Analysis of Stellar Parameter Uncertainty Estimates from Bootstrapping Neural Networks

P.G. Willemsen[1], **C.A.L. Bailer-Jones**[2], **T.A. Kaempf**[1]

(1) Sternwarte der Universität Bonn, Auf dem Hügel 71, 53121 Bonn, Germany
(2) Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany

Report no.: ICAP-PW-004
2004.09.08

Abstract: The derivation of stellar parameters by automated methods is only meaningful if some measure of confidence for the predicted value can be stated. In this work we introduce one possible method for obtaining standard errors and confidence intervals for stellar parameters $T_{\rm eff}$, $\log g$, [Fe/H] and extinction A_V as predicted by neural networks (NN). We applied the bootstrapping method to a feedforward NN for Blind Testing Cycle 2 medium band 1X and 2F photometry for end of mission magnitudes G=15 and 19 mag. We further tested whether the parametrization results can be improved if multiple noisy versions of a filter flux vector (for a given astrophysical parameter) are used in the training set. The obtained results show that the bootstrap standard errors only change significantly if the overall signal to noise ratio is high.

1 Introduction

A feed-forward neural network as used in these and earlier tests (see e.g. Willemsen et al. 2004, hereafter ICAP-PW-003) is a regression model that maps inputs into outputs. Because of this feature, the concepts of standard errors and confidence intervals can also be used for this kind of parametrizer. We describe how uncertainty measures for a parameter as predicted from a neural network can be found via the bootstrap method. Further information as well as comparisons between different techniques on confidence and prediction interval estimation for neural networks can be found e.g. in Heskes (1997), Leisch et al. (2000), Papadopoulos et al. (2000) and especially Dybowski & Roberts (2000) (see also Bishop (1995) for general discussions about neural network regression).

1.1 Neural networks performing regression

A regression as represented by a feed forward neural network model relates an input vector of observations $\mathbf{x}$ (with components x_i which here are the filters in the GAIA 1X MBP system, i $\in$ [1:11]) to one or several output values y (here stellar parameters $T_{\rm eff}$, $\log g$, and [Fe/H], plus extinction A_V). During training, the weights of the network are updated according to some measure of error between the network outputs y and the target outputs t which is (for regression) most often the sum-of-squares error

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K}[y_k(\mathbf{x}^n;\mathbf{w}) - t_k^n]^2 \tag{1}$$

The first sum runs over the N training patterns and the second is for the general case of K outputs. To make things easier, we will only consider one single output (k=1).

If the number of training patterns goes to infinity, the sum over the patterns can be replaced by an integral, yielding

$$E = \frac{1}{2}\int\int[y(\mathbf{x};\mathbf{w}) - t]^2 p(t,\mathbf{x})\ dt\ d\mathbf{x} \tag{2}$$

where $p(t,\mathbf{x})$ can be decomposed into the product $p(t|\mathbf{x})\ p(\mathbf{x})$, i.e. the (conditional) target data probability density of t given a particular $\mathbf{x}$ and the unconditional density of $\mathbf{x}$.

By defining the conditional average of the target data by $\langle t|\mathbf{x}\rangle \equiv \int t\ p(t|\mathbf{x})\ dt$ and decomposing the term $[y-t]^2$ in expression 1 into $[y - \langle t|\mathbf{x}\rangle + \langle t|\mathbf{x}\rangle - t]^2$, one obtains

$$E = \frac{1}{2}\int[y(\mathbf{x};\mathbf{w}) - \langle t|\mathbf{x}\rangle]^2\ p(\mathbf{x})\ d\mathbf{x} + \frac{1}{2}\int[\langle t^2|\mathbf{x}\rangle - \langle t|\mathbf{x}\rangle^2]\ p(\mathbf{x})\ d\mathbf{x} \tag{3}$$

The second term is independent on the network weights and can be neglected for this discussion. The first term however is important as it tells us that the minimum of the error is obtained if

$$y(\mathbf{x};\hat{\mathbf{w}}) = \langle t|\mathbf{x}\rangle \tag{4}$$

where $\hat{\mathbf{w}}$ is the estimated weight vector at the minimum of the error function. Equation 4 shows that, if N goes to infinity (and if the number of weights is sufficiently large to ensure unlimited flexibility), the network mapping $y(\mathbf{x};\mathbf{w})$ is given by the regression of t conditioned on $\mathbf{x}$. In real world applications, the training set is naturally limited so that a network trained on a dataset $(\mathbf{x}_n; t_n)$ and minimizing equation 1 will approximate the mean value for t conditioned on $\mathbf{x}$, i.e. the equality in 4 is relaxed to $y(\mathbf{x};\hat{\mathbf{w}}) \approx \langle t|\mathbf{x}\rangle$. It is thus evident that such a network can be regarded as a regression function. Note that the components relating the target to the input can be distuinguished into a *stochastic* term, which describes the random fluctuation of t about its mean and a *deterministic* component which is described by the functional relationship (or mapping) $y(\mathbf{x};\mathbf{w})$.

1.2 Uncertainties in training neural networks

There are basically two sources of uncertainty related with neural networks. These are (1) uncertainties in the training data and (2) limitations of the model. The training data inaccuracies result from the fact that the set is typically noisy and incomplete. The random sampling of templates from the associated population as well as from the random fluctuation of t about the average $\langle t|\mathbf{x}\rangle$ introduces uncertainties. Thus, the aquisition of a training set is already prone to sampling variation. Since each training set can yield very different sets of network weights $\hat{\mathbf{w}}$, there is a distribution of (estimated) network mappings $\hat{y}(\mathbf{x};\hat{\mathbf{w}})$ for given inputs $\mathbf{x}$.

The model limitiations arise from local minima of the error function as well as from a suboptimal training algorithm both of which can again result in a number of possible estimated weight vectors $\hat{\mathbf{w}}$. In principle, the choice of the model such as the overall topology (number of hidden layers/nodes) and the shape of the transfer functions contribute to the uncertainty in $\hat{\mathbf{w}}$. In what follows, it is assumed that the chosen network topology is optimal.

1.3 Bootstrapping

The bootstrap was introduced by Efron (1979) for estimating various sample properties such as bias, variance and confidence intervals for any population parameter estimate. Since then, this method was successfully applied in many areas of statistics, including linear (Freedman 1981) and nonlinear nonparametric regression (Härdle & Bowman 1988). We will give an introduction to this technique with special emphasis on its application to neural networks performing parametrization. For a more detailed discussion see Efron & Tibshirani (1993).

Given a (random) training sample $\mathbf{S}$ by pairs of inputs $\mathbf{x}$ and corresponding outputs y, i.e. $\mathbf{S} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)$, taken from a population F, we want to estimate the ensemble of model parameters of interest (network weights) $\theta = f(F)$, which could be done by calculating $\hat{\theta} = g(\mathbf{S})$ based on $\mathbf{S}$. The bootstrap is a data based method for statistical inference which allows us to determine the error of the network outputs as given from different values of $\hat{\theta}$. To obtain the bootstrap standard error, one needs to build bootstrap samples. A bootstrap sample $\mathbf{S}^*$ is a random sample of the same size N as the original sample which is created by randomly resampling $\mathbf{S}$ with replacement (**i.e.** $\mathbf{S}^* \subseteq \mathbf{S} \subset F$). In this way, one obtains B bootstrap samples the bth bootstrap sample given by $(\mathbf{x}_1^{*b}, y_1^{*b}), (\mathbf{x}_2^{*b}, y_2^{*b}), ..., (\mathbf{x}_N^{*b}, y_N^{*b})$. For each sample, we minimize $\sum_{n=1}^{N}[y_n^{*b} - y(\theta; \mathbf{x}_n^*)]^2$ (i.e. we derive B regression functions by training a network on each bootstrapping sample) yielding $\hat{\theta}^{*b}$. The (nonparametric) estimate of the bootstrap standard error (BSE; i.e. the variance of the distribution) for the nth predicted value is then found from

$$\mathrm{BSE}(\mathbf{x}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}[y(\mathbf{x}_n; \hat{\theta}^{*b}) - y_{\mathrm{boot}}(\mathbf{x}_n; \cdot)]^2} \tag{5}$$

with $y_{\mathrm{boot}}(\mathbf{x}_n; \cdot)$ being the bootstrap committee's regression given by $\frac{1}{B}\sum_{b=1}^{B} y(\mathbf{x}_n; \hat{\theta}^{*b})$.

It is also possible to calculate a confidence interval (and, with some more effort, even a prediction interval, see e.g. Dybowski & Roberts 2000) from this booststrap standard error via

$$y_{boot}(\mathbf{x}_n; \cdot) \pm t_{\mathrm{confidence}}\ \mathrm{BSE}(\mathbf{x}) \tag{6}$$

where $t_{\mathrm{confidence}}$ is a factor to be taken from the Student's t-distribution corresponding to the desired level of confidence (here 95 %) with the number of degrees of freedom equal to the number of bootstrap samples B (see Heskes 1997).

The architecture of the networks is the same for each bootstrap sample but each network trained on a specific bootstrap sample starts at a different position in weight space. This approach accounts for model uncertainty caused by local minima in the error surface and is thus a better measure of standard error than e.g. the Delta method or the Sandwich estimator (see Tibshirani 1995). The number of bootstrap samples is ordinarily in the range from 25 to

200 (see Efron & Tibshirani 1993 for a detailed discussion and examples where this does not hold). Note that the 'error on the error' is proportional to $\frac{1}{\sqrt{(B-1)}}$.

The presented bootstrap strategy is also called *Bootstrap pairs sampling* and should not be confused with *Bootstrap residual sampling.* For the latter, the model residuals $(y_i - \hat{y}_i)$ are taken as the sampling units. As outlined in Dybowski & Roberts (2000), residual sampling uses the assumption that the residuals are independent of the inputs which need not be the case. Due to this and other reasons (see especially Tibshirani 1995), the more robust pair sampling procedure was chosen for these tests.

Above, it is assumed that the ensemble of networks yields unbiased estimates of the (true) regression function. As discussed in Heskes (1997), this is not really the case (neural networks are biased estimators) since models trained on a limited number of patterns will always tend to oversmooth sharp peaks in the data. We here follow Heskes (1997) and assume that the bias component of the errors is small as compared to the variance component (see Sect. 3).

2 The data set and bootstrap simulations

The data used were those of Blind Testing Cycle 2 for G=15 and 19 mag. For completeness, a graphical representation of the astrophysical parameter (AP) combinations in the training and validation sets is shown in Fig. 1 (but see also Brown 2003).

A training set at a given magnitude has 20000 different APs with the corresponding 11 filter fluxes. There are 20 different noise versions (noise in the filter fluxes) of the training set. We performed several tests with different numbers of noisy training templates for a given astrophysical parameter combination (see Sect. 4). This procedure is motivated by the fact that additional noise can help to regularize the network, e.g. prevent overfitting (see e.g. Bishop 1995). The random resampling was done with a random number generator as given in Press et al. (1992).

For the general case (**case1**), we chose 1 of these 20 noise versions with 20000 inputs of filter fluxes (plus corresponding stellar parameters) and randomly sampled 80 bootstrap sets. For such a large number of patterns, 80 bootstrap samples may seem rather small. To allow for a sufficiently large amount of resampling (by which we mean that a pattern does not appear in the training set), we therefore ensured that 1000 patterns of the original 20000 were indeed missing in each bootstrap sample (*on average*, each training pattern is missing four times for the 80 bootstrap samples). The tests for this case were done for the 1X and 2F photometric system.

Additionally, we calculated the bootstrap errors from the same networks' outputs, but only for 20 bootstrap replications (instead of 80). This is referred to as **case1b**.

For the second case (**case2**), we did the same as above but, in case that an AP combination was randomly chosen into the bootstrap sample, we chose 5 noisy (filter flux) versions for this specific AP, i.e. each bootstrap sample is made up of 20000 × 5 (number of noise versions for given AP) training templates. In total, we created 20 bootstrap samples in this way for both magnitudes. Only the 1X system was considered for this case. Note that the random resampling was done over the APs and not over the noise versions!

For the third case (**case3**), we did the same as above, but chose in total 15 noise versions for each randomly resampled AP, i.e. each bootstrap sample is made up of 20000 × 15 inputs (number of noise versions for given AP). As before, we created 20 bootstrap samples in this way for both magnitudes but only for the 1X system.

The choice of the number of bootstrap replications is here mainly motivated by the limited amount of available computer time. Especially for the last case with 300000 templates per training set we had to choose a smaller number of replications, given that the training for one of these network takes very long.

For the validation, we chose only one pattern out of the 20 noisy versions available. For each bootstrap sample a network with architecture 11:13:13:4 was trained (11 filter flux inputs, two hidden layers each having 13 neurons and 4 outputs, one for each parameter). In addition, we trained networks with the same architecture but on the whole training set of 20000 AP combinations (i.e. without resampling). For case1, a committee of 5 networks was trained, while for the two other cases only single networks were set up. The code used was that of Bailer-Jones (2000).

Note that these networks are different from those used in Blind Testing Cycle 2 (results of Kaempf & Willemsen) which were specialized networks trained on specific ranges in parameter space and which had a larger number of hidden neurons.

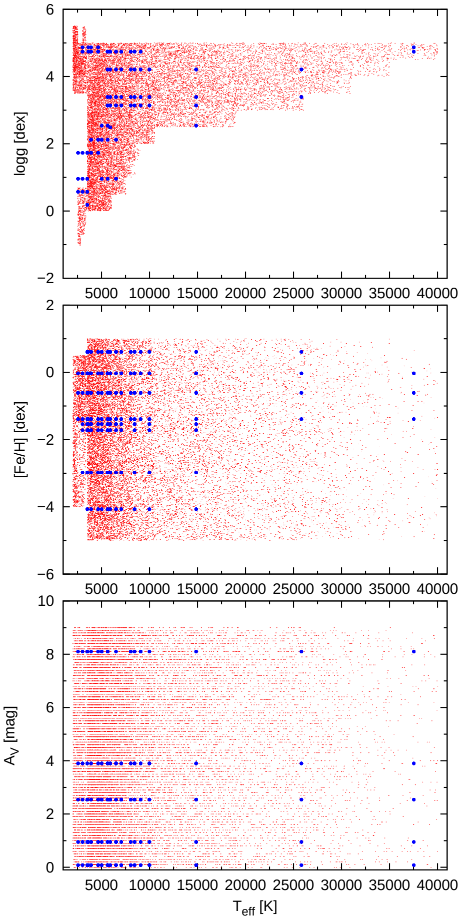

Figure 1: The astrophysical parameter grids of the training (small dots) and validation set (large dots) as used in the second cycle of blind testing and in this work.

3 Results and discussion

In the following the major results of the bootstrap standard error distributions and their correlations for different stellar parameters are highlighted.

3.1 Bootstrap standard errors for individual stars

Figs. 2 and 3 show *examples* of bootstrap standard error bars and confidence intervals for selected stars in the validation set. These results are for bootstrap samples with one filter flux noise version for a given AP (case1), i.e. 20000 inputs.

In general, it appears that the bootstrap standard errors (BSE) are larger for higher temperature objects. Note however, that the fractional error is about the same (~ 6 % at $G = 15$ mag) for all temperatures and that the systematic deviations appear to be stronger for hotter objects due to the chosen scale. The points for the 2F photometric system are not much different from those of the 1X system.

For gravity we see a systematic trend for the 1X system which is not found for the 2F photometry. Especially for the 1X case, we see that the precision of the networks is high (the bootstrap standard errors are small) while the accuracy is rather low (the computed values are systematically too large or too small).

In going to lower S/N, we observe an overall systematic trend in both filter systems where low gravities are strongly overestimated while at the same time, the BSE are smaller. This shows that a higher noise level in the input data does not necessarily mean a higher variance of the estimated output parameters.

There is something peculiar when looking at the errors for the different parameters. For those parameters which mostly affect the continuum of a stellar energy distribution ($T_{\rm eff}$ and A_V), we see that a lower S/N (going from $G = 15$ mag to $G = 19$ mag) results in equal or larger bootstrap standard errors. However, for parameters which are (to first order) only acting on the lines ([Fe/H] and $\log g$) the BSEs become significantly smaller for smaller S/N. This can also be seen from the distributions of the bootstrap standard errors as shown in Figs. 4 and 5 (both case1), or Figs. 6 and 7.

An explanation for these trends might be given by the fact that [Fe/H] and $\log g$ are much more affected by noise than those parameters which can be read from the continuum (we here primarily refer to $T_{\rm eff}$ since the above trends are not so significant for A_V). Moreover, given that $T_{\rm eff}$ acts on the whole stellar energy distribution, combinations of all or many of the 11 filters are used for the determination of this parameter, while for the line-based parameters only certain filters (centered on specific lines) are of relevance. An increase of the noise (in going from $G = 15$ to 19 mag) heavily deteriorates the signal (the information content) of the line sensitive parameters, thus making the filter flux combinations look almost equal (equal due to the noise) for different values of in these parameters. The networks trained on the different bootstrap samples cannot discriminate well the APs, therefore always ending up on almost the same values (the BSEs, which measure the variance about the bootstrap committee's mean value become small). The temperature information however can still be found in the continuum, thus allowing for a determination of this parameter. Increasing the overall noise in the input data therefore yields only a higher variance (BSE) for this parameter.

	G=15 mag			G=19 mag		
# noise templates	1	5	15	1	5	15
$\Delta(T_{\rm eff})$ [K]	−20	−18	10	228	235	247
$\Delta(\log g)$ [dex]	−0.02	0.01	0.02	−0.03	−0.02	−0.02
Δ([Fe/H]) [dex]	−0.31	−0.28	−0.32	−0.71	−0.69	−0.71
$\Delta(A_V)$ [mag]	0.04	0.05	0.04	0.18	0.18	0.15

Table 1: This table shows the changes of the systematic errors (for the 1X system) for different magnitudes and different numbers of noisy filter flux vectors for a given AP in the training set (1, 5 and 15). $\Delta(X)$ is the median of the difference between the bootstrap mean value for a given parameter and the true value, i.e. Δ(X) = median(mean$_{boot}$ - true(X)).

4 Bootstrap errors for multiple noisy filter flux versions in the training set

A network which is too complex tends to overfit the training data, giving bad generalization performance. As mentioned above, we therefore tested whether the regularization of the network can be improved if there are several noisy versions of a filter flux vector (for a given AP) in the training set.

Figs. 6 and 7 show the results for indiviudal stars for G = 15 and 19 mag, respectively. The bootstrap error distributions for the three cases are shown in Figs. 8 and 9 for the two magnitudes. In Table 1 we further state some measure for the systematic errors for the different cases.

For G= 15 mag, i.e. a high a S/N, we see that only the absolute systematic error of $T_{\rm eff}$ decreases while the others remain almost the same when there are several noisy flux vectors for a given AP in the training set. At G = 19 mag however, the absolute systematic error for $T_{\rm eff}$ increases while for the other parameters the systematics become smaller (for A_V and $\log g$).

Concerning the bootstrap errors we note from Figs. 8 and 9 that for G= 15 mag all BSEs increase by more than ∼ 10 % when there are several noisy flux vectors for a given AP in the training set. At G= 19 mag however, only the BSE of A_V increase significantly while the bootstrap errors for the other parameters remain the same or get slightly smaller.

This shows that multiple noisy templates in the training set only change the results for those cases where the overall S/N is rather high. This is sensible since for very low S/N the parametrization performance is always poor (due to less significant information in the training set and probably not because of model imperfections). Additional noise versions of a filter flux vector do therefore not help in the parametrization.

4.1 Bootstrap standard error correlations

Figs. 10 and 11 show the dependencies of the standard bootstrap errors for different parameters (only case1), while Figs. 12 and 13 show the dependencies of the overall parametrization errors (given as computed − true) versus the bootstrap errors (only case1, 2F system). To have some quantitative (albeit somewhat arbitrary) measure for the error's dependencies, we also state the correlation coefficients for each parameter pair. However, these numbers should not be overinterpreted since even a scatter for a small fraction of the points will naturally give other values. Moreover, it should be remembered that the plots reflect the underlying grid of validation stars (e.g. not all parameter combinations are represented).

For analysing such dependencies one has to consider how uncertainties in a regression are also caused by the input pattern in terms of physical stellar characteristics. For example, a hot star will almost always yield a high standard error for metallicity, independent of any sample distribution, initial weight setting etc., but simply due to the fact that there are almost no metal sensitive features in such spectra. As a result, the regression in this part of the parameter space is supposed to be almost random, yielding a high variance of the regression functions estimated by the neural network.

From Figs. 10 and 11 it can be seen that the parameters' estimated standard errors are correlated albeit with different degrees of strengths. That BSE(teff) is (weakly) correlated with BSE([Fe/H]) can be understood from the above said: hot stars, which have larger errors due to the smaller training grid density (and the overall similar spectral shape, expressed by the Rayleigh-Jeans approximation) at these temperatures do not show strong metal lines. The standard errors of $\log g$ seem to be only weakly correlated with the uncertainties in $T_{\rm eff}$. This is sensible since temperature information is drawn from the continuum (which can be well estimated in most cases) while $\log g$ is mostly a line sensitive feature (at least for certain temperatures) as is metallicity. This also explains why the errors of [Fe/H] and $\log g$ are strongly correlated.

Interestingly, we find that the errors of A_V and [Fe/H] or $\log g$ are strongly correlated while that of A_V and $T_{\rm eff}$ are not. Intuitively, one might have expected that large uncertainties in $T_{\rm eff}$ correspond to large errors in extinction A_V, given that both parameters act on the continuum in a similar way. An explanation is possibly given by the fact that extinction is mostly acting in the blue part of the spectrum (see extinction curves of e.g. Fitzpatrick 1999), while $T_{\rm eff}$ affects the whole spectral energy distribution so that a robust temperature estimation is possible even if extinction determination fails. Due to the CCDs sensitivity, the S/N is generally lower at blue wavelengths which is especially the case for low temperature (red and yellow) objects for which extinction cannot be derived easily. Given that for such objects $\log g$ is possibly derived from the (shallow) Balmer Jump or other features at blue wavelengths, we can understand that BSE($\log g$) $\sim$ BSE(A_V). In the same way, metallicity information is mostly available at bluer wavelengths (take the Stroemgren $m1$ index as an example which is commonly used to derive metallicities for red giants). Thus, low S/N at these wavelengths deteriorates both, A_V and [Fe/H].

A comparison of the distributions in Figs. 12 and 13 shows that the weak correlation between the overall parametrization errors and the bootstrap errors which can be seen at G = 15 mag almost totally levels off at $G = 19$ mag. It can also be seen that the BSEs of the line sensitive parameters ($\log g$ and [Fe/H]) generally become smaller for lower S/N, something which was discussed in Sect. 3.1.

4.2 Distributions of bootstrap replications

It is always useful and sometimes also necessary to look at the distribution of the individual bootstrap realizations. For example, outliers or heavily skewed distributions would rather call for some more robust measure of standard deviation than given in equation 5. In such a case one could consider to use a measure based on the quantiles of the distribution (note that this specific measure is biased, see e.g. Efron & Tibshirani 1993).

Fig. 14 shows the results for two stars with different temperature and gravity but equal metallicity and extinction (only case1). It can be seen that the distributions are rather well behaved, i.e. do not markedly look different from normal distributions. Note that for the gravity and extinction distributions of the second star (lower panels) the variance is rather large but more bootstrap replications would probably yield normal distributions.

5 Conclusions

The results show that the bootstrap method is applicable for the estimation of standard errors for stellar parameters as determined by neural networks. At this point, it must be mentioned that other methods for uncertainty estimation of predicted values exist. The most promising alternative is a Bayesian framework as suggested in e.g. Bishop & Qazaz (1997). Such an approach allows the noise variance itself to depend on the input variables, unlike the ususal assumption of a normal noise distribution with a constant variance (which can yield systematically underestimated noise variances). Bishop & Qazaz (1997) could show that this framework can significantly reduce such a bias. Future work on the estimation of error bars should therefore include a Bayesian approach.

From our results, we conclude that

- the bootstrap standard errors of the different parameters depend on each other, albeit to very different degrees of strength. The strongest dependencies are found for the bootstrap errors of [Fe/H], $\log g$ and A_V, which probably reflects the fact that these parameters are mostly derived from the blue part of the spectrum, i.e. a signal deterioration results in larger uncertainties for all three parameters.

- the bootstrap standard errors become smaller for overall smaller S/N most noticably for $\log g$ and [Fe/H]. This can probably be explained in that the networks cannot discriminate well between the filter flux vectors for different APs at overall low S/N, thus ending up at almost the same (wrong) value for each bootstrap replication.

- when using multiple noisy versions of a flux vector in the training set (at a given S/N) the regularization performance of neural networks as measured by the systematic errors can be improved but only for overall high S/N and only for $T_{\rm eff}$ while for the other parameters no relevant changes are seen. At lower overall S/N, small improvements were only observed for $\log g$ and A_V.

  Concerning the BSEs, a general increase was found for all parameters when there were several noisy templates in the training set but only for high S/N. For low S/N, the BSEs only seem to increase for A_V while for the other parameters they essentially remain the same.

References

Bailer-Jones, C. A. L. 2000, A&A, 357, 197

Bishop, C. 1995, Neural Networks for Pattern Recognition (Oxford University Press)

Bishop, C. M. & Qazaz, C. S. 1997, in Proceedings 1996, International Conference on Aritficial Neural Networks, ICANN'96, ed. von der Malsburg et al. (Springer)

Brown, A. 2003, Results of the second cycle of blind testing, Tech. rep., ICAP-AB-004

Dybowski, R. & Roberts, S. J. 2000, in Clinical Applications of Artificial Neural Networks, ed. R. Dybowski & V. Gant (Cambridge University Press.)

Efron, B. 1979, Ann. Statist., 7, 1

Efron, B. & Tibshirani, R. 1993, An Introduction to the Bootstrap (Chapman and Hall, New York)

Fitzpatrick, E. L. 1999, PASP, 111, 63

Freedman, D. A. 1981, Ann. Statis., 9, 1218

Härdle, W. & Bowman, A. 1988, J. Americ. Statist. Assoc., 83, 102

Heskes, T. 1997, in Advances in Neural Information Processing Systems, ed. M. Mozer, M. Jordan, & T. Petsche, Vol. 9

Leisch, F., Jain, L. C., & Hornik, K. 2000, in ETD2000, IEEE Computer Society Press, Los Alamitos, California, USA, ed. L. C. Jain

Papadopoulos, G., Edwards, P. J., & Murray, A. F. 2000, in ESANN'2000 proceedings - European Symposium on Artificial Neural Networks, Bruges

Press, W. H.and Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C (Cambridge University Press)

Tibshirani, R. 1995, Neural Computation, 8, 152

Willemsen, P., Kaempf, T., & Bailer-Jones, C. A. L. 2004, Identification and Parametrization of spectroscopic binaries by medium band photometry, Tech. rep., ICAP-PW-003

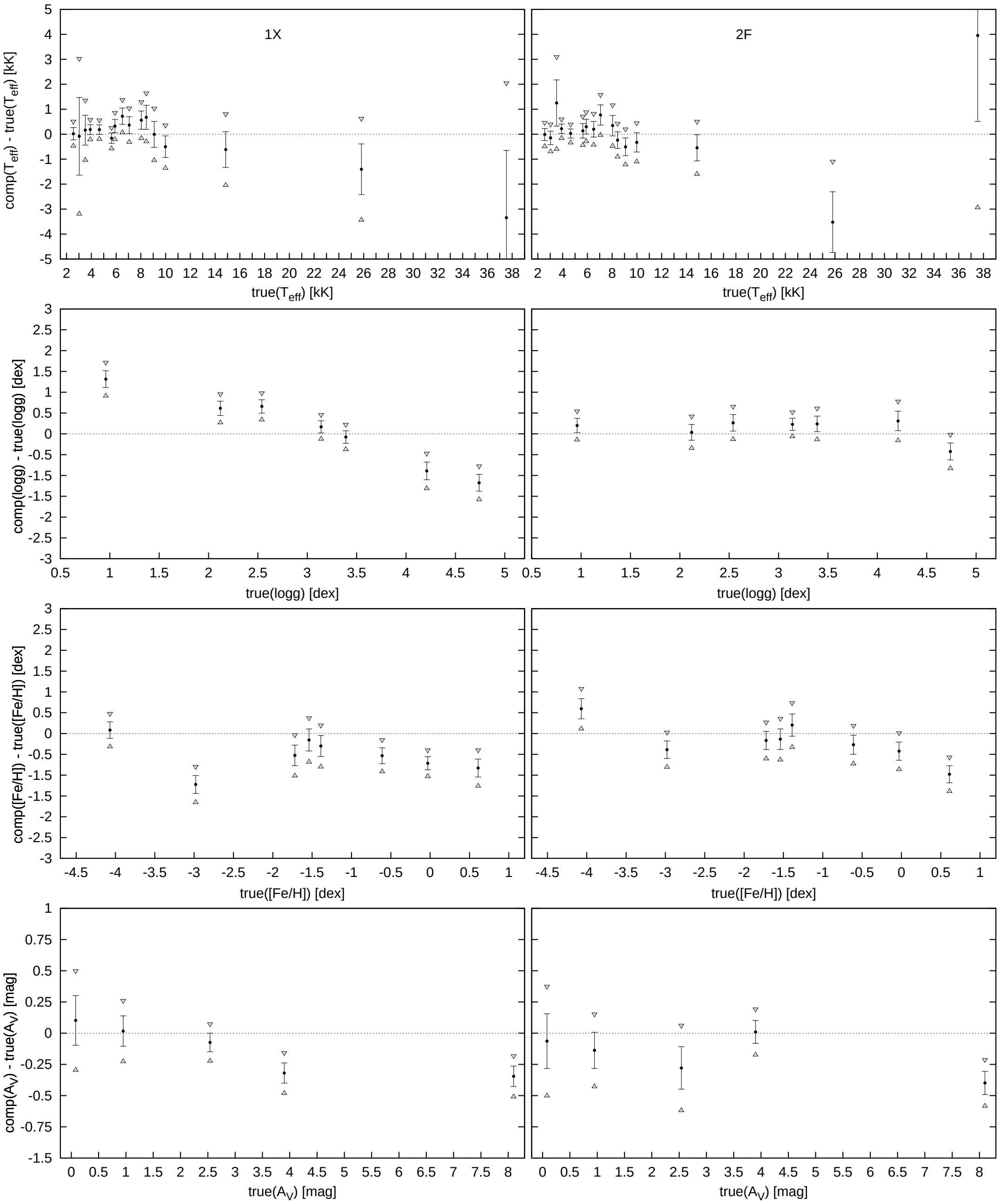

Figure 2: Shown is the difference between the network committee's mean value and the corresponding true stellar parameter for specific stars for case1 at G=15 mag for the 1X (left) and 2F system (right column). The error bars are the bootstrap standard errors and the triangles denote the limits of the corresponding 95 % confidence intervals. In the top panel, the results are shown for stars of different temperature (in units of kilo Kelvin), the other parameters fixed at A_V= 0.95 mag, [Fe/H]= -1.39 dex and $\log g\sim$ 4.5 dex for $T_{\rm eff}\geq$ 5000 K and $\log g$ = 1.73 dex for $T_{\rm eff}$ $\leq$ 5000 K. The second row is for different stellar surface gravities with the same metallicity and extinction and $T_{\rm eff}$= 5650 K. The metallicity results are for stars with $T_{\rm eff}$ = 5650 K and $\log g$ = 4.21 dex while those for extinction have $T_{\rm eff}$= 5650 K and $\log g$ = 2.54 dex.

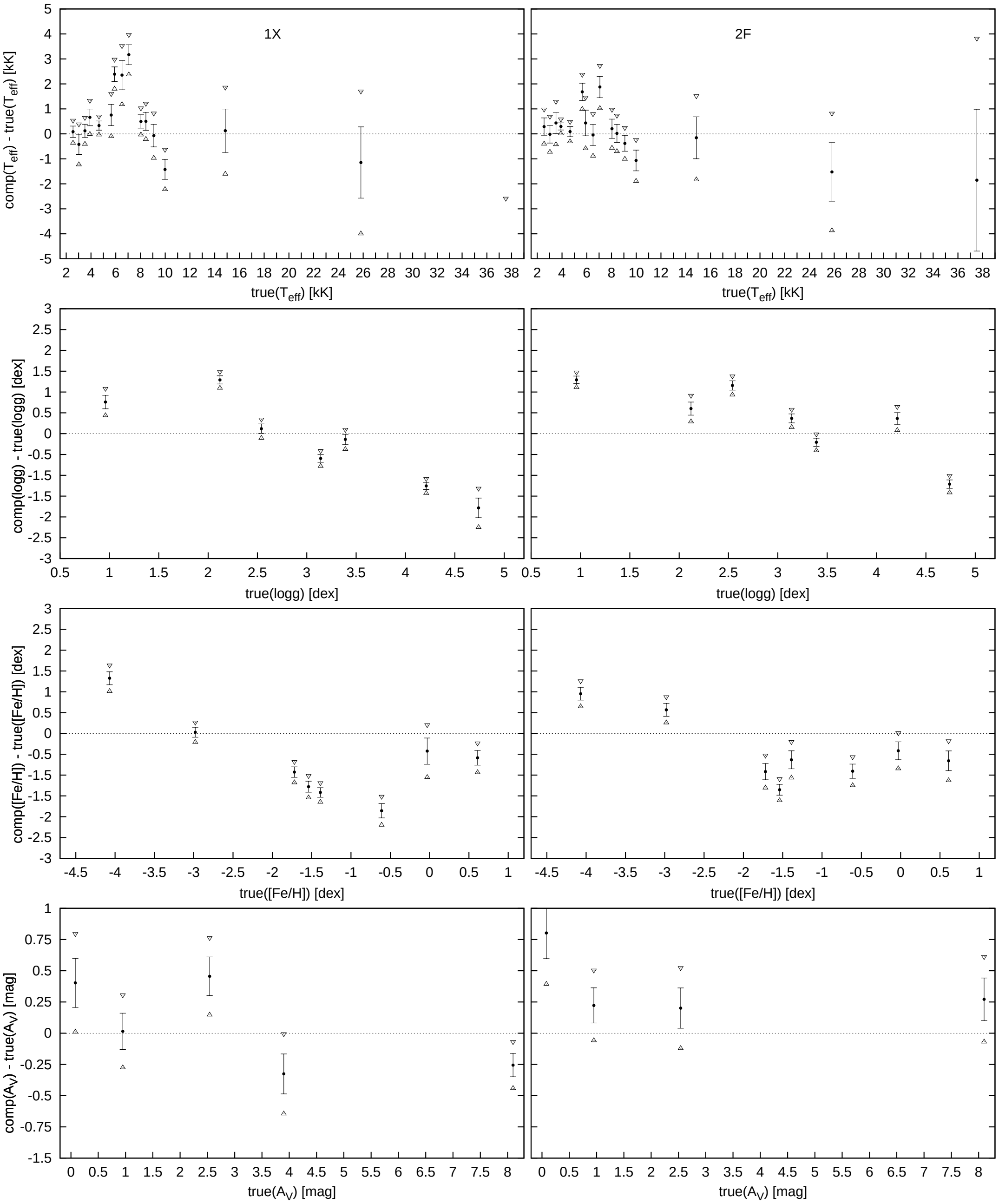

Figure 3: The same as in Fig. 2 but for G=19 mag. Note that the scale is the same as in Fig. 2 so that certain points fall outside the plotted range.

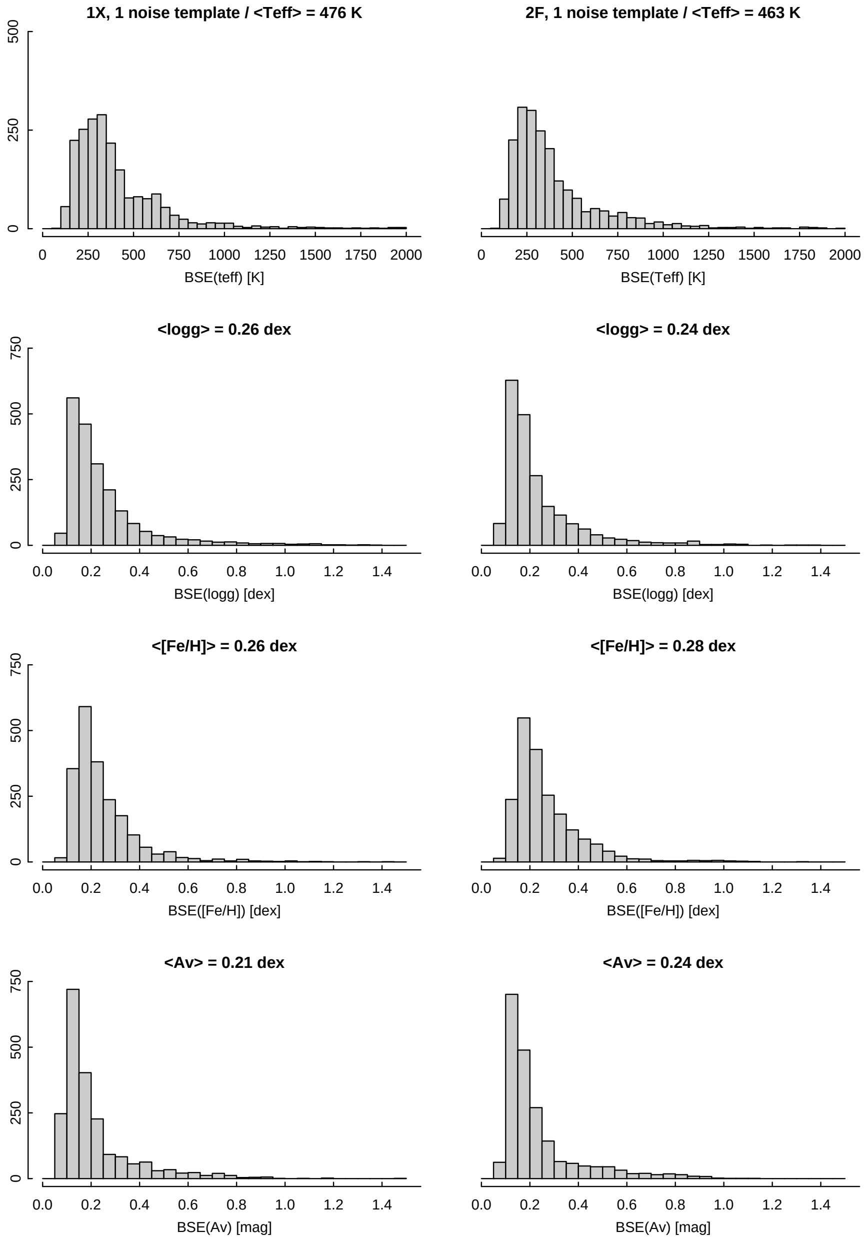

Figure 4: The distributions of the bootstrap standard errors BSE from top to bottom for the stellar parameters $T_{\rm eff}$, $\log g$, [Fe/H] and extinction A_V for G=15 mag and case1 (1 noise version of filter fluxes, 80 bootstrap replications). The left column is for 1X, the right for 2F photometry. The numbers in brackets show the mean values of the distributions.

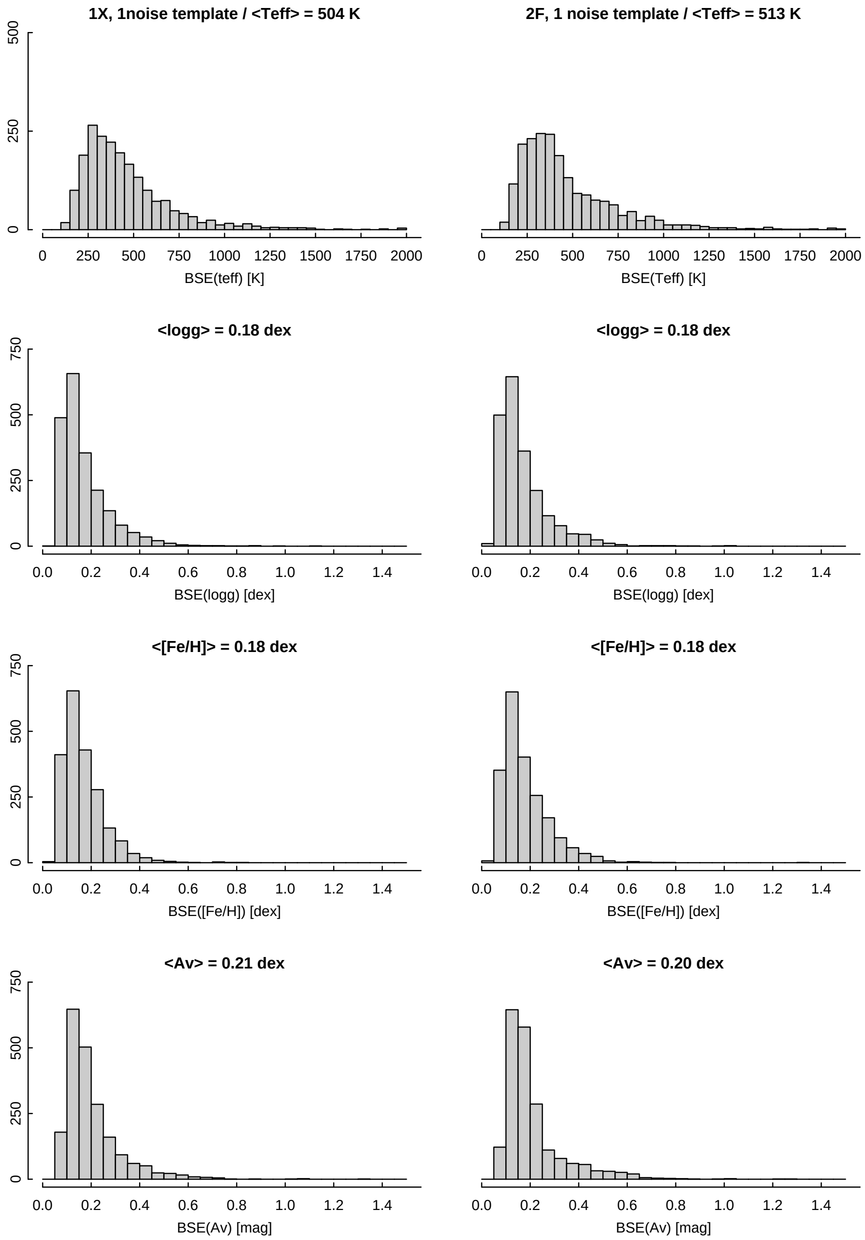

Figure 5: The same as in Fig. 4 but for G=19.

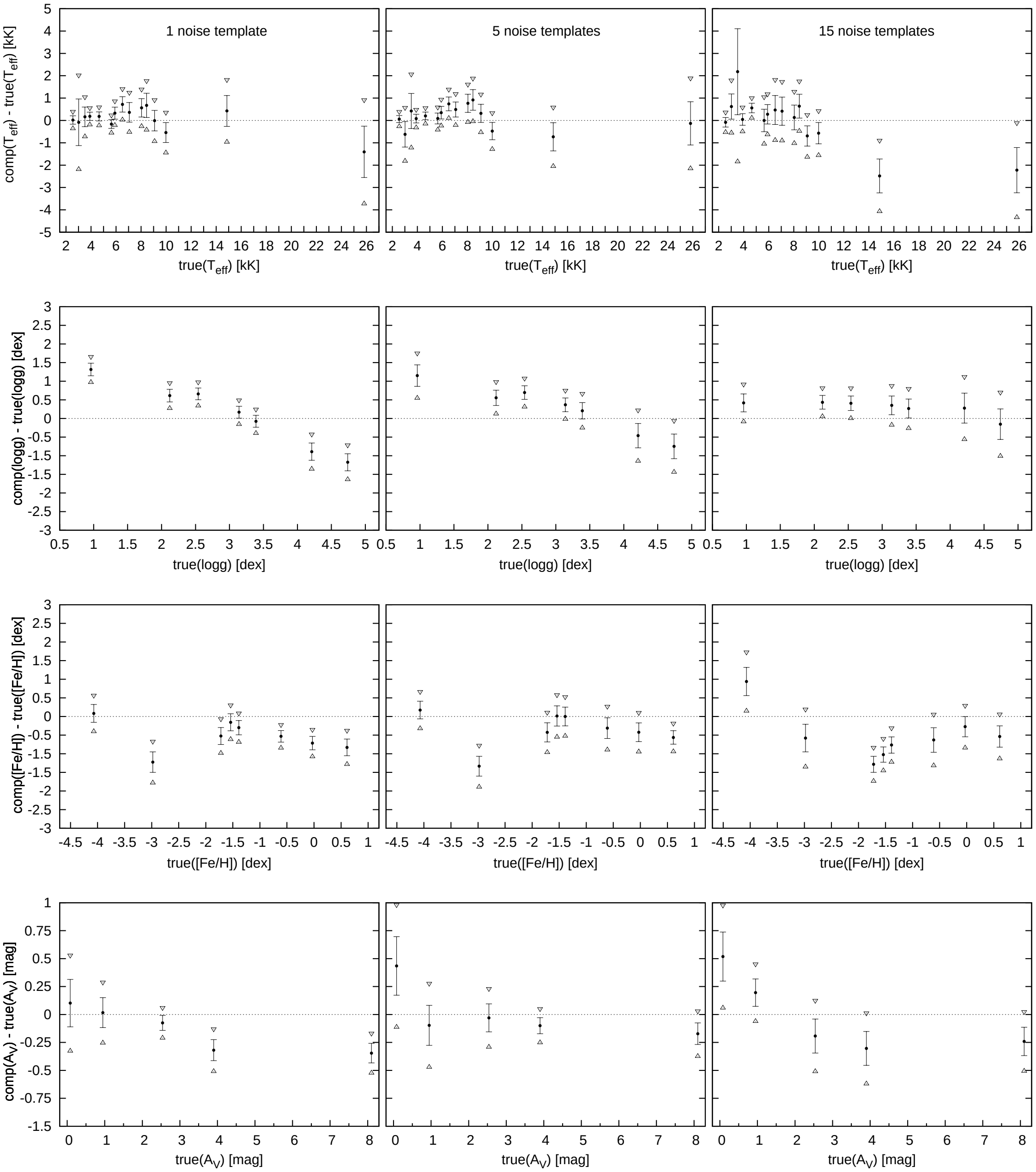

Figure 6: The same as in Fig. 2 ($G = 15$ mag, only 1X system) but for the different cases of multiple noisy flux vectors in the training set for a given AP (here refered to as noise templates). Left column for 1 noise template (case1b), middle for 5 (case2), right for 15 noise templates (case3) in the training set per AP.

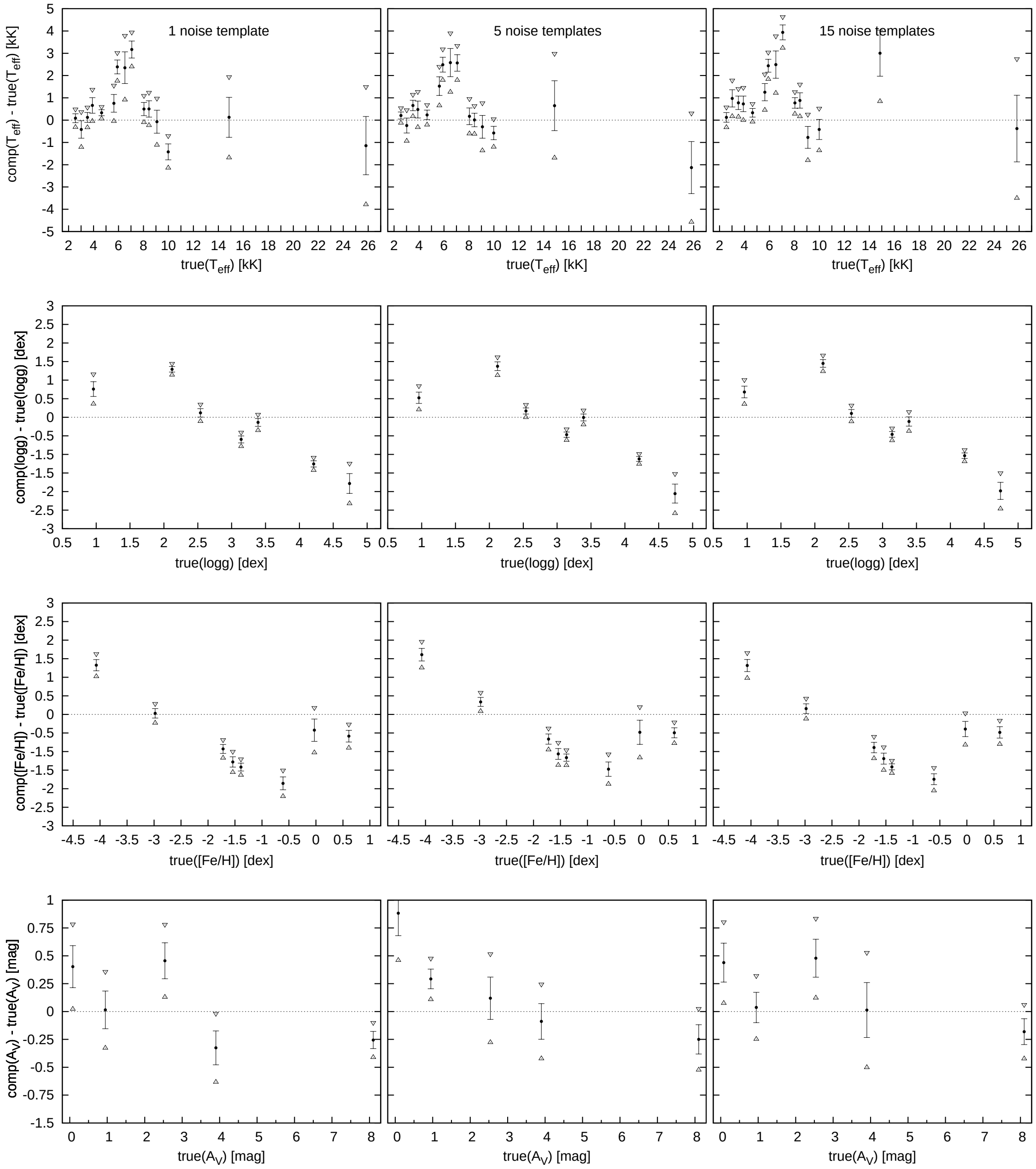

Figure 7: The same as in Fig. 6 but for $G = 19$ mag.

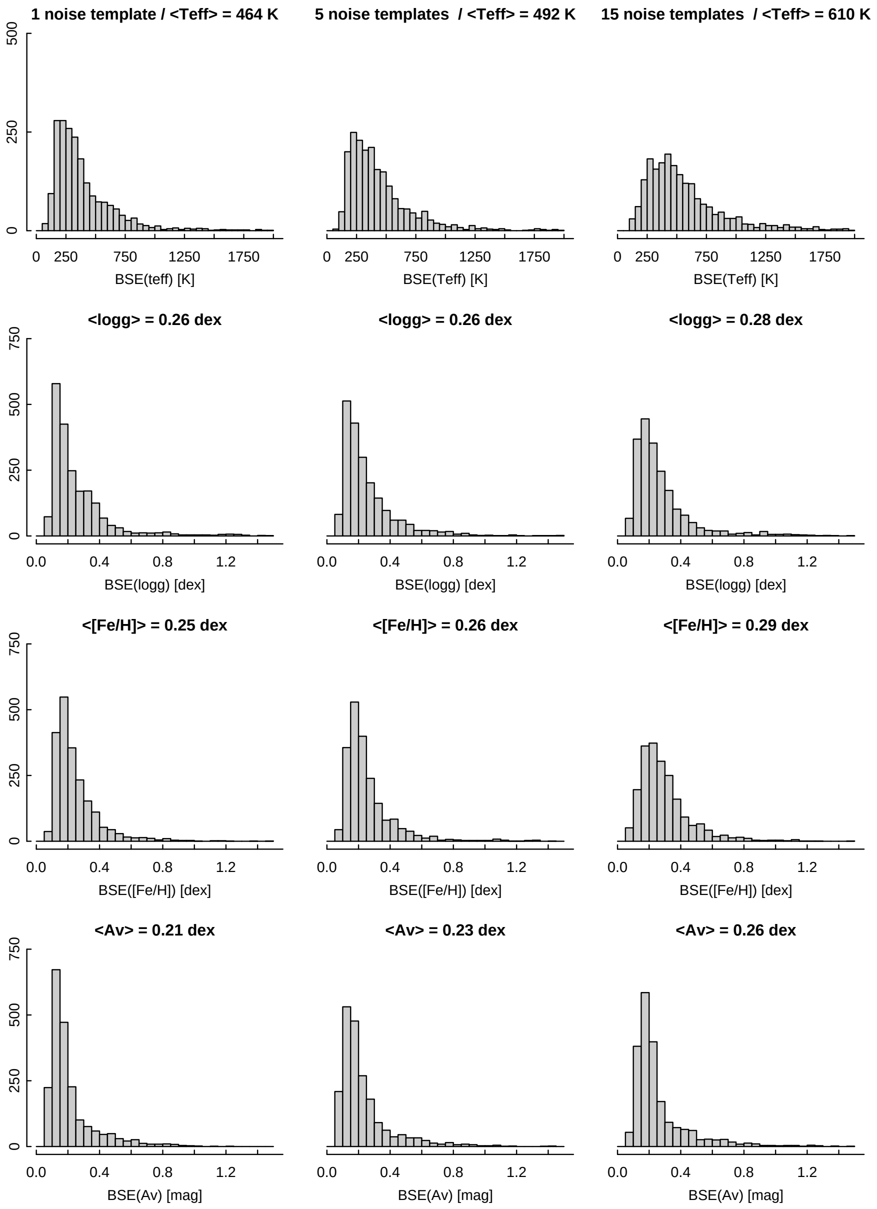

Figure 8: The distributions of the bootstrap standard errors BSE from top to bottom for the stellar parameters T_{eff}, $\log g$, [Fe/H] and extinction A_V for G=15 mag as in Fig. 4 but for the 1X system and for different numbers of noise versions (per AP) in the training set.

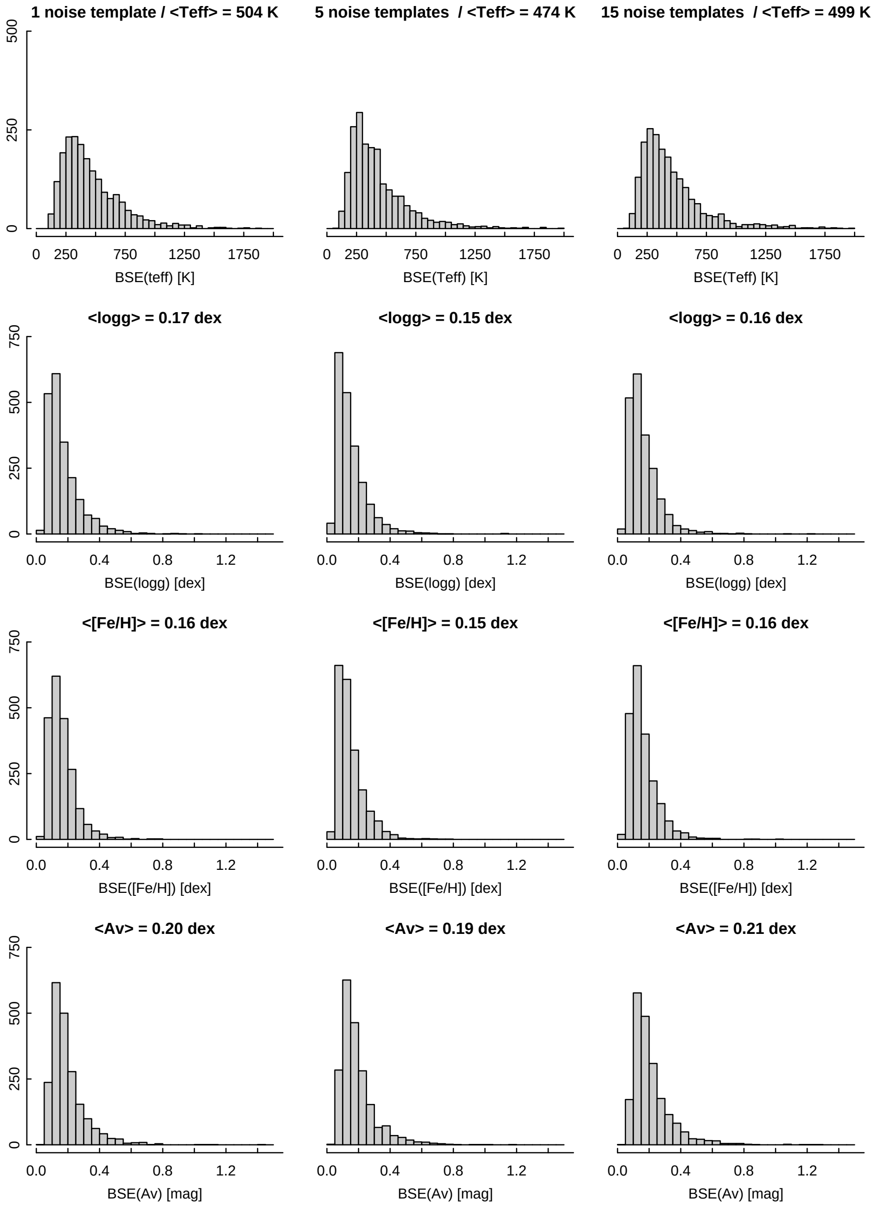

Figure 9: The same as in Fig. 8 but for G=19.

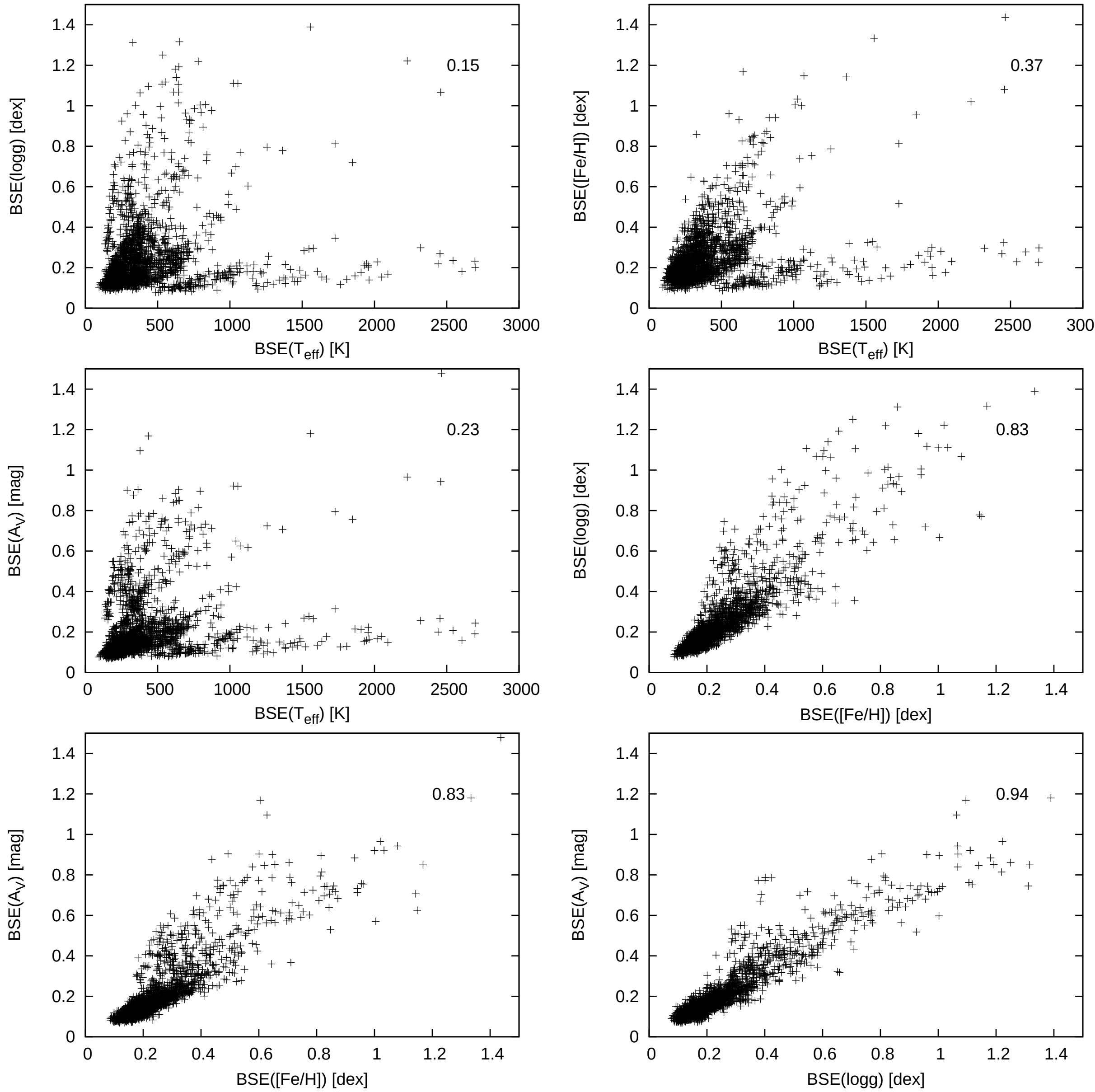

Figure 10: Shown are the correlations of the bootstrap standard errors (BSE) for the different parameters $T_{\rm eff}$, $\log g$, [Fe/H] and extinction A_V, for 1X photometry (G=15 mag, case1). The numbers in the upper right part of each plot are the correlation coefficients as calculated for the data in the plotted ranges.

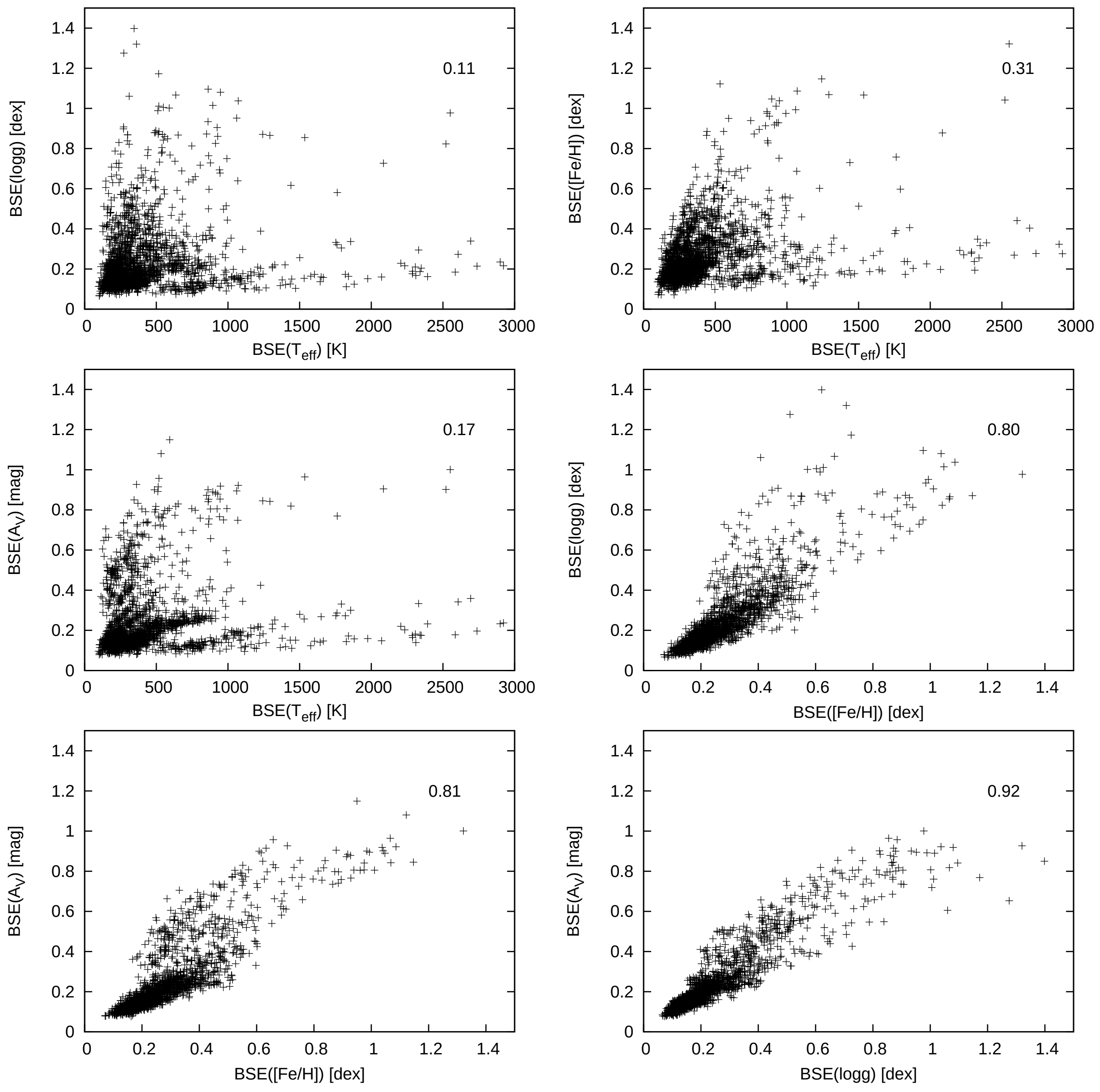

Figure 11: The same as in Fig. 10 but for 2F (G=15 mag) photometry.

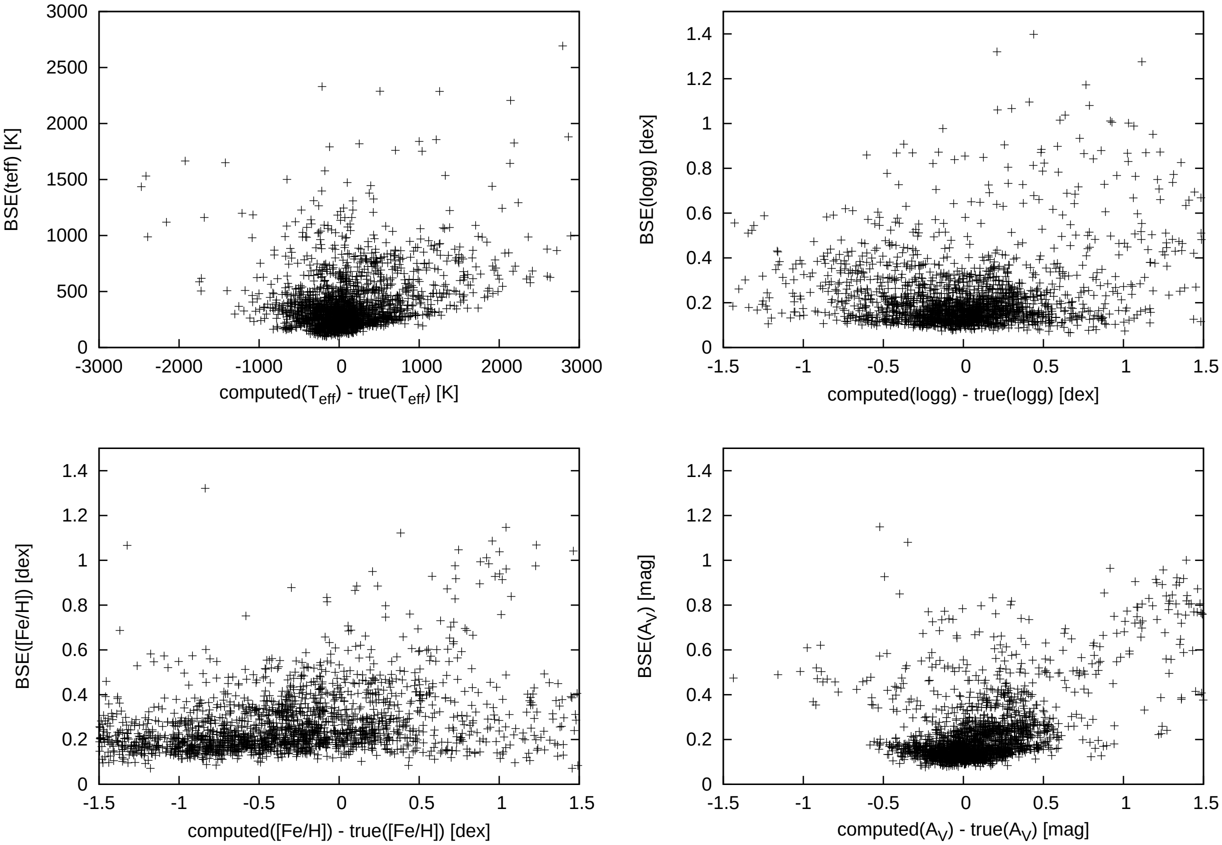

Figure 12: The parametrization error given as computed $-$ true versus the BSE for the astrophysical parameters. This plot is for $G = 15$ mag (case1) in the 2F system.

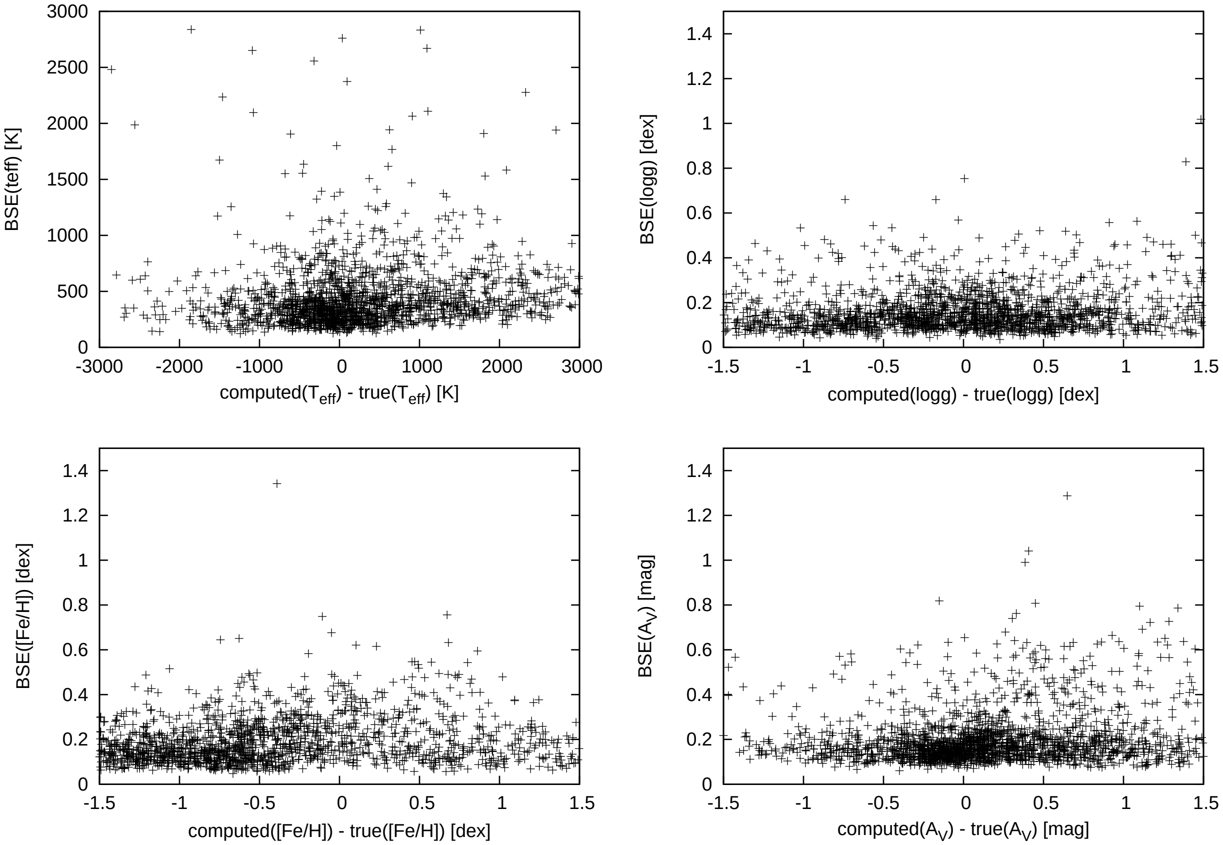

Figure 13: The same as in Fig. 12 but for $G = 19$ mag.

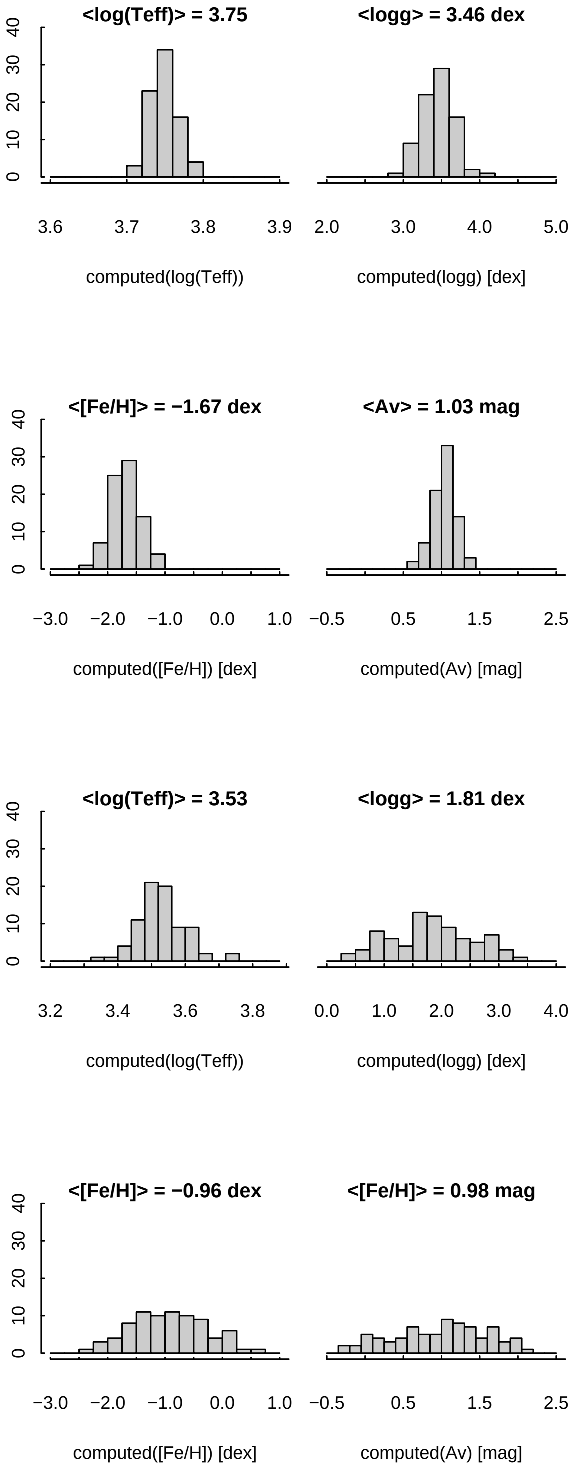

Figure 14: The distributions of 80 bootstrap neural network replications (case1, 1X) for two stars at G=15, with $\log g$ = 4.21 dex and $\log(T_{\rm eff})$ = 3.75 (top four panels) and $\log g$ = 1.73 dex , $\log(T_{\rm eff})$ = 3.55 (lower panels). A_V and [Fe/H] were fixed to 0.95 mag and −1.39 dex.