Response to Reviewer 1: Benoît Crouzy (Referee):

Thanks a lot for your comments here are our responses.

1) Page 1 line 10: selection of the training and test set. This is a potential major issue. The authors selected randomly scans over the years. In order to have good generalization of the algorithms it however is important to have not too close training and test data. The total number of raw scans out of which events were randomly drawn suggests that this is achieved but I would like the authors to comment on this question, even if only as a caveat to the community. Best practice could be to select a period isolated from the training set to select the test set (e.g. different year).

We agree that setting a period totally isolated for the test is the best practice and can test the generality of a model, but we wanted our training set to include most possible scenarios (different scans). In our training phase we had 3150 scans, and in the testing phase we had 850 scans. As the number of test sets is large (compared to the total number of selected scans), we have confidence in achieving a model capable of generalization. We also recommend that the model be retrained every year (adding new data to the existing pool of training data). Retraining with more (and newer) data will help with the overfitting problem.

2) Page 2 lines 3-8: please check for some repetitions (laser fluctuations)

We shortened the paragraph.

3) Page 2 line 17-18: I suggest to distinguish between supervised and non-supervised from the onset.

We changed the paragraph as follows:

Here, we propose a machine learning (ML) approach for level-0 data classification. The classification of lidar profiles is based on supervised ML techniques which will be discussed in detail in Section 2. Using an unsupervised ML approach, we also have examined the capability of ML to detect anomalies …

4) I would put this line earlier to distinguish between supervised and non-supervised techniques. In addition I would hint for non-specialists that unsuper- vised techniques are no silver bullet and can be expected to be less powerful due to the absence of training data.

Done!

5) Page 2 line 27: "clustering ML" and line 30 "These ML methods" please precise which methods.

We changed the sentence to: Both Zeng et al. (2018) and Nicolae et al. (2018) concluded that their proposed ML algorithms can classify large sets of data and can successfully distinguish between different types of aerosols.

6) Page 3 line 23: I find the sigma confusing (summation sign), especially when keeping signed differences.

We explained the equation with more details:

Formally, we are trying to learn a prediction function $f(x): x \rightarrow y$ which minimize the expectation of some loss function $L(y,f) = \Sigma_{i}^N (y^{true}_{i} - y^{predicted}_{i})$, where y^{true}_i is the actual value (label) of the classification for each data point, and $y^{predicted}_{i}$ is the prediction generated from the prediction function and N is the length of data-set \citep{bishop2006pattern}

7) Page 3 line 25: what about "matrix size (m,n)" or m x n ?

We changed to a matrix with size (m,n)

8) Page 4 line 13-14: please describe better the Kernel trick that make SVM so powerful. In the current form I do not find the description self-contained (non-uniform level of details). I would describe the parameters to be tuned (e.g. Cost, epsilon insensitive tube, ...)

We added the following to the paper which explains the SVM in more detail (there is also a colored marked up pdf where you can see the latex below rendered):

The SVM algorithm finds an optimal hyperplane that separates the data set into a distinct predefined number of classes \citep{vapnik2013nature}. For binary classification in a linearly separable data-set a target class $y_{i} \in \{1, -1\}$ is considered with a set of input data vectors ${\mathbf x_{i}}$. The optimal solution is obtained through maximizing the margin between the separating hyperplane and the data. It can be shown that the optimal hyperplane is the solution of the constrained quadratic equation:}

\begin{align}

minimize &: \frac{1}{2}||\vec{w}||^2 \\

subject &: y_{i}(\vec{w}.\vec{x_i} +b) \geqslant 1.

\end{align}

In the above equation the constraint is a linear model. To solve this constrained optimization problem, the Lagrange function can be built:

\begin{equation}

L(\vec{w}, b, \alpha) = \frac{1}{2} \norm{\vec{w}^2} - \sum_i \alpha_i
\left(y_i(\vec{w}.\vec{x}_i+b)-1\right)

\end{equation}

where $\alpha_i$ are Lagrangian multipliers. Setting the derivatives of $L(\vec{w}, b, \alpha)$ with respect to $\vec{w}$ and b to zero:

\begin{align}

\vec{w} &= \sum_i \alpha_i y_i \vec{x}_i \\

& \sum_i \alpha_i y_i = 0.

\end{align}

Thus we can rewrite the Lagrangian as:

\begin{equation}

L(\vec{w}, b, \alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i . \vec{x}_j

\end{equation}

It is clear that the optimization process only depends on the dot product of the samples.

Many real world problems involve nonlinear data sets in which the above methodology will fail. To tackle the non-linearity, using a non-linear function $\Phi (x)$ the feature space is mapped into higher dimensional feature space. The Lagrangian function can be re-written as:

\begin{align}

L(\vec{w}, b, \alpha) &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\vec{x}_i, \vec{x}_j)\\

k(\vec{x}_i, \vec{x}_j) &= \Phi(\vec{x}_i) . \Phi(\vec{x}_j)

\end{align}

where $k(\vec{x}_i, \vec{x}_j)$ is known as the kernel function. Kernel functions let the feature space be mapped into higher dimensional space without the need of calculating the transformation function (only the kernel is needed). This property makes them really powerful and easy to use. More details on SVM and kernel functions can be found in \cite{bishop2006pattern}

9) Page 4 line 25: please define all variables

We rewrote the section.

10) Page 5 line 3: maybe I missed it, but if you used LIBSVM or a derived tool please mention it, as this information could help other users

All the codes are written using the python scikit-learn package. We have explained it in the methodology section of the paper.

11) Page 5 line 25: I found this sentence somewhat disconnected.

We agree with you and have deleted the sentence.

12) Equation 3: define the class index.

What does he mean?

13) Page 5 line 29: H=0 usually means low entropy which could be seen as a pure prediction. Please clarify.

We rewrote the paragraph as follows:

where p_{i} represents a set of probabilities that adds up to 1. $H(x)=0$ means that no new information was gained in the process of splitting, and $H(x)=1$ means that maximum amount of information was achieved. Ideally, the produced leaves will be pure and have low entropy $H(x)=0$ meaning all of the objects in the leaf are the same.

14) General comment: I would summarize for all methods the hyperparameters to be tuned. This is currently well done only for some of the methods.

We added a short subsection addressing the comment:

Machine learning methods are generally parametrized by a set of hyper-parameters λ. An optimal set of hyper-parameters λ_{best} will result in an optimal algorithm which minimizes the loss function that formally can be written as:

\begin{equation}

\lambda_{best} = arg min L(X_{test}; A(X_{train}, \lambda))

\end{equation}

where A is the algorithm and X_{test} and X_{train} are test and training data. Searching to find the best set of hyper-parameters is mostly done by grid search method in which set of values on a predefined grid will be proposed. Implementing each of the proposed hyperparameters the algorithm will be trained, and the prediction results will be compared. Most algorithms have only few hyper-parameters. Depending on the learning algorithm, the size of training and test data sets the grid search can be a time consuming approach. Thus automatic hyper-parameter optimization has gain interest, details on the topic can be found else where \citep{feurer2019hyperparameter} }

15) Page 7 line 1: "a detailed description" this sentence and the introduction to various methods give the impression of ML as a closed list of techniques. As the Lidar community is not very

familiar to ML, I would mention that a vast number of other techniques exist. I would also mention ANNs and explain why those were not used in the present paper

The "a detailed description" refers to a Hastie et al., 2009's detailed description of all unsupervised learning methods and not to our explanation.

We added a paragraph in page 4, before explaining the ML algorithms which we used in this study:

\textcolor{blue}{Many algorithms have been developed for both supervised and unsupervised learning. In the following section, we introduce Support Vector Machine (SVM) , Decision Trees, Random Forests and Gradient Boosting Tree Methods as part of ML algorithms that we have tested for sorting lidar profiles. We also describe The t-distributed Stochastic Neighbour Embedding Method and Density-based spatial clustering of applications with noise (DBSCAN) unsupervised algorithms which were used in this paper.}

\textcolor{blue}{Recently, Deep Neural Networks (DNNs) have received attention in the scientific community. In the Neural Network approach the loss function computes the error between the output scores and target values. The internal parameters (weights) in the algorithm are modified such that the error becomes smaller. The process of tuning the weights continues until the error is not decreasing anymore. A typical deep learning algorithm can have hundreds of millions of weights, input and target values. Thus the algorithm are really useful when dealing with large sets of images and text data. Although, DNNs are power full tools, they are acting as black boxes and important questions such as what features in the input data are more important will stay unknown. For the purpose of this study we decided to use the classical machine learning algorithms.}

16) I would remove Figure 2 (too much detail in comparison with the rest of the chapter), but this is a matter of personal taste.

We prefer to have this figure in the article.

17) Page 8 line 15: from and not form

Thanks for pointing out this typo.

18) Section 3.1: see general comment above, how was it ensured that enough separation between training situations and test situations is achieved (scans taken the same day/hour might not always achieve this). This point needs to be discussed carefully

As we responded to the first comment, our training and test data-sets contain scans from different years, months, and hours to make sure that we can achieve the generalization.

19) Section 3.2: In my opinion the discussion on TP/FP/TN/FN does not belong to the results but to the methods.

As we wanted to show the confusion matrix under the result and we found out putting the definitions of TP/FP/TN/FN in the methodology will require us to insert another short subsection. So, we decided to just squeeze the definition in the result.

20) Page 11 line 6: to estimate

Fixed

21) Page 14 line 11: anomalies and not anomolies

Fixed

22) General comment: why not making use of thresholds in order to achieve finer objectives, eg. "never tagging good scans as bad"? For example, with SVMs one can use the distance to the hyperplane to select events with a good likelihood of correct classification.

As the other two algorithms did better in the test and evaluation phase we did not attempt to use the SVM as our primary approach of classification. However, we definitely agree with you that we can use the distance to hyperplane to achieve a probability approach rather than a solid binary classification.