Predicting Unitarity and Bounded from Below Constraints Using Machine Learning

Darius Jurčiukonis

Vilnius University, Institute of Theoretical Physics and Astronomy
darius.jurciukonis@tfai.vu.lt

September 15, 2023

UNI and BFB

There are two important theoretical constraints that one must impose on the scalar potential.

- **UNI**: all the (tree-level) scalar–scalar scattering amplitudes must respect unitarity.
- **BFB**: the potential must have a minimum, viz. they prevent the existence of directions in field space along which the potential is unbounded from below.

UNI

- It can be computed analytically.
- Typically, the computation requires determining the eigenvalues of scattering matrices.
- The computations are precise and fast.

BFB

- Analytical computation is feasible only for simple models.
- For complex models, minimization of the potential is required.
- The computations are often slow and imprecise.

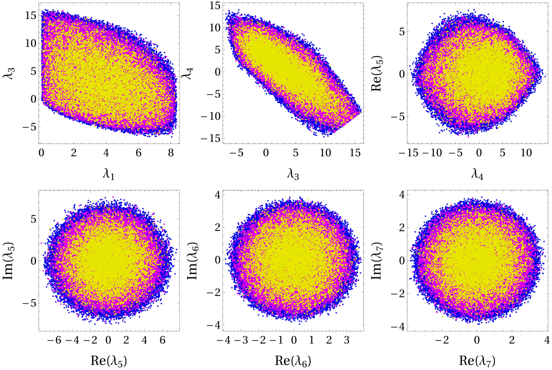
Parameters

Figure on right:

- blue: 10^6 samples
- magenta: 10^5 samples
- yellow: 10^4 samples

Table below:

- Percentage of true samples when lambdas are generated randomly (raw data).
- BFB-I: the percentage of true examples from raw data.
- BFB-II: the percentage from datasets refined by UNI conditions.



Model (λ's)	UNI	BFB-I	BFB-II	UNI+BFB
G2HDM (10)	1.02	23.3	3.25	0.033
LRM (13)	0.41	18.4	0.7	0.0028
A3HDM (14)	0.18	1.73	0.05	0.00009

Strategy

Machine learning is currently utilized across various fields; let's see if it can be applied to UNI and BFB calculations.

Three models with precise "analytical" procedures for BFB computations were investigated to verify the reliability of machine learning.

- General 2HDM [DJ and L.Lavoura, 1807.04244; M. Maniatis et al., hep-ph/0605184; I.P. Ivanov and J.P. Silva, 1507.05100].
- CP conseved Left-Right model [D. Fontes, DJ and L. Lavoura, 2212.12075; K. Kannike, 2109.01671].
- Aligned 3HDM [DJ, L. Lavoura, 2103.16635; P.M.Ferreira et al., 1711.02042].

There are two computational strategies:

- Train networks to predict BFB constraints using data that satisfy UNI constraints.
- Train networks to predict both UNI and BFB constraints simultaneously.

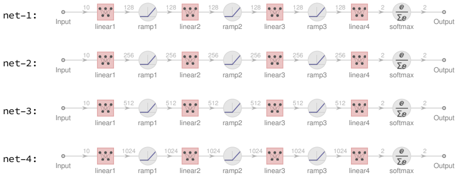
Technical details

For this research, a desktop computer was used.

- CPU: Intel® Core™ i9-13900K, 24 Cores (8P+16E) 2.2-5.8 GHz
- GPU: NVIDIA GeForce RTX 4090
- SOFT: Ubuntu 22.04 LTS; Wolfram Mathematica 13.2

Machine learning parameters:

- training data: $10^6 - 10^7$ samples
- bach size: $2^{14} = 16\,384$
- training rounds: up to 500
- networks: linear nets of 8 layers



- The initial network, net-1, is trained using raw data.
- This network, net-1, is subsequently used to prepare the training data for net-2.
- Both net-3 and net-4 are trained using the same data as net-2.
- These networks, net-3 and net-4, are then utilized to filter the predictions made by net-2.

The scalar potential

The most general scalar potential for the 2HDM is

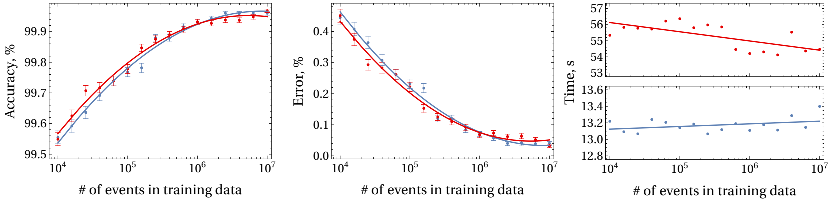
$$
\begin{aligned}
V \quad = \quad & \mu_1 \phi_1^\dagger \Phi_1 + \mu_2 \Phi_2^\dagger \Phi_2 + \left(\mu_3 \Phi_1^\dagger \Phi_2 + \mathrm{H.c.}\right) \\
& + \frac{\lambda_1}{2}\left(\Phi_1^\dagger \Phi_1\right)^2 + \frac{\lambda_2}{2}\left(\Phi_2^\dagger \Phi_2\right)^2 + \lambda_3\, \Phi_1^\dagger \Phi_1\, \Phi_2^\dagger \Phi_2 + \lambda_4\, \Phi_1^\dagger \Phi_2\, \Phi_2^\dagger \Phi_1 \\
& + \left[\frac{\lambda_5}{2}\left(\Phi_1^\dagger \Phi_2\right)^2 + \lambda_6\, \Phi_1^\dagger \Phi_1\, \Phi_1^\dagger \Phi_2 + \lambda_7\, \Phi_2^\dagger \Phi_2\, \Phi_1^\dagger \Phi_2 + \mathrm{H.c.}\right],
\end{aligned}
$$

where $\mu_{1,2}$ and $\lambda_{1,2,3,4}$ are real, while the remaining parameters are complex. The quartic part of the scalar potential comprises **10** real parameters.

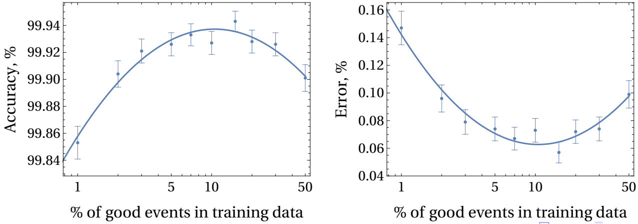
- Simple UNI conditions
- Algorithms for precise BFB computations
- Fast computations

Classifier measurements

Classifier measurements (accuracy, error and prediction time) as functions from the size of raw training data. Blue points represent net-1, while red points correspond to measurements from net-4.



Dependence of net-1's accuracy and error on the percentage of true samples in the raw training data.

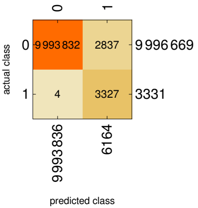


Classifier measurements

Classifier measurements for the net-1 (left) and net-2 (right).

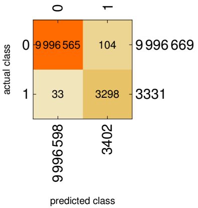
Classifier Measurements

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.9716 \pm 0.0005)\%$
Accuracy baseline	$(99.9667 \pm 0.0006)\%$
Geometric mean of probabilities	0.999 ± 0.000025
Mean cross entropy	0.00115 ± 0.000025
Single evaluation time	4.23 ms/example
Batch evaluation speed	1.09 examples/μs



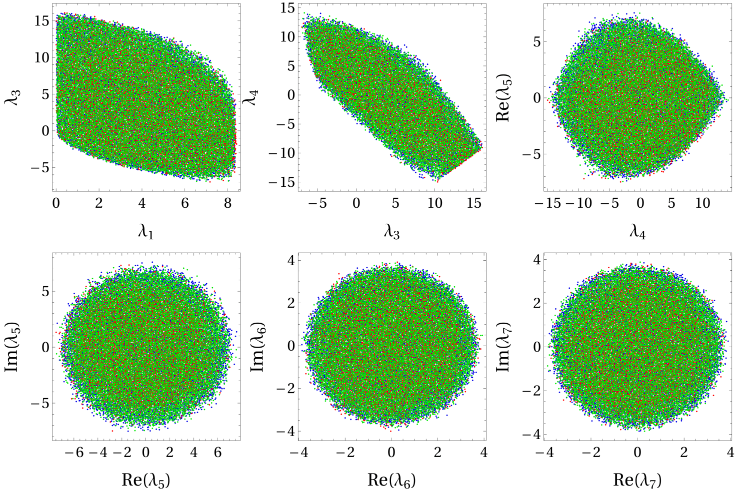
Classifier Measurements

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.99863 \pm 0.00012)\%$
Accuracy baseline	$(99.9667 \pm 0.0006)\%$
Geometric mean of probabilities	$1.00 \pm 3.5 \times 10^{-6}$
Mean cross entropy	$0.0000378 \pm 3.5 \times 10^{-6}$
Single evaluation time	4.2 ms/example
Batch evaluation speed	813. examples/ms



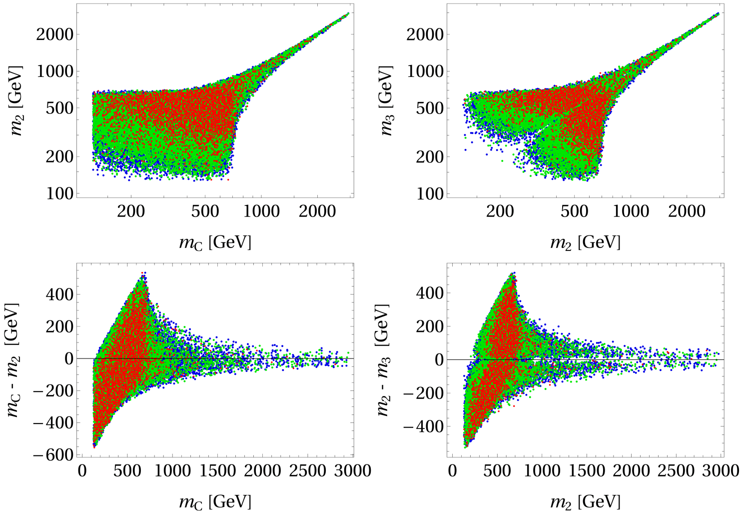
Lambdas

Scatter plots of λ values for the G2HDM.



Masses

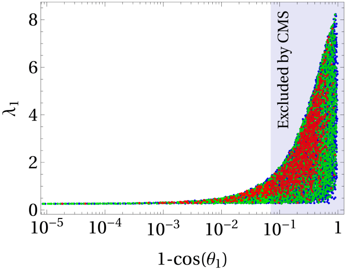
Scatter plots of Higgs masses for the G2HDM.

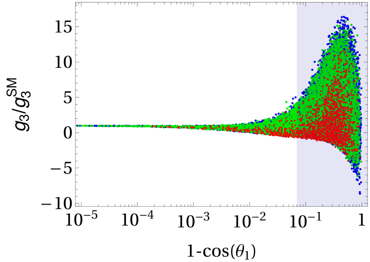


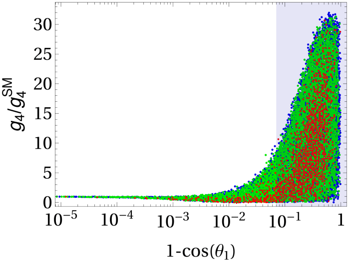
Couplings

The three-Higgs vertex [DJ, L.Lavoura, 1807.04244]:

$$
\begin{aligned}
g_3 \quad = \quad & \frac{v}{\sqrt{2}} \Big[\lambda_1 c_1^3 + (\lambda_3 + \lambda_4)\, s_1^2 c_1 \\
& + s_1^2 c_1 \left(c_3^2 - s_3^2 \right) \Re\lambda_5 - 2 s_1^2 c_1 c_3 s_3\, \Im\lambda_5 \\
& + 3 s_1 c_1^2 \left(c_3\, \Re\lambda_6 - s_3\, \Im\lambda_6 \right) \\
& + s_1^3 \left(c_3\, \Re\lambda_7 - s_3\, \Im\lambda_7 \right) \Big] .
\end{aligned}
$$







The scalar potential

The scalar potential in our LRM is of the form $V = V_H + V_\Phi + V_{H\Phi}$, where [D. Fontes, DJ, L. Lavoura, 2212.12075] [K. Kannike, 2109.01671]

$$
\begin{aligned}
V_H &= \mu_L\, H_L^\dagger H_L + \mu_R\, H_R^\dagger H_R \\
&\quad + \lambda_L\, H_L^\dagger H_L\, H_L^\dagger H_L + \lambda_R\, H_R^\dagger H_R\, H_R^\dagger H_R + \lambda_{LR}\, H_L^\dagger H_L\, H_R^\dagger H_R, \\
V_\Phi &= \mu_1 \operatorname{tr}\left(\Phi^\dagger \Phi\right) + \mu_2 \operatorname{tr}\left(\tilde{\Phi}^\dagger \Phi + \Phi^\dagger \tilde{\Phi}\right) \\
&\quad + \lambda_1 \left[\operatorname{tr}\left(\Phi^\dagger \Phi\right)\right]^2 + \lambda_2 \left\{\left[\operatorname{tr}\left(\Phi^\dagger \tilde{\Phi}\right)\right]^2 + \mathrm{H.c.}\right\} + \lambda_3 \left|\operatorname{tr}\left(\Phi^\dagger \tilde{\Phi}\right)\right|^2 \\
&\quad + \lambda_4 \operatorname{tr}\left(\Phi^\dagger \Phi\right) \operatorname{tr}\left(\tilde{\Phi}^\dagger \Phi + \Phi^\dagger \tilde{\Phi}\right), \\
V_{H\Phi} &= m_1 \left(H_L^\dagger \Phi H_R + H_R^\dagger \Phi^\dagger H_L\right) + m_2 \left(H_L^\dagger \tilde{\Phi} H_R + H_R^\dagger \tilde{\Phi}^\dagger H_L\right) \\
&\quad + \lambda_{3L}\, H_L^\dagger \Phi \Phi^\dagger H_L + \lambda_{3R}\, H_R^\dagger \Phi^\dagger \Phi H_R + \lambda_{4L}\, H_L^\dagger \tilde{\Phi} \tilde{\Phi}^\dagger H_L + \lambda_{4R}\, H_R^\dagger \tilde{\Phi}^\dagger \tilde{\Phi} H_R \\
&\quad + \lambda_{5L}\, H_L^\dagger \left(\Phi \tilde{\Phi}^\dagger + \tilde{\Phi} \Phi^\dagger\right) H_L + \lambda_{5R}\, H_R^\dagger \left(\Phi^\dagger \tilde{\Phi} + \tilde{\Phi}^\dagger \Phi\right) H_R .
\end{aligned}
$$

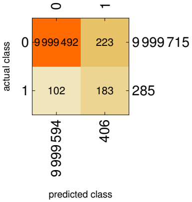
All parameters of the potential are real because of the assumed CP conservation. The quartic part of the scalar potential comprises **13** real parameters.

Classifier measurements

Classifier measurements for the net-1 (left) and net-2 (right).

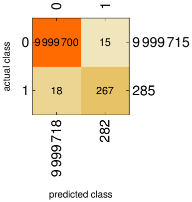
Classifier Measurements (net-1)

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.99675 \pm 0.00018)\%$
Accuracy baseline	$(99.99715 \pm 0.00017)\%$
Geometric mean of probabilities	1.00 ± 0.000011
Mean cross entropy	0.000150 ± 0.000011
Single evaluation time	4.39 ms/example
Batch evaluation speed	1.07 examples/μs



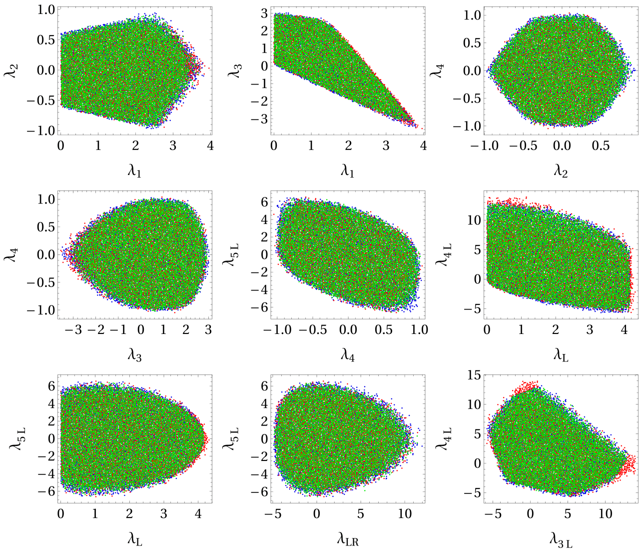
Classifier Measurements (net-2)

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.99967 \pm 0.00006)\%$
Accuracy baseline	$(99.99715 \pm 0.00017)\%$
Geometric mean of probabilities	$1.00 \pm 1.6 \times 10^{-6}$
Mean cross entropy	$8.52 \times 10^{-6} \pm 1.6 \times 10^{-6}$
Single evaluation time	3.65 ms/example
Batch evaluation speed	806. examples/ms



Lambdas

Scatter plots of λ values for the LRM.



The scalar potential

We consider following potential for the aligned 3HDM [DJ, L. Lavoura, 2103.16635]

$$V = \mu_1 \, \Phi_1^\dagger \Phi_1 + \mu_2 \, \Phi_2^\dagger \Phi_2 + \mu_3 \, \Phi_3^\dagger \Phi_3 + \left(\mu_4 \, \Phi_1^\dagger \Phi_2 + \mu_5 \, \Phi_1^\dagger \Phi_3 + \mu_6 \, \Phi_2^\dagger \Phi_3 + \mathrm{H.c.} \right)$$

$$+ \frac{\lambda_1}{2} \left(\Phi_1^\dagger \Phi_1 \right)^2 + \lambda_4 \, \Phi_1^\dagger \Phi_1 \, \Phi_2^\dagger \Phi_2 + \lambda_5 \, \Phi_1^\dagger \Phi_1 \, \Phi_3^\dagger \Phi_3 + \lambda_7 \, \Phi_1^\dagger \Phi_2 \, \Phi_2^\dagger \Phi_1$$

$$+ \lambda_8 \, \Phi_1^\dagger \Phi_3 \, \Phi_3^\dagger \Phi_1 + \left[\frac{\lambda_{10}}{2} \left(\Phi_1^\dagger \Phi_2 \right)^2 + \frac{\lambda_{11}}{2} \left(\Phi_1^\dagger \Phi_3 \right)^2 + \lambda_{13} \, \Phi_1^\dagger \Phi_1 \, \Phi_1^\dagger \Phi_2 \right.$$

$$\left. + \lambda_{14} \, \Phi_1^\dagger \Phi_1 \, \Phi_1^\dagger \Phi_3 + \lambda_{19} \, \Phi_1^\dagger \Phi_1 \, \Phi_2^\dagger \Phi_3 + \lambda_{22} \, \Phi_1^\dagger \Phi_3 \, \Phi_2^\dagger \Phi_1 + \lambda_{25} \, \Phi_1^\dagger \Phi_2 \, \Phi_1^\dagger \Phi_3 + \mathrm{H.c.} \right],$$

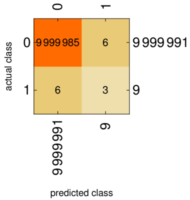
where $\mu_{1,2,3}$ and $\lambda_{1,4,5,7,8}$ are real and the remaining parameters are in general complex. The quartic part of the scalar potential comprises **14** real parameters.

Classifier measurements

Classifier measurements for the net-1 (left) and net-2 (right).

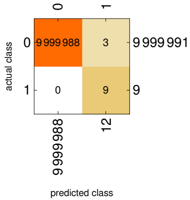
Classifier Measurements

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.999880 \pm 0.000035)\%$
Accuracy baseline	$(99.999910 \pm 0.000030)\%$
Geometric mean of probabilities	$1.00 \pm 8.1\times10^{-7}$
Mean cross entropy	$4.63\times10^{-6} \pm 8.1\times10^{-7}$
Single evaluation time	3.56 ms/example
Batch evaluation speed	1.36 examples/μs



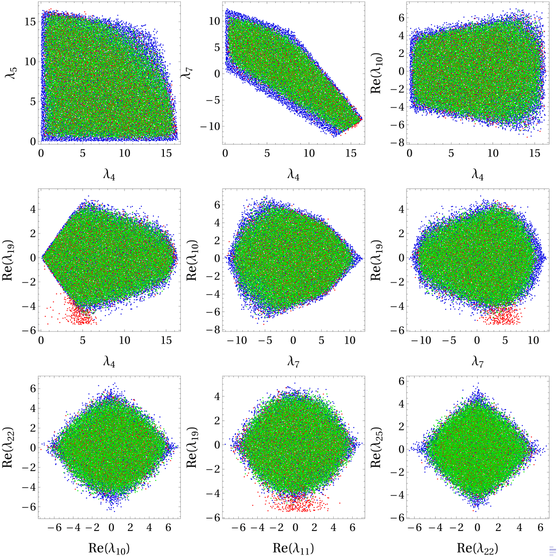
Classifier Measurements

Classifier method	Net
Number of test examples	10 000 000
Accuracy	$(99.999970 \pm 0.000017)\%$
Accuracy baseline	$(99.999910 \pm 0.000030)\%$
Geometric mean of probabilities	$1.00 \pm 4.2\times10^{-7}$
Mean cross entropy	$7.81\times10^{-7} \pm 4.2\times10^{-7}$
Single evaluation time	3.93 ms/example
Batch evaluation speed	826. examples/ms



Lambdas

Scatter plots of λ values for the aligned 3HDM.



Times

- The minimization of the scalar potential demands substantial computational resources.
- Neural networks have the potential to significantly accelerate these computations.
- Recalculating nets predictions using the minimization procedure ensures that computation times remain manageable.

Model	UNI+min.	neural nets	neural nets+min.	ratio-I	ratio-II
G2HDM	131	1.7	35	77	3.7
LRM	2224	18	78	123.5	28.5
A3HDM	3874	346	382	11.2	10.1

Table: Computation times (in seconds) needed to identify 1000 true samples. The second column presents the computation time employing both UNI and global minimization methods. The third column depicts the time taken using only neural networks. The fourth column outlines the time required when utilizing neural networks, followed by a check using global minimization.

Accuracy

Model	net-1	net-2	net-3,4	net-2 – net-4
G2HDM	52-55	96-97	97-98	> 99
LRM	42-47	91-92	91-93	> 98
A3HDM	13-15	90-92	90-91	> 97

Table: Percentage of true samples within predicted results (from raw data), verified using analytical UNI+BFB conditions.

Model	net-1	net-2	net-3,4	net-2 – net-4
G2HDM	96-97	98-99	97-98	> 99
LRM	84-86	94-96	95-97	> 99
A3HDM	69-74	92-95	92-96	> 98

Table: Percentage of true samples within predicted results (from UNI data), verified using analytical BFB conditions.

Conclusions

- This analysis demonstrates that machine learning techniques can effectively predict UNI and BFB constraints in multi-scalar models.
- Simple linear networks can achieve high prediction accuracy, though they require appropriately prepared and sizeable training data samples.
- Machine learning techniques can significantly reduce computing time in comparison to the global minimization technique.
- The Mathematica notebook containing examples of computations will be published soon.