Fink: Early supernovae Ia classification using active learning

M. Leoni¹, E. E. O. Ishida²*, J. Peloton¹, and A. Möller², ³

¹ Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France
e-mail: leoni@1al.in2p3.fr
² LPC, Université Clermont Auvergne, CNRS/IN2P3, 63000 Clermont-Ferrand, France
e-mail: emille.ishida@clermont.in2p3.fr
³ Centre for Astrophysics and Supercomputing, Swinburne University of Technology, Mail Number H29, PO Box 218, 3122 Hawthorn, VIC, Australia

Received 22 November 2021 / Accepted 14 April 2022

ABSTRACT

Context. The Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST) will produce a continuous stream of alerts made of varying sources in the sky. This data flow will be publicly advertised and distributed to scientists via broker systems such as FINK, whose task is to extract scientific information from the stream. Given the complexity and volume of the data to be generated, LSST is a prime target for machine learning (ML) techniques. One of the most challenging stages of this task is the construction of appropriate training samples which enable learning based on a limited number of spectroscopically confirmed objects.

Aims. We describe how the FINK broker early supernova Ia (SN Ia) classifier optimizes its ML classifications by employing an active learning (AL) strategy. We demonstrate the feasibility of implementing such strategies in the current Zwicky Transient Facility (ZTF) public alert data stream.

Methods. We compared the performance of two AL strategies: uncertainty sampling and random sampling. Our pipeline consists of three stages: feature extraction, classification, and learning strategy. Starting from an initial sample of ten alerts, including five SNe Ia and five non-Ia, we let the algorithm identify which alert should be added to the training sample. The system was allowed to evolve through 300 iterations.

Results. Our data set consists of 23 840 alerts from ZTF with a confirmed classification via a crossmatch with the SIMBAD database and the Transient Name Server (TNS), 1600 of which were SNe Ia (1021 unique objects). After the learning cycle was completed, the data configuration consisted of 310 alerts for training and 23 530 for testing. Averaging over 100 realizations, the classifier achieved ~89% purity and ~54% efficiency. From 01 November 2020 to 31 October 2021 FINK applied its early SN Ia module to the ZTF stream and communicated promising SN Ia candidates to the TNS. From the 535 spectroscopically classified FINK candidates, 459 (86%) were proven to be SNe Ia.

Conclusions. Our results confirm the effectiveness of AL strategies for guiding the construction of optimal training samples for astronomical classifiers. It demonstrates in real data that the performance of learning algorithms can be highly improved without the need of extra computational resources or overwhelmingly large training samples. This is, to our knowledge, the first application of AL to real alert data.

Key words. methods: data analysis – supernovae: general – methods: statistical

1. Introduction

Once large-scale photometric astronomical surveys successfully overcome the challenge of data acquisition, the next obstacle in the path for scientific exploration is to identify interesting candidates. This is especially important for researchers working on transient astronomy, where there is a window of opportunity when a potentially interesting target can be spectroscopically followed before becoming too dim to allow useful scrutiny. Depending on the specific astronomical target under investigation, this window can be as short as a few minutes.

Current surveys, such as the Zwicky Transient Facility¹ (ZTF), publicly advertise detected candidates through an alert data stream distributed to community brokers, whose task is to filter and process as well as to redistribute interesting candidates with added values to the astronomical community. A similar protocol will also be adopted by the upcoming Vera C. Rubin Observatory Legacy Survey of Space and Time² (LSST).

However, due to the large data volume expected from LSST, only seven full copies of the stream will be distributed. The recently selected LSST brokers are ALERCE ( Förster et al. 2021), AMPLE ( Nordin et al. 2019), BABAMUL, ANTARES ( Narayan et al. 2018), FINK³ ( Möller et al. 2021), LASAIR ( Smith et al. 2019), and Pitt-Google.

The FINK broker is being developed by an international community of researchers with a large variety of scientific interests, including multimessenger astronomy, supernovae (SNe), solar system, anomaly detection, micro-lensing, gamma-ray bursts’ optical counterparts, among others. It is currently processing and classifying alerts from ZTF as a test bed for Rubin operations. In this work, we give a detailed description of the early SN Ia classifier currently operating within FINK.

The module uses an active learning (AL) strategy to select an optimum set of alerts which is subsequently used as a training sample for a random forest classifier. The basic idea of AL is to allow the algorithm to identify which alerts should be labeled

¹ https://www.ztf.caltech.edu/
² https://www.lsst.org/
³ https://fink-broker.org
and included in the training set, according to their potential at improving the classifier. This philosophy is at the core of many of the recommendation systems currently in place, from pharmaceutical studies (Spjuth et al. 2021) to urban planning (Abernethy et al. 2018). In astronomy, it has proven to be effective in a few different scenarios (e.g., Solorio et al. 2005; Richards et al. 2012; Vilalta et al. 2017; Walmsey et al. 2020), including the classification of astronomical transients (e.g., Ishida et al. 2019; Kemamer et al. 2020) and anomaly detection (Ishida et al. 2021).

The remaining sections of this paper are organized as follows: Sect. 2 describes the construction of our data set. Section 3 describes all of the stages of our methodology, including filtering (Sect. 3.1), feature extraction (Sect. 3.2), classifier (Sect. 3.3), learning strategy (Sect. 3.4), and evaluation metrics (Sect. 3.5). We show our results in Sect. 4 and discuss their implications and future plans for FINK in Sect. 6.

2. Data

Our starting point was a data set containing all public alerts from the ZTF survey from November 2019 to March 2020 with confirmed classifications available through the SIMBAD catalog (Wenger et al. 2000) or Transient Name Server4 (TNS) database. TNS classified objects are spectroscopically confirmed while SIMBAD contains highly curated labels from both spectroscopic and photometric surveys (Loup et al. 2019), with the majority of its classifications taking spectroscopic redshift measurements into account (CDS, private communication). Each alert contains the last 30 days of history of photometric measurements for a given point in the sky. From the SIMBAD data set, this amounts to 5 004 378 alerts corresponding to 1 313 371 objects, with dominant classes being RRLyr (≈26%), eclipsing binaries (≈22%), and stars (≈19%). From the TNS data, there were 11 422 alerts corresponding to 11 329 objects, dominated by SNIa (52%) and SNII (19%).

From the TNS set, all alerts were used. Our experiments showed that including around five or six times the same number of alerts with a SIMBAD classification was enough to construct an informative training sample. Using a larger number of SIMBAD sources does not affect the main results of a binary classification study as the one presented here. In order to be conservative while not overusing computer resources, we randomly selected 75 000 alerts, corresponding to 41 076 objects5. In what follows, this combined set is referred to as raw data.

We used the photometric information contained in alerts, regarding ZTF broad-band filters [g, r, i], retrieved through the FINK API6. According to the FINK schema, these correspond to the alert id (candid), the time of observation in Julian dates (jd), the magnitude from PSF-fit photometry (magpsf), the error in magnitude (sigmapsf), and the ZTF filter (fid) as well as their TNS (TNS) and SIMBAD (cdsxmatch) associated classes. In order to avoid classification errors between the Ia and non-Ia classes, we did not use SN from Simbad. Thus all SN used in this work are spectroscopically classified objects from TNS. The full raw data set consisted of 86 422 alerts, corresponding to 52 405 unique sources.

Table 1 shows the population of classes within this data set. Alerts possessing a SIMBAD crossmatch were divided into three categories (star, multiple_objects6, and AGN-like) following SIMBAD’s classification scheme7,8,9. Those associated with SNe (not Ia) were allocated in the other_SN category, while all SN Ia are reported under the SNIa class (it comprises SNIa, SNe Ia-91T-like, SNe Ia-91bg-like, SNIa-CSM, SNIa-pec, and SNIax/02cx-like). Additional transients whose classification was provided via TNS were grouped under the other_TNS class, including cataclysmic variable stars, TDEs, among others. We made this raw data set publicly available, along with the necessary code to reproduce the results in this paper, through Zenodo10.

3. Methodology

We adapted the AL approach presented in Ishida et al. (2019) to optimize the construction of an effective training sample. Starting from their basic template pipeline of feature extraction, classification, and learning strategies, we performed a few adaptations to accommodate the higher complexity intrinsic to real data sets.

Since we are mostly interested in early classification (identifying SNIa before they reach maximum brightness), we began by selecting only epochs corresponding to the rising part of the light curve (Sect. 3.1). Subsequently, we developed a feature extraction method based on a sigmoid function (Sect. 3.2) which requires at least three observed epochs per filter, which is less than the minimum of five observations needed to fit the parametric function used in Ishida et al. (2019). The numbers and percentages of each subclass surviving this preprocessing are shown in Table 1 and Fig. 1.

3.1. Filtering

In order to isolate the alerts depicting light curves on the rise, we treated each filter separately. An exponentially weighted moving average (EWA) was used as a parametric function in Ishida et al. (2019). The numbers and percentages of each subclass surviving this preprocessing are shown in Table 1 and Fig. 1.

Table 1. Population types in our alert data set before and after feature extraction.

<table>
<thead>
<tr>
<th>Class</th>
<th>In raw data</th>
<th>After feature extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGN-like</td>
<td>6677 (7.7)</td>
<td>2961 (12.5)</td>
</tr>
<tr>
<td>SNIa</td>
<td>6269 (7.3)</td>
<td>1600 (6.7)</td>
</tr>
<tr>
<td>other_SN</td>
<td>3891 (4.5)</td>
<td>762 (3.2)</td>
</tr>
<tr>
<td>multiple_object</td>
<td>17 083 (19.8)</td>
<td>4419 (18.6)</td>
</tr>
<tr>
<td>other_TNS</td>
<td>964 (1.1)</td>
<td>174 (0.7)</td>
</tr>
<tr>
<td>star</td>
<td>51 538 (59.6)</td>
<td>13 859 (58.3)</td>
</tr>
<tr>
<td>Total</td>
<td>86 422</td>
<td>23 775</td>
</tr>
</tbody>
</table>

Notes. Values in parentheses indicate percentage of total sample. We note that the same object on the sky can emit several alerts. As a reference, the total number of objects in the raw data (after feature extraction) is 52 405 (15 751), while the total number of objects for SNIa in the raw data (after feature extraction) is 2287 (1021).

4 https://www.wis-tns.org/
5 The same object can emit several alerts over time, and each alert carries the photometric history over the last 30 days, thus allowing light curve analysis.
6 The number of SIMBAD objects to be used is a parameter input in our code and can be easily changed to allow further scrutiny, e.g., for users interested in multiclass experiments.
7 https://fink-portal.org/api
8 Multiple objects include eclipsing binaries, cataclysmic variables, and novae.
9 http://simbad.u-strasbg.fr/simbad/sim-display?data=otypes
10 We use the term AGN–like for all crossmatches related to AGNs: quasi-stellar objects (QSO), blazars, etc.
11 https://doi.org/10.5281/zenodo.5645609
average with a window of four points was performed, and its derivative (i.e., forward finite difference) was calculated. Epochs corresponding to a negative derivative were discarded. This process ensures that at least three points are on the rise, as one point will serve as the comparison baseline for the finite difference.

This procedure is successful in masking out the declining part of the light curve for transient events; however, it can fail to grasp details of periodic or oscillatory light curves (examples of outcomes from this filtering process are shown in Fig. 2 and the corresponding best-fit parameter values are shown in Appendix A). Nevertheless, it proved to be successful in filtering enough information to allow SN-like events to populate a specific area of the parameter space when combined with the feature extraction procedure described in Sect. 3.2.

3.2. Feature extraction

All surviving epochs were converted from ZTF magnitudes $m$ and magnitude error $\Delta m$ into SNANA\(^{12}\) (Kessler et al. 2009) flux units, $f$ and its corresponding error $\Delta f$ derived by the chain rule,

$$ f = 10^{\frac{m - 11}{5}}; $$

$$ df = 4 \frac{\ln(10)}{10} \Delta m \Delta f $$

$$ = 10^{10} \alpha \Delta m \exp\left(-\frac{am}{10}\right), $$

with $\alpha = 4 \times \ln 10 \sim 9.21034$. Subsequently, observations in each filter were independently submitted to a sigmoid fit $S$ at a given time $t_i$,

$$ S(t_i) = \frac{c}{1 + e^{-(\Delta t - b_i)}}, $$

where $\Delta t_i = t_i - \min(t)$ is the observation time of the $i$th data point since first detection. Employing a least square minimization routine and initialization values $a_0$, $b_0$, and $c_0$ defined by

$$ a_0 = \max(f) - \min(f), $$

$$ b_0 = \frac{1}{a_0} \log\left[\frac{c_0}{\min(f) - 1}\right], $$

$$ c_0 = \max(f), $$

where $N$ is the total number of points surviving the filtering for this alert and $t_{\text{max}}$ is the time of maximum flux for a given light curve, we obtained the three features for each alert and band from the best fit values of $a$, $b$, and $c$. The following three other features were extracted per band:

(i) the quality of the fit, represented by

$$ \text{MSE} = \sum_{i=1}^{N} \frac{(S_i - \bar{f}_i)^2}{S_i}, $$

where $S_i$ is the flux estimate for the $i$th epoch using the best fit values for $a$, $b$, and $c$, and tilde quantities ($\bar{S}_i$ and $\bar{f}_i$) generalized as $\tilde{X}_i$ are normalized values of $X_i$ defined by

$$ \tilde{X}_i = \frac{X_i}{\sum_j X_j}; $$

(ii) the mean signal-to-noise ratio ($S/N$),

$$ S/N = \frac{1}{N} \sum_{i=1}^{N} f_i / \Delta f_i; $$

(iii) the total number of epochs used in the fit, $N$.

In summary, for each alert we have a total of six parameters, $[a, b, c, \chi^2, S/R, N]$, per band. The input matrix was constructed by concatenating the parameters corresponding to $[g, r]$ bands for the same alert in one line.

Figure 3 shows the two-dimensional representation of objects in this parameter space. Our final input matrix is composed of 23 775 lines (alerts) and 12 columns (features); this corresponds to 15 751 unique sky objects. The composition of this alert sample in classes is shown in Table 1. From this table, we noticed that although the feature extraction reduced the data volume considerably, it maintained the overall proportion between classes present in the original raw data.

3.3. Classifier

Following the framework outlined in Ishida et al. (2019), we used a random forest classifier (Breiman 2001). This is an ensemble method which uses a number of decision trees (Breiman et al. 1984), constructed from different subsamples of the training set. Once the forest is constructed, the estimated classes for the target sample are determined by majority voting considering all trees in the ensemble. The method proved to be effective in a variety of data scenarios in astronomy (e.g., Richards et al. 2012; Ishida & de Souza 2013; Lochner et al. 2016; Möller et al. 2016; Calderon & Berlind 2019; Nixon & Madhusudhan 2020; Kuhn et al. 2021; Nakazono et al. 2021).

Random forest has a number of advantages when faced with complex data sets. Since it is based on decision trees, the trained model is fully interpretable, which allows each decision to be scrutinized. Moreover, for the specific purpose of this project, its most important quality is the sensitivity to small changes in the training set. Decision trees divide the parameter space into small regions around each object in the training sample. Thus, in the small training data regime, it quickly adapts classification results when faced with a small number of new labels. This is a crucial feature for any classifier which needs to work within an AL framework (Sect. 3.4). All random forest models trained in this work were constructed using 1000 trees, so as to ensure good convergence in a reasonable time.
Fig. 2. Alert data and sigmoid fit for alerts associated with a SNe Ia (objectId:ZTF20acqjmzk, candid: 1424192755315015009), a galaxy (AGN-like) (objectId:ZTF18aarybyq, candid: 1452439215115010000), an eclipsing binary (objectId:ZTF18aabtvle, candid: 127724716915010000), and a RRLyr star (objectId:ZTF18aaptrep, candid: 1385238050715015010). Points and error bars denote the observed values. Dashed lines show approximations obtained from the sigmoid fit (Sect. 3.2). The corresponding best-fit parameters are given in Appendix A.

Fig. 3. Distribution in feature space according to true alert classes. Each panel shows the same parameter in g and r bands (horizontal and vertical axis, respectively). The blue dots represent alerts from SN Ia sources, while the orange triangles represent all the other classes in our initial data set. We can observe that non-Ias occupy an overlapping and larger region of the parameter space than their Ia counterparts.

### 3.4. Active learning

Active learning (Settles 2012) is a branch of machine learning methods whose goal is to optimize classification results when labeling is expensive and/or time-consuming. In its pool-based approach, at each iteration the algorithm identifies objects which can potentially increase the information content available in the training sample if labels were provided. This list is sent to an annotator (which can be a human or a machine) who is in charge of labeling the queried sample. Once labels are available, the queried sample is added to the training, the machine learning model is retrained and the process is repeated. The iterations continue until the labeling resources (budget) are exhausted.
In the context of astronomical transient classifications, we started with a small number of classified transients in available catalogs. Among all unlabeled objects, the algorithm subsequently identified which of them would be more informative to the model if a classification (label) was provided. Once this queried object was identified and its classification was obtained, we added the new photometry and label to the training sample and retrained the learning model. Each iteration in this process is also called a “loop.” We highlight that, although here this exercise was performed in catalog data, our intent during LSST production was to obtain labels through spectroscopic follow-up.

Active learning strategies differ as to how the most informative objects are chosen (query strategy). In this work, we use uncertainty sampling (US, Sharma & Bilgic 2017), which identifies the objects closest to the decision boundary between classes at each iteration. We are focused on correctly identifying SNe Ia, thus, our experiment is designed as a binary classification problem (Ia versus non-Ia). At each iteration, the algorithm queries the object with the highest entropy (for a binary classifier, this corresponds to the object with a classification probability closest to 0.5. See, e.g., MacKay 2003).

We also performed experiments using a random sampling (RS) approach, where at each iteration objects were randomly selected from the target and added to the training sample. This allows us to better understand how much influence the decision process has on the final classification results.

### 3.5. Metrics

We evaluated our results using the classification metrics and nomenclature from the Supernova Photometric Classification Challenge (Kessler et al. 2010),

\[
\text{accuracy} = \frac{C_{\text{Ia}} + C_{\text{non-Ia}}}{N},
\]

\[
\text{efficiency} = \frac{C_{\text{Ia}}}{N_{\text{Ia}}},
\]

\[
\text{purity} = \frac{C_{\text{Ia}}}{C_{\text{Ia}} + W_{\text{non-Ia}}}
\text{ and }
\]

\[
\text{fom} = \text{efficiency} \times \frac{C_{\text{Ia}}}{C_{\text{Ia}} + W^* \times W_{\text{non-Ia}}},
\]

where \text{fom} stands for figure of merit. \(C_{\text{Ia}}\) is the number of correctly classified IAs, \(C_{\text{non-Ia}}\) denotes the number of correctly classified non-IAs, \(N\) represents the total number of objects in the target sample, \(N_{\text{Ia}}\) is the total number of IAs in the target sample, \(W_{\text{non-Ia}}\) is the number of wrongly classified non-IAs, and \(W^* = 3\) (Kessler et al. 2010) is a weight which penalizes false positives.

We stress that for the purposes of this study, these metrics were used for diagnostic purposes only and they were not used in the decision-making process.

### 4. Results

Starting from the initial data matrix described in Sect. 3, we repeated the same experiment 100 times. In each realization, we randomly selected ten objects (five SNe Ia and five non-Ia), to be used as the initial training sample, and evolved the system through 300 iterations. The final configuration consists of 310 alerts in the training sample and 23,425 in the test sample. This small number of iterations was enough to significantly improve the classification results while, at the same time, ensuring that the overall statistical properties of the target sample were maintained throughout the learning process.\(^{13}\)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Random sampling</th>
<th>Uncertainty sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.95 ± 0.01</td>
<td>0.97 ± 0.01</td>
</tr>
<tr>
<td>efficiency</td>
<td>0.24 ± 0.07</td>
<td>0.54 ± 0.01</td>
</tr>
<tr>
<td>purity</td>
<td>0.87 ± 0.08</td>
<td>0.89 ± 0.02</td>
</tr>
<tr>
<td>fom</td>
<td>0.17 ± 0.06</td>
<td>0.39 ± 0.01</td>
</tr>
</tbody>
</table>

**Notes.** Reported values correspond to the final state of the model, after 300 iterations, averaged over 100 realizations. A full representation of the classification results in intermediate steps is given in Fig. 4.

Qualitatively, these finds are in agreement with the full light curve results presented in Ishida et al. (2019, Fig. 6). However, we see a significant decrease in the effectiveness of the RS strategy when compared to the results archived by US. In Ishida et al. (2019), RS figure of merit results achieve between 70% and 80% of the levels reported by US, while Table 2 shows RS figures stabilizing in approximately 40% of US ones. Despite different feature extraction and data sets, this behavior is mostly driven by the composition of the target sample. The ideal binary classifier should be trained in an equally balanced data set. In Ishida et al. (2019), the sample available for query was composed of 22% SN Ia (Fig. 3 of that paper), while in our results they represent merely 7% of the objects available for query (Table 1). Thus, in that work, RS results in a training sample closer to the 50/50 fraction and consequently this leads to better classification results.

More details about the queried sample are shown in Table 3. The RS strategy, as expected, queries individual classes following their fraction in the original data (Table 1). On the other

\(^{13}\) In data sets where such properties cannot be sustained through the system evolution, one should use an experiment design with separated pool and validation samples, as shown in Kenamere et al. (2020).
Fig. 4. Evolution of classification results as a function of the AL loop. In all panels the orange and blue lines show the results from random and uncertainty sampling, respectively. The dashed regions mark one standard deviation over 100 realizations. The initial training sample in all experiments consisted of ten alerts (five randomly selected from each class) and at each loop only one object was added to the training sample (batch = 1).

Table 3. Percentages of each class in the final queried sample.

<table>
<thead>
<tr>
<th>Class</th>
<th>Random sampling</th>
<th>Uncertainty sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNIa</td>
<td>6.58 ± 1.28</td>
<td>44.80 ± 1.40</td>
</tr>
<tr>
<td>star</td>
<td>58.52 ± 2.89</td>
<td>21.58 ± 2.04</td>
</tr>
<tr>
<td>other_SN</td>
<td>3.17 ± 0.93</td>
<td>16.76 ± 1.70</td>
</tr>
<tr>
<td>multiple_object</td>
<td>18.84 ± 2.35</td>
<td>6.31 ± 1.28</td>
</tr>
<tr>
<td>AGN-like</td>
<td>12.15 ± 1.95</td>
<td>9.82 ± 1.95</td>
</tr>
<tr>
<td>other_TNS</td>
<td>0.82 ± 0.46</td>
<td>0.78 ± 0.36</td>
</tr>
</tbody>
</table>

Notes. Values show mean and standard deviation over 100 realizations.

Fig. 5. Classes of the queried alerts as a function of the learning loop. The plot shows queries selected by the uncertainty sampling strategy over 100 realizations.

Almost half of the alerts queried by US in each full learning cycle (~135) were Ias. Thus, we can estimate that from the 23 425 alerts in the test sample, Ias correspond to ~1460. A detailed analysis of the photometrically classified set shows that, in over 100 realizations, RS identifies 443 ± 145 alerts as Ias, while US photometrically assigns 896 ± 33 alerts to the SNIa class. Given that both strategies report similar purity values, this means an almost double increase in the number of SNIa correctly classified when the training sample is constructed using hand, US queries seven times more SNIa and more than five times as many the other_SN type. Figure 5 shows how the queried classes change as a function of the learning loop. In early stages the queries are basically random, thus only until loop 20 do we again see an agreement with the overall class population (larger fraction of classes star and AGN-like). Once information is gathered about the larger populations, learning is concentrated in the class of interest, with a larger fraction of SN Ia being queried in latter loops.
Fig. 6. Distribution of true classes among the alerts classified as SN Ia. The classifier was trained using the complete queried sample in addition to the initial training sample (after 300 learning loops) gathered by the random (orange) and uncertainty (blue) sampling strategies. The bar length shows the mean over 100 realizations. Numerical values are given in Table 4.

Table 4. Percentages of each true class in the final set of alerts photometrically classified as SNe Ia.

<table>
<thead>
<tr>
<th>True class</th>
<th>Random sampling</th>
<th>Uncertainty sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiple_objects</td>
<td>0.95 ± 0.66</td>
<td>0.66 ± 0.29</td>
</tr>
<tr>
<td>star</td>
<td>2.13 ± 1.29</td>
<td>1.51 ± 0.39</td>
</tr>
<tr>
<td>AGN-like</td>
<td>1.25 ± 1.39</td>
<td>0.53 ± 0.24</td>
</tr>
<tr>
<td>other_SN</td>
<td>9.50 ± 1.73</td>
<td>8.25 ± 0.74</td>
</tr>
<tr>
<td>other_TNS</td>
<td>0.35 ± 0.10</td>
<td>0.25 ± 0.09</td>
</tr>
<tr>
<td>SNIa</td>
<td>85.99 ± 3.89</td>
<td>88.80 ± 1.07</td>
</tr>
</tbody>
</table>

Notes. Values show the mean and standard deviations over 100 realizations.

5. Deployment in Fink

The experiments described above resulted in a set of optimized training samples. We used the one that produced the best performing model (accuracy: 0.97, efficiency: 0.53, purity: 0.91, form: 0.42) to train a random forest classifier, which was subsequently integrated into the broker. Since May 2020, FINK has been processing ZTF alerts using a previous version of this module and reporting candidates with an early SN Ia probability higher than 0.5 to TNS since November 2020. The user can access alerts and the light curves from their corresponding objects through the science portal14.

Following the procedure described in Sect. 3.2, in order for an alert to receive a score from this classifier, it is required for there to be at least three observed epochs in each filter. Thus the number of classifiable alerts is strongly influenced by the telescope observation strategy. In order to minimize the effect of this requirement, for the purpose of feature extraction, we also consider observed epochs which fail the broker’s quality cuts, but were included in individual alert history, thus allowing for alerts with at least one valid point, among the three required by the filtering process (Sect. 3.1), to be classified. In practice, we found that epochs that fail the broker’s quality cuts were due to a poor S/N rather than clear artifacts or pipeline errors (see Möller et al. 2021 for a discussion on the quality cuts). In order to tag an alert as an early SN Ia candidate, we have a set of six criteria15: (a) Ia probability larger than 50% from the early SN Ia module described in this work; (b) Ia probability larger than 50% from either of the deep learning classifiers based on SUPERNOVA (Möller & de Boissière 2020) deployed in FINK; (c) no match with galactic objects from the SIMBAD database; (d) probability higher than 50% from the RealBogus algorithm (Mahabal et al. 2019; Duev et al. 2019); (e) star-galaxy classification score from SExtractor above 40% (Bertin & Arnouts 1996); and (f) the associated object holds no more than 20 photometric measurements.

Alerts fulfilling the above criteria were advertised to the community via publication in TNS. Over this period (01 November 2020 to 31 October 2021), 809 early SN Ia candidates were reported, from which 535 were spectroscopically confirmed. Among the confirmed set, 459 (86%) were confirmed as SNIa while the remaining ones were shown to belong to other SN types16.

The reported TNS numbers are in agreement with our results (Table 4). However, one should keep in mind that there is still a human screening involved in these numbers, which cannot be modeled. Every observer has the opportunity to inspect the alert candidate and make their own decision about moving forward with the spectroscopic follow-up. Nevertheless, the agreement between the reported numbers and the results presented in this paper are encouraging.

In order to allow the use of adaptive learning techniques “on the fly”, we developed an automated infrastructure which identifies alerts that can improve the classifier. This downstream will be distributed to the community allowing further improvement to future classifications. We plan to extend this framework to enable the classifier to be automatically retrained as soon as target labels from spectroscopy are available in public repositories, such as TNS.

6. Conclusions

The impact of recent technological development and the rise of the era of “big data” has been known and widely discussed in

14 https://fink-portal.org
15 https://tinyurl.com/FinkIaFilter
16 Except one LBV star.
a variety of research fields. This new paradigm is certainly also felt in astronomy; however, there are peculiar aspects of astronomical data which require special attention. Among them is the discrepancy between spectroscopic and photometric resources. If we aim to use real data for training as well as testing, we strongly recommend adopting automatic learning strategies which can work in the small data regime.

Active learning is a viable alternative when the number of spectroscopically classified objects is very limited. Working with cataloged transients observed in the ZTF alert stream as a case study, our final model achieves a purity of 89% and an efficiency of 54% while using merely 310 alerts for training – the cited results were obtained from a test sample of 23 465 alerts, ≈1463 of which were SN Ia. Such results, derived from a static database of alerts, were corroborated on real-time ZTF data. In the last year, we reported promising SNe Ia to the TNS service, from which 86% were found to be SNe Ia from spectroscopic follow-up observations. We stress that this does not correspond to a complete sample (not all candidates were spectroscopically followed) and that we cannot model the influence of human screening while selecting which candidates will be followed. Nevertheless, it is an encouraging result which can be further scrutinized in a purely automated environments, similar to the one proposed by Street et al. (2018).

Such fully automated pipelines would allow us to remove the current bias present in spectroscopic samples and enable the construction of small, optimized training samples according to the astronomical object of interest. Moreover, this has the potential to result in interesting classified objects benefiting from multiple communities. Among the queried alerts, ≈45% were SNIa, and others correspond to alerts which can be easily mistaken by SN Ias. The study of such objects may reveal details about their physical processes as well as their relationship with the Ia class.

This work represents the first step in the development of an AL environment which has been in the core design of FInK since its conception. Nevertheless, we can imagine similar classifiers being specifically design to target other classes of transients. This would allow us to expand our knowledge about the zoology of astronomical transient sources starting from the classes we know – thus enabling an optimal scientific exploitation of the large volumes of data expected from the next generation of large telescopes.

Acknowledgements. We thank Johan Bregeon, Jean-Eric Campagne, Benoît Carry and Maria Puzhinskaya for comments and discussions on the original draft. This work was developed within the Fink community and made use of the Fink community broker resources. Fink is supported by LSST-France and CNRS/IN2P3. E.E.O.I. receives financial support from CNRS International Emerging Actions under the project Real-time analysis of astronomical data for the Legacy Survey of Space and Time during 2021–2022. M.L. acknowledges financial support from the Paris Saclay University, DSI.

References
### Table A.1. Best-fit parameter values found for the alerts shown in Figure 2.

<table>
<thead>
<tr>
<th></th>
<th>top left</th>
<th>top right</th>
<th>bottom left</th>
<th>bottom right</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Ia</td>
<td>AGN</td>
<td>EB*</td>
<td>RRLyr</td>
</tr>
<tr>
<td>objectId</td>
<td>ZTF20acqjmzk</td>
<td>ZTF18aarybyq</td>
<td>ZTF18aabtvle</td>
<td>ZTF18aaptrep</td>
</tr>
<tr>
<td>candid</td>
<td>1424192755315015009</td>
<td>1452439215115010000</td>
<td>1277247161915010000</td>
<td>1385238050715015010000</td>
</tr>
<tr>
<td>a_g</td>
<td>0.527456</td>
<td>-3172.429739</td>
<td>4.37929</td>
<td>0.224062</td>
</tr>
<tr>
<td>b_g</td>
<td>0.422352</td>
<td>3170.602659</td>
<td>-0.089541</td>
<td>-7.343463</td>
</tr>
<tr>
<td>c_g</td>
<td>1846.688226</td>
<td>4224.205181</td>
<td>3766.113687</td>
<td>5782.164818</td>
</tr>
<tr>
<td>snratio_g</td>
<td>6.089286</td>
<td>10.370073</td>
<td>10.216985</td>
<td>13.058625</td>
</tr>
<tr>
<td>nmsnratio_g</td>
<td>0.000135</td>
<td>0.00016</td>
<td>0.000044</td>
<td>0.000571</td>
</tr>
<tr>
<td>nrise_g</td>
<td>6.0</td>
<td>4.0</td>
<td>9.0</td>
<td>8.0</td>
</tr>
<tr>
<td>a_r</td>
<td>0.356387</td>
<td>4.057071</td>
<td>0.004883</td>
<td>0.021988</td>
</tr>
<tr>
<td>b_r</td>
<td>-1.713326</td>
<td>-0.604576</td>
<td>1076.327583</td>
<td>271.873134</td>
</tr>
<tr>
<td>c_r</td>
<td>1913.34661</td>
<td>22509.009715</td>
<td>1210968.725619</td>
<td>16897710.267757</td>
</tr>
<tr>
<td>nmsnratio_r</td>
<td>6.776962</td>
<td>23.731405</td>
<td>7.220319</td>
<td>11.432526</td>
</tr>
<tr>
<td>nmse_r</td>
<td>0.0000032</td>
<td>0.000033</td>
<td>0.000376</td>
<td>0.001975</td>
</tr>
<tr>
<td>nnrise_r</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

**Appendix A: Features' values for Figure 2**

Table A.1 shows the extracted features’ values for alerts shown in Figure 2.