R2: 1 General comments The paper compares a neural network cloud mask trained with 2D features to two operational cloud masks. The algorithm is trained with CALIOP data and uses a pseudo labeling method to deal with the issue that sunglint areas are not covered by the colocation dataset. The neural network cloud mask uses a large network but few sources of traditionally used ancillary data (most notably surface temperature is not included). Performance is very good except for small broken clouds; which given the 2D features is a bit counter-intuitive. The same network gives similar results for a large variety of surfaces.

2 Specific comments

1. The introduction is missing an important reference. The ESA cloud CCI algorithm also uses a neural network trained with CALIOP data for the cloud mask but with a different network structure, training, imager etc.

AR: Thank you for pointing this out. This work is highly relevant to our paper. We will plan to include reference to this in the final version. Below seems to be the appropriate citation for the work mentioned (but please correct us if not):

Sus, O., Jerg, M., Poulsen, C., Thomas, G., Stapelberg, S., McGarragh, G., Povey, A., Schlundt, C., Stengel, M., and Hollmann, R.: The Community Cloud retrieval for CLimate (CC4CL). Part I: A framework applied to multiple satellite imaging sensors, submitted to Atmospheric Measurement Techniques Discussions, pp. , 2017.

R2: 2. Line 61-65: Our approach aims to improve upon existing literature in several ways. Rather than relying on precomputed spectral, or textural features, we allow a neural network to learn relevant features from a local 3 pixel by 3 pixel image patch from all 16 moderate resolution VIIRS channels. The section is missing a motivation as to why it might be good to let the network learn the relevant feature itself. If the relevant features can be precomputed then the network can be made smaller and faster (fewer variables, fewer layers).

AR: Yes, we have not properly motivated this statement. It is also our experience that supplying precomputed to the neural network results in a network with fewer parameters after hyperparameter tuning. Both the ECM and the MVCM make use of cloud tests that involve some amount of feature engineering. In this paper we are starting with the assumption that we don't know all the relevant features for cloud detection. Despite the considerable amount of work has been done in the development of both the operational cloud masks, there might still be variability among the imager channels that is relevant to cloud detection and currently going unexploited (particularly that which involves three or more channels).

R2: 3. Line 70-75: Include short descriptions of the MVCM and ECM cloud mask methods. (Does not have to be here.)

AR: We agree that short descriptions of each should be added and will plan to do so. This will hopefully help readers understand differences between the methods that we write about later in the discussion section.

R2: 4. Line 185-210: Did you use any available software for training the network?

AR: Yes, we used the numpy, tensorflow, and keras python libraries. We will plan to add reference to them in the final version.

R2: 5. Line 234: Could the slightly overestimated cloud fraction in day time for MVCM be due to thin clouds not detected by 1km CALIOP data, but detected in the 5km CALIOP data and the MCVM? The chance of detecting these very thin clouds should be larger during day time.

AR: This is a great point. The 1km CALIOP Cloud products are less sensitive to optically thin cloud cover than the 5 km product, and it's not unreasonable to expect that one of the operational masks might detect clouds missed by the 1km product, but correctly identified by the 5 km product. We will plan to describe this in the results or discussion section.

R2: 6. Table 1: Add also at least TPR, TPN and CALIOP cloud fraction to a table. It is the 2019 data that is used for the table, right? Add info in the caption. Include also a table with results for the unfiltered data.

AR: We are assuming that by TPN, the reviewer means the TNR (the True Negative Rate). If this assumption is wrong, please correct us so we can properly address your comment.

We will remake this table with BACC, TPR, TNR and cloud fraction. It is indeed for the 2019 testing dataset. We did not initially include a table for the unfiltered dataset because it did not offer much additional insight for earlier versions of our model that wasn't apparent in our other tables and figures. However, if we don't include such a table, it could be interpreted as an obvious oversight since earlier in the text we emphasize the importance of the differences between the unfiltered and filtered datasets. At the very least we will plan to include the unfiltered table in the supplement. We will consider the unfiltered results more carefully before the final version and will move it to the main text if there are interesting differences to discuss.

R2: 7. Line 265: I find this surprising, I would have expected the 2D feature to be most useful for fractional clouds.

AR: We agree that this result is a bit counter-intuitive. We believe this is a limitation of using CALIOP as our source of labeled data. It is likely that small broken clouds are not well represented in our collocation database due to the size of these clouds and the time difference between when the two instruments observe them. Small horizontal displacements of these clouds between times that both VIIRS and CALIOP observe the same ground location could be mean that some of our labels for these types of clouds are more prone to error, and result in poor characterization of them from our neural network approach.

R2: 8. Line 306: You mention that Bayesian algorithms might be affected by climatological means. Considering that your method includes latitude could it not be that it too uses the latitude mean cloudiness from the two years of training data? Have you tested how much the network depends on latitude?

AR: Thanks for this comment! Very early on in this work we were using models that were somewhat more interpretable than the neural network we talk about in this paper (Gradient Boosting Machines). We found that including latitude changed how other features were used in very interesting ways. For example a difference between the 11µm and 8.6µm brightness temperature of 1 K had almost no impact at the equator, but was one of the most influential features in the model at the high latitudes. Of course, the neural network is a completely different model and there is absolutely no guarantee that usage of latitude is similar here, but this was our initial motivation for including it.

Motivated by this comment, we retrained our neural network without latitude. Surprisingly, the daytime results improved in our testing dataset by very small margins. The surface type with the largest difference during the daytime was (non-permanent) snow which increased from 95.5% to 95.8% using the filtered dataset. During the night, removing latitude overall worsened our results slightly more substantially. The largest change was, again, over (non-permanent) snow which decreased from 92.0% to 90.9%.

We additionally retrained the model a second time after removing latitude, solar-zenith angle, and sun glint zenith angle. This is because the distribution of solar zenith angle and sun glint zenith angle vary with latitude, and could leak some information to the model about the latitudinal mean cloudiness (low sun glint angles typically occur at low latitudes, for example).

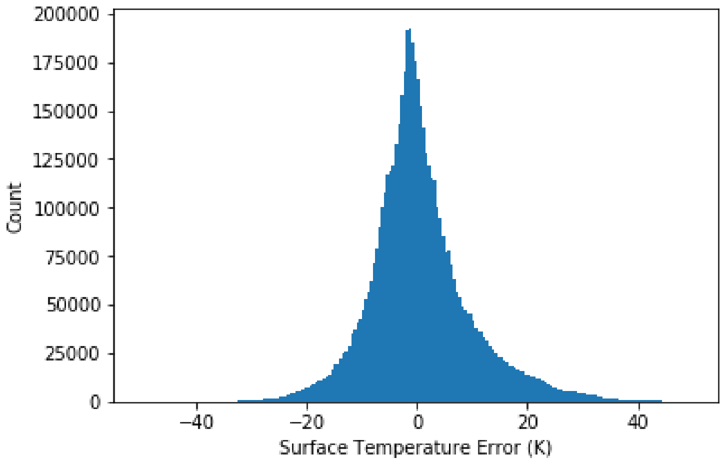
This model had very slightly worse results in the daytime than the model without latitude (the largest difference was -0.2%). The global nighttime BACC changed from 93.4 to 92.9 with most of the change coming from nighttime water surfaces with changed 93.2 to 92.4. The other surface types remained unchanged with differences within +/- 0.1%.

Overall, the model seems to mostly worsen in nighttime water scenes when removing latitude and information related to latitude. Considering these results, I think it is likely that our model depends on latitudinal mean cloudiness in some capacity over these areas. However, it is difficult to quantify whether it is serving a purpose similar to that of a climatological mean, or if it is changing the usage of other observations features (like we have previously observed in our other models).

R2: 9. Line 357: Can the latitude combined with sun zenith angle give a rough estimate of surface temperatures? Very impressive results for temperatures close to surface temperature.

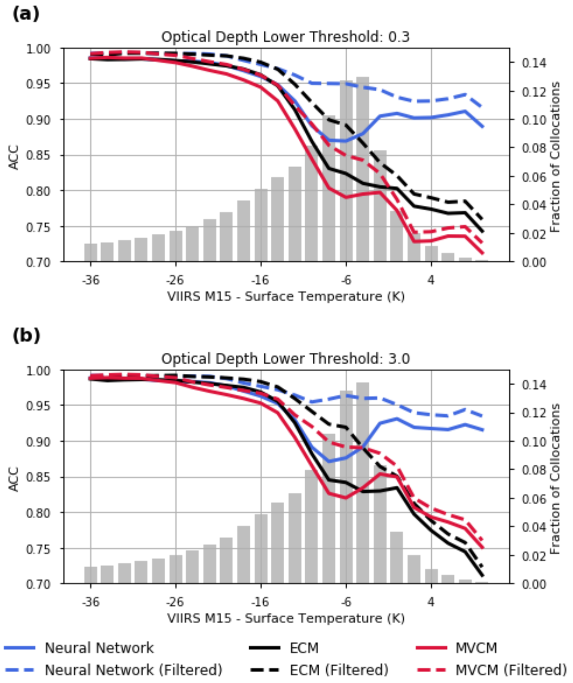
AR: This is a really great point that we did not think about. Even though we have not provided surface temperature directly to the model, perhaps the other information we have provided is utilized in a similar way that a mean surface temperature would.

I built a model that uses solar zenith angle, the land/water mask, sun glint angle, and the absolute value of latitude to predict the GFS surface temperature. It is trained in a similar way as the model in the manuscript but with fewer parameters. It has 3 fully connected layers of 20, 10 and 1 units, no dropout, a linear activation as the last layer, and mean squared error as the loss function.



On the 2019 testing dataset, the model produced a mean absolute error of 6.7 K and a mean squared error of 86.3 K. I think it is somewhat possible that this information is serving similar purpose as an extremely rough estimate of surface temperature. However, I do not know if a mean absolute error of 6.7 K is accurate enough to be especially useful in cloud detection.

To add another perspective to this question, I retrained the cloud detection model only using the 16 VIIRS channels and have recreated the thermal contrast figure (Figure 8). The differences in this figure and the one in the manuscript are fairly minimal (except that we added both datasets to address a later comment). The largest difference in performance with and without this information is a change of roughly -3% ACC around -10 to -16 K. Based on these small differences I think it is safe to rule out the idea that this information serves a similar purpose as surface temperature might.



R2: 10. Line 390: I agree it is not bad with a consistent TPR dependent only on the cloud. But optimizing TPR differences might mean making the TPR lower in easy conditions to match the *performance in more difficult conditions. Is it not equally important to keep TPN as constant as possible? I think this is what is more traditionally aimed at.*

AR: (Again we are assuming that TNR was meant instead of TPN, but please correct us if not). I think the usefulness of this metric perhaps depends on the application. There is indeed a tradeoff between performance with respect to CALIOP and cloud detection consistency. For operational nowcasting, we expect that users might not be especially concerned with detection consistency over different surface types or times of day. For climate applications, this might be a more important consideration. For example, a globally uniform increase in the amount of clouds with optical depth of about 0.1 might only be detected as a much larger increase over certain surface types with higher TPR for these specific clouds. This might result depictions of cloud cover change that don't align with reality. Minimizing TNR differences between different conditions could achieve a similar result, we think. TPR differences might be more useful since one could weigh the differences with respect to optical depth. For example, a large TPR difference at a high cloud optical depth is likely more problematic than one at a very low optical depth.

Either way, we recognize that our suggestion that TPR differences should be minimized in conjunction with other performance metrics might be an overstatement. We will think more carefully about this in the final version, and perhaps only suggest that it be used as a metric to identify detection consistency when that is a specific need of a cloud mask.

R2: 11. Line 425: For the validation data do you have sea ice cover to the north west of Greenland? Can the shrinking sea ice cover in the arctic be part of the explanation. If MVCM is trained on older data and assumes it to be sea ice, and the new NN approach trained on more recent data expects more water?

AR: Great question. Our understanding is that the MVCM uses separate decision pathways based only surface type and time of day. Quoting from the MVCM users guide (Frey et al. 2019): "Several ancillary data sets serve as inputs to the MVCM process, [...] Near-Real-Time SSM/I-SSMIS (Special Sensor Microwave/Imager-Special Sensor Microwave Imager/Sounder) EASE (Equal-Area Scalable Earth)-Grid Daily Global Ice Concentration and Snow Extent (NISE) files contain daily global gridded snow and ice extent." Assuming the correct up-to-date surface type is given to MVCM, we do not expect this to be an issue.

However, this does bring up another an interesting point: What if the surface type or surface temperature is not perfectly known? If the operational methods assume incorrect characteristics of the surface, then this could be a reason why our approach appears to perform better over scenes where the 11µm BT is close to the NWP surface temperature. Similarly, if the presence of sea-ice is incorrectly indicated, this might be problematic for approaches that otherwise depend on this information.

R2: 12. Line 433: The averages across space are weighted by the cosine of latitude expressed in radians. I do not understand what you mean here.

AR: Yes, this sentence is poorly-worded. When doing this analysis we compute the values on a grid with regular latitude/longitude spacing. Because of this, high-latitude grid cells represent a smaller surface area than the lower-latitude grid cells. To account for this we give a smaller weight to the higher latitude observations when averaging across the domain. To calculate the weights, we first convert the latitude of the grid cells to radians, and then take the cosine of those values. We will revise this sentence to hopefully make this more clear. Perhaps something like the following: "When calculating the mean cloud fraction, individual values on the regular latitude/longitude grid are weighted to account for differences in surface area between them."

R2: 13. Line 457: The pseudo-labeling model likely has low skill in such conditions due to the low contrast between a low-level fractionally cloudy pixel and the background. Did you consider using the ECM for the pseudo-labeling?

AR: We considered it, but did not actually do this analysis. When doing the analysis for this manuscript our goal was to demonstrate the effectiveness of the neural network as a stand-alone approach for cloud detection. It is clear that the ECM is better in very specific scenarios and has obvious advantages of being more interpretable. In practice, it would surely be better to use the ECM for pseudolabeling of sun glint scenes. However, if a cloud mask is needed for a sensor in which the ECM has not been validated for, we wanted to demonstrate how sun glint issues could be mitigated without it

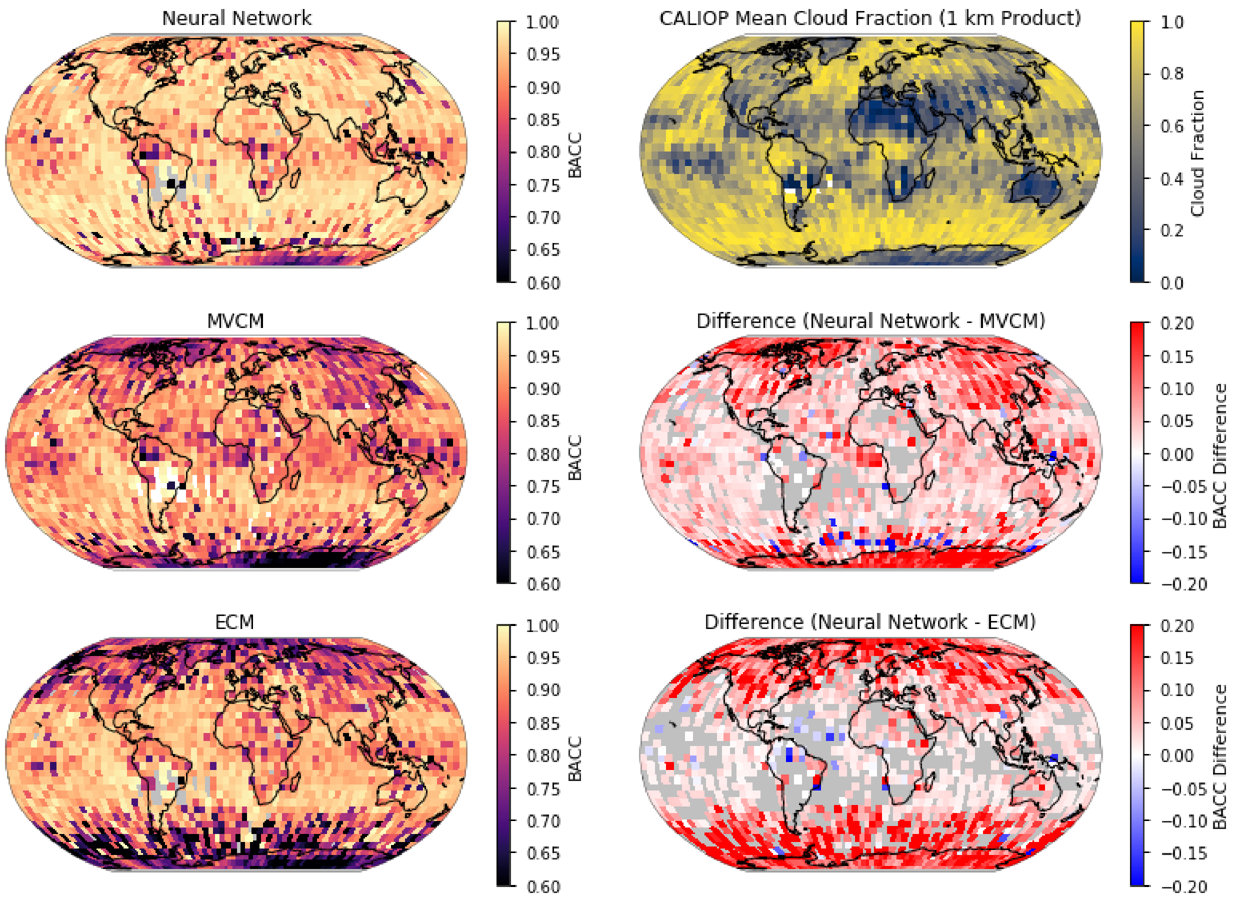
---

R2: 14. Line 537: Additionally, we have not evaluated how the neural network performs specifically in cloud-free scenes with high aerosol loading. We expect that this could depend largely on the ability for CALIOP to distinguish cloud from aerosol layers. Even if it does depend on CALIOP's ability should it not depend mostly on the VIIRS capabilities?

AR: Agreed. We wanted to mention the point that CALIOP's ability to distinguish cloud from aerosol would be an additional complicating factor. We now realize that these sentences, as written, are somewhat misleading. We will plan to rewrite this statement to something like this: "In addition to the challenge of distinguishing cloudy from cloud-free scenes with high aerosol loading using VIIRS measurements, we expect that CALIOP's ability to discriminate between them and provide accurate labels to the neural network model could be another complicating factor."

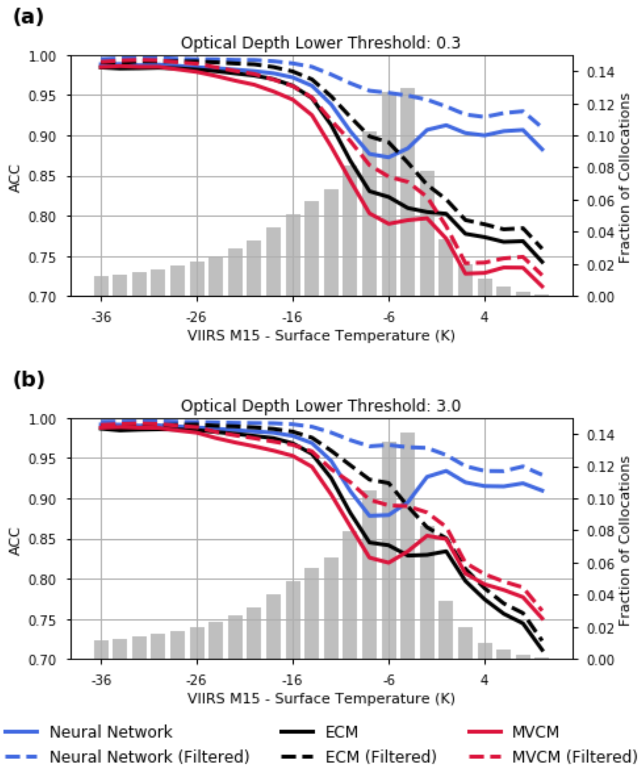
R2: 15. Figure 6: Consider adding figures also for the BACC.

AR: Other reviewers have asked for this as well so it is clear that we should add this figure (at the very least in the supplement if not in the main text). We initially did not include a figure with BACC since there were grid cells where the mean CALIOP cloud fraction was particularly high, and the BACC was mostly dependent on a relatively small amount of cloud-free CALIOP collocations (the Southern Ocean, for example.) Below is the same figure with BACC, and panel (b) is replaced with the CALIOP mean cloud fraction. Similar to the ACC figure in the manuscript, these maps are calculated using the filtered dataset.



R2: 16. Figure 8: Is this filtered or unfiltered data?

AR: It was filtered data, but we see no reason by we couldn't include both datasets here. See below for figure. Actually, there are interesting differences in the unfiltered data that that don't exist in the other. For the neural network and the MVCM there is either a global or local minimum around -6 to -8 K. We will plan to revise this figure with both datasets and add some discussion of this.



R2: 17. Have you tested to applid the NN on older data (2013) and was there a difference in performance?

AR: Not yet, but we plan to examine something similar to this in future work. Whether or not ML-based approaches generalize to data collected long after (or before, in this case) their training period is an interesting question. This perhaps partially depends on whether our approach learned variability that is specific to our training period (2016 and 2018) which seems clearly undesirable. Of course, consistent sensor calibration is also a significant concern with this.

R2: 18. Is execution time comparable with the operational cloud masks? Is it feasible to use for nowcasting?

AR: I wasn't particularly comfortable commenting on execution time in the manuscript since it might vary greatly across implementations or systems. Using a fairly recent GPU processing an entire 6-minute VIIRS scene with our neural network implementation took roughly 7 seconds and only hit roughly 50% utilization of the GPU. To contrast, using 2 threads of a very old (8-10 years?) CPU it took 90 seconds for the full scene and roughly 25 seconds using 12 threads. Nowcasting is certainly a possibility if the machine has access to a GPU or a modern CPU. If running on older hardware, the processing time might be unfortunately prohibitive. That being said, we are definitely not expert software engineers, so others may have more luck in writing fast processing code.

I am not familiar with the MVCM processing code. Based only on the method itself, I suspect the MVCM is the fastest, and that the neural network is by far the slowest if run on similar hardware.

R2: 19. A name of the method would be useful.

AR: Agreed. This might help make the text more concise and readable. We will spend some time thinking of an appropriate name, or at least an abbreviation to use in the text.

R2: 20. From my experience with NN cloud masks results often look less realistic close to the swath edges when comparing results to the RGB. In Figure 12 results look realistic also closer to the edges. Is this normally the behavior?

AR: Figure 12, specifically, is a cropped image close to the nadir track of VIIRS to focus on the area of sun glint.

Early experiments we performed with MODIS/CALIOP data showed issues with viewing angle, but this is expected due to MODIS only making near-nadir collocations with CALIOP. Håkansson et al. 2018 discuss this as well with cloud-top pressure/altitude.

VIIRS makes collocations with CALIOP at a larger variety of viewing angles (0 to 50 degrees in our dataset). Early on in the development of our model we included sensor zenith angle as a predictor. However, we noticed that performance with respect to CALIOP increased when it was removed so we left it out. Aside from the regional analysis over Greenland, we have not extensively analyzed full scenes to the swath edges aside from a set of 20-30 scenes that we run as a “sanity check”. In these scenes we have not noticed particularly unrealistic behavior at swath edges.

R2: 3 Technical corrections • Line 53: Häkansson should be Håkansson (several places)

AR: An embarrassing mistake on our part! We will make sure this is fixed in the final version.