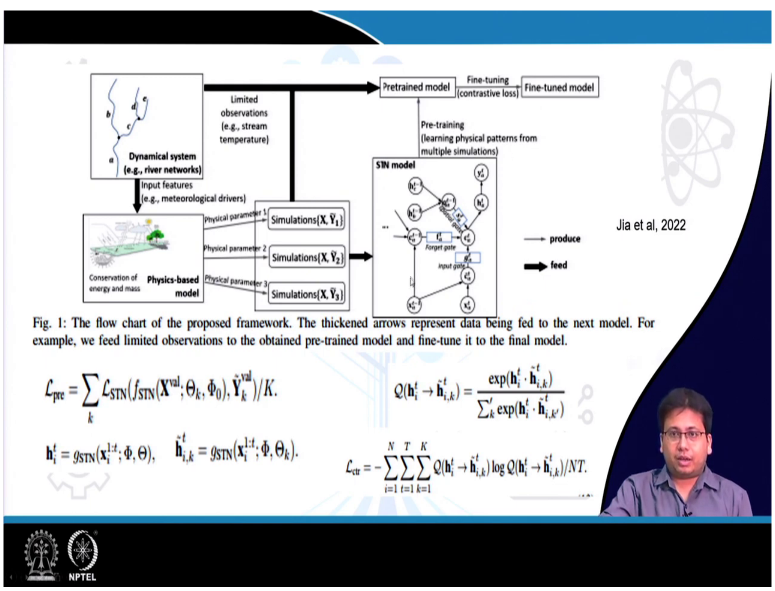
So, this is not necessarily tied to the GLM which we consider. I mean the GLM can be like this can be used to for estimating the parameters of the GLM. Apart from that there might be some process based hydrological models for river basic for river flows also. So, their parameters can also be estimated using this framework.

So, they have developed a STN or spatio-temporal network like this is not very different from the concept of these Markov random fields. The spatiotemporal Markov random fields which we talked about earlier which has the edge functions and so on. Of course, this is not a probabilistic model.

So, it does not have the edge potentials instead it has certain gating variables and these gating variables are used to pass the data across the spatio-temporally neighboring locations. So, the input is the various meteorological variables at a particular location i in a lake or river system and the output is some kind of target variable.

In this particular case in this particular paper, they have considered the water temperature at that depth or at that location as the like as the output variable. And of course, what that output variable from the model will be depends on the parameter setting k.

(Refer Slide Time: 36:59)



So, let us say this is the physics-based model input to it are the different covariates and so on. Its output is the temp the water temperature at different depths, but those outputs are going to be different for different values of the physical parameters.

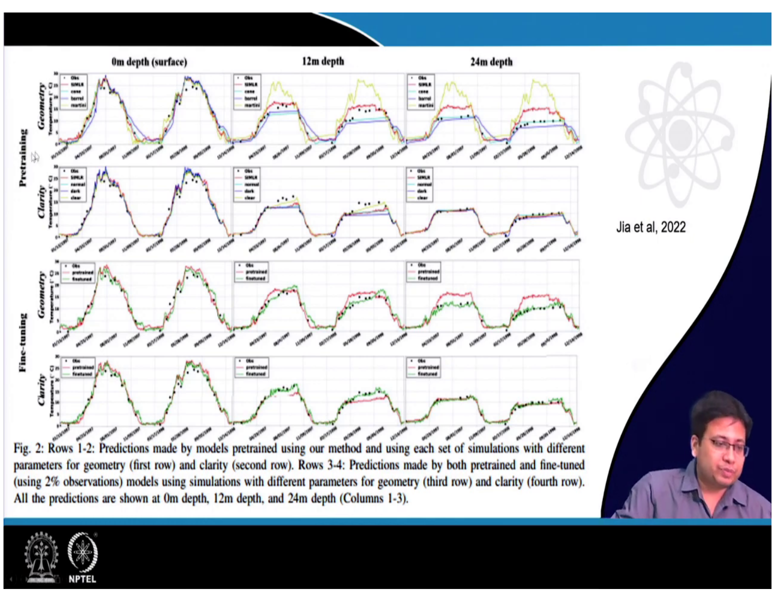
So, the like the this STN model this is actually learnt to what it does is, it learns a mapping that is like if the model has a certain inputs of the covariant values and then what parameters does it need to produce a particular output? That is the question that this model tries to answer.

So, let us say that it already knows what should be the like what is the ideal output that is what is the water temperature at the different depths it? Now it sees whether the model is able to actually produce those water temperature values or not for different choices of the parameters and that is framed as the learning problem.

So, the like then the what it does is it provides these parameters. So, that it the I mean it provides the parameter the physical parameter settings such that the physics-based model will be use those able to use those parameters along with the initial condition to produce what is the desired or what is the true output variables that is the water temperature.

So, its like; it its like fine tuning the model by choosing the parameters appropriately.

(Refer Slide Time: 38:50)



And so, like after this is done as you can see the this is again validated by measuring the temperature at different depths over a period of time. And it shows that and they have shown that if they like can provide the if the parameters of the model are estimated in this particular way, then it is able to do a better job of the simulation that is the this temperature profile comes as close to the observations as like or closer to the observations than what was earlier.

So, basically these three papers like in a sense they are a bit complementary. So, like here the aim of the first two papers is to emulate the physics-based model which is the general lake model and the third paper what it does is its aim is to provide the optimal values of the parameters to this model.

So, that it can perform well. The second paper actually shows that it is able to outperform the general lake model while the third paper actually discusses a way in which the general lake model can choose its parameters or receive its parameters. So, that it will be able to perform better, ok. So, this brings us to the end of this lecture.

(Refer Slide Time: 40:14)

In the following lecture also, we will see some other ways in which these machine learning models including the physics-inspired machine learning is able to or is used for emulating different process model. So, till then bye.