Summary of the first paper:
Title: Deep k-nearest neighbors - towards confident, interpretable and robust deep learning

So from what I have understood this paper discusses a method to improve deep neural networks in three different domains: Confidence, Robustness and Interpretability. Before getting into the paper itself, to understand the paper better, I first had to look more deep into the definition of these 3 domains. So here is my understanding:

1. Confidence: Confidence is about how sure our model is with respect to its predictions. For example, if we give a photo of a cat to it, and it labels it as a cat, how sure it is that the model is a cat. This can be important because sometimes in order to understand model's strength, we try to fool the model with giving not-very-straight-forward examples. For instance, in the example of the model predicting cats and dogs, we can use photo editing softwares like photoshop to mix a photo of a cat with a dog, like adding ears or tail of a dog to it, and then use that photo to test the model. When we use our dog/cat model to label this photo, whether it labels it as a cat or a dog, we expect it to have a lower confidence in its prediction.

2. Interpretability: So interpretability in this context is related to how much we are able to understand why the model makes decisions. Like in the same example in confidence, the cat/dog model, it is important for us to know that if the model labelled a photo as a dog or a cat, why it did so, for example, which pixels of that it was looking at. This can help us to understand the rationale behind model making decisions.

3. Robustness: Again going back to the example of our cat/dog model, this term "robustness" refers to how strong our model is with respect to the tricky inputs that we might give to it sometimes. A robust model can more easily understand if we are trying to trick it with giving an input like image of a cat mixed with some features of a dog with photoshop.

So after understanding these terms, it was easier for me to go through the paper itself. So what authors have tried to do in this paper is enhancing deep neural networks in three criteria that I mentioned and explained above with using a KNN algorithm inside it. Based on what I have understood so far, the output of each layer of neurons in a deep neural network (which is used as the input for another layer). So basically, at each layer of DNN, a KNN algorithm is run on the output of each layer of neural network to compare it with the outputs of the training data at the same layers, and it finds the k-nearest neighbours to it. With comparing the output of the test data with the output of training data,it can determine the confidence, robustness and interpretability of the DNN's explanation.

Summary of the second paper:
Time series classification: nearest neighbor versus deep learning models

Given my own research topic (using satellite imagery and data from gauges on-earth to predict floods), this paper was more related to my field of research and more interesting for me. Also,comparing with previous paper, this one was relatively simler and easier to understand. So basically what the author did was used two machine learning methods (K-nearest neighbours and deep learning models) on time series data and compared the results of them together.

Summarizing both methods to the best of my understanding, the nearest neighbours method compare the time series with some labelled examples and then label each trend based on similarities that it finds. On the other hand, deep learning models (in this paper, recurrent neural networks and covolutional neural networks learn patterns directly from the data. The second group (deep learning models) showed a better performance in classification of time series, but the downside to them is that theta re much more computationally expensive, and less understandable, comparing with K-nearest neighbours.

This does not mean that deep learning methods are always better and preferred to nearest neighbours method. The choice of model to use for different tasks can depend on the amount of data we want to train our model on, the accuracy we required from the model, and the hardware resources that we have to train our models on.

Suggestion to improve KNN performance

Since my research area is using satellite imagery data for flood prediction purposes, I'm gonna give an example/suggestion in that context. So we know that traditional KNNs use a simple distance metric, and the spatial and temporal context of data, which is a very important feature of satellite imagery is usually not affecting the outcome of the KNN methods. So my idea is developing a new KNN model with spatial-temporal weights.