Characterizing the performance of the SPHERE exoplanet imager at the Very Large Telescope using deep learning

L. Bissot[a, d], J. Milli[c], E. Choquet[b], F. Cantalloube[c], P. Delorme[c], D. Mouillet[c], G. Louppe[e], and O. Absil[a]

[a]Institut d'Astrophysique et de Géophysique, University of Liège, Belgium
[b]Aix Marseille Université, CNRS, CNES, LAM, Marseille, France
[c]Université de Grenoble Alpes, CNRS, IPAG, 38000 Grenoble, France
[d]Institute for Information and Communication Technologies, Electronics and Applied Mathematics, Catholic University of Louvain, Belgium
[e]Montefiore Institute, University of Liège, Belgium

ABSTRACT

The Spectro-Polarimetric High- contrast Exoplanet REsearch (SPHERE) instrument is a high-contrast imager designed for detecting exoplanets. It has been operational at the Very Large Telescope since 2014.
To make the most of the extensive data generated by SPHERE, improve future observation planning, and advance instrument development, it is crucial to understand how its performance is affected by various environmental factors. The primary goal of this project is to use machine learning and deep learning techniques to predict detection limits, measured by the contrast between exoplanets and their host stars. Two types of models will be developed : random forest models and Multi-Layer Perceptron (MLP) models. The aim is to better understand the relationship between input parameters and detection limits, providing deeper insights into this field. Additionally, a neural network will be used to capture uncertainties in the input features, thus providing confidence intervals for its predictions.

Keywords: SPHERE, High Contrast Imaging, Machine Learning, Neural Network, Exoplanet, Very Large Telescope

1. INTRODUCTION

Over the past two decades, advancements in adaptive optics, coronagraphy, optical manufacturing, wavefront sensing, and data processing have led to the development of a new generation of high-contrast imagers and spectrographs for large ground-based telescopes. Among these, the SPHERE instrument stands out as one of the most productive.

Designed for the ESO Very Large Telescope (VLT) in Chile, SPHERE includes an extreme Adaptive Optics (AO) system, a stable common path interface, various coronagraphs, and three specialized science instruments. The Integral Field Spectrograph (IFS) and the Infrared Dual-band Imager and Spectrograph (IRDIS) efficiently cover the near-infrared range to facilitate the search for young planets. The third instrument, ZIMPOL, focuses on visible polarimetric observations to detect reflected light from exoplanets and scattered light from debris disks. Together, these instruments allow for the detailed study of circumstellar environments at unprecedented angular resolutions and contrasts in both visible and near-infrared wavelengths.[1]

Since its installation in 2014, the instrument has been collecting a wealth of data from all its observations. This study focuses on observations made with the IRDIS instrument using the Angular Differential Imaging (ADI) technique[2] in the H2H3 bands. Our work draws inspiration from Xuan et al[3] (2018), which aimed to characterize the performance of the NIRC2 vortex coronagraph on Keck II, an instrument designed to directly image exoplanets and circumstellar disks in the mid-infrared bands.

Further author information:
E-mail: ludo.bissot@uliege.be, Github: `https://github.com/lbissot/Master-Thesis`
A more detailed manuscript of this work can be found at `http://hdl.handle.net/2268.2/19340`

Indeed, to fully exploit the vast SPHERE database, optimize future observation scheduling, and advance instrument development, it is crucial to thoroughly comprehend the relationship between instrumental performance and various environmental parameters. These parameters include atmospheric turbulence intensity, wind velocity, observation duration, pointing direction among others. Understanding these dependencies would allow us to optimize the potential of SPHERE's capabilities.

This project's principal goal is to use machine learning and deep learning approaches to forecast detection limits in terms of contrast between exoplanets and their host stars.
A dataset will be built and two types of models will be created to achieve this goal: random forest models and multilayer perceptron models. The goal is to create a greater understanding of the relationship between input attributes and detection limits, resulting in better insights and knowledge in this domain.

Section 2 details the data collection process and the creation of the dataset used as input for the machine learning models. Section 3 provides a comprehensive overview of key machine learning concepts, particularly focusing on the models used in this work. Section 4 explains the methodology employed to construct consistent and smoothly operating models. Finally, the results are presented in Section 5.

2. DATASET BUILDING

The telescope takes a sequence of pictures during an observational sequence. These separate snapshots are then assembled into a consolidated master cube, which serves as a container for the raw astronomical images coming from the same observation.
Those data-cubes, accessible through the High-Contrast Data-Center (HCDC), undergo a series of processing steps. This involves the correction of faulty pixels and the subtraction of the background noise. Afterward, the images are realigned with respect to the center of the star, and the stellar speckle field is reconstructed before being subsequently removed. The data being processed through this reduction pipeline is consistently accessible through the HCDC at each stage of the reduction process.

2.1 Contrast Curves : Objective Function

A contrast curve measures sensitivity to off-axis companions by evaluating the sensitivity limits in terms of contrast at different angular separations. The contrast is the ratio of flux of a given off-axis companion to the flux from the central star. Sensitivity limits are determined by placing circular apertures at various radial distances from the star. The Signal-to-Noise Ratio (SNR) is calculated as the ratio of the flux in an aperture to the standard deviation of flux in other apertures at the same separation, estimating the radial noise statistics.

For this project, contrast curves are obtained from the SPHERE IRDIS instrument using the H2H3 bands and the cADI reduction algorithm. Figure 1 shows an example of contrast curve.

Contrast curves vary by observation due to different factors affecting instrument performance and can be divided into two regimes: background noise-limited and speckle noise-limited. The distance at which a target hits the background limit depends on integration time, target magnitude, turbulence conditions, quality of AO correction, etc. Performance in the speckle noise-limited regime is influenced by a wider range of factors than the background noise limit.

To emphasize the speckle noise-limited regime, a maximum separation value of 3 arcseconds is set uniformly for all the observations. Moreover a consistent separation discretization is applied to all the curves. To avoid extrapolation, the minimum separation is set to the highest value among all observed minimums. Finally, a logarithmic spacing was used in order to prioritize small separation values.

As shown in Figure 2a, some curves deviate significantly from the expected range of achievable contrasts, likely due to observational errors or circumstellar object contamination. These outliers need to be removed. The mean log-deviation from the median curve is computed, and curves with deviations above a threshold (1 in this case) are discarded.

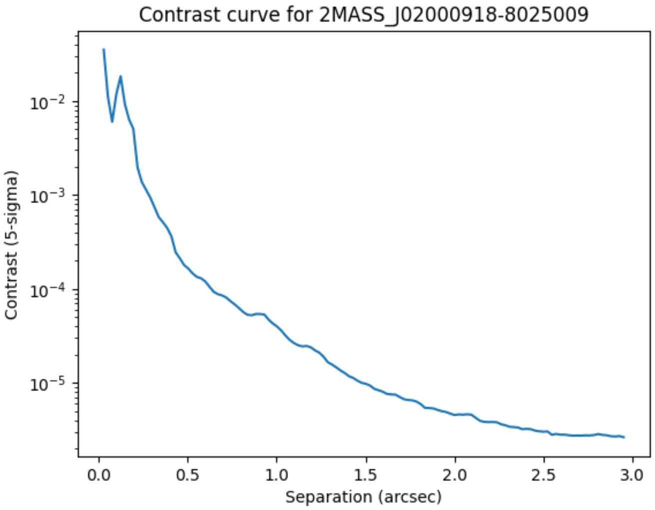
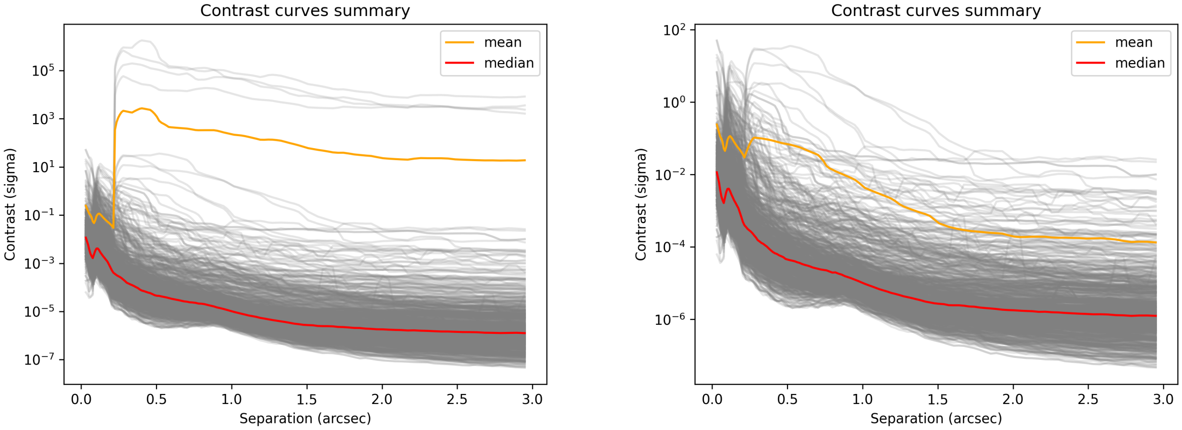


Figure 1: Example Contrast Curve



(a) With outliers

(b) Without outliers

Figure 2: Contrast curves summary with and without outliers

2.2 Features : Input of the models

The goal here is to gather a large number of explanatory variables that will help predict the contrast limits and thus will serve as inputs to the Machine Learning (ML) model. In other words variables which might have an impact on the contrast value are sought.

Three main data sources will be used for feature selection: the headers from the `fits` files where the contrast curves are stored, the Paranal Astronomical Site Monitoring (ASM), and the SIMBAD database. The different features along with their sources are listed in Table 1. Note that some observations may have missing features, which is common when building a dataset from scratch.

Many header keywords have been made accessible by the HCDC. However, because the reduction algorithms are periodically updated, some keywords are only available in specific versions of the data products. Consequently, some features, such as the Strehl Ratio (SR), statistics about the airmass, and other potentially important features, are not used in this work due to their high percentage of missing values.

Variable	**Source**	**Missing Percentage(%)**
Observing Conditions		
τ_0 (median and std)	ASM	11.57
Seeing (median and std)	ASM	11.57
WFS Frame Rate	FITS Header	0.00
Deformable Mirror Temperature	FITS Header	1.53
Primary Mirror Temperature	FITS Header	0.00
Dome Temperature	FITS Header	0.00
Ambient Windspeed	FITS Header	0.00
Ambient Relative Humidity	FITS Header	0.00
Ambient Air Temperature	FITS Header	0.00
Parallactic Angle Amount	FITS Header	0.00
Observation Parameters		
WFS Spectral Filter	FITS Header	0.00
WFS Spatial Filter	FITS Header	0.00
Sub-Integrations Number	FITS Header	0.00
Integration Time	FITS Header	0.00
Target Parameters		
Stellar Magnitude H Band	SIMBAD	4.25
Stellar Magnitude G Band	SIMBAD	1.18
Separation	FITS Data	0.00

Table 1: Input Features

2.3 Dataset Pre-Processing

The dataset is divided into three subsets: the training set, the validation set, and the test set. The training set is used to train the machine learning model. The validation set helps assess the model's performance, ensuring it is not overfitting or underfitting and determining when to stop training. The test set is reserved for evaluating the model's generalization to new, unseen data.

Handling missing values is another crucial aspect of preprocessing. The K-Nearest Neighbors (KNN) algorithm is used to impute missing values. Each sample's missing values are imputed using the mean value from the k nearest neighbors found in the training set. Two samples are close if the features that neither is missing are close. For the models that will predict the contrast values, $k = 5$ is chosen, and for the models that will try to capture uncertainty in the data, $k = 3$.

In neural network applications, feature normalization is important to prevent issues like vanishing or exploding gradients during training thus all the quantitative input variables are normalized and have values varying between 0 and 1.

Imputation and normalization parameters are calculated based on the training set and then applied to the training, validation and testing sets to maintain the integrity of the model evaluation process. This ensures that the model is not influenced by the validation and test sets, preserving the independence of these sets for accurate assessment.

3. MODELS

Models play an essential role in enabling intelligent decision-making and prediction in machine learning. Two models will be introduced in this section: Random Forests[4] and Neural Networks.[5]

3.1 Random Forest

The first model used to predict contrast values is a Random Forest. A Random Forest is an ensemble learning method used for both classification and regression tasks. It combines the predictions of multiple decision trees to produce more accurate and robust results. While decision trees are easy to use and interpret, they are prone to overfitting and generally perform less effectively compared to more complex models. Random Forests mitigate these issues by enhancing robustness in their predictions and reducing overfitting by averaging the predictions of the the individual trees.

3.2 Neural Network

Neural networks simulate the connections between neurons by assigning varying weights to these connections, enabling them to process information and make predictions. An artificial neuron, also called perceptron,[6] can be modeled as a mathematical function to simulate the behavior of biological neurons.

$$f(\mathbf{x}) = \sigma(\sum_i w_i x_i + b)$$

Here, σ is a non-linear activation function, $\mathbf{x}$ represents the input features vector, and w_i are the weights that need to be tuned. The perceptron is the fundamental building block of all neural networks.

Single neurons are not more expressive than linear models, but when interconnected, they form a Multi-Layer Perceptron (MLP), which is a complex non-linear parametric model. In an MLP, neurons are organized in interconnected layers. The configuration of these layers, including the number of units and outputs, are hyperparameters specific to the problem.

The loss function measures the discrepancy between the neural network's predictions ($\hat{y}$) and the true target values (y). In regression problems, the typical assumption is that the conditional distribution of the target variable y given the input $\mathbf{x}$ follows a normal distribution $p(y|\mathbf{x}) = \mathcal{N}(y; \mu = f(\mathbf{x}, \theta), \sigma^2 = 1)$ where f is parameterized by a neural network. To train this neural network, a suitable loss function is derived using maximum likelihood. Given the training data $\mathbf{d}$ and neural network parameters θ, the loss function is derived and the Mean Squared Error (MSE) is recovered as seen in Equation 1.

$$\begin{aligned} \text{argmax}_\theta p(\mathbf{d}|\theta) &= \text{argmax}_\theta \textstyle\prod_{\mathbf{x_i}, y_i \in \mathbf{d}} p(y_i|\mathbf{x_i}, \theta) \\ &= \text{argmin}_\theta \textstyle\sum_{\mathbf{x_i}, y_i \in \mathbf{d}} (y_i - f(\mathbf{x}, \theta))^2 \end{aligned} \tag{1}$$

On the other hand if the goal is to account for aleatoric heteroscedastic uncertainty in the data, then instead of generating point estimates $\hat{y} = f(\mathbf{x})$, the aim is to model the full conditional density $p(y|\mathbf{x})$. Assuming a Gaussian distribution it comes:

$$p(y|\mathbf{x}) = \mathcal{N}(y; \mu(\mathbf{x}), \sigma^2(\mathbf{x}))$$

where $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ are parametric functions to be learned.

Then using maximum likelihood the loss function is derived in Equation 2.

$$\begin{aligned} \text{argmax}_\theta p(\mathbf{d}|\theta) &= \text{argmax}_\theta \textstyle\prod_{\mathbf{x_i}, y_i \in \mathbf{d}} p(y_i|\mathbf{x_i}, \theta) \\ &= \text{argmin}_\theta \textstyle\sum_{\mathbf{x_i}, y_i \in \mathbf{d}} \frac{(y_i - \mu(\mathbf{x_i}))^2}{2\sigma^2(\mathbf{x_i})} + \log(\sigma(\mathbf{x_i})) + C \end{aligned} \tag{2}$$

This function includes the squared difference between the actual value y_i and the predicted mean $\mu(\mathbf{x_i})$, similar to the MSE, but with additional terms to account for the standard deviation, offering a more comprehensive representation of prediction uncertainty.

Those loss functions cannot be minimized analytically, so numerical methods like gradient descent are used. Gradient descent iteratively adjusts the model's parameters to minimize the loss. Starting with an initial parameter set (θ_0), the parameters are updated as follows:

$$\theta_{t+1} = \theta_t - \gamma \nabla_\theta \mathcal{L}(\theta_t)$$

Here, γ is the learning rate. This iterative process continues until the model's parameters converge to an optimal solution. The choice of initial parameters and the learning rate is crucial for successful optimization.

4. METHODOLOGY

First and foremost, it is important to distinguish between two different representations of the dataset. The dataset used as input for the models is denoted as $\mathbf{X}$ and has dimensions $(m \times n)$, then m represents the number of observations, or in this context, the number of telescope observations ($m = 843$). Each observation results in one contrast curve, so there are m contrast curves. On the other hand, n stands for the number of features from Table 1 (n=18), excluding the separation.

As mentioned in a previous section, all the contrast curves share the same separation vector, meaning that no discriminatory information between the different observations is provided by those separation vectors. In this regard, two different approaches were studied in this work:

1. **Predicting the Entire Contrast Vector :** The model predicts the entire contrast vector for each observation, where each element y_i corresponds to the separation s_i of the shared separation vector, as seen in Table 2. The motivation for this approach is to a prediction per observation, making the model easier to tune.

2. **Predicting Contrast at Specific Separations :** The model takes as input all the features along with a specific separation value and predicts the contrast at that separation, as depicted in Table 3. Although this model is more flexible than the first one, it was somewhat harder to train.

Observation ID	$\mathbf{X}$			$\mathbf{y}$
1	x_1^1	...	x_n^1	$[y_1^1, y_2^1, ..., y_N^1]$
2	x_1^2	...	x_n^2	$[y_1^2, y_2^2, ..., y_N^2]$
...		...		...
m	x_1^m	...	x_n^m	$[y_1^m, y_2^m, ..., y_N^m]$

Table 2: Vector contrast prediction

Observation ID	$\mathbf{X}$				$\mathbf{y}$
1	x_1^1	...	x_n^1	s_1	y_1^1
1	x_1^1	...	x_n^1	s_2	y_2^1
1	x_1^1	...	x_n^1	...	...
1	x_1^1	...	x_n^1	s_N	y_N^1
2	x_1^2	...	x_n^2	s_1	y_1^2
2	x_1^2	...	x_n^2	s_2	y_2^2
2	x_1^2	...	x_n^2	...	...
2	x_1^2	...	x_n^2	s_N	y_N^2
...		...		...	...
m	x_1^m	...	x_n^m	s_1	y_1^m
m	x_1^m	...	x_n^m	s_2	y_2^m
m	x_1^m	...	x_n^m	...	...
m	x_1^m	...	x_n^m	s_N	y_N^m

Table 3: Single value contrast prediction

Then, training the random forest model is straightforward, utilizing the `fit()` function from the `scikit-learn` library.
For the neural networks, the training pipeline follows a standard procedure: a maximum number of epochs is set, and observations are shuffled at the beginning of each epoch. The dataset is then divided into batches, ensuring the total number of data points is divisible by the batch size. Each batch undergoes a forward-pass for prediction, criterion evaluation (Equation 1 for contrast prediction and Equation 2 for uncertainty), and a backward-pass to update model parameters. This process repeats for all batches within an epoch, with the epoch loss being the average of batch losses.

In neural network training, the validation set is used at the end of each epoch to evaluate the model's performance on unseen data, helping to identify underfitting or overfitting. A stopping criterion is set where the model is saved if it achieves a new minimum validation score. If the validation score rises, indicating potential overfitting, training is stopped if no improvement is seen within a set number of epochs. This ensures good generalization and prevents overfitting.

The validation set also helps fine-tune both random forest and neural network models by testing various hyperparameters. Finally the set of hyperparameters that results in the lowest validation loss is chosen, optimizing the model's performance.

5. RESULTS

As depicted in Table 4, the random forest outperforms both neural networks on the test set. Interestingly, the neural network predicting a single contrast value has a higher MSE indicating less precision compared to the network that outputs a vector. However, the single-output neural network performs better on average when considering the Mean Absolute Error (MAE), as its predictions are closer to the actual values. The higher MSE is due to significant errors being heavily penalized.

Restricting the random forest to split interior nodes using only a subset of the input features appears to be a robust approach for achieving good results. In this case, only one feature is used to split each interior node, and the strategy is to use the best possible split, which is handled by scikit-learn. Using all features resulted in poorer performance compared to the neural networks.

Model	**MSE**	**MAE**
Random Forest (single)	0.1848	0.2540
Neural Network (single)	0.2632	0.3040
Neural Network (vector)	0.2446	0.3264

Table 4: Results of the models predicting the contrast on the test set

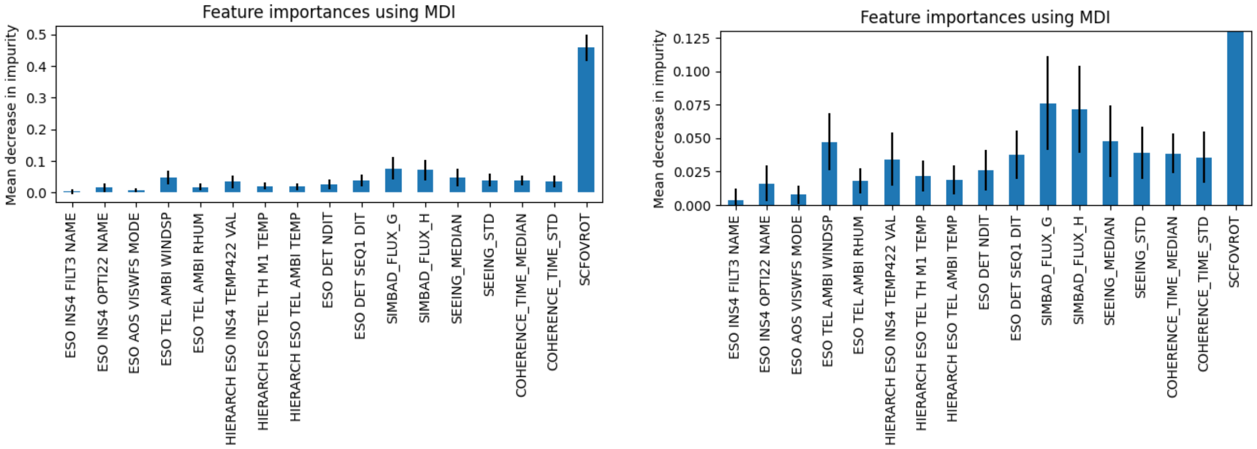
In decision trees, impurity measures how mixed the target variable is within a node. The lower the impurity, the purer the node, meaning it is more homogeneous with respect to the target variable. When a decision tree splits a node based on a feature, it aims to reduce impurity. The average decrease in impurity brought about by a feature across all trees in an ensemble is used to quantify the importance of that feature in predicting the target variable. This method is called Mean Decrease Impurity (MDI), and a theoretical study of this technique has been conducted by Louppe et al. (2013).[7] Figure 3 illustrates the importance of different features based on this method.

The two main limitations of MDI are its potential bias towards high cardinality features and its unreliability with highly correlated features. Therefore, another technique known as feature permutation has been used to investigate whether the same results would be obtained. Permutation feature importance is a method for evaluating how much each feature contributes to a model's performance on a dataset. This technique is especially helpful for complex or non-transparent models. It works by randomly shuffling the values of one feature at a time and then measuring how much the model's performance score decreases as a result.[4] Results of this method can be found in Figure 4.

As expected, the field of rotation is the most explanatory input parameter of the model for ADI reductions. Next are the stellar magnitudes and wind speed, which have a direct impact on the speckle residuals and photon noise limit. Another interesting observation is that the wind speed at the observatory seems to have a similar impact as the seeing, which was not expected.

Figure 5 displays random examples of predictions made by the predictive models.

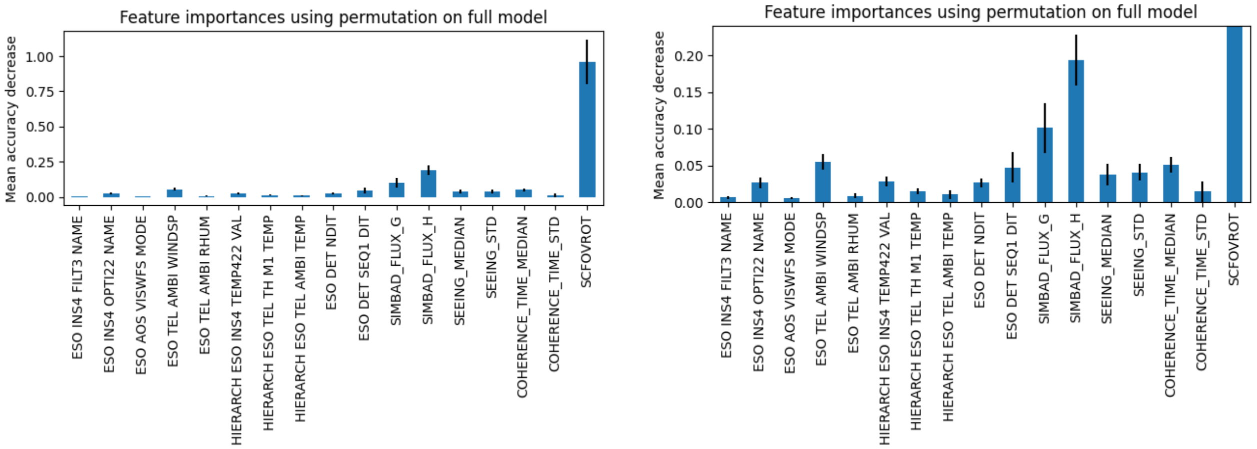
In some observations, the predicted contrast curves closely resemble the actual ones (e.g. Figure 5(f)). This can be attributed to the fact that different observations of the same target star, under different conditions, are conducted. Thus, the same target can appear in both the training and testing sets. It's important to clarify that this is intentional. The aim is to gain a deeper understanding of the impact of various input features on



(a)

(b)

Figure 3: Feature importance using Mean Decrease Impurity



(a)

(b)

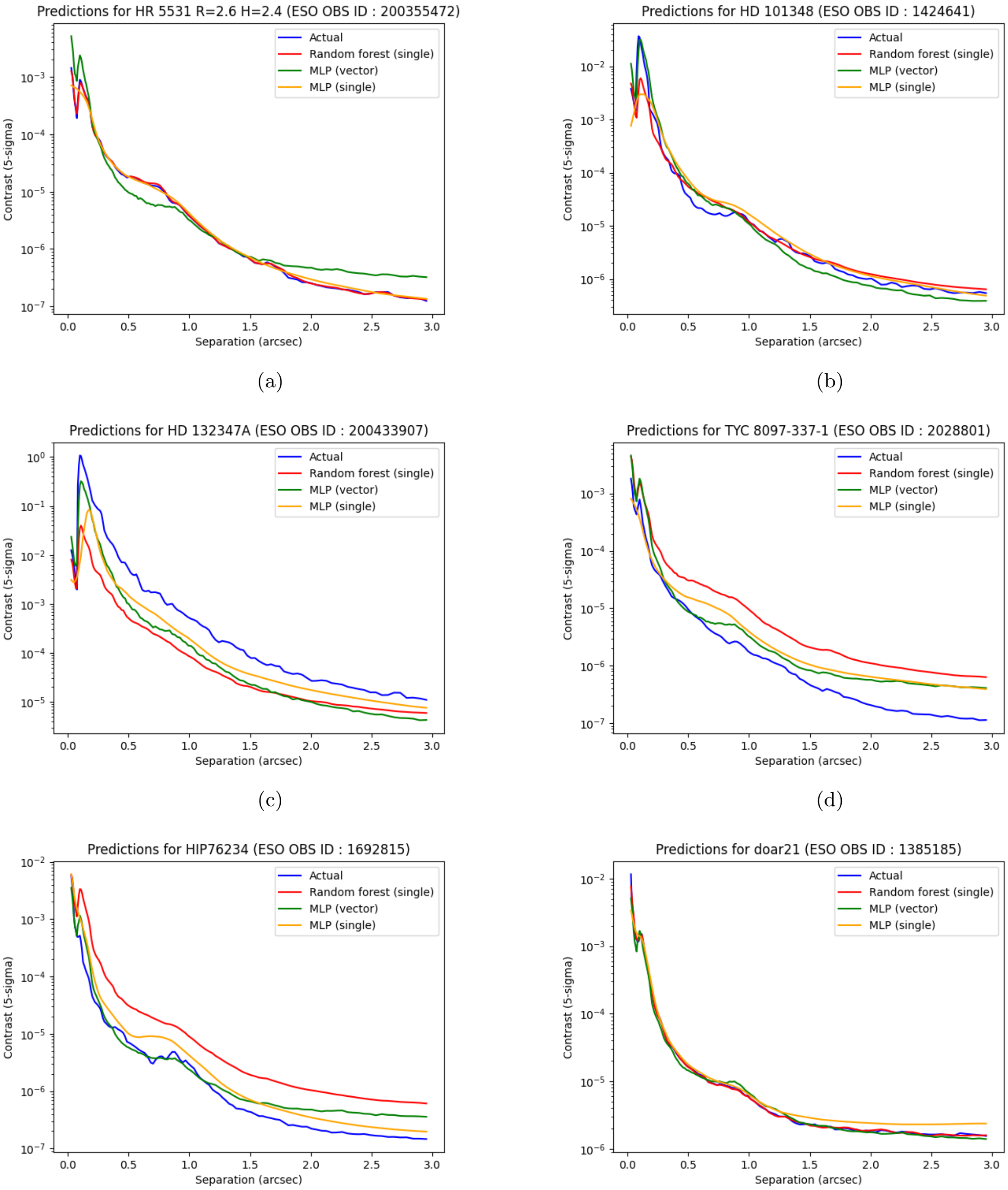
Figure 4: Feature importance based on feature permutation

the contrast value, such as atmospheric conditions and observing strategies. Therefore, in this context, the same target may be observed at different times and under different observing conditions, rendering the observations independent even for the same targets.

Figures 9a, 9b and 9c examine the predictions at three separation values: 0.25 arcsec, 0.8 arcsec, and 2 arcsec. At 0.25 arcsec, typically the first spike in contrast curves, the random forest predictions often exceed actual values, except for unusually high contrasts. At 0.8 arcsec and 2 arcsec, where contrast values are smaller and closer to the noise limit, the random forest predictions are more linear but still tend to overestimate the actual values. One limitation observed is the model's inability to effectively identify outliers, often underpredicting high actual contrast values.

For the neural network predicting single contrast values (Table 3), predictions are generally closer to the identity line across all separations, indicating less bias in predicting smaller or larger contrast values. However, the predictions are more dispersed compared to the random forest, reflecting less precision on average. The neural network predicting a vector of contrasts (Table 2) exhibits similar behavior to the single-output model, maintaining closeness to the identity line but also showing some dispersion in predictions. These observations align with the loss values presented, highlighting differences in precision and bias among the models.

As mentioned in Section 3.2, neural networks are used for two distinct objectives in this study: contrast

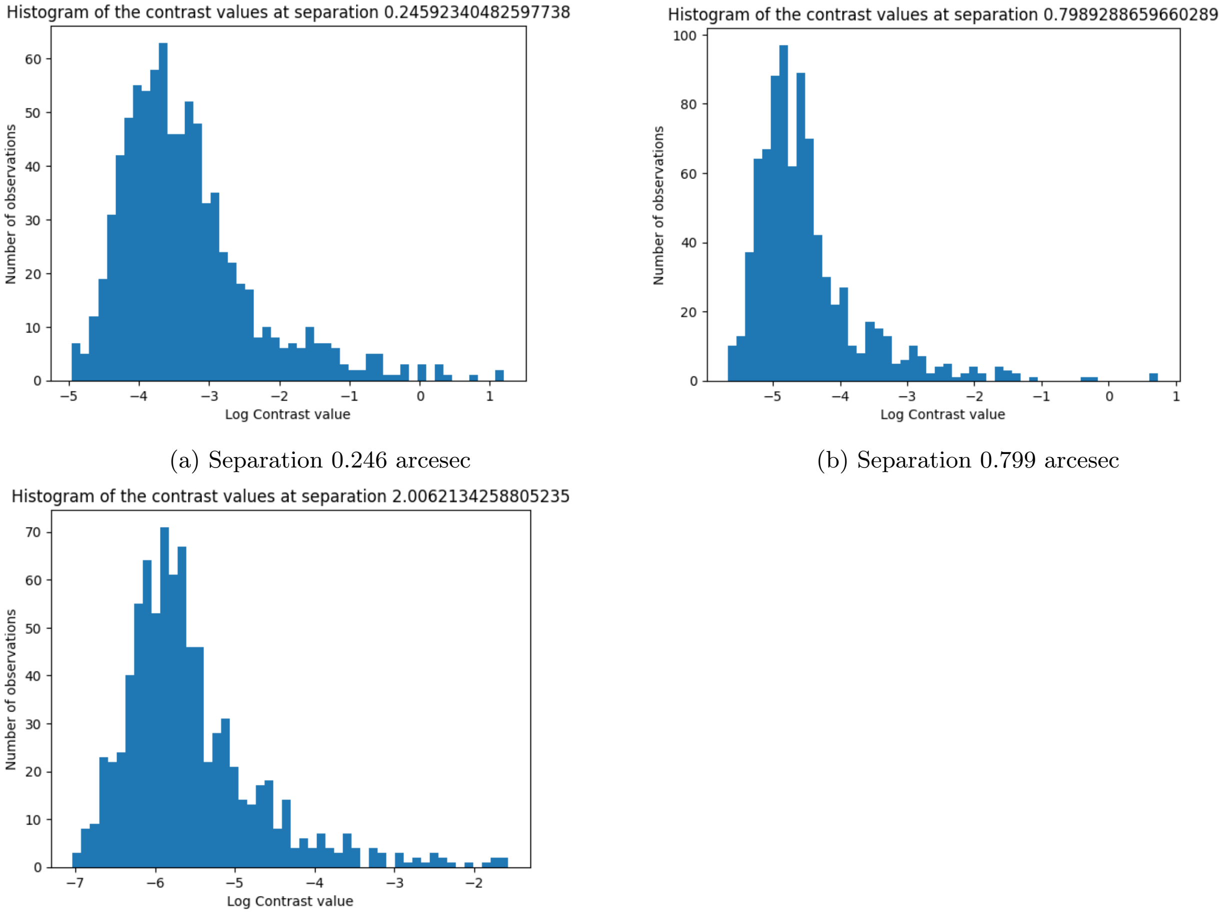


(a) (b) (c) (d) (e) (f)

Figure 5: Predictions of the different models (test set)

prediction and uncertainty capture. For contrast prediction, a neural network is trained using the loss function defined in Equation 1. On the other hand, for uncertainty capture, the neural network uses the loss function specified in Equation 2.

Figure 10 shows the predictions made by the model capturing the uncertainty. The red curve represents the


Histogram of the contrast values at separation 0.24592340482597738

Histogram of the contrast values at separation 0.7989288659660289

Histogram of the contrast values at separation 2.0062134258805235

(a) Separation 0.246 arcesec

(b) Separation 0.799 arcesec

(c) Separation 2.006 arcesec

Figure 6: Histograms of the (log) contrast values at different separations

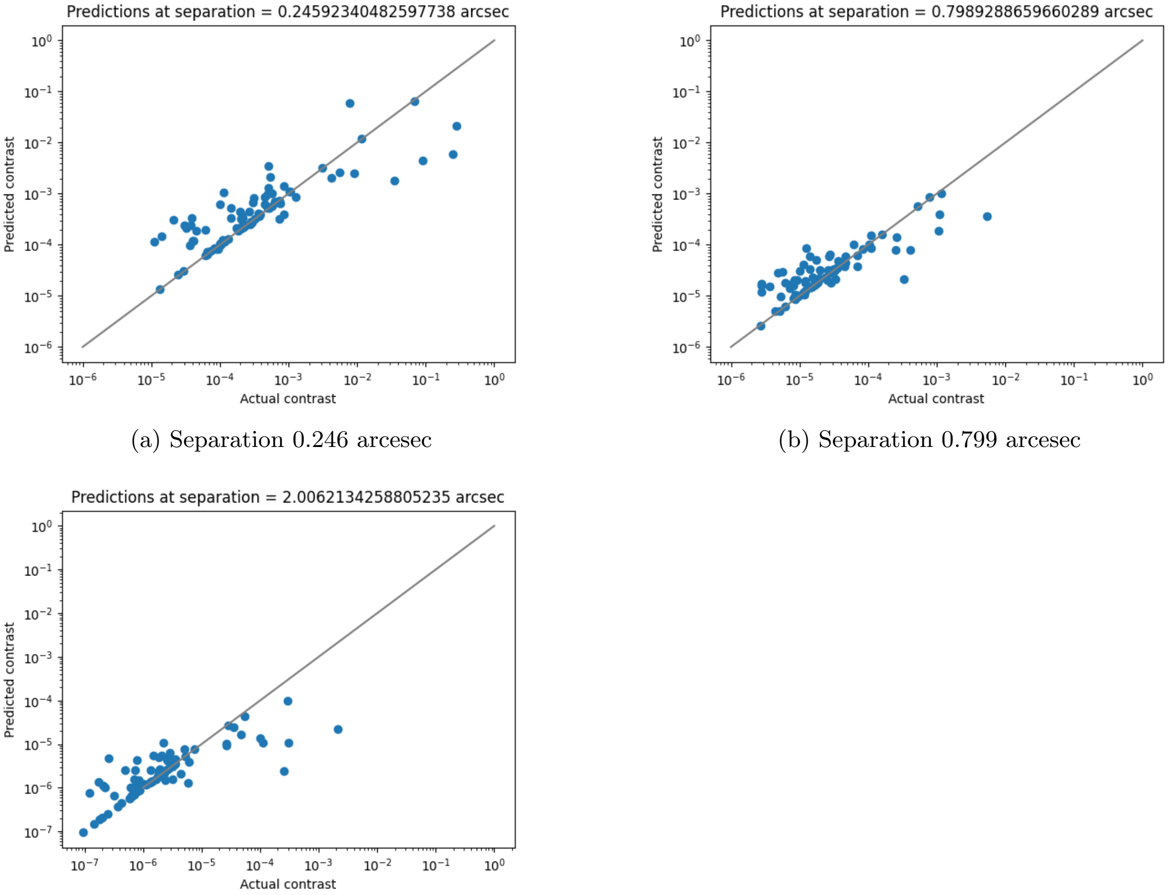
mean of the distribution, which can be identified as the prediction of the model, while the blue area around it represents the confidence interval $[\mu - \sigma; \mu + \sigma]$.

6. CONCLUSION

In this study, a dataset was built from scratch using observations from SPHERE in the H2H3 bands and the Angular Differential Imaging (ADI) technique. The objective functions, which are the values to be predicted, are the contrast limits distinguishing an exoplanet from its host star. To predict these contrast values, explanatory variables were gathered from various sources and fed into the models. This process resulted in a dataset comprising 843 observations, each with 17 features.

In the second part of this work, random forests and neural networks were employed to predict contrast. Due to data limitations, a relatively simple Multi-Layer Perceptron architecture was used. The initial vector-output neural network showed promise but lacked flexibility, leading to the development of a single-output network. All these different models along with the database are publicly available on github.

The final results indicated that random forests offered slightly better precision than neural networks, though both models exhibited similar overall performance with different failure points. Notably, predicting contrast was not the sole objective; capturing uncertainty, particularly in neural networks, was also a focus. The results for uncertainty prediction were encouraging, suggesting potential for further research in this area.



(a) Separation 0.246 arcesec

(b) Separation 0.799 arcesec

(c) Separation 2.006 arcesec

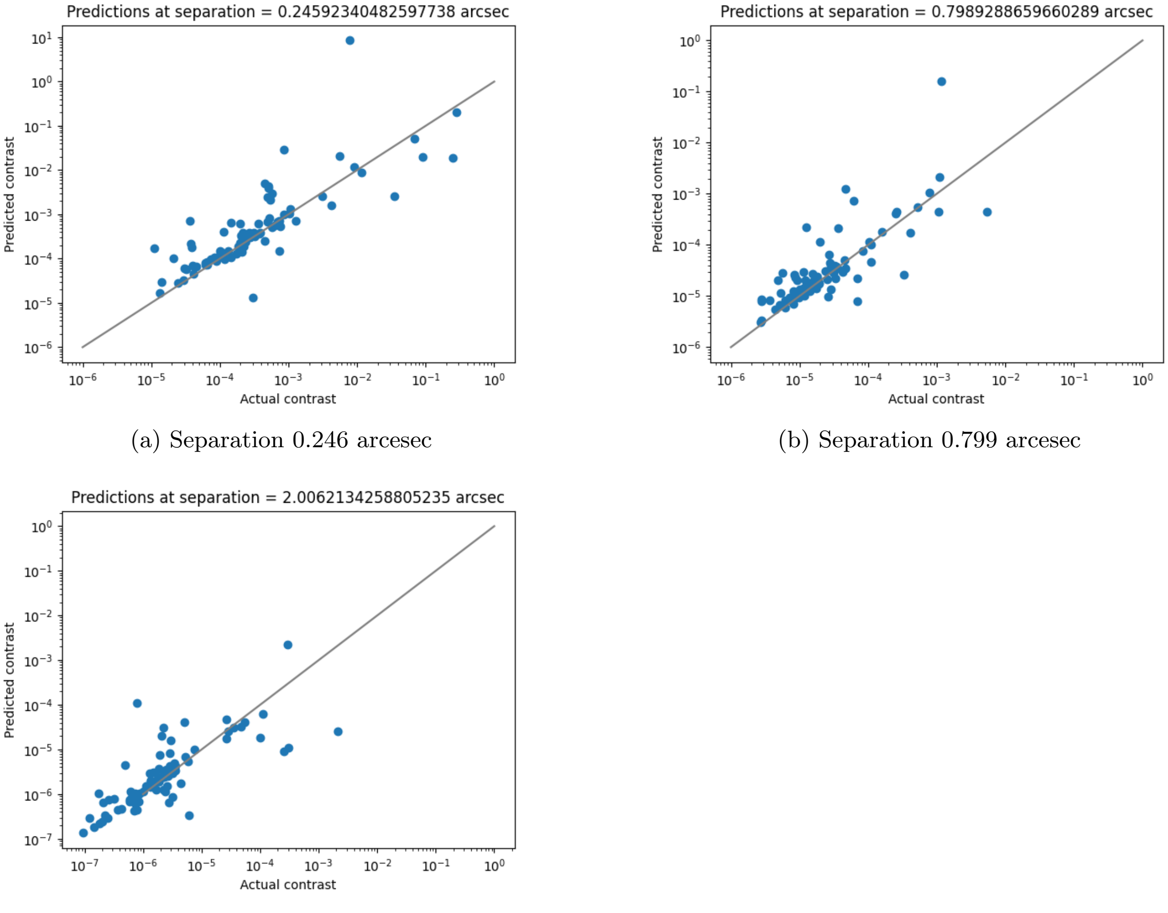
Figure 7: Contrast actual vs predicted values (Random Forest)

In conclusion, while better results might have been achievable with more data, advanced modeling techniques, and extensive fine-tuning, the study provides a solid foundation. Future work could build on this by exploring more advanced approaches for uncertainty prediction, such as outputting a prediction vector with the mean and a covariance matrix to capture deeper data relationships.

7. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Program (ERC Grant Agreement n. 337569), and from the Wallonia-Brussels Federation (grant for Concerted Research Actions).

This work has made use of the the SPHERE Data Centre, jointly operated by OSUG/IPAG (Grenoble), PYTHEAS/LAM/CESAM (Marseille), OCA/Lagrange (Nice), Observatoire de Paris/LESIA (Paris), and Observatoire de Lyon.[8910]



(a) Separation 0.246 arcesec

(b) Separation 0.799 arcesec

(c) Separation 2.006 arcesec

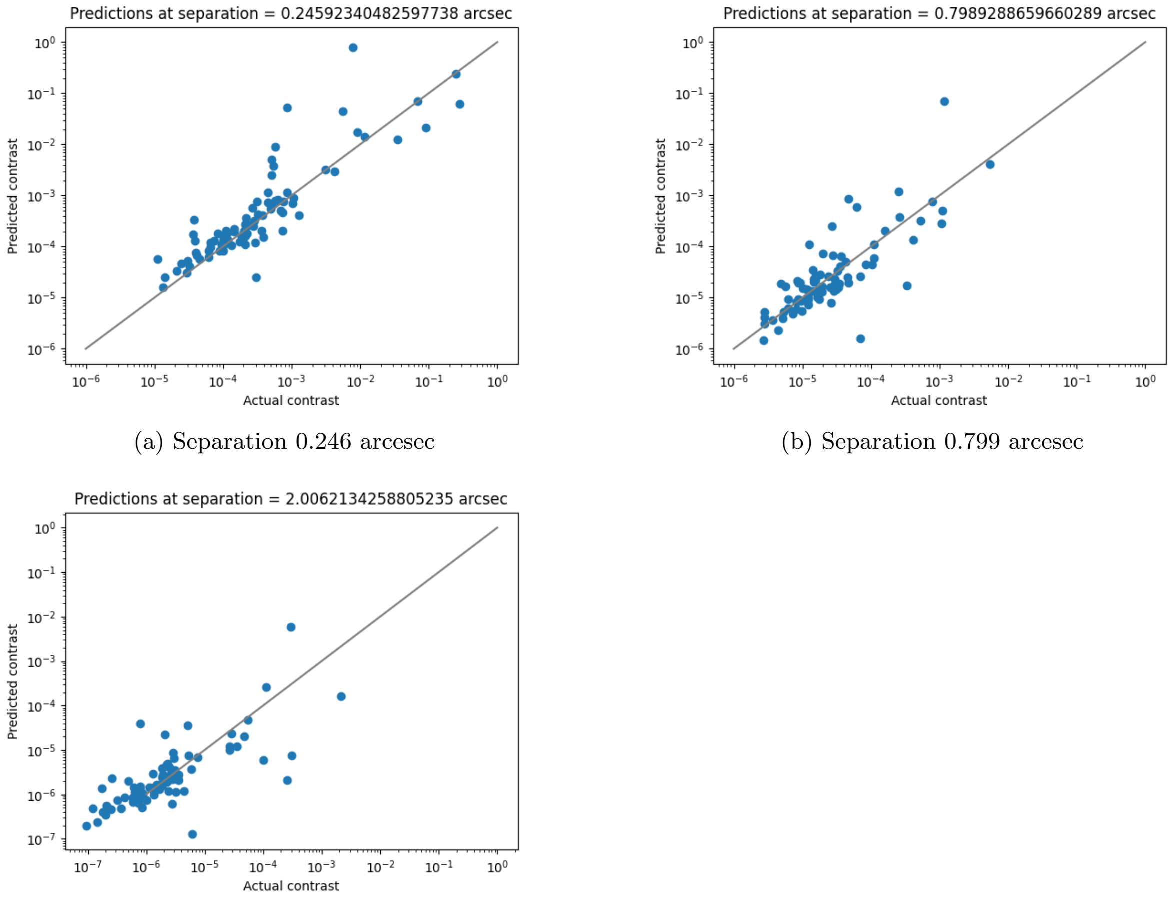
Figure 8: Contrast actual vs predicted values (MLP single)

REFERENCES

[1] Beuzit, J. L., Vigan, A., Mouillet, D., Dohlen, K., Gratton, R., Boccaletti, A., Sauvage, J. F., Schmid, H. M., Langlois, M., Petit, C., Baruffolo, A., Feldt, M., Milli, J., Wahhaj, Z., Abe, L., Anselmi, U., Antichi, J., Barette, R., Baudrand, J., Baudoz, P., Bazzon, A., Bernardi, P., Blanchard, P., Brast, R., Bruno, P., Buey, T., Carbillet, M., Carle, M., Cascone, E., Chapron, F., Charton, J., Chauvin, G., Claudi, R., Costille, A., De Caprio, V., de Boer, J., Delboulbé, A., Desidera, S., Dominik, C., Downing, M., Dupuis, O., Fabron, C., Fantinel, D., Farisato, G., Feautrier, P., Fedrigo, E., Fusco, T., Gigan, P., Ginski, C., Girard, J., Giro, E., Gisler, D., Gluck, L., Gry, C., Henning, T., Hubin, N., Hugot, E., Incorvaia, S., Jaquet, M., Kasper, M., Lagadec, E., Lagrange, A. M., Le Coroller, H., Le Mignant, D., Le Ruyet, B., Lessio, G., Lizon, J. L., Llored, M., Lundin, L., Madec, F., Magnard, Y., Marteaud, M., Martinez, P., Maurel, D., Ménard, F., Mesa, D., Möller-Nilsson, O., Moulin, T., Moutou, C., Origné, A., Parisot, J., Pavlov, A., Perret, D., Pragt, J., Puget, P., Rabou, P., Ramos, J., Reess, J. M., Rigal, F., Rochat, S., Roelfsema, R., Rousset, G., Roux, A., Saisse, M., Salasnich, B., Santambrogio, E., Scuderi, S., Segransan, D., Sevin, A., Siebenmorgen, R., Soenke, C., Stadler, E., Suarez, M., Tiphène, D., Turatto, M., Udry, S., Vakili, F., Waters, L. B. F. M., Weber, L., Wildi, F., Zins, G., and Zurlo, A., "SPHERE: the exoplanet imager for the Very Large Telescope," **631**, A155 (Nov. 2019).

[2] Marois, C., Lafrenière, D., Doyon, R., Macintosh, B., and Nadeau, D., "Angular Differential Imaging: A Powerful High-Contrast Imaging Technique," **641**, 556–564 (Apr. 2006).

[3] Xuan, W. J., Mawet, D., Ngo, H., Ruane, G., Bailey, V. P., Choquet, É., Absil, O., Alvarez, C., Bryan, M.,



(a) Separation 0.246 arcesec

(b) Separation 0.799 arcesec

(c) Separation 2.006 arcesec

Figure 9: Contrast actual vs predicted values (MLP vector)

Cook, T., et al., “Characterizing the performance of the nirc2 vortex coronagraph at wm keck observatory,” *The Astronomical Journal* **156**(4), 156 (2018).

[4] Breiman, L., “Random forests,” *Machine learning* **45**, 5–32 (2001).

[5] Krogh, A., “What are artificial neural networks?,” *Nature biotechnology* **26**(2), 195–197 (2008).

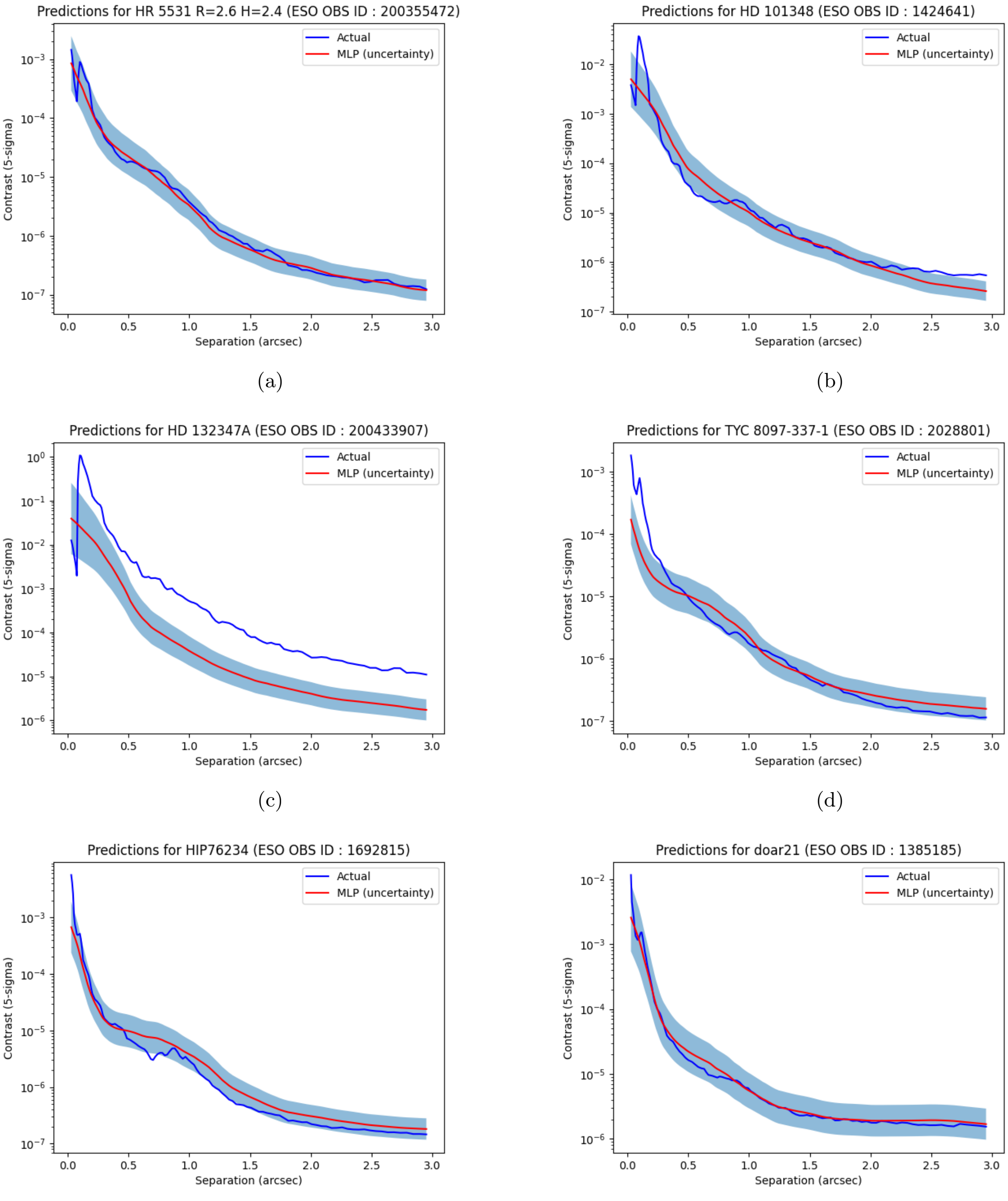
[6] Rosenblatt, F., [*The perceptron, a perceiving and recognizing automaton Project Para*], Cornell Aeronautical Laboratory (1957).

[7] Louppe, G., Wehenkel, L., Sutera, A., and Geurts, P., “Understanding variable importances in forests of randomized trees,” *Advances in neural information processing systems* **26** (2013).

[8] Delorme, P., Meunier, N., Albert, D., Lagadec, E., Coroller, H. L., Galicher, R., Mouillet, D., Boccaletti, A., Mesa, D., Meunier, J.-C., et al., “The sphere data center: a reference for high contrast imaging processing,” *arXiv preprint arXiv:1712.06948* (2017).

[9] Galicher, R., Boccaletti, A., Mesa, D., Delorme, P., Gratton, R., Langlois, M., Lagrange, A.-M., Maire, A.-L., Le Coroller, H., Chauvin, G., et al., “Astrometric and photometric accuracies in high contrast imaging: The sphere speckle calibration tool (specal),” *Astronomy & Astrophysics* **615**, A92 (2018).

[10] Maire, A.-L., Langlois, M., Dohlen, K., Lagrange, A.-M., Gratton, R., Chauvin, G., Desidera, S., Girard, J. H., Milli, J., Vigan, A., et al., “Sphere irdis and ifs astrometric strategy and calibration,” in [*Ground-based and Airborne Instrumentation for Astronomy VI*], **9908**, 975–986, SPIE (2016).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10: Predictions of the model capturing uncertainty (test set)