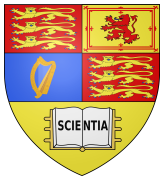


Imperial College London

Masters Project

Computational search for high redshift quasars

Using artificial neural networks to find the most distant objects in the Universe

Author:
Edoardo Vittori
CID: 00729563

Supervisor:
Dr. Daniel Mortlock

June 7th, 2016

Abstract

High redshift quasars (HZQs) with $z \gtrsim 6$ are extremely rare; they are the farthest objects which can be detected by current instruments. The high redshift confines them to the first billion years from the Big Bang and the study of their emission can reveal secrets about the Universe at a very early age. However, advanced computational techniques are required to process the large amounts of modern survey data used for the identification of HZQs.
The purpose of this work is to use machine learning, and specifically artificial neural networks (NNs), for the search of such rare HZQs.
A toy model was initially set up to compare two neural network classification tools (Matlab and Skynet) with each other, and against the Bayesian probabilistic method. Following the preliminary assessment, Matlab was chosen; its customisation required several thousand lines of software code.
The neural network specialisation and optimisation was conducted with the use of two data sets: one deriving from UKIRT Infrared Deep Sky Survey (UKIDSS) cross-correlated with the Sloan Digital Sky Survey (SDSS); the other consisting of simulated HZQs with $z \gtrsim 6$.
Every object in the dataset was identified by eight attributes: four measurements coming from different filters, each with its own error. Having as first priority the avoidance of false negatives - to not lose any HZQs - innovative computational methods were designed: by averaging or adding the results of independent NN classification cycles; by complementing the classification of each object with the probability of being an HZQ; by mapping each object from a point-like entity into a Gaussian distribution around the original value. The stability of the classification results was further enhanced by assigning to the data points of the training set a weight inversely proportional to their error. These methods proved to be capable of identifying correctly all of the target HZQs (completeness of 100%), with a of 95 to 99% classification rate (ratio between selected objects and initial population).
Neural Networks offer a stable, robust, and flexible tool in the search for the rare HZQs. They represent an ideal platform to quickly and effectively analyse big quantities of data, and will be fundamental in the next generation astronomical surveys.

Acknowledgements

I would like to express to my deepest gratitude to my supervisor Dr. Daniel Mortlock for his support, expert guidance and encouragement throughout my research.

The work presented is my own, unless explicitly stated otherwise.

Edoardo Vittori

Dedication

I dedicate my dissertation work to my family and friends. A special feeling of gratitude to my loving parents whose support has been invaluable and my brothers who have never left my side. I give special thanks to my friend Andrea Bolle, I will always appreciate all he has done.

You can't connect the dots by looking forward; you can only connect them by looking backwards.

Steve Jobs

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AV Average method

DR8 Data Release 8 (SDSS)

EP Empirical Probability method

ESA European Space Agency

FN False Negative

FP False Positive

HZQ High Redshift Quasar ($z \gtrsim 6$)

IC InClusion method

LAS Large Area Survey

NASA National Aeronautics and Space Administration

NNs Artificial Neural Networks

ROC Receiver Operating Characteristic

SDSS Sloan Digital Sky Survey

SMBHs Super Massive Black Holes

TN True Negative

TP True Positive

2D Two Dimensional mapping

UKIDSS United Kingdom Infrared Deep Sky Survey

UKIRT United Kingdom InfraRed Telescope

ULAS Union List of Astronomy Serials

z Redshift

1 Introduction

A quasar, or quasi-stellar radio source, is a distant object powered by a black hole a billion times larger than our sun (see (Rees 1984)). [1]. Quasars were first discovered in 1963 with 3C273 in the constellation Virgo (Schmidt 1963). Since then, they have played a key role in studying black holes, galaxy evolution, reionization, and cosmology (Peacock 1998). In particular, high redshift quasars with $z \gtrsim 6$ are considered probes capable of revealing the secrets of the early Universe.

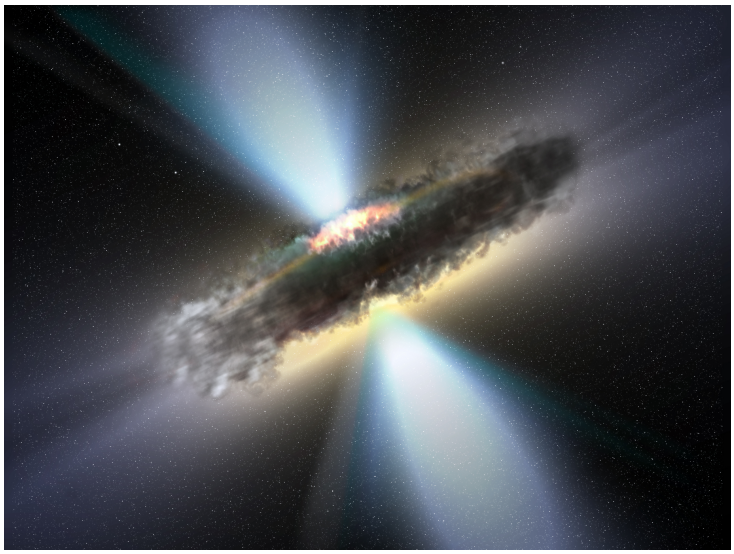


Figure 1: *Artist's rendering of a quasar. Gas and dust form a torus around the central black hole, with clouds of charged gas above and below.*
Credit: NASA/ESA.

Quasars are part of a class of objects known as "active galactic nuclei" since they require supermassive black holes (SMBHs) to power them. All galaxies have SMBHs at their centres; active galactic nuclei are the fraction actively growing. When material gets too close to the black hole it forms an accretion disk (see Figure 1). Because of the extremely high pressure the matter in the accretion disk heats up to millions of degrees, blasting out enormous amounts

[1]Not highlighted hyperlinks available throughout the text, on figures, sections and references.

of radiation and jets of material which can be detected millions of light-years away. Quasars are so bright that they can outshine all other sources of light, including entire surrounding galaxies (Rees 1984), thus making them the furthest objects detectable.
HZQs with $z \gtrsim 6$ (hereafter referred to as HZQs) are situated in the epoch of cosmological reionization which ended a billion years after the Big Bang; they have helped prove that hydrogen in the early Universe was indeed neutral (Fan et al. 2006).
The most distant HZQ is ULAS J1120+0641. With a redshift of 7.08, it is found 13 billion light-years from Earth (thus appearing to us as it was approximately 700 million light-years after the Big Bang)(Mortlock et al. 2011). The closest quasar identified to date is Markarian (Mrk) 231 (1969), $z = 0.04$, located 581 million light-years away from earth (there is no quasar in our galaxy and its central blackhole appears to be quiescent) (Schmidt 1963).
The colours of stars and other astronomical objects differ from quasars, but the practical problem of photometrically looking for HZQs derives from the large population amongst which the search needs to be performed - tens of objects in samples of hundreds of millions. Surveys take images of a vast region of sky to then automatically identify and process the light sources. Modern surveys are capable of gathering several terabytes of data on a daily basis. Data management and its classification thus becomes a real challenge in astronomy and requires advanced computational techniques.

The Bayesian method used in Mortlock et al. (2012), which calculates the probability that an object is an HZQ, proved to be very efficient; the most distant quasar ever discovered was identified using this probabilistic search technique. The "limitation" of this method is that it is built around the theoretical distribution of HZQs and it must be tuned for any given survey. This research effort concentrates on expanding the options by exploring the cutting edge computational tools of machine learning and specifically of artificial neural networks. NN methods are not new for quasar searches, but this is the first known application to HZQs with $z \gtrsim 6$.
A number of papers explore the use of NNs for the search of quasars. Starting from "Neural Networks in Astronomy" (Tagliaferri et al. 2003), different

groups have applied the NN techniques: "Selection of quasars candidates from combined radio and optical survey using neural networks" (Carballo et al. 2004); "Photometric identification of quasars from the Sloan Survey" (Sinha et al. 2006); "Selection of quasar candidates from combined radio and optical surveys using neural networks" (Carballo et al. 2008); "A photometric catalogue of quasars and other point sources in the Sloan Digital Sky Survey" (Abraham et al. 2012), but all of those focus on quasars with a maximum redshift value of 4. By concentrating instead on quasars with $z \gtrsim 6$ there is a new challenge linked to the small target population of HZQs.
Machine learning techniques, and in particular NNs, are an ideal way to quickly and effectively analyse big quantities of data and will be necessary for the coming generation of astronomical surveys, such as the Large Synoptic Survey Telescope (Ivezic et al. 2008).
Although optical spectrum will ultimately be required to confirm the quasar classification, an optimised computational selection of candidates permits the reduction of telescope time.

2 Machine Learning

There are two main categories of machine learning: unsupervised and supervised learning. In unsupervised learning there is no desired output value; the aim is to extrapolate a function to describe a hidden structure from unlabelled data. In supervised learning, the goal is to infer a function from a labelled training set (consisting of training examples which include an input object and a desired output value) and then apply the function to unlabelled data (test set). Supervised learning can be further divided into classification and regression; in classification, the labels have discrete values, while in regression the labels are continuous.
This research focuses on supervised classification problems, as objects in the universe are a discrete set of data points.

There are several machine learning approaches, some of the most established are:

- Decision trees
- Neural networks (NNs)
- Case-based reasoning
- Genetic algorithms

These examples can be divided in two major categories: eager learning and lazy learning. In eager learning, the system creates a general approximation to the target function during training. As an example, when using the neural network approach, a network is created during training and then applied to the new instances. A similar reasoning can be used with decision trees.
However, lazy learning methods (such as case-based reasoning) store the instances, so that generalising beyond the data is postponed until an explicit request is made. Lazy learning methods construct a different approximation to the target function for each encountered query instance. For an in-depth introduction to machine learning the reader is referred to (Mitchell 1997).
The structured approach of eager learning and in particular NNs has been considered appropriate for the computational search of HZQs.

2.1 Artificial Neural Networks (NNs)

NNs are a family of machine learning models inspired by the brain. They consist of a group of interconnected nodes, with each one processing the information it receives and passing it to other neurons through a weighted connection. The most basic neural network - with one neuron - is called a Perceptron (see Figure 2). In a perceptron, the inputs form a weighted sum which, after being passed through an activation function, gives the output as a boolean variable.

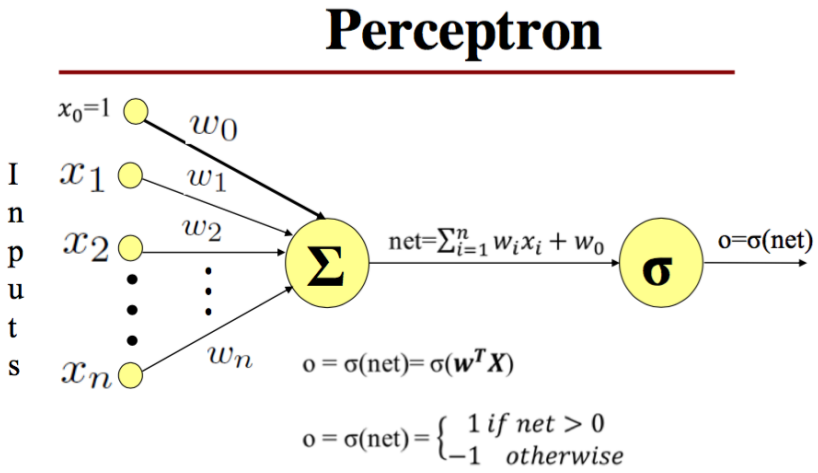


Figure 2: *Perceptron -*$(x_1, ..., x_n)$ *are the inputs,* $(w_0, ..., w_n)$ *are the weights,* $\Sigma =$ *is the neuron which consists of the dot product of the inputs and the weights,* σ *is a sign/step function.*
Credit: Maja Pantic (2015)

More neurons or even hidden layers between the inputs and the outputs (with a variable number of neurons in each layer) can be added to approximate more complex systems. These complex networks are called "multilayer feed forward neural networks" ("feed forward" means that inputs are sent to the neuron and are processed into an output - there are no cycles in the network). Figure 3 shows an example of a network with 3 inputs, 4 outputs, and 3 hidden layers with 5 neurons.
For a broader overview of machine learning see (Mitchell 1997) or (Shiffman 2012).

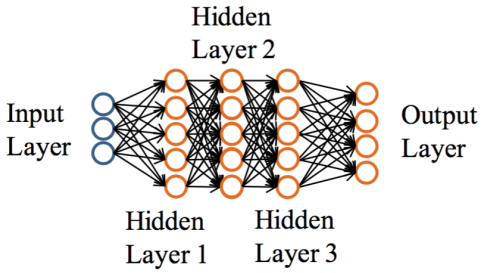


Figure 3: *4-layer feed-forward network. The network presents 3 inputs, 4 outputs, and 3 hidden layers with 5 neurons in each layer.*
Credit Maja Pantic (2015).

2.2 Evaluation and testing of the classification

The next paragraphs will describe the instruments used to assess and compare the performance of classification models. The four tools used are: the confusion matrix, the ROC curve, the Two-Colour diagram, and the Comparison Histogram.

2.2.1 Confusion matrix

A confusion matrix is a table that portrays the performance of an algorithm. It presents the True Positives (TP) and the True Negatives (TN) on the main diagonal, and the False Positives (FP) and False Negatives (FN) on the non-diagonal. Figure 4 shows an example of a confusion matrix with two classes: the diagonal displays the number of correctly classified examples, and the non-diagonal portrays the number of incorrectly classified ones.
This can be easily extended to the multi-dimensional case as will be shown in the toy model in the next section (Figure 9).

Classification Measures There are several types of classification measures to include: completeness/recall, efficiency/precision and accuracy/classification rate.

- Completeness/Recall $\frac{TP}{TP+FN}$.
- Accuracy/Classification rate $\frac{TP+TN}{TP+TN+FP+FN}$.
- Efficiency/Precision $\frac{TP}{TP+FP}$.

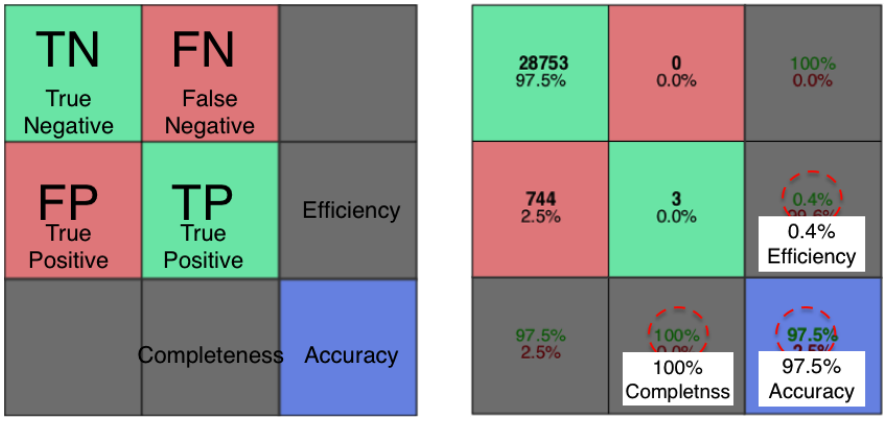


Figure 4: *Confusion matrix. An example on the right with the corresponding definitions on the left.*

Considering HZQs as true positives, the main objective is to maximise completeness to avoid the loss of potential candidates. The second priority is to reach the best possible efficiency and classification rate by minimising false positives. Efficiency is necessarily low because of the rarity of HZQs.

2.2.2 ROC curves

The receiver operating characteristic (ROC) is a metric used to check the quality of classifiers. The ROC is calculated by applying threshold values across the interval [0,1]. For each threshold, two values are calculated: the true positive ratio (the number of outputs greater than or equal to the threshold, divided by TP+FP, as defined in the previous section), and the false positive ratio (the number of outputs less than the threshold, divided by TN+FN). The ROC graph is a plot of the false positive ratio vs true positive ratio for each threshold.
As Figure 5 shows, the closer the line is to the top left edge, the better the classification for that specific class is ranked (high values of True Positive rate and low values of False Positive rate). If the line is on the $x = y$ axis, then it means that approximately 50% of the examples are classified correctly. The results from the ROC curves and confusion matrices may be slightly different. The ROC curves depend in fact on the threshold and so the value between 0

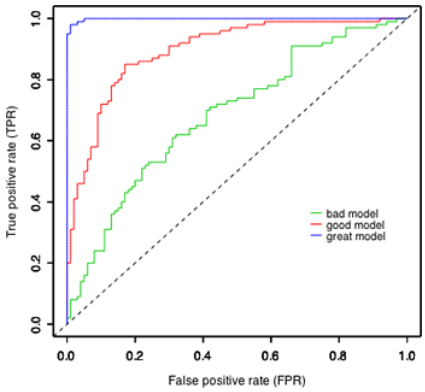


Figure 5: *ROC curve example.*
Credit: Jack Weiss (2010)

and 1 attributed to each class. In the confusion matrix, instead, a "winner" is assigned to each output value i.e. the one with the highest value. This difference can be noticed while analysing graphs in the next section on the toy model; it is therefore important to keep in mind that confusion matrices and ROC curves are independent methods used to analyse the effectiveness of the classifier. For a more detailed overview of ROC curves and confusion matrices see (Fawcett 2006).

2.2.3 Two-Colour diagram

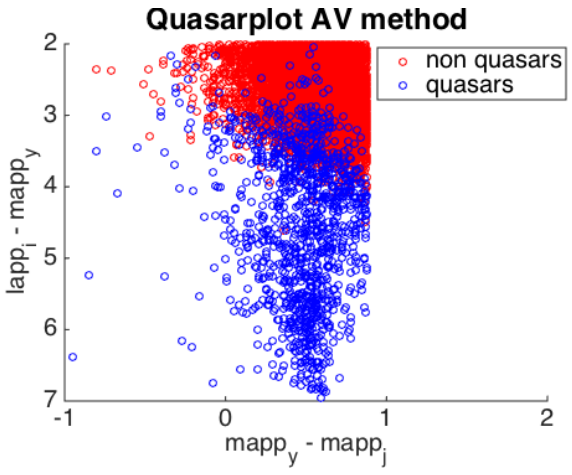


Figure 6: *Two-Colour diagram example*

The diagram in Figure 6 is used to give a visual evidence of the classifi-

cation results: red indicates the objects classified as non-HZQs and blue the HZQs. A more in-depth astronomical interpretation will be offered in section 3.1.
This type of graph is used in section 6, but not in the toy model.

2.2.4 Comparison Histogram

The Comparison Histogram is a graph created for the scope of this project:to compare the Bayesian inference calculated in (Mortlock et al. 2012), with the NN classification.

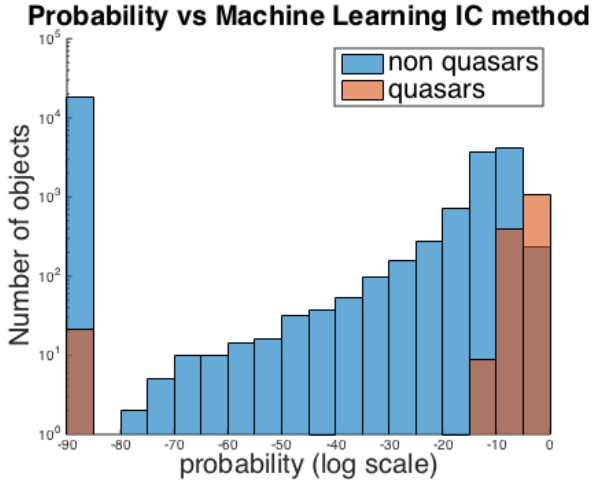


Figure 7: *Comparison Histogram example*

Figure 7 is constructed as the superposition of two histograms, the blue and the orange. Both histograms have on the x-axis (the logarithm of) the probability that the objects are classified as HZQs, calculated with the Bayesian inference method. The blue histogram represents the objects which have been classified by the neural network as non-HZQs; the orange histogram represents the objects which have been classified by the neural network as HZQs. The total number of objects classified with the Bayesian probability is the sum of the blue and orange histogram columns. The brownish colour represents the superposition of the blue and orange NN classification. The points with probability zero are those in the last column to the left.
The interpretation of this particular comparison histogram in Figure 7 is that the majority of the objects classified as HZQs by the neural network coincide

in general with the ones that are assigned a high probability. Instead, just over ten objects with null probability of being HZQs are classified as HZQs by the NN.
This type of comparison is used in section 6, but not in the toy model.

2.3 Bayesian inference as a comparison to NNs

Bayesian inference has already been successfully implemented to search for the rare HZQs with $z \gtrsim 6$ in (Mortlock et al. 2012) and this project represents an alternative to it. The probabilisitic approach is considered in this research as a reference and a comparison to the results obtained. It is a classification method based on Bayes' theorem (equation 1). Prior knowledge is combined with observed data to determine the final probability of a hypothesis. In a classification problem, the aim is to find the probability that a set of data points d are in a specific category or class C: $\Pr(C|d)$.
Bayes' theorem states that:

$$\Pr(C|d) = \frac{\Pr(d|C)\Pr(C)}{\Pr(d)} \propto \Pr(d|C)\Pr(C) \tag{1}$$

Where $\Pr(C)$ is the prior probability, $\Pr(d|C)$ the likelihood and $\Pr(C|d)$ the posterior probability. Using proportionality, it is possible to exploit the fact that $\sum_{c_i \in C} \Pr(c_i|d) = 1$ to find the posterior probabilities.
For details on Bayes' techniques and marginal likelihood, the reader is referred to an advanced text such as (Carlin & Louis 2000).

Marginal likelihood Given a set of independently and identically distributed data points $\bar{x} = (x_1, ..., x_n)$ where $x_i \sim Pr(x_i|\theta)$ and θ is a random variable described by $\theta \sim Pr(\theta|\alpha)$ then

$$Pr(\bar{x}|\alpha) = \int_\theta Pr(\bar{x}|\theta)Pr(\theta|\alpha)\mathrm{d}\theta \tag{2}$$

is the marginal likelihood of $\bar{x}$.

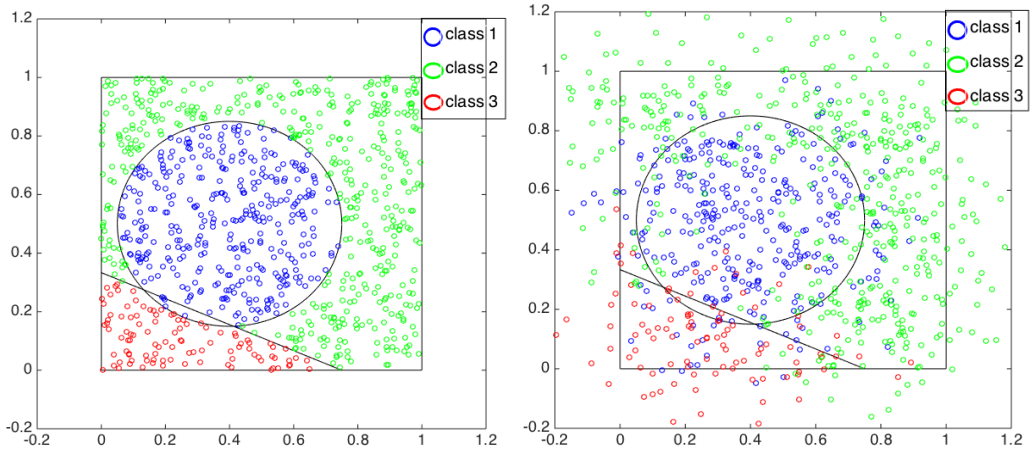
2.4 Application to a toy model

The toy classification problem is based on the three-way classification set created by Radford Neal, which can be found in Neal (2014) and has been used

with similar purposes in (Graff et al. 2013) section 4.2. The data examples consist of four random samples from a uniform distribution (x_1, x_2, x_3, x_4) on the unit square. The unit square is divided into 3 parts:

- Class 1: if the Euclidean distance between (x_1, x_2) and the point $(0.4, 0.5)$ is less than 0.35.
- Class 2: if $0.8x_1 + 1.8x_2 < 0.6$.
- Class 3: if both conditions are false.

The points: (x_3, x_4) are not utilised for the classification but are included simply to be consistent with the toy model and to verify how the model behaves using these extra parameters. White noise: σ is added to the training and testing set. The classification is reproduced with different values of σ. In Figure 8a there is the standard distribution of the data with no noise, while in Figure 8b there is an example of a test data sample with $\sigma = 0.12$.



(a) *Data distribution no noise* (b) *Data distribution, noise ($\sigma = 0.12$)*

Figure 8: *The figures represent the data distribution: 900 points subdivided in the three classes (blue = class 1, green = class 2, red = class 3).*

The same classification is also reproduced with heteroskedastic (i.e. variable) σ. Most of the classification is done using a training set of 1000 elements and a test set of 300 elements.

2.4.1 Skynet

Skynet is an efficient and robust neural network training tool which can train deep networks. The following parameters are suggested from (Graff et al. 2013):

- 1 hidden layer with 8 neurons.
- Data whitening (normalising of the values) before input.

2.4.2 Matlab Neural Network Toolbox

The Matlab Neural Network Toolbox is a much more general tool that supports different types of supervised and unsupervised learning. The following paragraph will describe the optimisation of the parameters available in the toolbox.

The initial optimisation considered a large variety of different parameters and techniques:

1. Training function and internal parameters.
2. Number of hidden layers with the respective number of neurons in each layer.
3. A regularization value.
4. Preprocessing and postprocessing steps.

Training functions. These following training functions were compared:

- *trainscg* - scaled conjugate gradient backpropagation.
- *traingd* - gradient descent backpropagation.
- *traingda* - gradient descent with adaptive learning rate backpropagation.
- *traingdm* - gradient descent with momentum backpropagation.
- *trainrp* - resilient backpropagation.

Some of the training functions have internal parameters that must be optimised including: the learning rate, the ratio to increase/decrease learning rate, the momentum constant, and the increment/decrement to weight change. The parameters in each training methodology are continuous. To maximise efficiency, the most typical values which are suggested on (Mitchell 1997) were considered.
Several classifications were run to compare each parameter and choose the optimal ones. *trainscg* gave the best classification and so was picked to continue the analysis.

Hidden layers. Another important parameter to optimise was the number of hidden layers and the number of neurons in each network. Differently from other parameters, they tend to give results which differ depending on the training set. For this reason an algorithm was set up to create several different combinations of hidden layers and neurons in each layer, and to pick the optimal one every time a network is trained. The values looped through were:

- 1 to 5 hidden layers.
- 10, 20 and 30 neurons in each hidden layer.

Following is an example of the Matlab code for a basic classification cycle.

```
%% NN cycle on variable hidden layers
for nH = 1:length(nHidden);
    net = feedforwardnet(nHidden{nH}, 'trainscg');
    net.performParam.regularization = 0.25;
    net = configure(net,x2,output);
    net.trainParam.epochs = 700;
    [net, tr] = train(net, x2,output);
end
```

Regularization. A regularization with a parameter of 0.25 was introduced to minimise overfitting and improve the classification.

Preprocessing and postprocessing. Preprocessing and postprocessing steps (i.e data whitening) were implemented on the inputs and targets.

2.4.3 Bayesian inference

This paragraph contains the analytical calculations to find the probability that each data point belongs to a specific class:

$$\Pr(C = i|(\hat{x}_1, \hat{x}_2)), \tag{3}$$

where i = (1, 2, 3) are the possible classes and $\hat{x}_1, \hat{x}_2$ are two random samples from a uniform distribution with added white noise (x_1, x_2 are the clean versions).
By Bayes' theorem (equation 1):

$$\Pr(C = i|(\hat{x}_1, \hat{x}_2)) = \frac{\Pr((\hat{x}_1, \hat{x}_2)|C = i)\Pr(C = i)}{\Pr((\hat{x}_1, \hat{x}_2))} \propto \Pr((\hat{x}_1, \hat{x}_2)|C = i)\Pr(C = i) \tag{4}$$

To be in category 1, the two-dimensional Euclidean distance between (x_1, x_2) and the point $(0.4, 0.5)$ needs to be less than 0.35. This means a total area of $\pi \times 0.35^2 = 0.3848$. Therefore:

$$\Pr(C = 1) = 0.3848451 \tag{5}$$

To be in category 2, then $0.8x_1 + 1.8x_2 < 0.6$. It gives an area of $\frac{1}{8}$, but we also have that the overlap between region 1 and 2: $\Pr(C = 0 \cap C = 1) = 0.0072622$ which means:

$$\Pr(C = 2) = 0.1250000 - 0.0072622 = 0.1177377. \tag{6}$$

Finally:

$$\Pr(C = 3) = 1 - (0.3848451 + 0.1177377) = 0.4974230. \tag{7}$$

What remains to be calculated is $\Pr((\hat{x}_1, \hat{x}_2)|C = k)$. Using the marginal likelihood function, equation 2 from section 2.3:

$$\Pr((\hat{x}_1, \hat{x}_2)|C = k) = \int \Pr(\hat{x}_1, \hat{x}_2|x_1, x_2)(x_1, x_2|C_k)\mathrm{d}x_1\mathrm{d}x_2 \tag{8}$$

$$= \int \Pr(x_1, x_2|C_k)\Pr(\hat{x}_1|x_1)\Pr(\hat{x}_2|x_2)\mathrm{d}x_1\mathrm{d}x_2 \tag{9}$$

Knowing that $\hat{x}_j \sim N(x_j, \sigma^2)$ for $j = 1, 2 \implies (\hat{x}_j|x_j) \sim N(x_j, \sigma^2)$.
Which means:

$$= \int \Pr(x_1, x_2|C_k) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_1 - x_1)^2}{2*\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_2 - x_2)^2}{2*\sigma^2}} \mathrm{d}x_1 \mathrm{d}x_2 \tag{10}$$

Given a specific class, there is a 100% certainty that the clean data is inside that class, so what changes are the limits of integration in each class. For example: $\Pr(x_{1obs}, x_{2obs}|C_1)$ is 0 if $0.8x_{1obs} + 1.8x_{2obs} < 0.6$ and 1 anywhere else. Following this reasoning we obtain:

$$\Pr((\hat{x}_1, \hat{x}_2)|C = 1) = \int_{x_1, x_2 \in A} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_1 - x_1)^2}{2*\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_2 - x_2)^2}{2*\sigma^2}} \mathrm{d}x_1 \mathrm{d}x_2 \tag{11}$$

$$\Pr((\hat{x}_1, \hat{x}_2)|C = 2) = \int_{x_1, x_2 \in B} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_1 - x_1)^2}{2*\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_2 - x_2)^2}{2*\sigma^2}} \mathrm{d}x_1 \mathrm{d}x_2 \tag{12}$$

$$\Pr((\hat{x}_1, \hat{x}_2)|C = 3) = \int_{x_1, x_2 \in C} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_1 - x_1)^2}{2*\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\hat{x}_2 - x_2)^2}{2*\sigma^2}} \mathrm{d}x_1 \mathrm{d}x_2 \tag{13}$$

Where :

$$A = \sqrt{x_1^2 + x_2^2} < 0.35$$

$$B = 0.8x_1 + 1.8x_2 < 0.6$$

$$C = [0, 1] \times [0, 1] - A \cup B$$

These integrals cannot be solved analytically, so Matlab was used to solve them numerically for each point in the test set.

2.4.4 The two methods compared against Bayesian inference

The following criteria are used to compare between the three methods:

- Confusion matrices
- ROC curves
- Mislabelled points

The criteria were compared for different values of constant noise (increasing from 0.02 to 1), as well as for heteroskedastic noise, for different maximum bounds that the noise could take (increasing from 0.1 to 5). The following figures and graphs did not give absolute results as each time the classification

was run, a different training and testing set was generated. Training a network on the same set can generate differences and classification results can vary even when using the same network. Nevertheless given the randomness in the creation of the sets all the results were approximately the same.

Constant noise If the value of noise was k, then the "noisy point" $\hat{x}_i$ was generated by:

1. Obtaining a random sample y_i from a normal distribution with variance k: $N(0, k)$.
2. Adding the random value to the data point: $\hat{x}_i = x_i + y_i$.

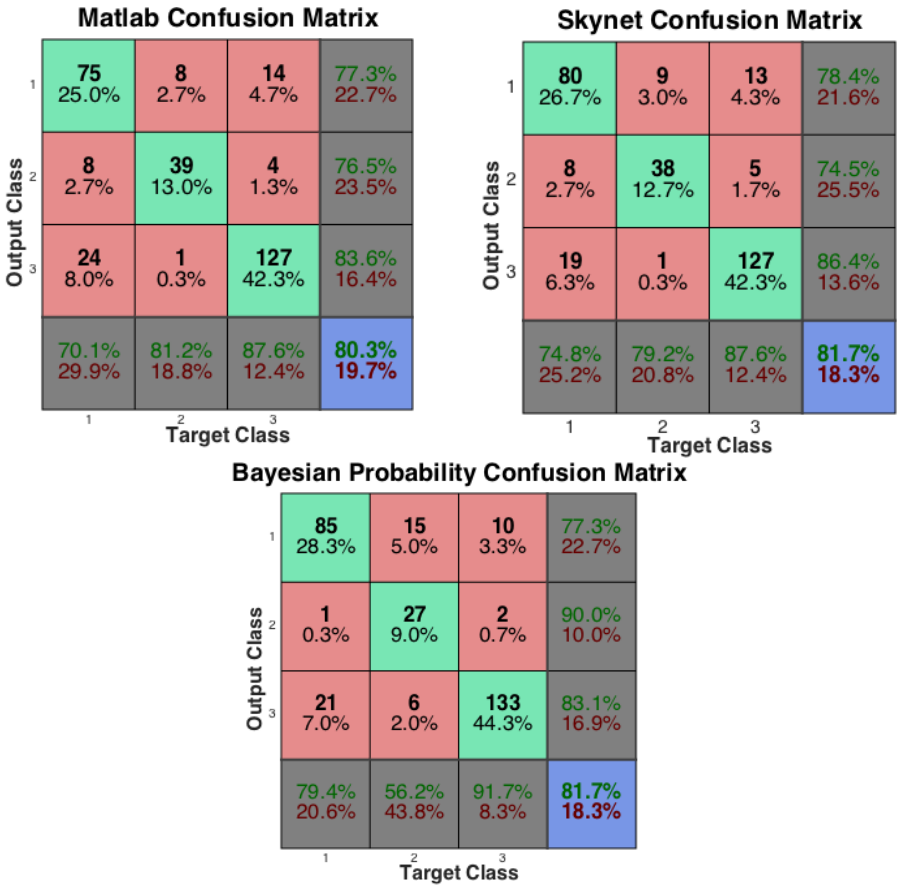


Figure 9: *Confusion matrices for the three classification methods, data with constant noise:* $\sigma = 0.12$

The results using a constant noise with $\sigma = 0.02$ were that all three methods performed extremely well with a classification rate of 96.0% for Matlab,

97.0% with the Bayesian inference method and again 97.0% with Skynet. The analysis did not give much insight or information on the different classification systems as the overall outcome was really high. Figures 9, 10 and 11 show the results i.e. the confusion matrices, ROC curves and scatterplots of the mislabelled points when $\sigma = 0.12$.

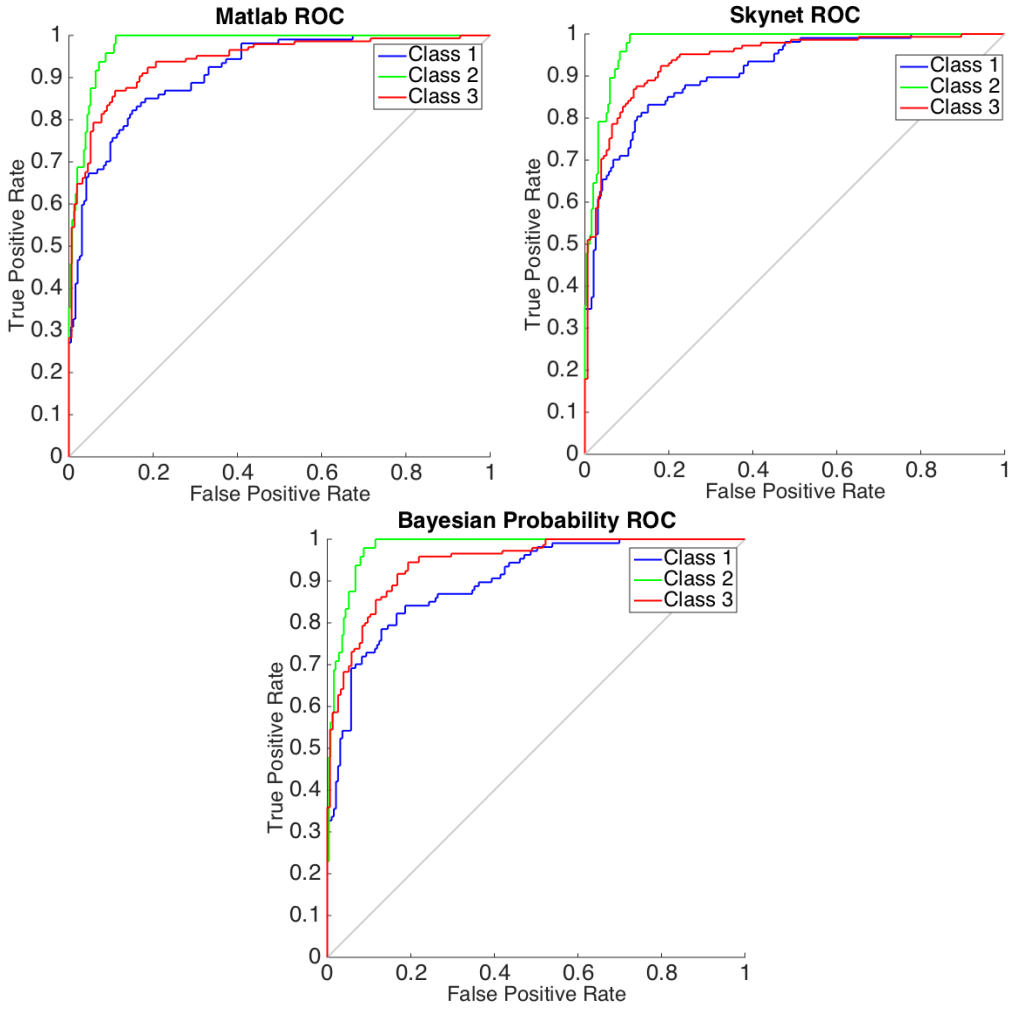


Figure 10: *ROC curves for the three classification methods, data with constant noise:* $\sigma = 0.12$.

This time Skynet and the Bayesian inference method with a classification rate of 81.7% outperformed Matlab which achieved 80.3%.

The ROC curves (Figure 10) show that class 2 (green) is the best classified one. While this may not seem consistent with the confusion matrices, the scatter plots show that very few of the green dots are mislabelled compared to the others. The first row of the scatter plots (Figure 11) shows the misclassified points and their erroneous classification, while the second row shows misclassified points in the correct classification. The mislabelled points are located around the boundaries between the different categories, and the ones misclassified tend to be the same ones in all three methods. The misclassified points are the ones that were sent outside their class because of the added noise. These behaviours tend to show themselves more evidently as σ increases.

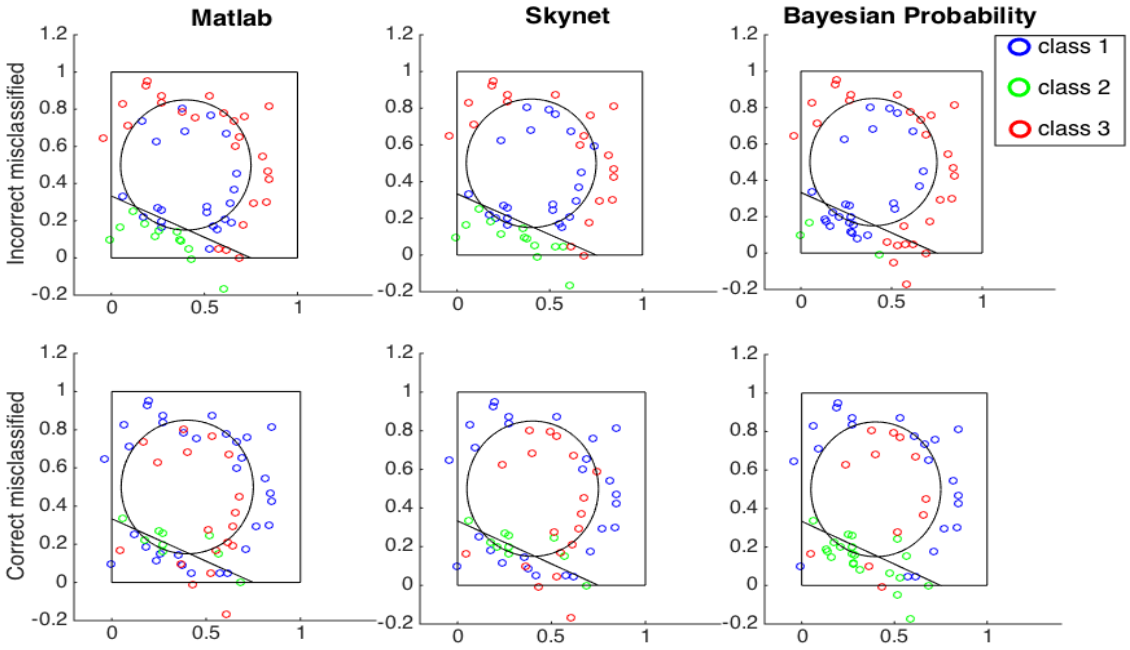


Figure 11: *Mislabelled data with constant* $\sigma = 0.12$. *The first row has the misclassified points with their incorrect classification while the second row represents the same misclassified points but labelled with their correct classification. Going from left to right the points are classified using Matlab, Skynet and Posterior Probability.*

A test with $\sigma = 10$ resulted in Skynet and Bayesian inference methods sending all the points to category 3. Matlab instead classified in a more diversified way but all three methods gave a classification rate of approximately 50%.

Heteroskedastic noise To add heteroskedastic noise to the data, the following procedure was used for each data point (d_i):

1. A random sample x_i was taken from a uniform distribution on $(0, \theta)$.
2. The noise value y_i was obtained by sampling from a normal distribution with variance x_i: $N(0, x_i)$.
3. The point with noise was defined as $\hat{d}_i = d_i + y_i$.

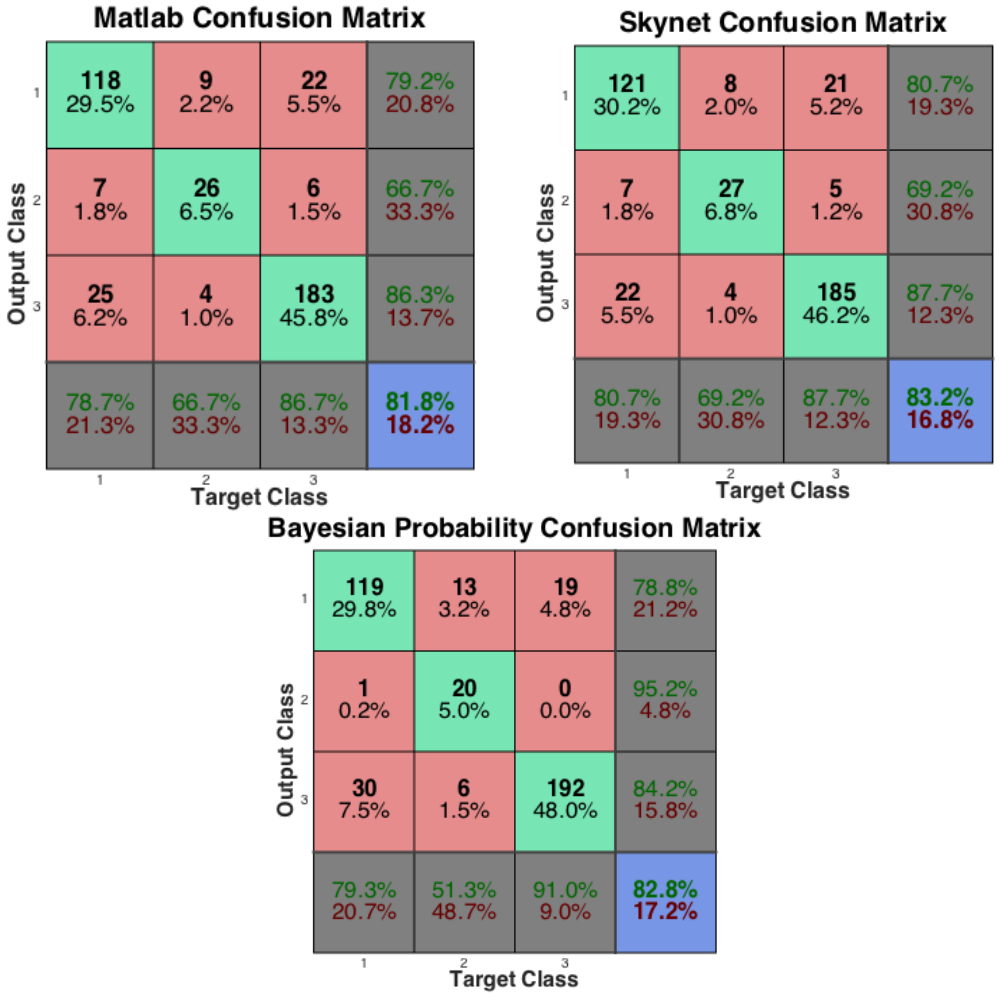


Figure 12: *Confusion matrices, data with heteroskedastic noise:* $\sigma \in (0, 0.2)$.

This process was repeated several times, each time with a different parameter for the random sample from the uniform distribution: $\theta \in (0.1, 5)$.

The confusion matrices and ROC curves where the original random sample was taken from a uniform distribution on $(0, 0.2)$ are shown in Figures 12 and 13.

The results are similar to a constant σ of value 0.12. Skynet seems to perform slightly better than Matlab and the Bayesian inference method (they have accuracies of 83.2%, 82.2% and 81.8% respectively). The ROC curves instead show better results for the Bayesian inference. When increasing the upper limit for the noise, the Bayesian inference tends to give better results. This is to be expected because when calculating the probability, it has prior knowledge of the noise value.

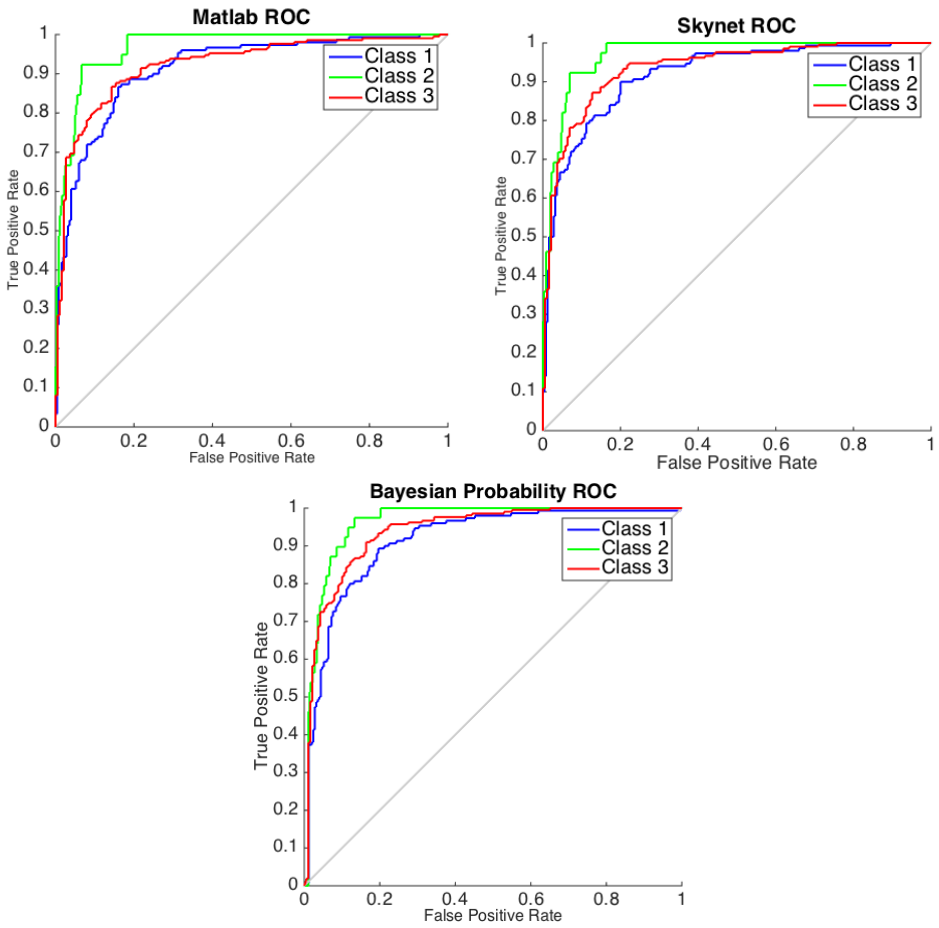


Figure 13: *ROC curves, data with heteroskedastic noise:* $\sigma \in (0, 0.2)$

2.5 Choice of computational tool

Summarising the results, both machine learning methods work well compared to the Bayesian inference classification. While it was hard to understand which one works best when dealing with low values of noise, the Bayesian method was superior with high values of noise and when it was heteroskedastic.
The best classifier between Matlab and Skynet was also difficult to determine, in fact they give very similar results. Matlab was the choice because of its ease of customisation compared to Skynet.

3 Astronomical data

This chapter gives an overview of the survey, of the HZQ selection process and of the database available. Then there will a more in-depth description of the data and its architecture.

3.1 Surveys and HZQ pre-selection process

The search for high redshift quasars is in itself a long process which starts with the creation of surveys.
The database used derives from the UKIRT (United Kingdom Infrared Telescope), Infrared Deep Sky Survey (UKIDSS), Large Area Survey (LAS) (Lawrence et al. 2007), cross-correlated with the Sloan Digital Sky Survey (SDSS) (York et al. 2000). This means that the area covered by LAS was overlapped with the SDSS survey to create a unique near-infrared catalog that concentrated the effort on high redshift targets. The cross-matched area was about 2270 deg^2 (5.5% of the total sky that is 41,253 deg^2) and produced 2×10^7 catalogued sources.
Figure 14 shows the theoretical distribution of the quasars: the set of lines starting from the centre of the graph. The farthest line to the right represents the theoretical distribution of brown dwarfs. The problem is that when photometric measurements are taken a measurement error displaces the true value; this is where the additional challenge comes in (see also 3.4).
The dotted line in Figure 14 represents a "preselection cut". This "cut" is bounded above by $i - Y = 2.80$ and is bounded on the right by $Y - J = 0.88$. The objective of the top bound is to eliminate all quasars with $z \lesssim 6$ while the right bound tries to separate brown dwarfs from HZQs. i, Y and J will be better explained in the following section.
Once the "preselection cut" was applied to the astronomical data, the data set was reduced to about 30000 objects which was the data set used in this work.

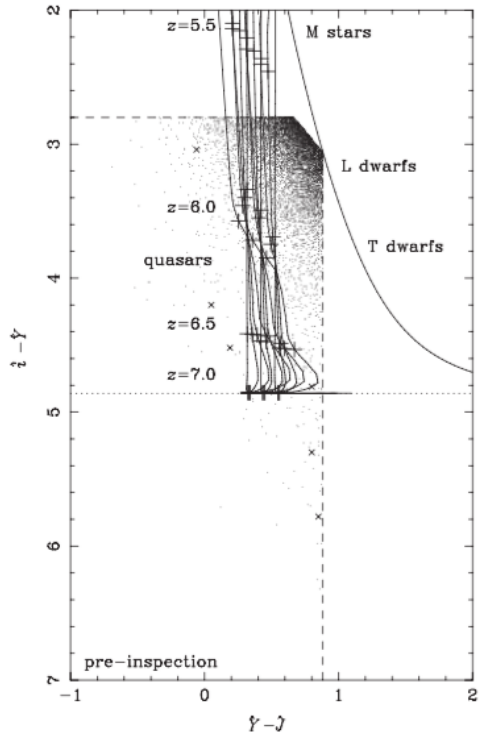


Figure 14: *Diagram showing the UKIDSS DR8 LAS point sources. They are quasar candidates as they are found inside the dashed lines which correspond to the initial pre-selection cuts. The set of lines which start inside the pre-selection cut represent the theoretical distribution of quasars, while the line tangent to the right corner of the cut represents the distribution of the brown dwarfs. (The data used in this report has only the right cut and not the top one - as can be seen in the plots of section 6; the idea is that the algorithm is capable of doing the rest) Credit: Mortlock et al. (2012)*

Beyond the scope of this paper: once the computational process has been completed, each candidate is examined visually by the researcher, measured photometrically and classified again using a second computational search. A final spectroscopical observation will verify with certainty for the completion of the process. Previous searches have been done using Bayesian inference as in Mortlock et al. (2012).

3.2 Spectral characteristics of HZQs

The spectral characteristics of quasars can be understood by analysing Figure 15. The y-axis is the relative flux and the x-axis is the observed wavelength.

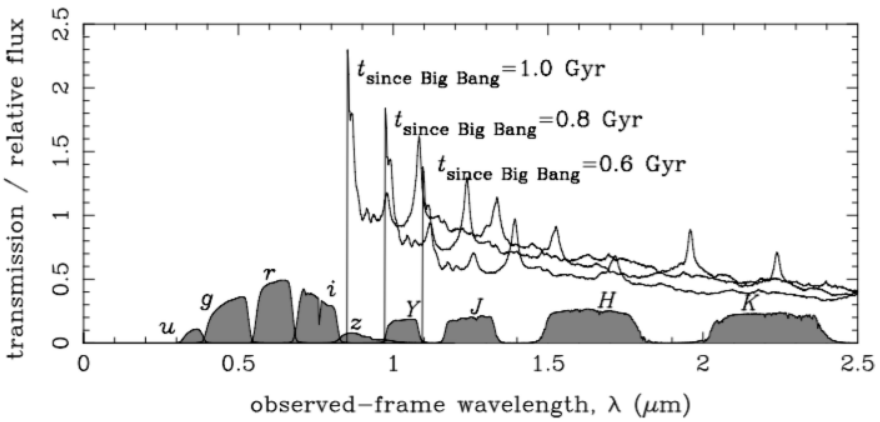


Figure 15: *Simulated spectra of quasars compared to the transmission curves of the SDSS optical filters (u, g, r, i, z) and the UKIDSS near-infrared filters (Y, J, H, K). The three quasars are at different distances and hence different look-back times, as labelled. The normalisation is arbitrary, although the decrease in flux level with distance is realistic. The least distant of these quasars might be seen in SDSS images (specifically the z band), but the most distant would not. All three simulated quasars would be visible in appropriately deep UKIDSS exposures.*
Credit: (Mortlock 2014)

What is unique to the spectrum of a high redshift quasar is that it has a distinct break. In Figure 15, short wavelengths correspond to "recent time": moving towards the right is like going back in time. Looking at the three spectra, the one with the break around 0.9 is the most "recent", and the labels show to what year they correspond to. It is interesting to understand why these spectra contain a break: initially in the Universe, hydrogen atoms were neutral and thus absorbed the light at low wavelengths emitted by the quasars (and other luminous objects). Around 1 billion years after the Big Bang (which corresponds to 1.0 Gyr in this case) hydrogen was completely ionised and so the emitted light stopped being absorbed. The rest-frame (i.e. the distortion due to redshift) of the break is $0.1216\mu m$, the wavelength of

all light is increased by cosmological expansion; the relationship is: $\lambda_{obs} = (1 + z)\lambda_{rest}$ where λ is the wavelength. The Universe is seven times larger now than when when it was one billion years old, and so the breaks in the spectra of these quasars are seen at an observed-frame wavelength of $\sim 7 \times 0.1216\mu m \simeq 0.85\mu m$. The breaks are expected to go from the i filter up to the filter J (Hewett et al. 2006).

3.3 Data sets

Two main data sets were used, one with astronomical data (also referred to as candidates set, real objects set, test set - it contains 30000 data points) and the other with simulated high redshift quasars (also referred to as simulated quasars set - contains 9000 data points). The cross-correlated astronomical object data set included values from the SDSS survey cross-correlated with the UKIDSS survey as explained above. The simulated HZQs were necessary in order to have a sufficient number of objects to perform the necessary training.

Simulated $z \gtrsim 6$ HZQ data set Each simulated quasar was generated by:

1. Drawing a redshift, z, and absolute magnitude M_{1450} from the best-fit evolving quasar luminosity function given in (Willott et al. 2010).

2. Drawing a continuum slope, s, and line-strength, l, from the empirical distribution of these properties found by (Hewett & Wild 2010).

3. Using the empirical quasar spectral energy distributions (SEDs) from (Hewett & Wild 2010) to calculate the fluxes F_i, F_z, F_Y and F_J that this quasar would have in the SDSS i and z filters and the UKIDSS Y and J filters.

4. Choosing a location on the sky within the areas covered by the SDSS and UKIDSS surveys to obtain a value for the noise σ_i, σ_z, σ_Y and σ_J in the four bands.

5. Generating measured fluxes in each of the above bands according to $F_{b,obs} \sim N(F_b, \sigma_{b^2})$, where b is i, z, Y or J.

6. Converting from the measured flux and noise level in flux units to magnitudes.

3.4 Project data structure

Every data point in both the candidates or simulated quasars data set contained four photometric measures in different filters (i, z, Y, J) each with its own measurement error as can be seen in Table 1.

	i	i_{err}	z	z_{err}	Y	Y_{err}	J	J_{err}
data point 1	24.34	0.54	18.55	2.61	17.54	0.02	16.96	0.02
data point 2	24.29	1.27	19.22	6.83	18.19	0.04	17.65	0.04
data point 3	24.36	0.58	20.16	0.12	18.63	0.04	18.09	0.05

Table 1: Data structure example

Photometry measurements and error data When obtaining photometry - the value of the flux - from a point source in the sky, the flux is measured by summing the light received from the object and subtracting the contribution from the sky background in the same region of the sky. The simplest technique, known as aperture photometry, consists in summing the pixel counts within an aperture centred on the object, and subtracting the product of the nearby average sky count per pixel and the number of pixels within the aperture. This gives the raw flux value of the target object. Unfortunately as with most measurement procedures, there is noise involved for a variety of factors; for example the night sky is not completely dark and the detector emits a thermal signal. This "extra" luminosity sums to the brightness of the sources and makes them seem brighter. The "extra" brightness is estimated by looking at a pixel with no luminous sources next to it and then is subtracted from the sources. The estimation is where the error comes in. This measurement error is present in the data sets next to the photometric values. For more information on astronomical photometry, refer to (Budding & Demircan 2007).

4 NN classification algorithm

The NN classification algorithm code was designed on the following priorities:

1. Avoid false negatives thus maximising the completeness.
2. Minimise false positives thus maximising the efficiency.

Completeness and efficiency are defined in section 2.2.1. The Matlab Neural Network Toolbox needed customisation and this required several thousand lines of software code.

Training set The first step in using a neural network for a classification problem was defining a training set. The training set was forcibly defined as a mixture of both positive (high red shift quasars) and negative data examples (non-high red shift quasars) which means combinig a subset of the simulated quasars with a subset of the real observations. The data set with real astronomical objects (also referred to as candidates set) coincided with the test set.

Optimal training sets depended on the balance between the number of data points from the candidates file and the simulated quasars.
The approach followed was:

1. To keep the candidates set size fixed while varying the simulated quasars set size (starting from the complete set, to a minimum of 200).
2. To keep the simulated quasars set size fixed while the candidates set varied (from a maximum of 10000 to a minimum of 200).

Even though the size was kept constant, every time a network was trained, the test set was different and always picked randomly to keep the results as generalised as possible.
After several tests, the general guideline discovered was that the best results were obtained when the two subsets were of a similar size and about 1000 examples.
Keeping this configuration as a base and then increasing the amount of examples from the candidates and decreasing the number of simulated quasars

improved the efficiency.

The optimal balance for the training set was found to be with approximately 2000 candidates and 1000 simulated quasars.

Data artefacts category - unsuccessful attempt In the candidates file data set there were actually three different classes; the main ones were HZQs and non-HZQs. The non-HZQs however could be further subdivided into "good/real" astronomical objects and data artefacts. Unfortunately, the problem with introducing such a third category was that the objects known to be data artefacts were only 682, a small proportion of a potentially much bigger population that did not permit any valuable statistic.

4.1 NN algorithm code level 1

The following parameters were chosen from the ones available in Matlab (for the list of available choices see section 2.4.2):

- The training function *trainscg*.
- Variable hidden layers - see below (already described in section 2.4.2 but briefly summarised below).
- Regularization values between 0.25 and 0.28.
- Use of preprocessing and postprocessing steps.

Variable hidden layer algorithm code. A customised code (see the code listing), enabled to obtain flexible neural networks adapting to the specificity of the individual training sets. The customisable architecture consisted of:

- 1 to 5 hidden layers.
- 10, 20 and 30 neurons in each hidden layer.

Following is an example of the Matlab code for a basic classification cycle.

```
%% NN creation cycle on variable hidden layers
for nH = 1:length(nHidden);
    net = feedforwardnet(nHidden{nH}, 'trainscg');
    net.performParam.regularization = 0.27;
    net = configure(net,x2,output);
    net.trainParam.epochs = 700;
    [net, tr] = train(net, x2,output,{},{},EW);
end
```

4.2 NN algorithm code level 2 - AV and IC

When classifying, the priority was to protect completeness i.e. to avoid false negatives (FN).
The classification as a result of an NN cycle depended on a variety of factors:

- The training set used.
- The size of the training set.
- The balance between candidates and simulated quasars.
- The parameters.

Even the same network may give slightly different classification results even if completeness remains consistent.

Running the classification while increasing the ratio candidates/simulated HZQs showed that eventually the completeness (our first priority) was compromised. A range was found for which, when repeating independent cycles, mixed results were obtained. Beyond this range, an unrecoverable drop-off was been recorded. The solution therefore was to design a repetition of independent NN cycles with a specific ratio so that stability was obtained either by averaging the results of each cycle or by the union of the resulting HZQs in the individual classification cycle sets. The two methods Average (AV) and InClusion (IC) are described below.
Notice that the optimised parameters presented were not an exact figure but a range which is as restrictive as possible; changing slightly the parameters changes slightly also the results.

4.2.1 Average (AV) method

Summarising the AV algorithm code, the logic was to:

1. Create multiple training sets with real objects and simulated quasars of comparable size, for example 2000 candidates and 1500 simulated quasars.
2. Train a different network for each of the different training sets.
3. Classify the candidates file using each network.
4. Create a final classification which is the average of the network outputs.

For a discussion on the results see section 6

4.2.2 InClusion (IC) method

Summarising the IC algorithm code, the logic was to:

1. Create multiple training sets with real objects and simulated quasars of comparable size. For example 18 training sets with about 2000 candidates and 800 simulated quasars.
2. Train a different network for each of the different training sets.
3. Classify the candidates file using each network.
4. Create a final classification which included all the objects classified as HZQs from the different networks.

This method, since it identified as HZQ each object classified as such, ensured a 100% completeness but a lower efficiency than the AV method. For a discussion on the results see section 6

4.3 NN algorithm code level 3 - EP

A superiority of the Bayesian method compared to NNs is that an actual probability is calculated for each object. So far the NN classification effort offered only a black and white answer; a probability was obtained by an additional augmentation of the code. The Empirical probability (EP) method algorithm logic was to:

1. Create a large number of training sets with real objects and simulated quasars of comparable size. For example 150 training sets with about 2000 candidates and 900 simulated quasars.
2. Train a different network for each of the different training sets.
3. Classify the candidates file using each network.
4. Count the number of times each object was classified as HZQ and divide by the total number of networks.

The final result was then similar to a probability; the objects considered as HZQs were those with a probability greater than 0.5. The code listing is an example of the design of the core of the EP algorithm.

```
num1 = length(candidates);
num2 = length(bnets);
results = cell(2,num2);
sumthis = cell(1,num2);
sum = zeros(1,num1);
for loop = 1:num2
    bneti = bnets{loop};
    results{loop} = bneti(candidates');
    sumthis{loop} = results{loop}(2,:)>results{loop}(1,:);
    sum = sum + sumthis{loop};
end
probability1 = sum/num2;
A schematic was prepared to summarise the correlation ...
    between the algorithm code and its
relevant functionalities.
```

NN classification algorithm block diagram. A schematic diagram (Figure 16) was prepared to summarise the correlation between the algorithm code and its relevant functionalities.

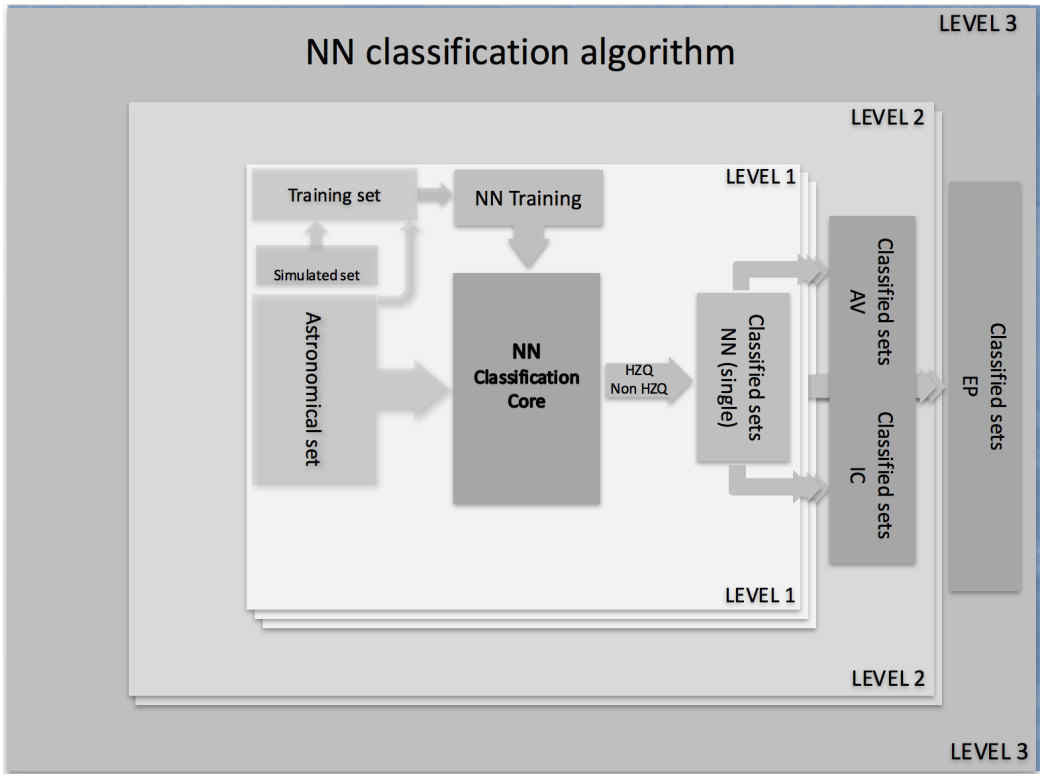


Figure 16: *In the middle of the diagram is the core of the calculation, which the training and classification phases. To the left are the inputs (Astronomical data set, simulated data set, training set). To the right are the outputs as classified results (AV, IC and EP).*

5 NN classification algorithm enhancements

Up to now, the training of the neural network and the classification of the candidates has been done using the four photometric measurements of each object. Other information present in the data set (i.e. measurement error) was not taken into account. However, it was possible to use the additional valuable information in several instances: during training, during classification or in any combination of the two.
The following sections briefly describe the usage of the error data as extra inputs. At first, the measurement error was used during the training phase; then, by means of a new transformation method denominated 2D-mapping, the measurement error was used to make the point-like source into a Gaussian distribution.
To the best of the author's knowledge, there is no established research which includes the measurement errors into the classification process.

5.1 Modifying the training algorithm

There are different ways to train a neural network function, and the choice of the optimal one is fundamental. As mentioned in 2.4.2, with the toy problem the following training functions have been tested: *trainscg*, *traingd*, *traingda*, *traingdm* and *trainrp*. *trainscg* was selected as it gave the best results.
The gradient descent training method *traingd* resurfaced though while studying the articles (Czarnecki & Podolak 2013) and (Reed et al. 1992).
The paper (Czarnecki & Podolak 2013) introduces a very interesting way of using noise during training. While this method was not implemented, it was still taken into consideration, and studied at length; it has given inspiration for the design that follows.

5.2 Upgrading the training phase with weighted inputs

One of the basic ideas behind the training is that the network weights are constantly modified by each example during training. Examples based on data with high measurement error are highly uncertain and should not be

taken into consideration as much as the others. This intuitive reasoning was the basis of the method implemented. The first experiment was performed through the use of a variable learning rate, described in the backpropagation appendix A. The idea was to define a high learning rate corresponding to small measurement error and a low learning rate corresponding to a high measurement error; unfortunately the learning rate was not applicable to the training function *trainscg*.
The alternative was to input a different weight for each example during training. Instead of modifying the learning rate every time, each training example was assigned a weight inversely proportional to the size of the error of each measurement; more specifically a sigmoid function sent low errors to a weight close to 1 and high errors to a weight close to 0. This last upgrade was actually implemented.
As there were four measurements for each object, the average of was used and converted to a weight. The optimised coefficients of the sigmoid function were found after several trials.
While this technique did not drastically improve the classification results it greatly improved the stability of the classification and the likelihood that the HZQs were correctly classified.

Error data as additional input to the training set - unsuccessful attempt To specialise the NNs the error data described in 3.4 was included, making it a total of 8 inputs: the 4 measurements previously used, plus their corresponding 4 errors. Unfortunately this caused the networks to get confused and consistently label one of the HZQs as a non-HZQ; also the ROC curves revealed to be a lot worse than before. Overall, adding the error as input was unsuccessful.

5.3 2D-mapping (2D) method

The astronomical data set is a collection of point-like objects. If the measurement value of each object is considered as the mean of a Gaussian distribution, and the corresponding measurement error as the variance, the point-like source is converted into a Gaussian distribution. This mapping provided an

increase of the "relevance" of the candidate HZQs during the classification phase: a false target would be further penalised while a real HZQ would increase its chances of being focused. This method, named 2D-mapping, proved to be the most promising, and it was implemented by:

1. Creating multiple training sets with real objects and simulated quasars of comparable size. For example 10 training sets with about 2000 candidates and 1500 simulated quasars.
2. Training a different network for each of the different training sets.
3. Generating for each point in the test set 50 other points, by sampling from a normal distribution with his original point as the mean and measurement error of the original point as the variance.
4. Classifying every point.
5. Taking the average of each of the 50 data points so to reduce to a classification of the original size.
6. (After repeating this procedure for each of the 10 networks) taking the average of the ten outputs to create a final classification or picking the output with the best classification.

A fundamental correction to this method was to limit this "spreading" process (step 3) only to examples which had a noise level less than 0.13 (this was the optimal limit reached after several trials). Results for this method will also be presented section 6.
This method can be used by itself - with an individual network - or "mixed" with AV and IC. The results for the mixed methods can be found in appendix D.

NN classication algorithm enhancements block diagram Figure 17 is a diagram representing a schematic setup of the algorithm enhancements using error data.

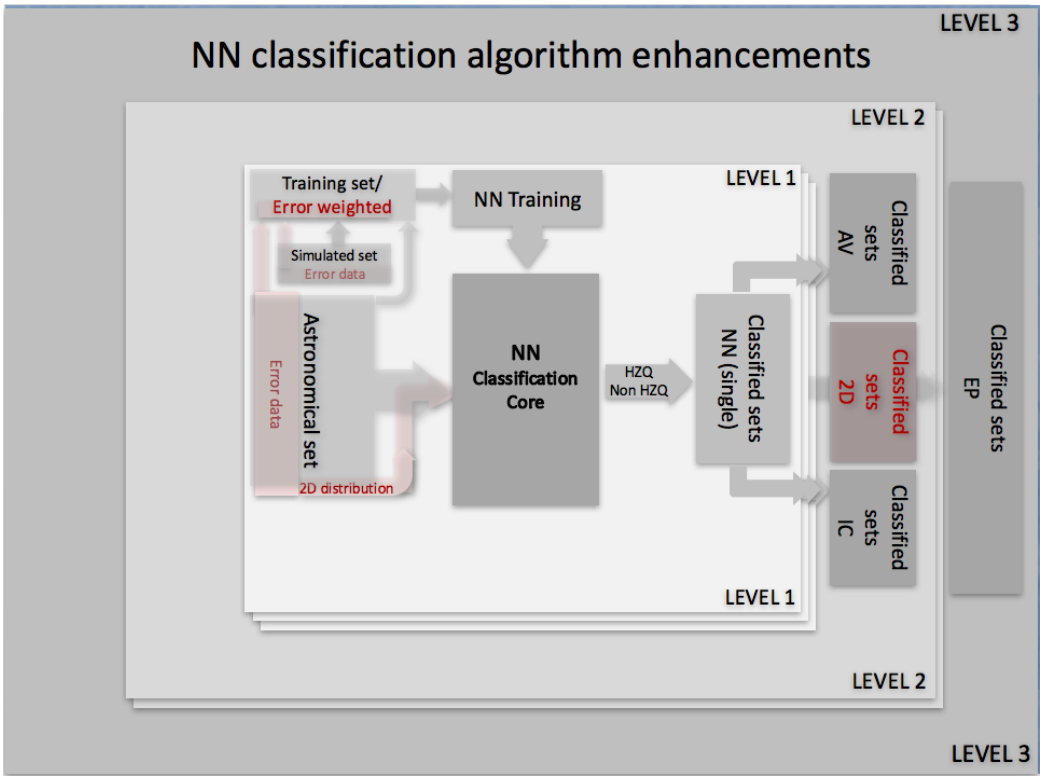


Figure 17: *The enhancements are depicted in colour (pink). The error data offers very valuable information. It was used both to weigh the input data before the training (in order to emphasise the importance of the training with low error) and to convert from a point-like object into a 2D Gaussian distribution distribution (thus maximising the chances of having the correct focus on any HZQ).*

Future improvements Possible computational designs to include measurement error are found in Appendix A.

6 Results

The four main methods implemented were: the average (AV), the inclusion (IC), the 2D-mapping (2D), and the empirical probability (EP).
The weighted training examples discussed in section 5.2 were also used. For additional verification of the robustness of the classification, a random subset of the simulated quasars data was added to the candidates file during classification; these results are presented below as well.
As mentioned for the toy example, all the following figures and graphs do not give absolute results since a different training set and test set is generated each time the classification is run. Training a network on the same set can generate differences. Classification results can differ (even if very slightly) also when using the same network and the same test data. Even though optimal parameters differed for each method, the same were used for a better comparison. The parameters are as following:

- Regularization of 0.27.
- 700 simulated quasars and 2000 candidates in each training set.
- 10 different networks.

Classification results: the completeness was always 100%. The efficiency and classification rate can be found in the table below.

Method	Efficiency	Classification rate
2D (0.13)	1.2%	99.2%
Bayesian	0.4%	97.5%
2D (0.9)	0.3%	96.3%
EP	0.3%	96.0%
AV	0.2%	95.9%
IC	0.2%	95.0%

Table 2: *Classification results*

The Bayesian results in the above table assume candidates with probability $p \geq 0.1$ as HZQs (identifying the 3 HZQs, 744 false positives).

Appendix B contains results obtained with a different data set with ~ 9000 objects and 5 HZQs and Appendix C include ~ 500 simulated quasars added to the test set.

6.1 AV results

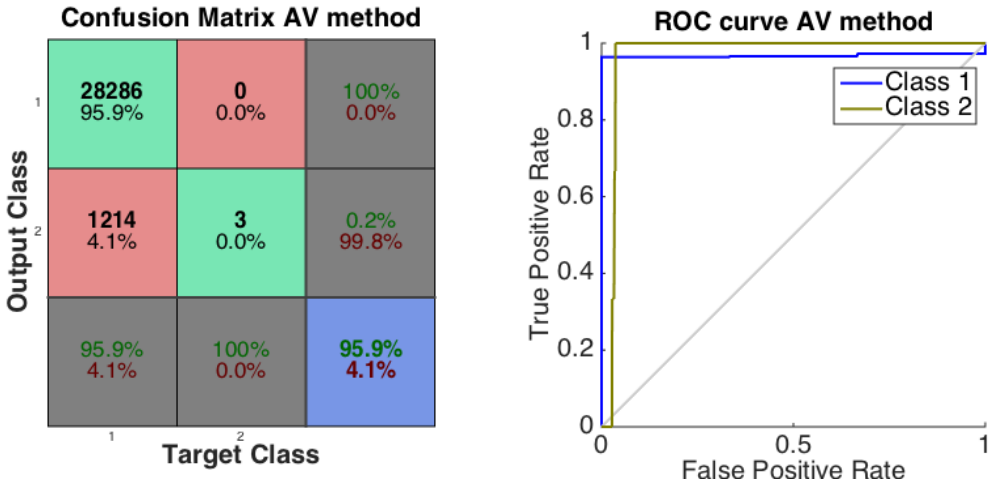


Figure 18: *Completeness of 100% and classification rate of 95.9%.*

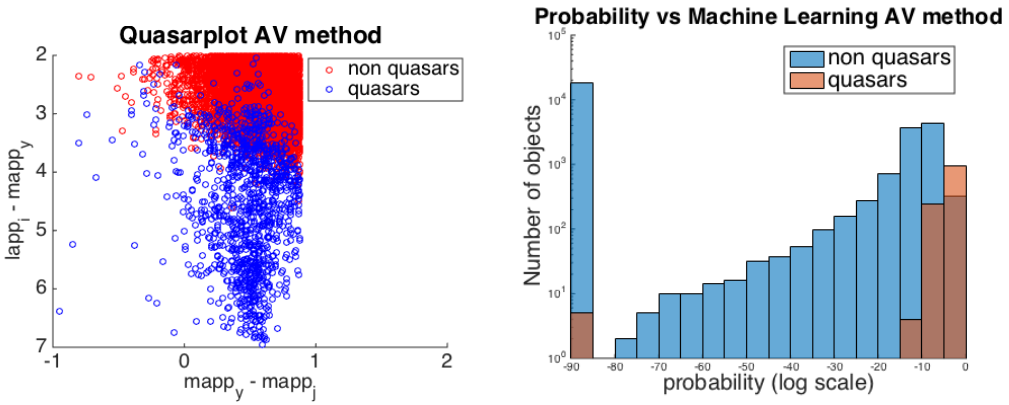


Figure 19: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4. The majority of the candidate HZQs are inside the "preselection cut".*

6.2 IC results

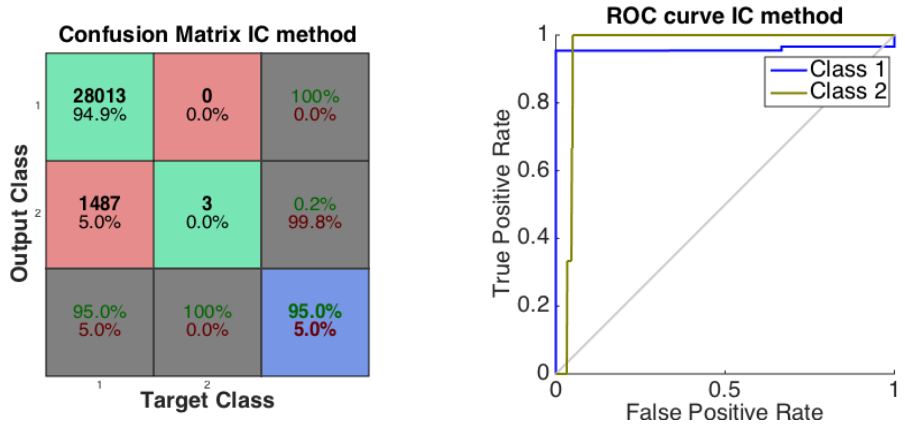


Figure 20: *Completeness of 100% and classification rate of 95%.*

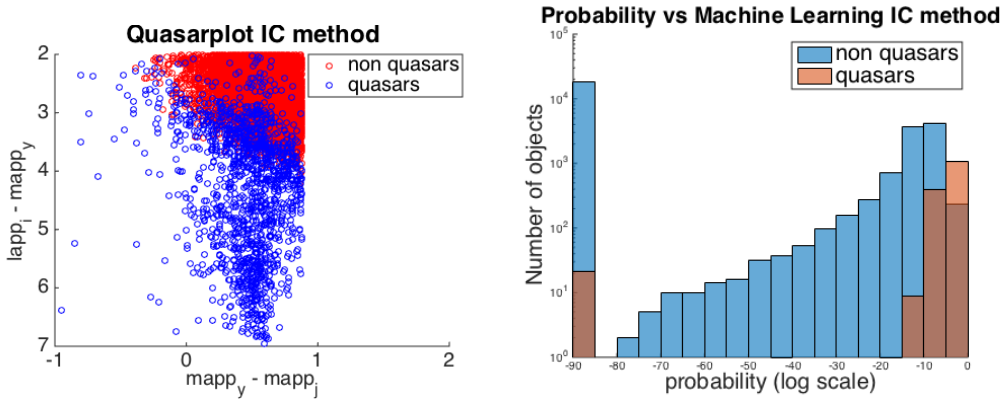


Figure 21: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4. The majority of the candidate HZQs are inside the "preselection cut".*

6.3 2D results

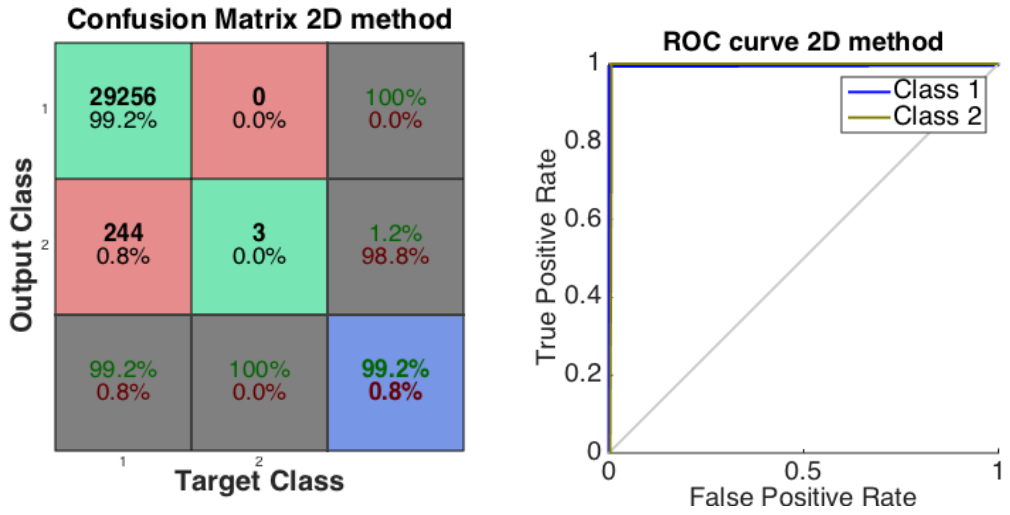


Figure 22: *2D method using the best network with limit 0.13. Completeness of 100% and classification rate of 99.2%.*

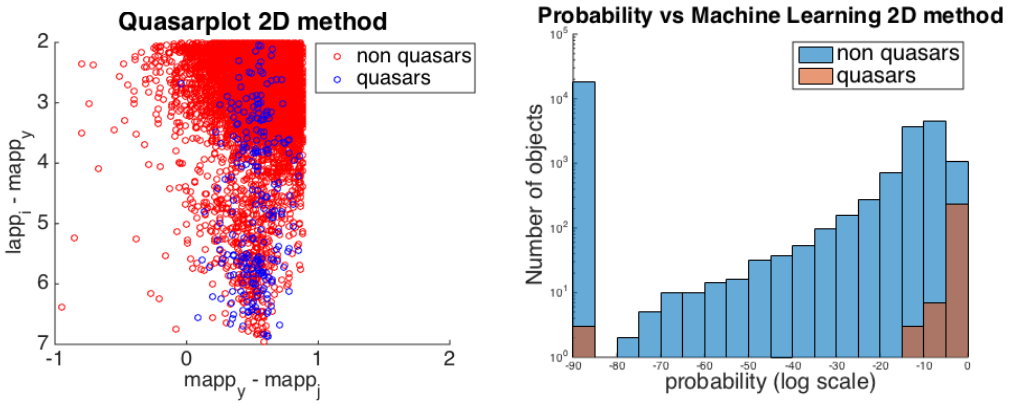


Figure 23: *2D method with limit 0.13. A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4. The two colour plot on the right emphasises how few candidate HZQs result from the classification compared to the total population.*

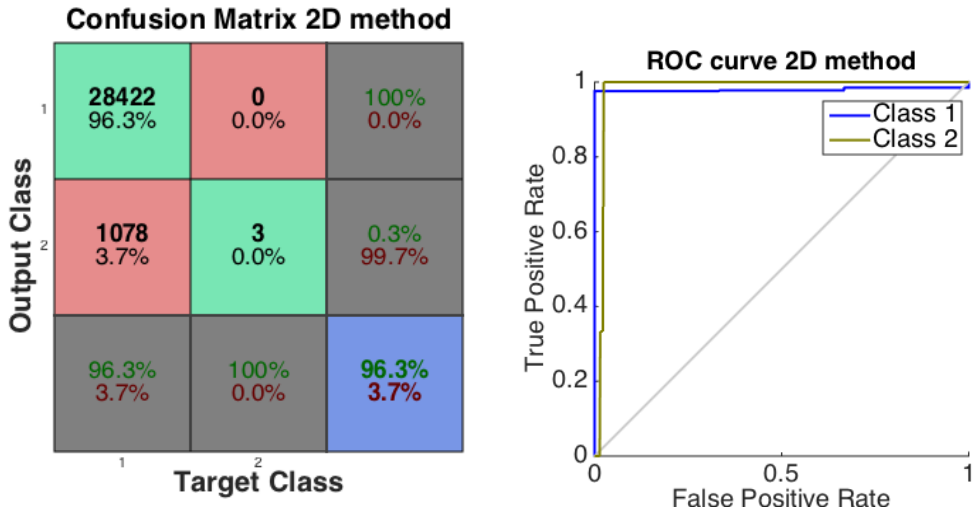


Figure 24: *2D method using the best network with limit 0.9. Completeness of 100% and classification rate of 96.3%. While the classification is still great, it proves how raising the limit to 0.9 induces many more false positives.*

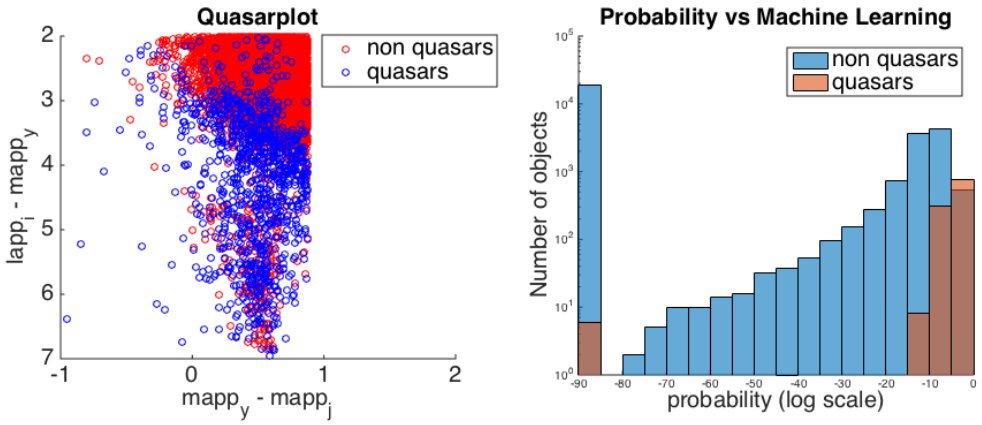


Figure 25: *2D method with limit 0.9. A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4. The two colour plot on the right emphasises how few candidate HZQs result from the classification compared to the total population.*

6.4 EP results

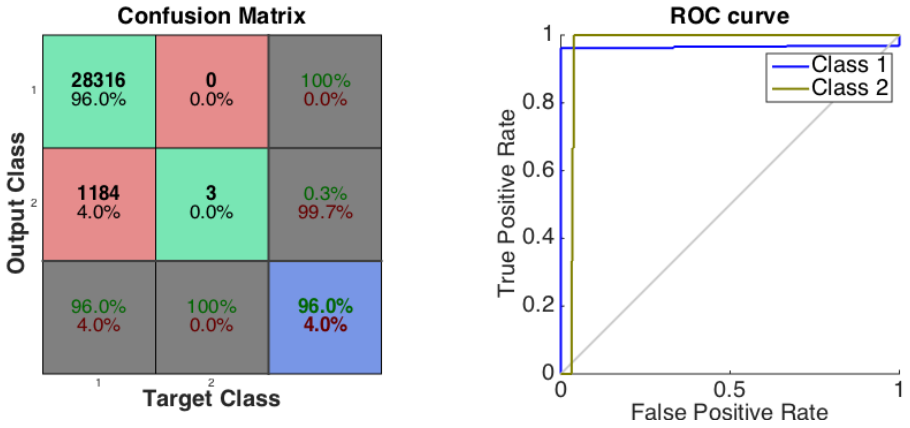


Figure 26: *Parameters: regularization of 0.27, 700 simulated quasars and 2000 candidates in each training set; 150 different training sets. Completeness 100% and classification rate 96.0%. The 3 HZQs are classified correctly and are assigned a probability of 0.6400, 0.8600 and 0.9267.*

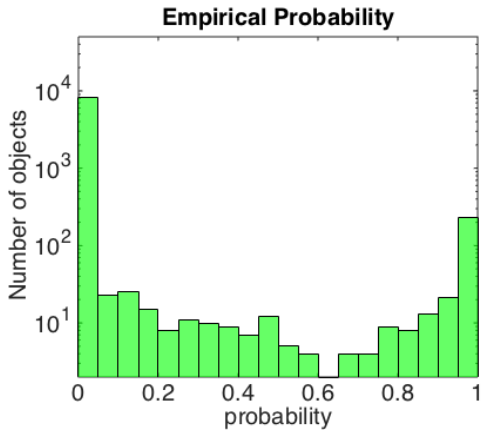


Figure 27: *Histogram of the probabilities calculated with the EP method. The majority of objects are always classified either as HZQs or non-HZQs, this proves that the different neural networks tend to have fairly consistent classifications.*

6.5 Robustness of neural networks

Other experiments executed to try and improve the classification consisted in applying transformations to the data.
The first attempt was using "colours", where a "colour" is defined as the difference between two measurements made at different wavelengths. The principle was to emphasize the difference between HZQs and non-HZQs by relying on the "preselection cut" plot in Figure 14 which has the $(i-Y)$ and $(Y-J)$ colours on the axes. The data was converted as $(i-Y)$, $(Y-J)$, $(z-Y)$, and $(i-z)$. The same tests as before were performed using this new format.
The second idea was to transform the data from magnitudes to fluxes, which were the original units in which the data was obtained. This was done through the relationship: $m = -2.5\log_{10}(\frac{F}{F_{ref}})$ where $F_{ref} = 3631JY$ and $1Jansky = 1Jy = 10^{-26}WHz^{-1}m^{-2}$ - (Lupton et al. 1999), (Fukugita et al. 1996).

The final result in both cases was that the classification was extremely similar to the original one, proving that NNs are robust and flexible tools.

7 Conclusions

NN methods have been implemented in the quest for HZQs with $z \gtrsim 6$. Different options have been finalised. Starting from the Matlab Neural Network Toolbox, iterative computational strategies have been designed, by averaging (AV method) or by inclusion (IC method). The latter was specifically designed to protected against False Negatives, an unacceptable loss due to the rarity of the HZQs.

The 2D mapping approach was found to be the most effective. Each candidate was converted from a point-like object into a 2D figure through Gaussian distribution which has the original point as the mean and the measurement error as the variance. This permitted either better rejection of false targets or a better focus of real HZQs and resulted into the highest obtainable performance. Increasing the number of computational cycles offered the opportunity to complement the classification by creating for each object a probability of being an HZQ. This was defined the EP method.

A wise usage of the measurement error further enhanced the classification performance. The training phase was in fact optimised by giving to the input a weight inversely proportional to its measurement error.

The code architecture meaningful enough to achieve the desired performance required different levels of iterations. Starting from a single NN cycle (level 1), in order to obtain sufficiently stable classifications on AV, IC and 2D a minimum of 10 NN cycles were required (level 2). When instead in addition to the classification an associated probability was pursued, the stability of the results imposed a minimum of a few hundreds iterations (level 3). Fine tuning of the parameters was one of the major challenges throughout the optimisation of the algorithm made possible by the remarkable flexibility of the neural networks.

The final classification results included completeness of 100% (perfectly protecting against False Negatives). Classification rate in the range of 95-99% was excellent, significantly reducing telescope time. The efficiency remained low due to extremely limited number of the target HZQs.

Overall, the NN performance was excellent and comparable with the Bayesian inference, offering robustness and additional flexibility. It is recommended

to consider modifying the NN training algorithm to include the error data. Neural networks are an ideal way to quickly and effectively analyse big quantities of data and will be necessary for the coming generation of astronomical surveys, such as the Large Synoptic Survey Telescope (Ivezic et al. 2008).

References

Abraham, S. et al. (2012), 'A photometric catalogue of quasars and other point sources in the sloan digital sky survey', *Monthly Notices of the Royal Astronomical Society* **419**(1), 80–94.

Budding, E. & Demircan, O. (2007), *Introduction to Astronomical Photometry*, 2nd edn, Cambridge University Press, Cambridge, New York.

Carballo, R. et al. (2004), 'Selection of quasar candidates from combined radio and optical surveys using neural networks', *Monthly Notices of the Royal Astronomical Society* **353**(1), 211–220.

Carballo, R. et al. (2008), 'Use of neural networks for the identification of new $z \geq 3.6$ QSOs from FIRST-SDSS DR5', *Monthly Notices of the Royal Astronomical Society.* **391**, 369.

Carlin, B. & Louis, T. (2000), *Bayes and Empirical Bayes Methods for Data Analysis, Second Edition*, Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis.

Czarnecki, W. M. & Podolak, I. T. (2013), Machine Learning with Known Input Data Uncertainty Measure, *in* '*Computer Information Systems and Industrial Management', r Heidelberg, pp. 379–388.*

Fan, X. et al. (2006), 'Constraining the Evolution of the Ionizing Background and the Epoch of Reionization with $z \sim 6$ Quasars. II. A Sample of 19 Quasars', The Astronomical Journal **132***(1), 117.*

Fawcett, T. (2006), 'An Introduction to ROC Analysis', Pattern Recogn. Lett. **27***(8), 861–874.*

Fukugita, M. et al. (1996), 'The sloan digital sky survey photometric system', The Astronomic Journal **111***, 1748.*

Graff, P. et al. (2013), 'SKYNET: an efficient and robust neural network training tool for machine learning in astronomy.', Monthly Notices of the Royal Astronomical Society .

Hewett, P. & Wild, V. (2010), 'Improved redshifts for SDSS quasar spectra', Monthly Notices of the Royal Astronomical Society **405**, *2302–2316.*

Hewett, P. et al. (2006), 'The UKIRT Infrared Deep Sky Survey ZY JHK photometric system: passbands and synthetic colours', Monthly Notices of the Royal Astronomical Society **367**, *454–468.*

Ivezic, Z. et al. (2008), 'Lsst: from science drivers to reference design and anticipated data products'.

Lawrence, A. et al. (2007), 'The UKIRT Infrared Deep Sky Survey (UKIDSS)', Monthly Notices of the Royal Astronomical Society **379**, *1599–1617.*

Lupton, R. et al. (1999), 'A modified magnitude system that produces well-behaved magnitudes, colors, and errors even for low signal-to-noise ratio measurements', The Astronomic Journal **118**, *1406.*

Maja Pantic, S. P. (2015), 'Course 395: Machine learning'. Lecture Notes.

Mitchell, T. M. (1997), Machine Learning, *1st edn, McGraw-Hill, Inc., New York, NY, USA.*

Mortlock, D. (2014), 'Finding the Most Distant Quasars Using Bayesian Selection Methods', Statistical Science **29***(1), 50–57.*

Mortlock, D. et al. (2011), 'A luminous quasar at a redshift of z = 7.085', Nature **474***(7353), 616–619.*

Mortlock, D. et al. (2012), 'Probabilistic selection of high-redshift quasars', Monthly Notices of the Royal Astronomical Society **419**, *390–410.*

Neal, R. (2014), 'A three-way classification problem'.
***URL:** http://www.cs.toronto.edu/ radford/fbm.2004-11-10.doc/Ex-netgp-c.html*

Peacock, J. (1998), Cosmological Physics, *Cambridge University Press.*

Reed, R. et al. (1992), Regularization using jittered training data, in *'Neural Networks, 1992. IJCNN., International Joint Conference on', Vol. 3, pp. 147–152.*

Rees, M. J. (1984), 'Black Hole Models for Active Galactic Nuclei', Annual Review of Astronomy and Astrophysics **22***, 471–506.*

Schmidt, M. (1963), '3c 273 : A star-like object with large red-shift', Nature **197***, 1040.*

Shiffman, D. (2012), The Nature of Code*, D. Shiffman.*
***URL:** http://natureofcode.com*

Sinha, R. P. et al. (2006), Photometric identification of quasars from the sloan survey, in *'Highlights of Astronomy', Vol. 2 of* Proceedings of the International Astronomical Union*, pp. 609–609.*

Tagliaferri, R. et al. (2003), 'Neural networks in astronomy', Neural Networks **16***(3-4), 297–319.*

Willott, C. J. et al. (2010), 'The Canada-France High-z Quasar Survey: Nine New Quasars and the Luminosity Function at Redshift 6', The Astronomical Journal **139***(3), 906 – 918.*

York, D. et al. (2000), 'The Sloan Digital Sky Survey: Technical summary', The Astronomical Journal **120***(3), 1579–1587.*

Appendix A. Customised NN algorithm by error incorporation

As mentioned in section 5, the paper (Czarnecki & Podolak 2013) introduces a very interesting way of using noise during training. While this method was finally not implemented for this research, it was still taken into consideration. The following paragraph describes the use of the conjugate gradient backpropagation training to which this method is applied. The basic idea of the training is also what *trainscg* - scaled conjugate gradient backpropagation is based on.

Backpropagation Section 2.1 explains the basic layout of a network: inputs, outputs and hidden layers in between, and how each layer is connected to the other by weights. These weights are what is being optimised during training.
Specifically in the stochastic backpropagation algorithm:

1. The weights are randomly initialised.
2. The first example is propagated forward through the network.
3. The error (output - target value) is calculated.
4. This error is propagated backwards throughout the network.
5. Each weight is updated depending on how big the error is.
6. This procedure continues until a stopping condition is reached.

Stochastic gradient descent differs from gradient descent as it updates the weights while iterating one training example at a time, instead gradient descent updates the weights after running all the training examples. Gradient descent means that it uses the gradient of the error to find where its change is steepest and then it moves in that direction. Each time a training example goes through a network, a weight w is updated as: $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$, where $\Delta w_{ji} = -\eta \nabla E_d$. ∇E_d is the gradient of the error function which for the stochastic gradient descent algorithm is defined as:

$$E_d(w) = \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2 \tag{14}$$

where t_k is the target value of unit k and o_k is the output value of unit k given example d. There are two slightly different values of Δw_{ji} which depend on whether the units being considered are considering are on the output layer or one of the hidden layers.
For the hidden layers we get that:

$$\Delta w_{ji} = \eta(t_j - o_j)o_j(1 - o_j)x_{ji} \tag{15}$$

where x_{ji} is the ith output to unit j.
Instead for the output layers the result is:

$$\Delta w_{ji} = \eta o_j(1 - o_j) \sum_{k \in \text{Downstream}(j)} \frac{\partial E_d}{\partial net_k} w_{kj} \tag{16}$$

Where $k \in$ Downstream(j) means all the units which take as input the x coming from unit j, so all the units in the layer after j. ∂net_k is the weighted sum of inputs for unit k: $\sum_i w_{kj}x_{kj}$. (Mitchell 1997)

Error in backpropagation The proposal of paper (Czarnecki & Podolak 2013) was to modify the gradient descent backpropagation algorithm by introducing the modified error function:

$$E_d(w) = \sum_{k \in outputs} (t_k P(k|\hat{x}_d) - o_k)^2 \tag{17}$$

where $P(k|x_d)$ represents the probability that example $\hat{x}_d$ is a noisy exemplification of example x_d: $P[\hat{x}_d = x_d + n | i \in 1, ..., M]$.

Appendix B. Results with new candidates

The results presented in this section use the same networks as section 6 applied to a new candidates set with 8562 data points and 5 HZQs taken from the same survey described in section 3.1, but with a different data release. As a reference point, the Bayesian inference method had recognised 234 objects as possible HZQs (and correctly classified the 5 HZQs).
Considering the classification rate (completeness is always 100%), the best classification results are given by:

Method	Efficiency	Classification rate
Bayesian	0.4%	97.2%
2D (0.6)	0.3%	96.7%
EP	0.3%	96.6%
AV	0.2%	96.4%
IC	0.2%	95.4%

Table 3: *Classification results for the new data set*

B.1 AV

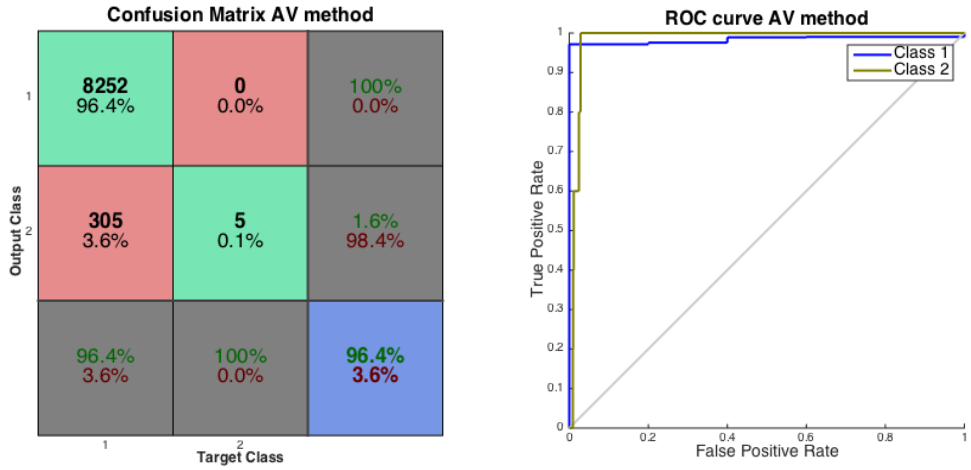


Figure 28: *Completeness of 100% and classification rate of 96.4%.*

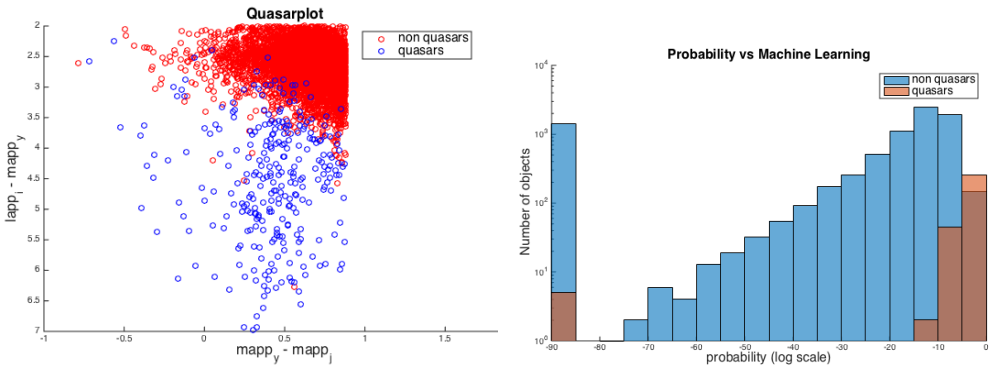


Figure 29: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4.*

B.2 IC

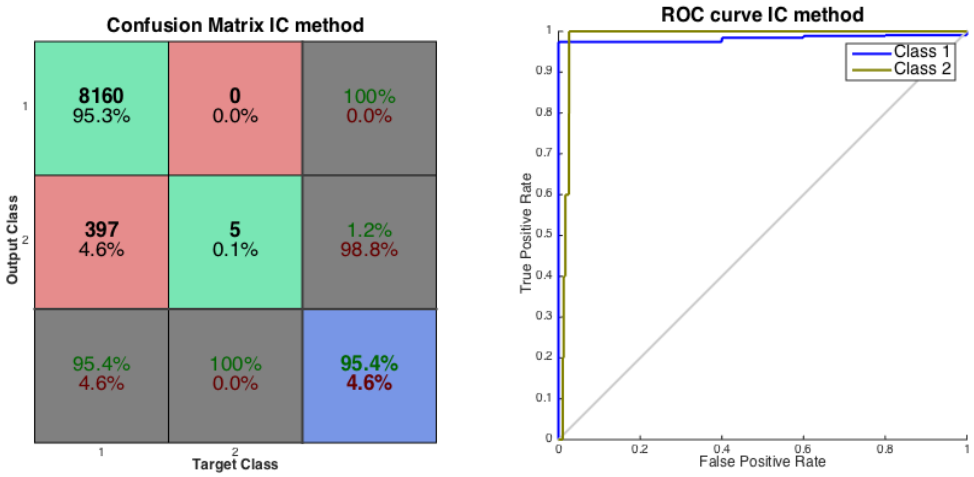


Figure 30: *Completeness of 100% and classification rate of 95.4%*

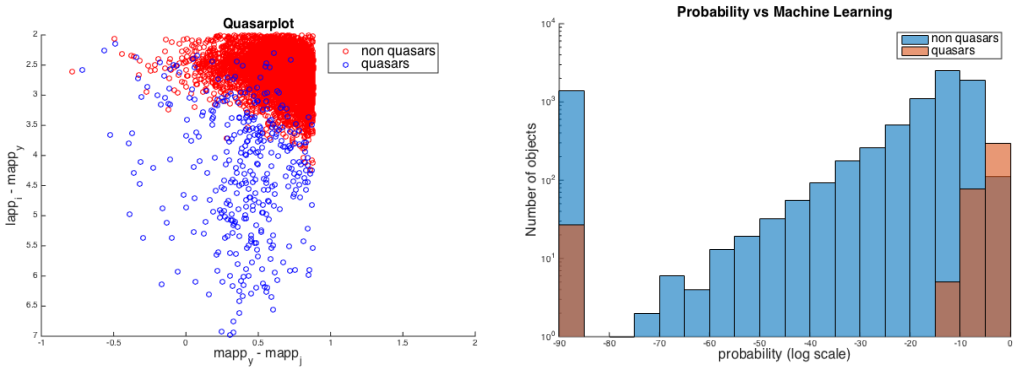


Figure 31: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4.*

B.3 2D

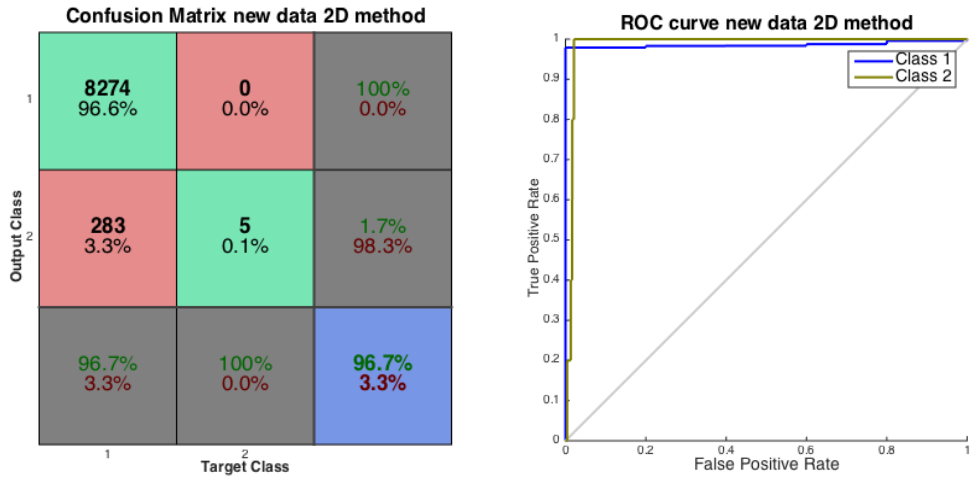


Figure 32: *2D method using the best network - with limit 0.6. Completeness of 100% and classification rate of 96.7%.*

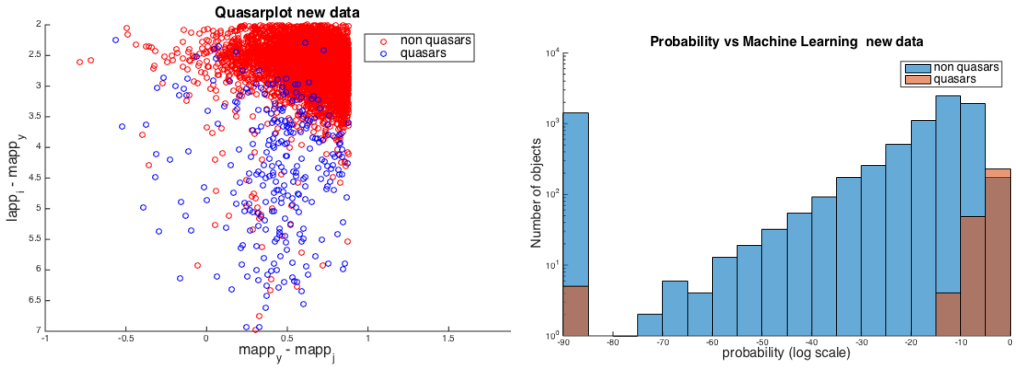


Figure 33: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4.*

B.4 EP

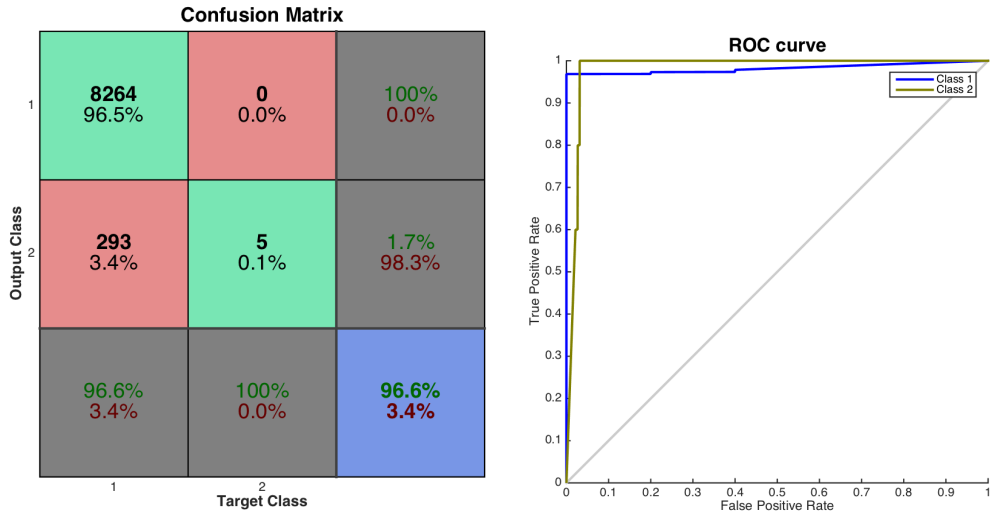


Figure 34: *Completeness of 100% and classification rate of 96,6%. Parameters: regularization of 0.27, 700 simulated quasars and 2000 candidates in each training set; 150 different training sets.*

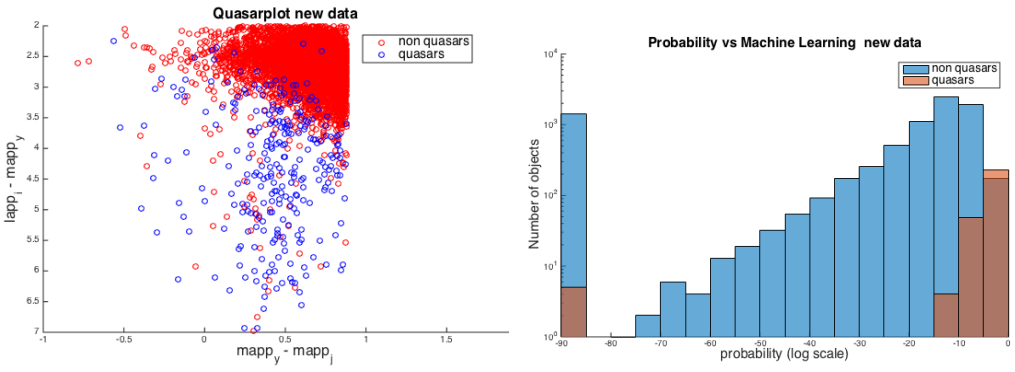


Figure 35: *A detailed explanation of the left figure can be found in section 2.2.3 and of the right figure in section 2.2.4.*

Appendix C. Simulated quasars

The results in this section consist of using the same networks as section 6, but adding to the test set which is classified 502 simulated quasars chosen randomly. As a reference point, the Bayesian inference method recognised 295 of the simulated quasars. Ranked in order of highest completeness we have:

Method	Efficiency	Classification rate	Completeness
IC	24.3%	94.7%	95.0%
AV	27.7%	95.8%	91.9%
EP	29.1 %	95.9%	91.5%
2D(0.9)	29.8%	96.3%	87.7%
Bayesian	40%	96.7%	58.8 %
2D (0.13)	43.6%	98.1%	36.8%

Table 4: *Classification results with extra simulated quasars (in order of highest completeness).*

When adding the simulated quasars, the methods which do not consider the measurement error are superior as they are able to obtain a higher completeness. It might be necessary to verify how the noise for the simulated quasars is created.

C.1 AV

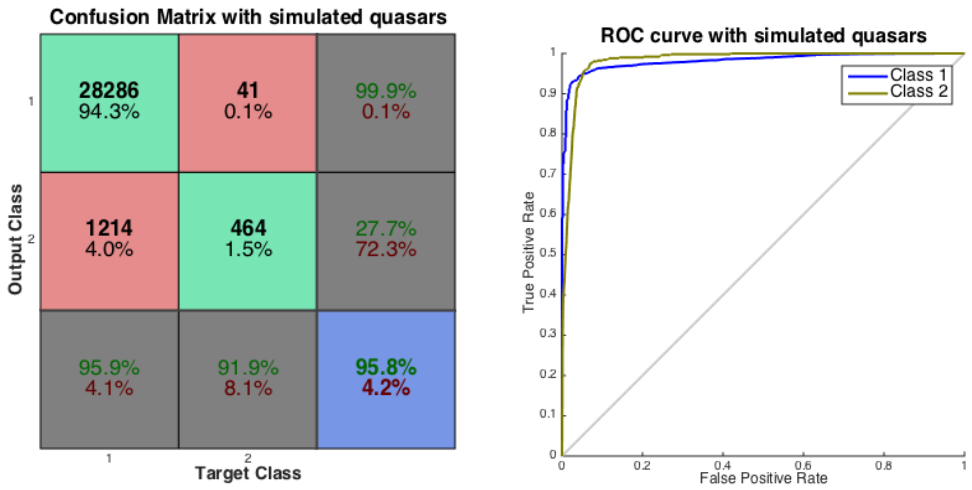


Figure 36: *461 of the 502 simulated quasars are recognised.*

C.2 IC

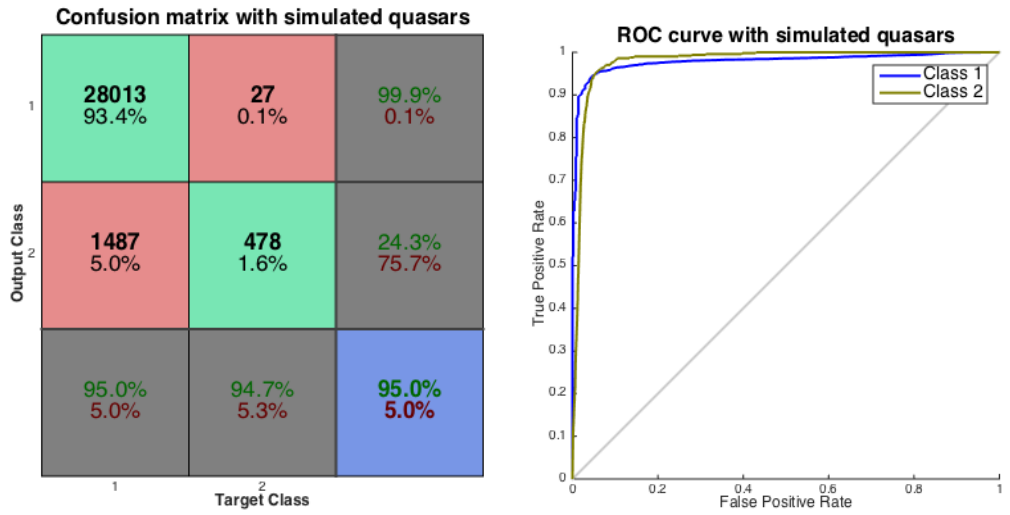


Figure 37: *475 of the 502 simulated quasars are recognised. This result shows the true power of this method. Only 27 simulated HZQs were erroneously classified as non-HZQs (false negatives) compared to the 41 before (and the 207 of the Bayesian Inference)*

C.3 2D

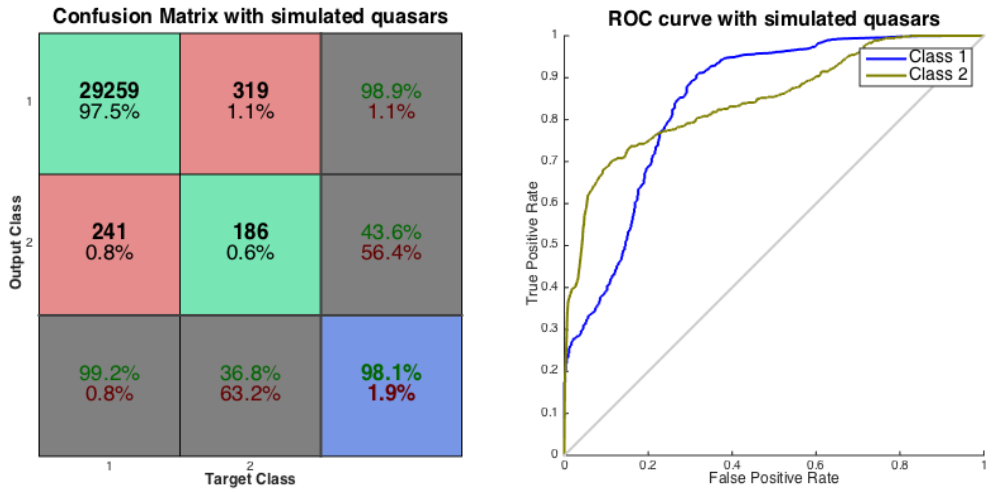


Figure 38: *2D method with limit 0.13; 183 of the 502 simulated quasars are recognised.*

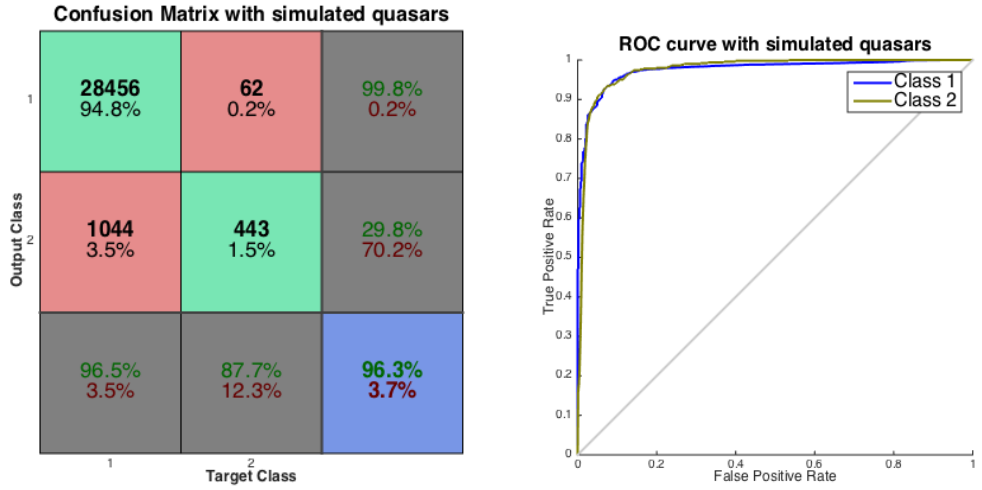


Figure 39: *2D method with limit 0.9; 440 of the 502 simulated quasars are recognised.*

C.4 EP

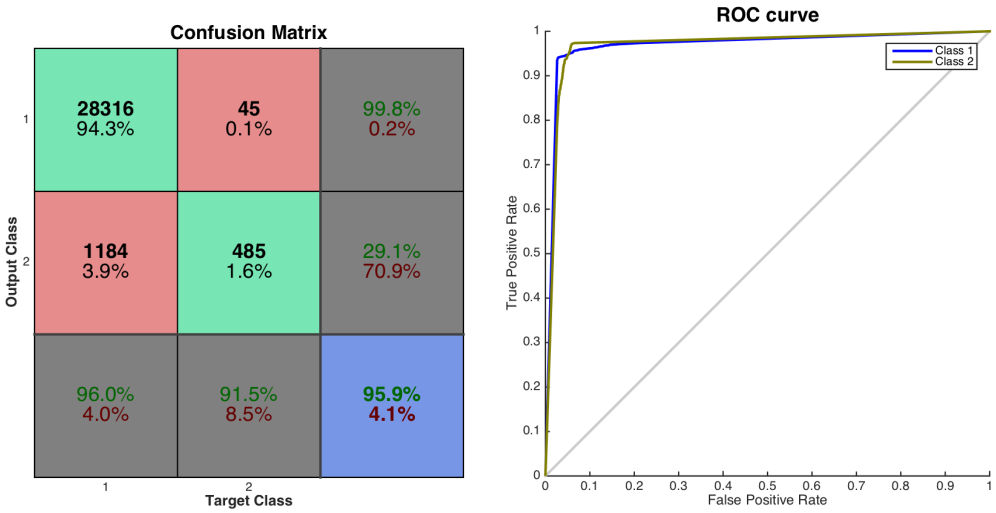


Figure 40: *482 of the 527 simulated quasars are recognised.*

Appendix D. Mixed methods

The graphs below are the experiment of mixing various classification methods together. The training parameters used are the same ones as before:

- Regularization of 0.27.
- 700 simulated quasars and 2000 candidates in each training set.
- 10 different networks.

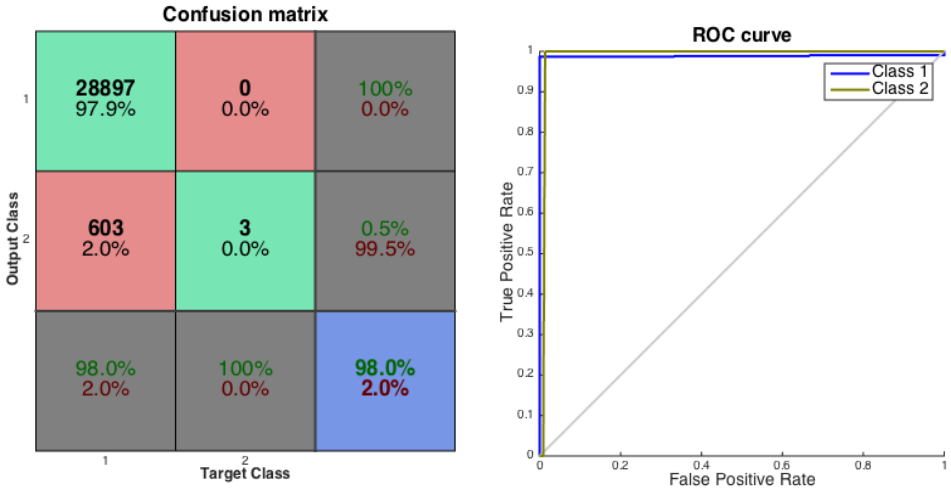


Figure 41: *Confusion matrix and ROC curve for plot of the 2D method mixed with IC method: the point was first expanded, then classified. All the points classified as quasars at least once by the different networks were selected. Finally, the extra points were averaged to reduce them back to one point.*

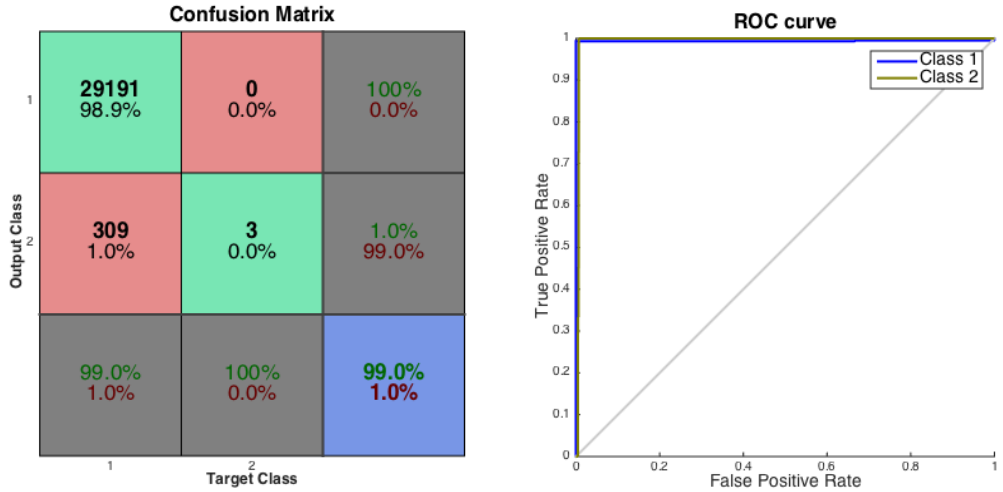


Figure 42: *Confusion matrix and ROC curve for plot of the 2D method mixed with IC method: this time the 2D method was done first, then the inclusion of all the HZQs was taken.*

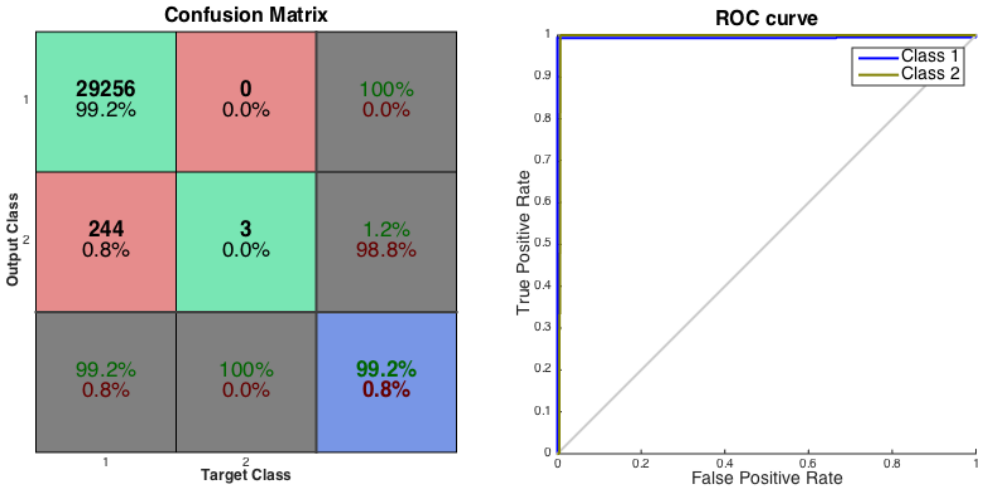


Figure 43: *Confusion matrix and ROC curve for plot of 2D method mixed with AV method: the 2D method was done first, then the average of the results was taken.*