A&A 665, A99 (2022)
https://doi.org/10.1051/0004-6361/202243668

Astronomy & Astrophysics

SNGuess: A method for the selection of young extragalactic transients

N. Miranda[1], J. C. Freytag[1], J. Nordin[2], R. Biswas[3], V. Brinnel[2], C. Fremling[4], M. Kowalski[2,5], A. Mahabal[4,6], S. Reusch[2,5], and J. van Santen[5]

[1] Institut für Informatik, Humboldt-Universität zu Berlin, Rudower Chaussee 25, 12489 Berlin, Germany
e-mail: nicolas.miranda@hu-berlin.de
[2] Institute of Physics, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489 Berlin, Germany
[3] The Oskar Klein Centre, Department of Physics, Stockholm University, AlbaNova, 106 91 Stockholm, Sweden
[4] Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA
[5] Deutsches Elektronen-Synchrotron, 15735 Zeuthen, Germany
[6] Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA

Received 29 March 2022 / Accepted 9 July 2022

ABSTRACT

Context. With a rapidly rising number of transients detected in astronomy, classification methods based on machine learning are increasingly being employed. Their goals are typically to obtain a definitive classification of transients, and for good performance they usually require the presence of a large set of observations. However, well-designed, targeted models can reach their classification goals with fewer computing resources.
Aims. The aim of this study is to assist in the observational astronomy task of deciding whether a newly detected transient warrants follow-up observations.
Methods. This paper presents `SNGuess`, a model designed to find young extragalactic nearby transients with high purity. `SNGuess` works with a set of features that can be efficiently calculated from astronomical alert data. Some of these features are static and associated with the alert metadata, while others must be calculated from the photometric observations contained in the alert. Most of the features are simple enough to be obtained or to be calculated already at the early stages in the lifetime of a transient after its detection. We calculate these features for a set of labeled public alert data obtained over a time span of 15 months from the Zwicky Transient Facility (ZTF). The core model of `SNGuess` consists of an ensemble of decision trees, which are trained via gradient boosting.
Results. Approximately 88% of the candidates suggested by `SNGuess` from a set of alerts from ZTF spanning from April 2020 to August 2021 were found to be true relevant supernovae (SNe). For alerts with bright detections, this number ranges between 92% and 98%. Since April 2020, transients identified by `SNGuess` as potential young SNe in the ZTF alert stream are being published to the Transient Name Server (TNS) under the `AMPEL_ZTF_NEW` group identifier. `SNGuess` scores for any transient observed by ZTF can be accessed via a web service `https://ampel.zeuthen.desy.de/api/live/docs`. The source code of `SNGuess` is publicly available `https://github.com/nmiranda/SNGuess`.
Conclusions. `SNGuess` is a lightweight, portable, and easily re-trainable model that can effectively suggest transients for follow-up. These properties make it a useful tool for optimizing follow-up observation strategies and for assisting humans in the process of selecting candidate transients.

Key words. methods: data analysis – supernovae: general – cosmology: miscellaneous – cosmology: observations – astronomical databases: miscellaneous

1. Introduction

The study of transient astrophysical events has made it possible to understand explosive phenomena not accessible in terrestrial laboratories, and to map out the evolution of the Universe. Type Ia supernovae (SNeIa) are particularly important in this context. Their use as standardizable candles has allowed astrophysicists for already more than two decades to accurately measure the distance of remote regions of the Universe, and they provide the opportunity to gain insight into the mechanisms of stellar death (Riess et al. 1998).

A new generation of astronomical surveys, including the currently operating Zwicky Transient Facility (ZTF; Bellm et al. 2018), the All-Sky Automated Survey for SuperNovae (ASAS-SN; Kochanek et al. 2017), the Asteroid Terrestrial-impact Last Alert System (ATLAS; Tonry et al. 2018), the Dark Energy Survey (DES; Abbott et al. 2019), the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Kaiser et al. 2002), and the upcoming Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) conducted on the *Vera C. Rubin* Observatory, give the community unprecedented real-time access to observations of these events.

These facilities perform automated photometric observations of many sources in large regions of the sky, and then distribute the data to the community via alert streams. Subsequently, the observational sources that are deemed to be of particular interest are inspected in more detail by photometric and/or spectroscopic follow-up observations. Follow-up observations allow researchers to precisely identify and characterize astrophysical phenomena.

Spectroscopic resources are limited, especially when compared to the large number of astronomical transients that are photometrically detected and distributed as alerts. The fraction of photometric candidates that are spectroscopically classified is therefore rapidly decreasing with improved photometric

surveys. Explosive phenomena such as SNe and other fast transients often have a short lifetime (usually weeks or months) and their behavior evolves in a matter of days, or even hours.

Early detection and follow-up of explosive transients is particularly relevant in the study of SNeIa. It is widely agreed upon in the community that SNe of this kind are the result of thermonuclear explosions that originate in the interaction between two progenitor stars: a white dwarf and a companion. However, there remain open questions regarding the exact nature of both these progenitors and of the mechanism of the resulting explosion (Maoz et al. 2014).

Many of these questions can only be studied from observations that take place early in the lifetime of the explosive transient phenomenon. For instance, early photometric observations of a SNeIa can constrain the radii of the possible progenitor white dwarf star and its companion (Nugent et al. 2011). The speed of the increase in luminosity after the explosion and its duration are a useful probe of the inner and surrounding distribution of material at the core of the white dwarf (Dessart et al. 2014). Also, strong emission at certain wavelengths early after explosion may indicate particular interactions between the ejected material and the companion star (Kasen 2010).

Having greater insight into the aforementioned physical phenomena would vastly improve the ability of astrophysicists to calibrate SNeIa for their use as standardizable candles, and allow a better understanding of the evolution of large-scale structures of the Universe.

In general, automated methods for the classification of astronomical transients based on photometric data are expected to become critical tools both for parsing the large alert streams in real-time streams and for understanding the full, final observed sample. In recent years, photometric classification challenges such as PLAsTiCC (Hložek et al. 2020) and SNPhotCC (Kessler et al. 2010) have succeeded in bringing together the expertise of both the astronomical and machine learning communities in the development of new state-of-the-art tools.

Methods commonly used for time-series classification broadly fall into two groups: feature-based and data-driven (or nonparametric). Feature-based classifiers rely on inferring qualities and substructures from the measurements, according to previously defined (or engineered) functions, and then using these as a representation of the time series. Most features are statistical and structural metrics that can be calculated over a collection of measurements in time (Schäfer & Leser 2020). In time-domain astronomy, feature functions are selected for their application based on their ability to reflect specific characteristics of time-variable astrophysical phenomena. We can find some examples of feature-based classifiers in ALeRCE (Sánchez-Sáez et al. 2021), `Avocado` (Boone 2019), and Dai et al. (2018).

On the other hand, data-driven methods operate directly on the measurements of the time series, and do not depend on explicitly defined features. Thus, in this case it is the process of building these features that is explicitly defined, and this is integrated into the automated learning method itself. We can find some examples of data-driven methods in Charnock & Moss (2017), Mahabal et al. (2017), `PELICAN` (Pasquet et al. 2019), `RAPID` (Muthukrishna et al. 2019), `SCONE` (Qu & Sako 2022), `snmachine` (Alves et al. 2022), `SuperNNova` (Möller & de Boissière 2020), and `SuperRAENN` (Villar et al. 2020).

Another way to categorize these methods is according to which phase of the transient explosion they would focus on in order to classify according to transient type. Methods such as the one developed by Dai et al. (2018) take light curves that display the complete life cycle of the explosive transient as input, while those like *SCONE* can take partial time series as input (Qu & Sako 2022). Others such as `RAPID` are designed for both cases (Muthukrishna et al. 2019).

Even though they are successful in classifying transient candidates according to photometric data, these models are built on complex architectures (especially in the case of deep neural networks), which makes it hard for researchers to grasp the underlying logic of their decisions. Furthermore, most of these models require several observations in order to achieve their high classification performance, making them less suited to the identification of young transients.

The astrophysical transient research community has already recognized the importance of follow-up observation strategy planning, given the limited resources and the difficulty of obtaining reliable labels from which supervised classification algorithms can learn. To this end, active learning strategies have recently been used to design training sets for machine learning classifiers, in the context of the peculiar data environment of astronomical transient detection (Ishida et al. 2019; Kennamer et al. 2020; Leoni et al. 2022; Carrick et al. 2021).

Some of the classification algorithms previously mentioned, such as RAPID and `snmachine`, have the specific use case of performing fine-grained classification between SN subtypes. Therefore, they were trained and tested using explosive transient data only. It is assumed, in these cases, that other types of transients, such as moving objects, cataclysmic variable stars (CV), and active galactic nuclei (AGN) can easily be removed by comparison with historical observations obtained far back in time. However, this is hard to do for surveys that are in their starting phase.

For some surveys, once nontransient sources of variability such as AGN and variable stars are discarded, the resources available are sufficient to classify nearly all explosive candidates that reach a certain peak magnitude. For instance, in the case of ZTF, the value of this peak magnitude lies around 18. In the case of LSST, the Time-Domain Extragalactic Survey (TiDES; Swann et al. 2019) is expected to follow up all explosive transients detected with magnitudes $r_{\rm AB} \lesssim 22.5$ at peak.

For these reasons, here we study the effectiveness of an alternative method that is able to provide an informed guess of whether a particular variable candidate will become an explosive transient that is relevant for follow-up. This guess should take place at early observation times and without the need for many historical observations or redshift-related catalog information.

This article is structured as follows. In Sect. 2, we introduce the astronomical and classification concepts (the latter in the context of machine learning) used in this text. In Sect. 3, we discuss the motivation and goals taken into consideration when designing and implementing `SNGuess`. Sections 4 to 7 describe the main steps followed in order to train the model that is at the core of `SNGuess` (see Fig. 3). Section 8 gives details of how `SNGuess` is actively selecting transients and how the results have been made available to the community and Sect. 9 evaluates the performance. Finally, in Sect. 10 these results are summarized and we list important challenges to tackle in forthcoming work.

2. Terminology

This study makes use of terminology both from astronomy and from computer science; specifically from the domains of data science and machine learning, and of classification tasks (or alternatively, information retrieval). In this section, we

proceed to introduce concepts that are commonly used in transient astronomy and in the classification process[1].

2.1. Astronomical concepts

Hereafter, we refer to "alerts", which are data packets issued at a certain point in time by an astronomical survey or a highly automated observation facility, such as the ZTF. A "stream" or "alert stream" is a set of alerts generated and distributed by a given observation facility during an interval of time.

An individual measurement performed by an instrument at the observation facility is called an "observation". In optical astronomy, an observation is represented by an image of a region of the sky taken by the camera of the instrument at a particular time. Point sources of variability in the sky are typically identified from a "difference image"; which is an image that is created through subtracting a newly made observation from a reference image. A reference image is an aggregation of observations of the same region of the sky but generated at previous moments in time. A variable source is identified by the survey every time a point source is recognized in a difference image. A point source in the difference image which exceeds some survey-specific signal-to-noise-ratio (S/N) threshold is called a "detection"[2].

We use the terms "astronomical source" or "astronomical object" to refer to the astrophysical entity or phenomenon from which a set of detections of a certain point or region in the sky seems to have originated. In the context of astronomical alerts, we call these "candidates", to reflect the fact that they may or may not end up being relevant for follow-up observations and subsequent characterization as a particular kind of astrophysical phenomenon.

Alerts are generated by an automated survey each time a detection is made in a difference image. Observational data of the detection are then complemented with other contextual data into an alert and distributed (Bellm et al. 2018). The alerts may contain metadata related to the observation process itself (this mostly concerns the instrument that was used and the observing conditions) or related to the candidate (such as its location or its proximity to other previously identified sources).

The nature of the observations contained in the alert will depend on the instrument that was used. For instance, ZTF uses an optical instrument, and therefore its observations are photometric in nature. By "light curve" we refer to a representation of a time series as a vector of photometric measurements at a certain band in wavelength and their associated time-stamps. Light curves can have measurements in multiple bands, and they usually have irregular sampling (see Fig. 1).

If a candidate has a recognizable behavior only during a certain time duration (usually days or weeks) before returning to a "normal" or baseline behavior, we say that the behavior is "transient" in nature, or that we observe a transient candidate or phenomenon. Events, or detections, can be caused by different phenomena, including "proper" transients (e.g., SNe), reoccurring variables (e.g., AGN), moving objects in the Solar System, cosmic rays, or pure noise due to for example misaligned references.

Alerts are represented by binary files containing metadata and contextual information for a single detection or event. This event is usually a change in the position or luminosity of a particular point in the sky (Masci et al. 2019). Each alert includes at least a unique identifier and data related to the particular science observation such as, for instance, a cutout image of the observation region (Juric et al. 2020). Additionally, in surveys such as ZTF and LSST, alerts include some listing of prior variability connected to the same source. ZTF alerts contain the photometry of any prior detection at the same position during the last 30 days. ZTF alert packages are represented and structured in JSON format. Some of its values are: a unique identifier for the alert, candidate-specific metrics from the image-subtraction process, image-specific metadata, nearby astronomical objects according to different catalogs, and image cut-outs centered on its location. There is typically a one-to-many relationship between candidates and their alerts. That is, a new alert for a candidate is issued every time a new valid observation is made. Therefore, many alerts are usually produced for a candidate during its transient lifetime.

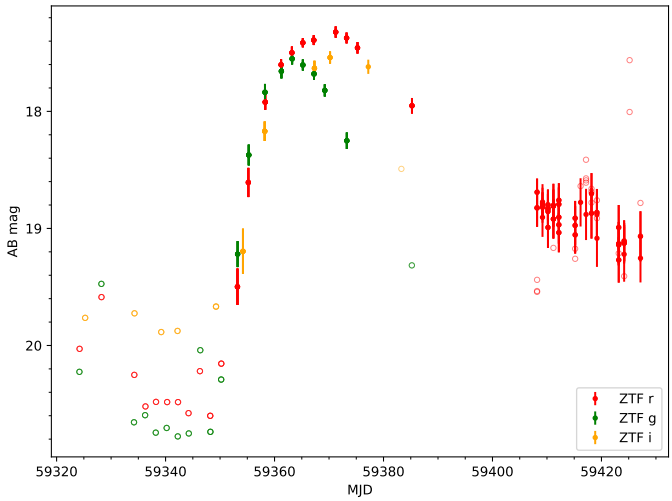

Fig. 1. Example of a light curve obtained from a ZTF alert; in this case candidate ID ZTF21abbyhvw. This particular candidate has detections in three bands. The irregular sampling that is characteristic in astronomical observations can be seen. Lightly colored hollow points correspond to nondetections.

Table 1. Definitions of some common classification assessment metrics.

Precision	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$
Recall	$\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$
$F1$-score	$\frac{2\mathrm{tp}}{2\mathrm{tp}+\mathrm{fp}+\mathrm{fn}}$
MCC	$\frac{\mathrm{tp}\times\mathrm{tn}-\mathrm{fp}\times\mathrm{fn}}{\sqrt{(\mathrm{tp}+\mathrm{fp})(\mathrm{tp}+\mathrm{fn})(\mathrm{tn}+\mathrm{fp})(\mathrm{tn}+\mathrm{fn})}}$

Notes. tp stands for true positive examples, tn for true negatives, fp for false positives, and fn for false negatives.

2.2. Classification concepts

In the context of information retrieval, precision is defined (Chicco & Jurman 2020) as the fraction of retrieved documents that are relevant (see Table 1). Its dual metric, recall, is defined as the fraction of relevant samples that are correctly retrieved. This definition can be directly applied to classification tasks if

[1] These are introductions to terminology for a cross-community context. Any reader already familiar with these concepts should skip this section and continue with Sect. 3

[2] In *Rubin* Observatory nomenclature, this is called a "source" instead of a "detection". Readers familiar with that nomenclature should bear this in mind for the remainder of the article.

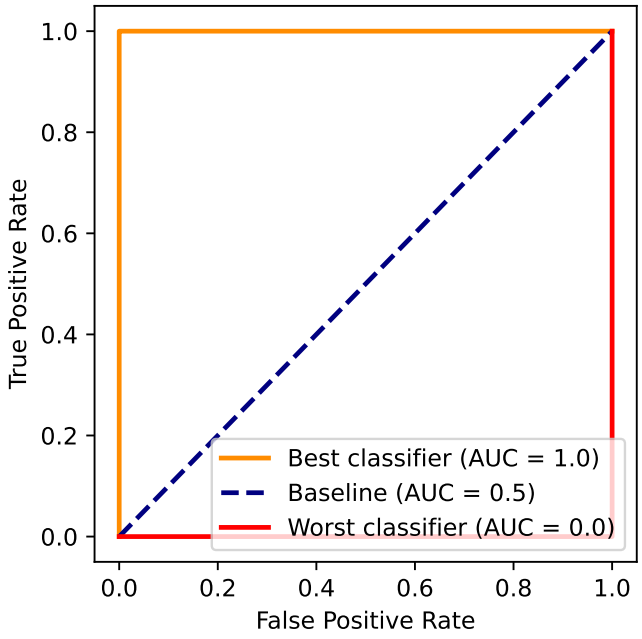

Fig. 2. Example of ROC curves with their respective AUC. Here, we see the curves that correspond to the performance of the best possible classifier (AUC = 1.0), a perfectly random or baseline classifier (AUC = 0.5), and the worst possible classifier (AUC = 0.0).

we interpret relevant and irrelevant samples as positive and negative classes, respectively. The $F1$-score, in turn, is defined as the harmonic mean between precision and recall. The Matthews correlation coefficient (MCC) is a statistical rate that is used as a score for classification tasks. It has the advantage of being more informative and truthful in evaluating binary classifications than the accuracy or the $F1$-score (Chicco & Jurman 2020).

The Receiver Operating Characteristic (ROC) curve (see Fig. 2) is the result of plotting true positive rates against false positive rates while gradually changing the threshold value returned by the classification model and used for distinguishing between classes. A ROC curve is plotted in a two-dimensional Cartesian coordinate system, where the horizontal axis represents the false positive rate and the vertical axis represents the true positive rate. Both rates can take values ranging from 0 to 1; and therefore, the ROC curve is bounded by the unitary quadrant of the plane.

In a perfect binary classification procedure, if the threshold condition for class distinction is incrementally relaxed (i.e., subsequently more and more examples start being considered as belonging to the positive class), almost all marginal increases in true positive rates occur with no increase in false positive rate. A classification like this will be represented by a curve in the ROC diagram that has the highest possible area under it, and it will have a value very close to (0, 1) at some point. This curve will be similar to a Pareto cumulative distribution with an infinite *shape* (α) value.

In contrast, if a classification is the worst possible, almost all marginal increases in false positive rate occur with no associated increase in true positive rate. When plotted in a ROC diagram, this curve will be very close to $y = 0$ in most of its points, and it will have an area under the curve that is close to zero.

A random binary classification where a given example has the same probability of being assigned to either of the two classes (e.g., flipping a balanced coin each time to decide which class to assign an example) should, if statistically significant, increase the true positive and the false positive rate by the same marginal amount. This constitutes the baseline performance, and when plotted in the ROC curve it describes a line that corresponds to the identity function (it goes from (0, 0) to (1, 1)).

The ROC curve can be summarized by a single number: the area under the curve (AUC, in this case the ROC AUC). This number is used by itself as a metric to assess classification performance, and its value ranges from 0 (worst possible performance) to 1 (best possible performance).

3. Motivation

The task of classifying light curves has two different goals, depending on whether the potentially observed transient is at an early or late stage of its lifetime. The scope and the challenges in these two cases are distinct from each other. In the case of late stage classification, the transient has already faded by definition. That is, we collected all observations for the full span of the transient's life, and the goal is to achieve precise classifications, even up to the subtype level. How fine-grained the classification should be depends on the specifics of each science use case. When a sufficient number of observations are available, it is trivial to identify Solar System objects, satellites, or other variables simply by looking at their light curve but not transient sources that are irrelevant for further analysis.

On the other hand, classifying transients at their early stage is fundamentally different: the goal here is to find relevant phenomena as early as possible to the beginning of their lifetime in order to request additional dedicated observations quickly. That is, we are interested in detecting quickly developing extragalactic objects in their infant or early phase. For such objects, we possess a limited number of measurements, and therefore, it is not trivial to distinguish them from irrelevant candidates. The likelihood of mistaking a transient for another kind of variable object is even higher when the survey is just starting its operations. This is also the case when there are not many historical observations available overall. In this context, our main sources of contamination are satellites, Solar System objects, intragalactic objects, and other types of nontransient variable sources, with the possible added presence of observation artifacts. Furthermore, if we are only focused on following-up young SNe, then distant and old but still bright SNe may be a source of confusion as well.

We investigate if simple but specialized methods could prove successful in terms of achieving early stage classification. In particular, the goal is to develop a model that works as a filter in terms of selecting young SNe as candidates for fast and/or automatic follow-up. In this case, noise will consist mostly of artifacts, nonextragalactic transients, and nontransient variable sources. In our work, the main goal is to reduce the number of false-positive selections by as much as possible, as the resources for follow-up observations are highly limited.

The selection of relevant transients for follow-up is a stage dominated by intense human labor. Any method that may assist in this process should be a valuable contribution in improving the efficiency of the whole transient detection and characterization pipeline.

Our selection method, which we call `SNGuess`, consists of extracting a feature vector from each alert of a set, and then organizing the vectors into a feature matrix. In a next step, a set of labels that indicate the relevance for follow-up of each one of the alert candidate objects is obtained (one label for each one of the feature vectors) and is then used to construct a target vector. The target vector takes Boolean values (`true` when the alert candidate is relevant, `false` when it is not relevant). The feature matrix and

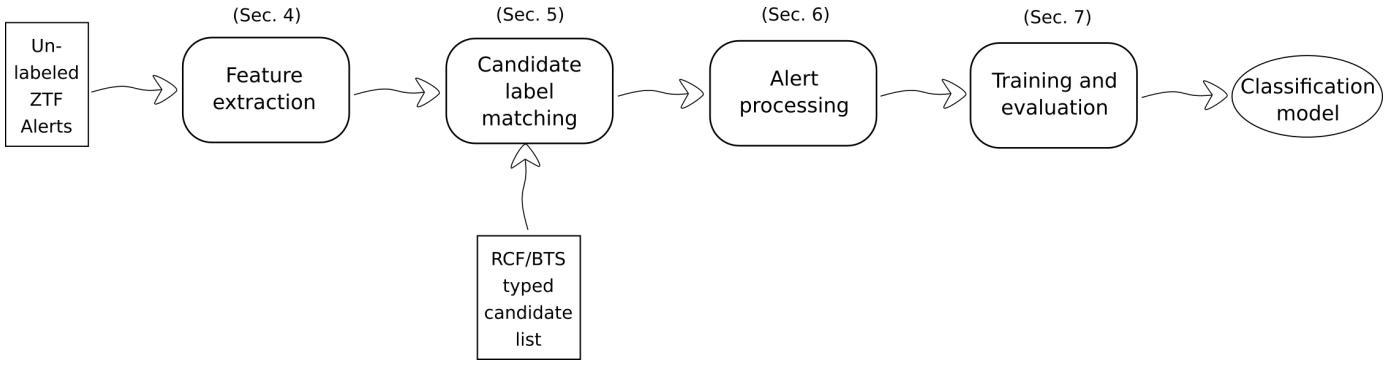

Fig. 3. Steps for `SNGuess` model generation. Each step in the process is described in a separate section, as indicated above the boxes.

the target vector provide the input for supervised training of a boosted-trees classification model.

Once the model is trained, vectors of the same feature functions are extracted; this time from a different set of alerts to be analyzed. These vectors are subsequently fed to the trained model in order to obtain a set of prediction scores. The higher the value of a particular prediction score, the more interesting and relevant the candidate is for follow-up.

The main motivation for developing the `SNGuess` method is the scientific use case of selecting young, extragalactic, and nearby SNe for additional follow-up observations. Almost all of them become bright at the peak of their transient lifetime. For our present analysis, we assume that the ZTF Bright Transient Survey (BTS; Fremling et al. 2020; Perley et al. 2020) has classified most of those transients to date, as this survey aims to classify all transients that at some point become brighter than 18.5 in magnitude. We are interested in the feasibility of automatically detecting those transients even earlier in time. Therefore, we use labels from BTS to indicate relevance for follow-up of individual transients.

The process used to generate the core classification model of `SNGuess` can be separated into the following phases (see Fig. 3):

1. Alert pre-selection and feature extraction (Sect. 4);
2. Candidate label matching (Sect. 5);
3. Alert processing (Sect. 6);
4. Training and evaluation (Sect. 7).

In the following sections, we discuss details of these steps together with the data sets used.

4. Alert pre-selection and feature extraction

`SNGuess` was trained and tested with data from the ZTF survey[3]. ZTF generated on average more than 150 000 alerts per night for its public survey between May 2018 and July 2019. An initial pre-selection was made to only include alerts located outside the Galactic plane and with at least two detections and a `RealBogus` score larger than 0.3. `RealBogus` is an automatic classification system designed to identify observational artifacts. Additionally, known Solar System objects flagged by the Minor Planet Center (MPC) are discarded. This filtering process is similar to those typically performed by real-time observational surveys, and minimizes the impact from Solar System objects and stellar variability in the Galaxy. The result of this filtering process is a set of 261 417 astronomical alerts generated by the ZTF stream. This set contains photometric detections that took place between May 2018 and July 2019.

We define a set of features (numerically measurable properties) to be extracted from the alert set that are relevant to our context. These are simple to calculate from alerts, regardless of whether they contain data for few or for many observations. Table 2 shows the full list of used features and their description.

The feature set can be divided in two subsets. The first one includes features that are related to the photometric observations contained in the alert. These are calculated using functions that depend only on the photometric points. Some examples of this kind of feature are the slope, color, duration, and peak magnitude of the light curves. The second subset contains the features that consist of values relating to the alert metadata or, more specifically, related to the transient candidate from which the photometric observations originate, such as its location in the sky, or its distance to the nearest astronomical source in a particular catalog in the PanStarrs (Kaiser et al. 2002) and *Gaia* (Perryman et al. 2001) catalogs. Most of these numerical values are already present in the alert.

In general, the values of the alert metadata features tend to remain constant between alerts that refer to the same transient candidate. In contrast, the values of the photometric features usually differ between two alerts, depending on how the light curve evolves as more observations of the candidate are received.

The precision of photometric features varies strongly depending on the candidate luminosity. Even though the 5σ detection limits of the ZTF is around 20 to 21 magnitudes, there can still be large uncertainties for detections that are fainter than 19.5. Therefore, it is not guaranteed that an alert with a detection of that magnitude will be distributed. For this reason, any calculation made for these detections will have large uncertainties associated and problems with incompleteness.

Some of the features take Boolean values, such as the `peaked` field, which indicates whether the alert's light curve has a peak or not. Other features have integer or floating point values, such as the number of detections (`ndet`) or the distance to the nearest astronomical source (`distnr`).

The output of the feature calculation phase is one vector of feature values for each alert. These feature vectors are arranged as matrix rows to form a feature matrix.

5. Candidate label matching

The classification at the core of `SNGuess` is performed by a supervised model. However, none of the alerts that are generated by ZTF have information on the type of the observed candidate.

[3] In the future, ZTF alerts will contain forced photometry data. This study was made based only on alert photometry.

Table 2. List of features used by `SNGuess`.

Name	Type	Description	Source
tPredetect	Time	Time between final good upper limit and first detection	Photometric
tLC	Time	Duration (time between first and most recent detection)	Photometric
ndet	Int	Number of significant detections	Photometric
peaked	Bool	Is the lc estimated to be declining?	Photometric
pure	Bool	No significant nondetections after first detection	Photometric
rising	Bool	Max brightness close to the most recent detection	Photometric
norise	Bool	No (significant) detected rise	Photometric
hasgaps	Bool	The light curve has a gap between detections of at least 30 days	Photometric
mPeak	Mag	Magnitude at peak light (any band). Only calculated if peaked==True	Photometric
mDet	Mag	Magnitude at first detection (any band)	Photometric
mLast	Mag	Magnitude of the current (i.e. latest) detection (any band)	Photometric
cPeak	g–r	Color at peak (if peaked and with g and r)	Photometric
cDet	g–r	Color at detection (if with $g+r$)	Photometric
cLast	g–r	Color at last detection (if with $g+r$)	Photometric
slopeRise g, r	Mag/time	g or r mag slope between detection and peak (none if norise)	Photometric
slopeDecline g, r	Mag/time	g/r magnitude slope between peak and last detection (none unless peaked)	Photometric
rb (med)	Float	Median Real Bogus (all detections)	Photometric
drb (med)	Float	Median deep Real Bogus (if available, all detections)	Photometric
distnr	Pixel	Distance to nearest astronomical source in reference image	Metadata
magnr	Mag	Magnitude of nearest astronomical source in reference image	Metadata
classtar	Float	Star/Galaxy classification score from SExtractor	Metadata
sgscore1	Float	Star/Galaxy score of closest astronomical source from PS1 catalog	Metadata
distpsnr1	Arcsec	Distance to closest astronomical source from PS1 catalog	Metadata
sgscore2	Float	Star/Galaxy score of next to closest astronomical source from PS1 catalog	Metadata
distpsnr2	Arcsec	Distance to next to closest astronomical source from PS1 catalog	Metadata
neargaia	Arcsec	Distance to closest astronomical source from *Gaia* DR1	Metadata
maggaia	Mag	*Gaia* (g-band) magnitude of closest astronomical source from *Gaia* DR1 catalog	Metadata

Notes. The features are separated into *photometric* (created based on the transient light curve provided in the alert) and *metadata* (created based on the properties contained in the ZTF alerts of nearby detections in the references and in astronomical catalogues). While most of the features are self-explanatory, their explicit definitions can be found in Appendix C.

Furthermore, no single metadata field in the alert can be directly used as a reliable indicator of follow-up relevance. This is why candidate type information has to be obtained from external data sources. Then, candidate relevance, and therefore a suitable relevance label, can be inferred from the candidate type, depending on what type of object should be followed up. In our case, these objects are young extragalactic SNe.

Two external type lists are used for this purpose. First, a list of 4578 labeled candidates from the BTS survey is obtained, and these are then cross-matched with the candidates in the training set of ZTF alerts. The BTS classifications are mapped into ten general classes, shown in Table 3. This mapping is required as some labels have an interrogation mark suffixed, indicating that the type assigned to the candidate is uncertain. Some labels simply indicate that no type was found for the candidate (e.g., `None`, `unknown`). Other labels are very similar to each other or may refer to very specific subtypes (e.g., `SLSN-I` and `SLSN-I.5`, `SN Ib`, and `SN Ib/c`, etc). The primary goal of `SNGuess` is the detection of SN-like objects, and to do this at an early moment in their life cycle, when subclassification is highly unreliable. Furthermore, a training set with overly fine-grained labeling results in classes with very few examples. Additionally, we use a second set with 495 candidates typed as CV that were retrieved in March 2020 as a source of labels to perform the matching process described above.

We note that the training sample contains nearby SNe eventually classified by BTS (and therefore with positive labels), fainter but real astrophysical sources, and nonastronomical objects (both without labels).

Table 3. Main types assigned from BTS to the alert data set used for training `SNGuess`.

Type	No. of candidates	No. of alerts
SN Ia	974	16942
SN II	215	5022
AGN	162	5573
CV	58	693
SN IIn	53	1476
SN IIP	45	1790
SN Ic	38	813
SN IIb	34	611
SN Ia 91T-like	31	854
SN Ib	24	611

6. Alert processing

Once the features are extracted from the ZTF alerts and the candidate type information is obtained, we process the alerts based on their observations and metadata. First, if a metadata field contains a measurement value that is nonphysical, it is replaced with a `null` value.

Table 4. Alert groups by number of detections and their cut values.

No. of detections	`tPredetect` [JD]	`tLC` [JD]
2	≤3.5	≤3.5
3	≤3.5	≤6.5
4	≤3.5	≤6.5
5	≤3.5	≤10
6	≤3.5	≤10
[7, 100]	≤10	≤90

Notes. A separate model is trained for each row.

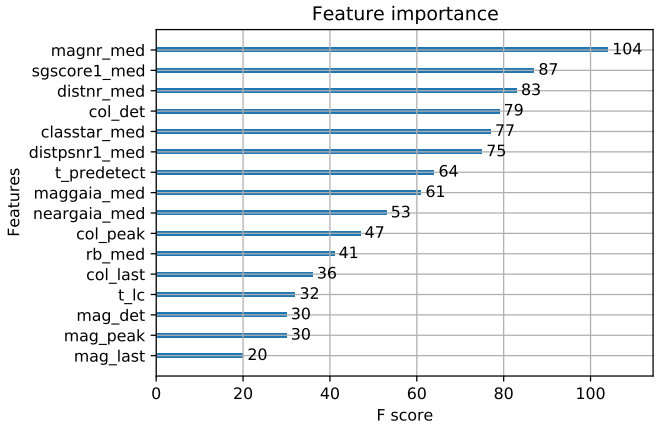

Fig. 4. Ranking of most important features for classification with the 7–100 detections classifier, according to the $F1$-score metric.

Alerts belonging to candidates with less than six "final" detections are removed from the training set. Most likely they would not have been part of any follow-up campaign (e.g., due to poor weather later), and thus risk biasing the training set. All alerts of candidates with sufficient data to eventually be classified (if sufficiently bright) are included individually.

In a subsequent step, alerts are separated into six different groups according to the number of their detections (`ndet`). For each one of these groups, we exclude those alerts that exceed a preset maximum duration between the first and most recent detection (`tPredetect`) and a duration between final good upper limit and first detection (`tLC`). These time ranges were chosen to be longer than the nominal ZTF northern sky survey cadence, meaning that an upper limit (i.e., nondetection) or previous detections should exist during normal operations. This additional filtering rejects detections made after a gap in the observations (e.g., due to visibility, scheduling issues, or weather). We perform this filtering because we are training `SNGuess` with mostly human-generated labels as a gold standard. We assume that an expert human agent is unable to systematically detect young transients without having a baseline to which its recent explosive behavior can be compared. The corresponding groups and their maximum values are shown in Table 4.

Finally, alerts that had a first detection with a magnitude below 16 are excluded from the training data set; these alerts were consistently found to be due to stellar variability and the same cut is applied in the live ZTF alert stream. Initially, the training set contains 261 417 alerts for 45 765 different candidates. At the end of the data processing phase, the resulting set contains 109 587 alerts for 35 883 candidates.

7. Training and evaluation

We use the XGBoost (Chen & Guestrin 2016) implementation of the Gradient Boosted Trees algorithm to obtain a decision tree ensemble model for the classification. Models trained by this algorithm have proven to be state of the art in classification, showing low out-of-sample errors in a variety of domains (Zhang et al. 2017). They can handle missing feature values out of the box, which simplifies the preliminary processing phase by making it unnecessary to replace them with generic values or to extrapolate them from existing data.

The Gradient Boosted Trees algorithm receives as input a series of hyper-parameters that control how the training takes place, and puts constraints on model parameters to keep the model from over-fitting to the training data. The first task of the learning phase is to explore different combinations of hyper-parameter values in order to estimate how well the models generated by the training algorithm perform and how this performance changes with each one of these combinations, before selecting the best hyper-parameter values.

Numerical ranges for all of the hyper-parameters are defined in order to explore the space of possible hyper-parameter combinations (see Table 5). Each one of the hyper-parameters is then assigned a uniform probabilistic distribution within its range. Next, a randomized search is performed in order to obtain an optimal hyper-parameter value combination. That is, m points are randomly sampled from the joint distribution over all hyper-parameter variables. Each one of these points represents a combination of values for all hyper-parameters. For each one of them, training and evaluation of the model is performed with five-fold cross-validation. The mean and variance of the classification performance metric for all cross-validation folds are calculated before continuing with the next point sampled from the hyper-parameter joint distribution. The model that is selected as the best is the one that obtains the highest performance metric across all its cross-validation folds.

This training and evaluation process is performed once for every group of alerts (alerts grouped by their number of detections; see Sect. 6). For groups of alerts that have six or fewer detections, the following features are used as input for training: `mDet`, `mLast`, `tLC`, `rb`, `cDet`, `tPredetect`, `distnr`, `magnr`, `classtar`, `sgscore1`, `distpsnr1`, `neargaia`, and `maggaia`. For the final group, with alerts that have between 7 and 100 detections, all of the features indicated above are used as input for training, plus: `mPeak`, `cLast`, and `cPeak`.

After the training and evaluation phase, one set of boosted trees is produced for each one of the six groups into which the alerts were separated according to their number of detections. It is important to note that this division is done on an alert-by-alert basis, and not on an object-by-object basis. For instance, if an object has seven detections, one of its alerts will be considered in the two-detections group to start with, but later another one of its alerts will be considered in the 7–100 group (and its other alerts in the respective groups in between). Table 5 shows the optimal hyper-parameters obtained after performing cross-validated training and evaluation over every alert group by number of detections.

XGBoost allows the user to easily generate a ranking of feature importance for the classification model. Figure 4 shows a list of the most important features for classification in the 7–100 detections classifier.

Table 5. Hyper-parameter space used for tuning the XGBoost model.

Hyper-parameter	Min. value	Max. value	2 det.	3 det.	4 det.	5 det.	6 det.	7 to 100 det.
Column sample by level	0.01	1.0	0.01	1.00	1.00	1.00	1.00	0.00
Column sample by tree	0.1	1.0	0.74	0.75	0.69	0.92	0.75	0.74
Gamma	0.0	40.0	0.14	1.65	3.81	2.49	1.65	0.13
Learning rate	0.0005	0.5	0.38	0.17	0.10	0.32	0.17	0.38
Maximum depth	2	12	12	9	3	11	9	12
Minimum child node weight	0.0001	0.5	0.32	0.04	0.42	0.21	0.04	0.32
Subsample	0.01	1.0	0.79	0.48	0.60	0.36	0.48	0.79

Notes. The tuning process was done via a randomized search, and final optimal hyper-parameters obtained per model by number of detections. 200 points were randomly selected from the joint distribution of uniform probabilities within each hyper-parameter range. A five-fold cross validation was then performed for each of them and the mean performance was calculated. The point with the highest mean performance was selected as the optimum combination of hyper-parameter values.

8. Model deployment

SNGuess was designed as a light-weight system that is easy to both retrain and use. The data and procedures described so far have been made public in an SNGuess Git repository[4]. Users can access Python Jupyter notebooks with all of the steps in the data processing and training pipeline of SNGuess[5]. The results and plots of this article can be locally replicated, or the model retrained over a different set of features or labels.

SNGuess has been deployed as part of the AMPEL real-time alert processing platform (Nordin et al. 2019), and has been running on the servers of *DESY* in Zeuthen, Germany, since April 2020. Users can directly access the generated SNGuess scores of ZTF transients through the AMPEL API[6]. Alternatively, users can install the core AMPEL packages and run the trained SNGuess model on raw ZTF light curves directly[7].

Finally, SNGuess is used as one of the components of the selection process for young ZTF transients which are autonomously submitted to the Transient Name Server (TNS[8]) under the sender name AMPEL_ZTF_NEW. The score assigned by SNGuess is a floating point value between 0 and 1, where a score closer to 1 indicates that the candidate is more relevant for follow-up observations than one with a score closer to 0.

9. Testing

We evaluated the performance of SNGuess over a set of 173 402 alerts for 8969 candidates, received from October 8, 2020 to August 15, 2021. These alerts remained after applying the same filter for poor-quality observations, as discussed in Sect. 6.

Four classes of alerts were defined in order to generate performance metrics for SNGuess:

– *Actual positive.* any alert of a candidate eventually targeted by BTS;

– *Actual negative.* any alert of a candidate not targeted by BTS;

– *Predicted positive.* any alert with an SNGuess score above a certain threshold;

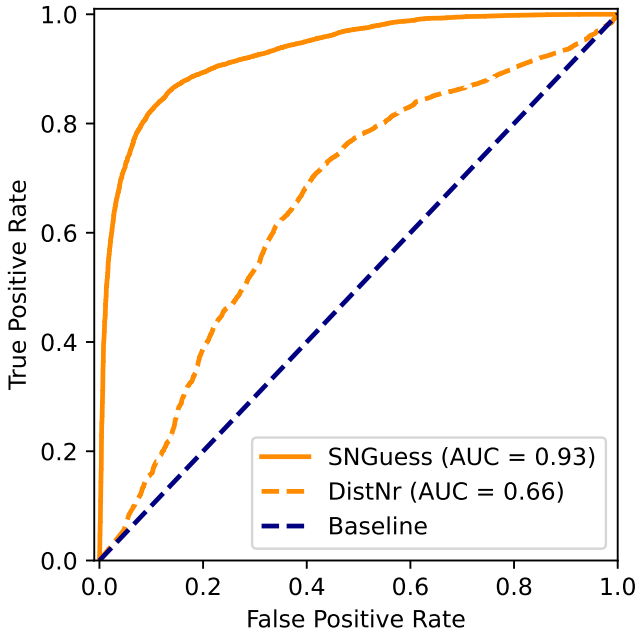

Fig. 5. ROC curve for SNGuess with the test data set (see Sect. 2.2). The area under the curve (ROC AUC) value of 0.93 summarizes a good performance across different score thresholds for distinguishing between relevant and nonrelevant candidates. As a comparison point, we also see the ROC curve for performing a simple logistic regression classification with just the distance to nearest source as an independent variable. We can see that SNGuess shows a better performance than this simple classification (ROC AUC = 0.66).

– *Predicted negative.* any alert with an SNGuess score below or equal to the threshold.

The use of BTS labels as a measure of the "true" outcome assumes that this survey was 100% efficient in terms of selecting transients by a human scanner. There are several reasons why this precision is not achieved in practice: observations of transients might have halted prior to reaching the BTS magnitude threshold due to weather or visibility, the transient might be intrinsically faint (or reddened), or it could be located in the core of a bright galaxy. It is therefore likely that the evaluation carried out here undervalues the SNGuess performance.

Gradually increasing the SNGuess acceptance threshold while keeping record of the true-positive and false-positive rates produces the ROC curve shown in Fig. 5. The ROC AUC value of 0.93 reflects good performance over a variety of different

[4] https://github.com/nmiranda/SNGuess
[5] https://github.com/nmiranda/SNGuess/tree/main/notebooks
[6] https://ampel.zeuthen.desy.de/api/live/docs
[7] https://github.com/AmpelProject/Ampel-HU-astro/blob/main/notebooks/ampel_api_run_T2BrightSNProb.ipynb
[8] https://www.wis-tns.org/

Table 6. Performance metrics for SNGuess over the test data set for different models, by number of detections of their alert training data set.

Metric	2 det.	3 det.	4 det.	5 det.	6 det.	7 to 100 det.	All models
Precision (best = 1.0, worst = 0.0)	0.473	0.563	0.642	0.675	0.671	0.962	0.876
Recall (best = 1.0, worst = 0.0)	0.537	0.781	0.827	0.877	0.898	0.830	0.821
$F1$-score (best = 1.0, worst = 0.0)	0.503	0.655	0.722	0.763	0.768	0.891	0.848
MCC (perfect = +1.0, random = 0.0, inverse = –1.0)	0.398	0.539	0.596	0.641	0.617	0.787	0.728

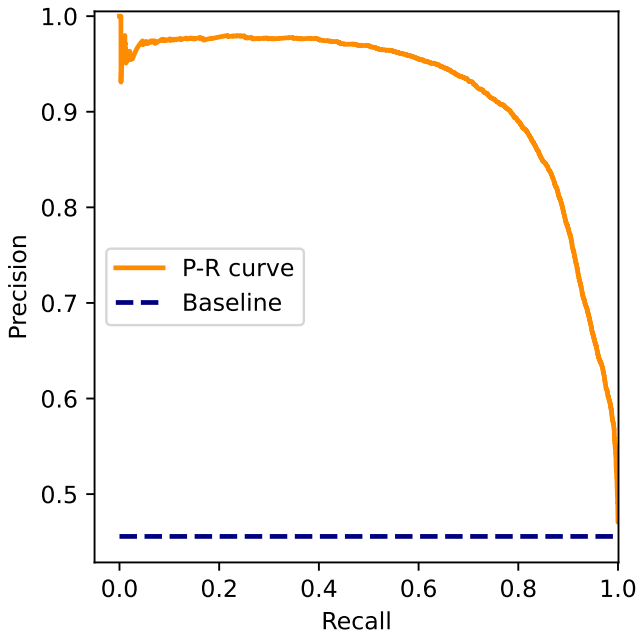

Fig. 6. Precision–recall curve for selection by SNGuess from the test data set. The baseline performance shown by the blue dashed line corresponds to the ratio between relevant and total number of examples, and is an indicator of the imbalance between classes.

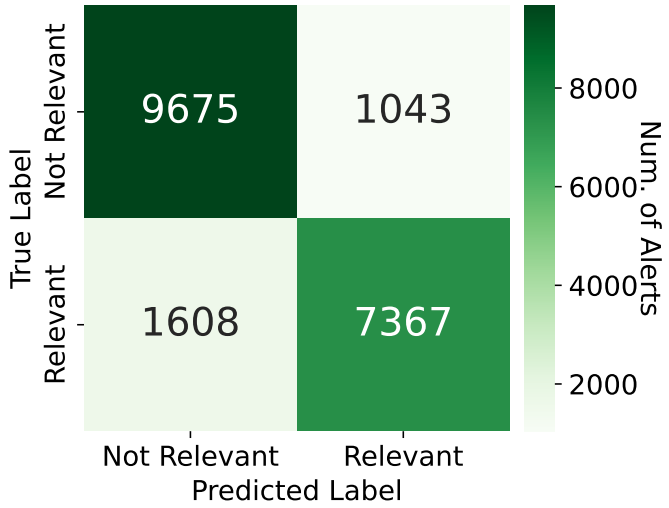

Fig. 7. Confusion matrix of the selection by SNGuess from the test data set. The number in each box corresponds to the number of alerts fulfilling the SNGuess input criteria (see text for these).

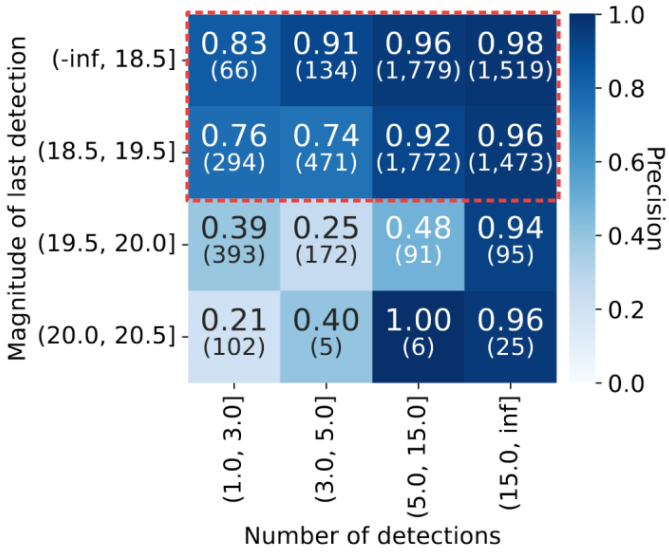

Fig. 8. Precision of the selection from alerts grouped by number of detections and magnitude of last detection for alerts fulfilling SNGuess run criteria. The area highlighted in red, with last magnitude below ~19.5, correspond to alerts that are expected to have small photometric uncertainties. For fainter objects, ZTF photometry can exhibit significant uncertainties and variable detection efficiency.

transient ages and luminosities. We draw the same conclusion for the precision–recall curve of the selection, as shown in Fig. 6. We realize that the curve is quite close to that of a perfect classification.

If we fix the score threshold to 0.5, and define all examples with a higher score as belonging to the positive selection or prediction class, we obtain a classification that produces the confusion matrix displayed in Fig. 7. We observe that false positives are less common than false negatives. As we can see in Table 6, SNGuess has a good performance in several metrics that are commonly used in binary classification tasks. Some of them, such as the $F1$-score and the Matthews correlation coefficient (MCC), are particularly important because of their robustness to class imbalance (Chicco & Jurman 2020).

Astronomical use cases for a tool like SNGuess would typically include program limits in terms of the desired age or brightness of follow-up targets. Figure 8 shows the precision of the selection by SNGuess from alerts grouped by number of detections and magnitude of last detection. The number of detections works as an effective indicator of the potential age of the transients (which cannot be fully known at the time of the alert generation) as the nominal ZTF MSIP survey observes each field twice ($g + r$) every third day. The number of examples in each bin is displayed in parentheses in order to give an idea of the statistical significance. We again note that these are conservative numbers in that a fraction of the negative labels actually correspond to real extragalactic transients.

SNGuess consistently provides a precision above 90% and approaching 100% for transients suitable for follow-up with

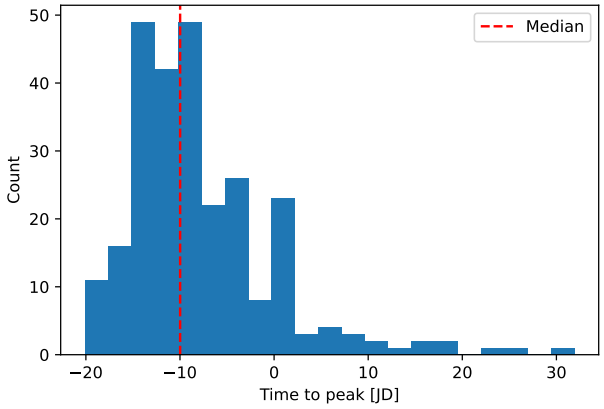

Fig. 9. Distribution of SNeIa candidate selection times with respect to light curve peak time. Two outlier points with extreme values were excluded from the plot: `ZTF21abbxtwv` and `ZTF18acydlux`. Both candidates show gaps of more than 30 days around peak magnitude in their light curves, which likely caused the peak-finding algorithm of BTS to give an incorrect detection time value. Similar observational irregularities are found for most SNe with a detection phase after peak.

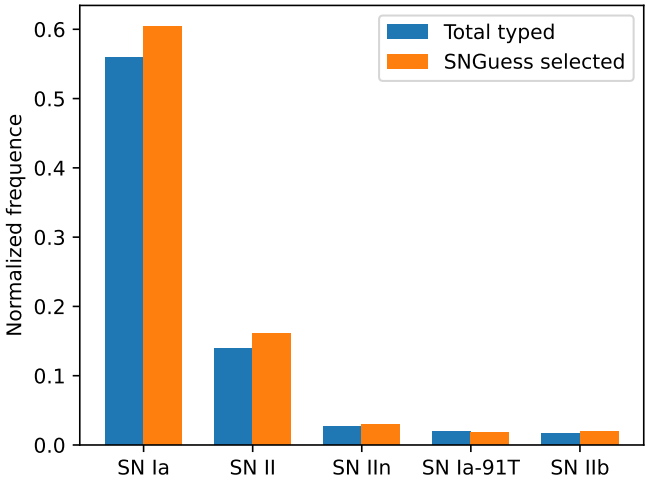

Fig. 10. Comparison between the distribution of all typed candidates vs. SNGuess-selected candidates in the test data set; limited to the five most common types. The distributions are normalized in order to make it easier to compare relative biases.

smaller scale facilities (magnitude <19.5) and of intermediate age (5+ detections). Our results show that `SNGuess` in this parameter range can, for example, efficiently reduce the amount of real-time scanning needed by human observers.

The numbers obtained in the lower two rows of the table in Fig. 8 correspond to the area with large statistical uncertainties (see Sect. 6), which will directly impact the feasibility of classification. In particular, due to the ZTF alert distribution mechanism, measurements in this brightness range might not be reported (even if eventually recognized by BTS). While the results in Fig. 8 are displayed in terms of the magnitude of last detection, the mean or typical magnitude of each alert is likely to be even fainter for the first two columns (as most real transients are increasing in brightness here), while it is likely to be brighter for the final column (where most transients have passed their peak).

Science programs looking to target young extragalactic transients for immediate follow-up could make use of `SNGuess` to carry out automatic observations already at first detection for objects with magnitudes in bands g or r at < 19.5 with reasonable precision (~75%). There is a drop in efficiency for young transients at larger magnitudes, and therefore `SNGuess` is more suitable here for use as an efficient first filter prior to a human decision regarding whether a transient should be targeted by the larger follow-up facilities.

Finally, the number of significant ZTF detections that caused an alert to be generated is not identical to the real age of a transient, which is typically the focus of scientific interest. As SNeIa have a reasonably well-defined rise time to peak light (18 ± 2 days), we can use the phase of candidate detection for transients later classified as SNeIa to estimate the time at which `SNGuess` would have selected a transient for follow-up. Figure 9 shows the distribution of candidate-selection times with respect to the time of light curve peak. Candidate selection time is defined as the detection time (`jd_last`) of the earliest alert for a particular candidate selected by `SNGuess`, and this is compared with the time of peak as catalogued by BTS. The median time of selection for SNIa candidates is close to 10 days before peak. The early selection purity can easily be improved through, for example, matching to external catalogs of nearby galaxies, a step which was not done here in order to maintain the general applicability of the basic method.

One should keep in mind that only a subset of the candidates that have been selected by BTS have also been assigned a type, and so by performing any analysis that involves confirmed types of candidates we are immediately subject to biases in the selection of BTS. In future work, we will enrich our data with further sources of labels, and also train and evaluate with data sets from other surveys, in order to have a more general assessment of the inherent type biases in the selection performed by `SNGuess`.

However, we can still compare the type distribution of all the candidates that have been classified with the type distribution of the candidates selected by `SNGuess` from the test data set. Figure 10 shows this comparison for the five most common types. We see that even though the selection performed by `SNGuess` follows the overall distribution of the selection of BTS, the former is slightly more biased toward type Ia and type II SNe.

10. Conclusions

`SNGuess` is a light-weight machine-learning system designed to assist in the selection of transients for potential follow-up observation from high-throughput alert streams generated by modern astronomical surveys. `SNGuess` was designed to work based on the alert content alone and without any external information. The output could therefore easily be further improved, for example, by taking into account redshift information and/or the rejection of galaxy core/AGN variability.

The average precision of `SNGuess` in terms of correctly predicting a candidate as being an interesting extragalactic transient is 88%, but this precision increases to 92% to 98% when examining transients brighter than 19.5 mag with five or more detections. This shows that `SNGuess` can be directly used for surveys looking to streamline the selection of interesting candidates for follow-up observations. The precision score obtained for transients in the same brightness class but with fewer detections is slightly lower, at ~75%. In addition, `SNGuess` is being used to search for faint, infant transients with few detections. However, the lower precision in this magnitude range (~30%) means that results need to be combined with visual inspection or catalog matching. Results in this magnitude range are affected by the increasing ZTF photometric uncertainties, which make both human and autonomous detection challenging.

As precision is a metric that is highly sensitive to class imbalance, we confirm that SNGuess produces good results according to other metrics, such as recall (0.821), $F1$-score (0.848), and MCC (0.728).

SNGuess was developed to help identify candidates of explosive transients for spectroscopic follow-up as detected by the ZTF, but can easily be retrained to parse transients detected by the LSST at the *Rubin* Observatory. TiDES is expected to be able to follow up all explosive transients detected by the LSST with magnitudes of $r_{\rm AB} \lesssim 22.5$ at peak (Swann et al. 2019). While the spectroscopic sample obtained from this may require further augmentation or combination with other samples (Carrick et al. 2021), an algorithm similar to SNGuess may be used to select appropriate candidates, particularly in the early part of the survey.

Our evaluation procedure is limited, as our sets of alerts are not fully labeled (we do not have types for most of the candidates) and that the true labels are only obtained from the BTS survey. Now that SNGuess is active we plan to carry out a dedicated follow-up of a subset of the SNGuess candidates in order to further evaluate, calibrate, and refine its results.

Our method may be improved by implementing a better strategy for hyper-parameter space search when training the classification model. Furthermore, active learning could be incorporated in our pipeline. This would allow the use of human-in-the-loop and automatic follow-up results of transient observations as feedback for improving the quality of the classification. It could also allow us to retrain the model with more transient candidate sources for labeled data. We would also like to explore different methods for estimating the performance of selection methods over partially labeled data sets by evaluating semi-supervised or unsupervised approaches.

The Python source code for SNGuess is fully available in the public repositories of AMPEL, with additional data sets and notebooks that generate the results shown in this article[9]. The results of the live instance of SNGuess that is running on the servers at DESY are freely accessible through the AMPEL Open API[10]. In addition, the repository contains an example notebook that shows how to obtain an SNGuess score for any ZTF candidate alert currently stored in the *DESY* archives[11].

Acknowledgements. We acknowledge the efforts of the BTS survey of the California Institute of Technology (Fremling et al. 2020; Perley et al. 2020), and of individual astronomers of the Humboldt-Universität zu Berlin in assigning types to transient candidates. Their results were instrumental to our supervised machine learning procedure. N. Miranda acknowledges the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS).

References

Abbott, T. M. C., Allam, S., Andersen, P., et al. 2019, ApJ, 872, L30
Alves, C. S., Peiris, H. V., Lochner, M., et al. 2022, ApJS, 258, 23
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2018, PASP, 131, 018002
Boone, K. 2019, AJ, 158, 257
Carrick, J. E., Hook, I. M., Swann, E., et al. 2021, MNRAS, 508, 1
Charnock, T., & Moss, A. 2017, ApJ, 837, L28
Chen, T., & Guestrin, C. 2016, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16 (New York, NY, USA: Association for Computing Machinery), 785
Chicco, D., & Jurman, G. 2020, BMC Genomics, 21, 6
Dai, M., Kuhlmann, S., Wang, Y., & Kovacs, E. 2018, MNRAS, 477, 4142
Dessart, L., Blondin, S., Hillier, D. J., & Khokhlov, A. 2014, MNRAS, 441, 532
Fremling, C., Miller, A. A., Sharma, Y., et al. 2020, ApJ, 895, 32
Hložek, R., Ponder, K. A., Malz, A. I., et al. 2020, arXiv e-prints [arXiv:2012.12392]
Ishida, E. E. O., Beck, R., González-Gaitán, S., et al. 2019, MNRAS, 483, 2
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Juric, M., Axelrod, T., Becker, A. C., et al. 2020, LSE-163, 71
Kaiser, N., Aussel, H., Burke, B. E., et al. 2002, in Survey and Other Telescope Technologies and Discoveries, 4836 (International Society for Optics and Photonics), 154
Kasen, D. 2010, ApJ, 708, 1025
Kennamer, N., Ishida, E. E. O., González-Gaitán, S., et al. 2020, in 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 3115
Kessler, R., Bassett, B., Belov, P., et al. 2010, PASP, 122, 1415
Kochanek, C. S., Shappee, B. J., Stanek, K. Z., et al. 2017, PASP, 129, 104502
Leoni, M., Ishida, E. E. O., Peloton, J., & Möller, A. 2022, A&A, 663, A13
Mahabal, A., Sheth, K., Gieseke, F., et al. 2017, in 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 1
Maoz, D., Mannucci, F., & Nelemans, G. 2014, ARA&A, 52, 107
Masci, F. J., Laher, R. R., Rusholme, B., et al. 2019, PASP, 131, 018003
Möller, A., & de Boissière, T. 2020, MNRAS, 491, 4277
Muthukrishna, D., Narayan, G., Mandel, K. S., Biswas, R., & Hložek, R. 2019, PASP, 131, 118002
Nordin, J., Brinnel, V., van Santen, J., et al. 2019, A&A, 631, A147
Nugent, P. E., Sullivan, M., Cenko, S. B., et al. 2011, Nature, 480, 344
Pasquet, J., Pasquet, J., Chaumont, M., & Fouchez, D. 2019, A&A, 627, A21
Perley, D. A., Fremling, C., Sollerman, J., et al. 2020, ApJ, 904, 35
Perryman, M. A. C., de Boer, K. S., Gilmore, G., et al. 2001, A&A, 369, 339
Qu, H., & Sako, M. 2022, AJ, 163, 57
Riess, A. G., Filippenko, A. V., Challis, P., et al. 1998, AJ, 116, 1009
Sánchez-Sáez, P., Reyes, I., Valenzuela, C., et al. 2021, AJ, 161, 141
Schäfer, P., & Leser, U. 2020, Data Mining and Knowledge Discov., 34, 1336
Swann, E., Sullivan, M., Carrick, J., et al. 2019, The Messenger, 175, 58
Tonry, J. L., Denneau, L., Heinze, A. N., et al. 2018, PASP, 130, 064505
Villar, V. A., Hosseinzadeh, G., Berger, E., et al. 2020, ApJ, 905, 94
Zhang, C., Liu, C., Zhang, X., & Almpanidis, G. 2017, Expert Syst. Applic., 82, 128

[9] https://github.com/nmiranda/SNGuess/tree/main/notebooks
[10] https://ampel.zeuthen.desy.de/api/live/docs
[11] Authentication for access to the DESY archives is currently mediated via GitHub. If you don't have a GitHub user account with an active membership in the ZTF or AMPEL organizations, please send an e-mail to ampel-info@desy.de in order to request access.

Appendix A: Data inputs and mathematical formalization

Chen & Guestrin (2016) define a data set with n examples and m features $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$, $|\mathcal{D}| = n, \mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}$ as input for their Gradient Boosted Trees learning model. We refer to $\mathbf{x}_i$, the *input vector*, and y_i, the *target value*.

When trying to set our photometric time-series data to comply with this format, we immediately face an issue: our data are composed of l time series of variable length $\mathcal{T} = \{\mathbf{t}_j\}$, where $\mathbf{t}_j = (t, v)^{p_j},\ t \in \mathbb{R}^+,\ v \in \mathbb{R},\ p_j = |\mathbf{t}_j|,\ j \in [1, l],\ l \in \mathbb{N}^+_*$. We denote an astronomical time series, or *light curve* with $\mathbf{t}_j$. In the previous definition, t is the *time stamp* of a particular light curve measurement, and v is the measurement value obtained at this time. Multiple $\mathbf{t}_j$ vectors can belong to the same astronomical candidate. This is due to the fact that photometric measurements are made for different band pass ranges, and measurements made in different bands do not necessarily have the same time stamp. In other words, a particular candidate has one light curve per band.

The approach we took to deal with this issue is to define a set of functions $\mathcal{F} = \{f_k\},\ k \in [1, m]$, and then to apply this set of functions to all of the light curves for each candidate. This means that the i-th candidate will be assigned an input vector $\mathbf{x}_i = (f_1(\mathcal{T}_i), (f_2(\mathcal{T}_i), \ldots (f_m(\mathcal{T}_i))$, where $\mathcal{T}_i$ is the set of light curves belonging to the i-th candidate. We refer to $\mathcal{F}$, our set of *feature functions*.

Appendix B: Data inputs, data structure formalization

There are several ways to store multivariate irregular-length time series as structured data. Let us suppose we have two candidates, for which we have measurements in at least one of g, r, and i pass bands. If we want to put these measurements in a data table format, our first idea would be to assign each one of the candidates to a single row of the table. This we call a *wide table* format. The result would then be of the form shown in Table B.1.

In this case, t_m would be the length of the longest light curve in our data set. The dimensions of this table, in general, would then be $n \times b \times t_m$, where n is the number of candidates and b is the number of pass bands. The advantages of this representation are its simplicity (all values are readily accessible) and the fact that it already has a format that can be accepted by most machine learning models (see Appendix A). In this case the table has one single obvious index: column **id**.

A wide table arrangement is useful when it is desirable to preserve the intuition present in general data science applications, that is, that each row of a table corresponds to an independent and identically distributed (i.i.d.) entity in the domain input. The disadvantage of this representation is the sparsity of the table, as in most cases we have many fewer measurements in some pass bands than in others. In addition, the length of the light curves varies greatly between candidates. Because of this we are forced to fill the missing values with zeroes.

Another possible configuration would be what is called a *nested table* configuration. In this arrangement, as in the wide table format, each row corresponds to a candidate. However, in this case the measurements are grouped in columns according to their type. Thus, we have one column for each band, and each value of the table contains the entire time series of the corresponding candidate at a certain band. Applying to our example case, the result would that displayed in Table B.2.

The advantage of this nested configuration is that the data are much less sparse, as a particular table value is empty only in cases when one of the candidates does not have any measurements for a given pass band. Furthermore, we have no constraints on the length of a given time series, as its full content is saved in a single cell of our table, and therefore we can easily store light curves of different lengths.

The disadvantage of this configuration is that we are no longer storing simple native data types in the table cells, but a composite series that has to be codified or serialized for storing its values, and there is no standard way of doing this. Also, we cannot perform mathematical operations with it directly, as it has no trivial matrix representation.

We can have a compromise between the two cases if we use what we call a *semi-wide table* format. In this arrangement, there is a row for each combination of candidate and time of observation. Additionally, it has one column for each one of the observation bands. This way, each time stamp is associated with one or more observation values, depending on how many bands were observed at the very same instant as indicated by the time stamp. Applying to our example case, the result would be the one shown in Table B.3.

Again, the advantage of this format is that we have no constraint over time series length, and that we have no issues storing time series of different lengths. Also, it is very easy to append new values to a table in this format, as any new value is just a new row in the table, and the rows do not need to be in any particular order.

The disadvantage of this format is again sparsity, as there are almost no cases where for a particular instant in time there are measurements in more that one pass band. In other words, almost no observations of a particular candidate in different bands share exactly the same time stamp. Therefore, many rows of the table will have zeroes in at least two of its columns. Another disadvantage is that we no longer have a single column index. In this case we may consider a composite index made out of tuples of the columns **id** and **time**.

Yet another possible configuration is what we call *long table* format. In this arrangement, a single row in the table corresponds to a single observation, and corresponds to a combination of candidate ID, time stamp, and band. Applying to our example, the result would be that shown in Fig. B.4.

The advantages of this configuration are the same as the previous one, with the difference that in this case the flexibility is even greater. We can arbitrarily append new measurements to the table as new rows, the rows do not need to be in any particular order, and we are not forced to fill with empty or zero values at all.

The disadvantage of this configuration is its length ($O(n \times b \times t_m)$) and the fact that if we want to set an index to the table this will have to be composite of more than one column. If we can guarantee that for a single candidate there is not more than one measurement at a given time, then we can set **id** and **time** as index. If not, then the index will have to be composed of the columns **id**, **time**, and **pass band**.

Another common format to use for this type of data is what we call *dictionary (or hashed index) format*. The idea is to have one single dictionary entry for each one of the candidates in our data set. The key of the entry is its **id**, and its value is a table with all the measurements for this candidate, as shown in Table B.5.

The advantages of this configuration are that, if we only (or most of the time) access by the candidate id, then it can be very efficient to work with this kind of structure. In this case, we retain the flexibility of the previous cases and we also have the ease of access of the wide table format.

Table B.1. Wide table format.

id	g_t1	...	g_tm	r_t1	...	r_tm	i_t1	...	i_tm
A	g_t1(A)	...	g_tm(A)	r_t1(A)	...	r_tm(A)	i_t1(A)	...	i_tm(A)
B	g_t1(B)	...	g_tm(B)	r_t1(B)	...	r_tm(B)	i_t1(B)	...	i_tm(B)

Table B.2. Nested table format.

id	g	r	i
A	g_t1(A) ... g_tA(A)	r_t1(A) ... r_tA(A)	i_t1(A) ... i_tA(A)
B	g_t1(B) ... g_tB(B)	r_t1(B) ... r_tB(B)	i_t1(B) ... i_tB(B)

Table B.3. Semi-wide table format.

id	time	g	r	i
A	t1	g(A, t1)	r(A, t1)	i(A, t1)
A	t2	g(A, t2)	r(A, t2)	i(A, t2)
...				
A	tA	g(A, tA)	r(A, tA)	i(A, tA)
B	t1	g(B, t1)	r(B, t1)	i(B, t1)
B	t2	g(B, t2)	r(B, t2)	i(B, t2)
...				
B	tB	g(B, tB)	r(B, tB)	i(B, tB)

Table B.4. Long table format.

id	time	pass band	value
A	t1	g	g(A, t1)
A	t2	g	g(A, t2)
A	t3	r	r(A, t3)
...			
A	tA	g	g(A, tA)
B	t1	g	g(B, t1)
B	t2	i	i(B, t2)
B	t3	g	g(B, t3)
...			
B	tB	r	r(B, tB)

Table B.5. Dictionary format.

dict key	dict value		
id(A)	**time**	**pass band**	**value**
	t1	g	g(A, t1)
	t2	g	g(A, t2)
	...	...	...
	tA	g	g(A, tA)
id(B)	**time**	**pass band**	**value**
	tA	g	g(A, tA)
	t1	g	g(B, t1)
	...	...	...
	tB	r	r(B, tB)

The disadvantage in this case is that, again, we are in the same case as with the nested table, where the values of the dictionary entries may not necessarily of the same type of the dictionary; and thus we need to unpack the data structure of the value in order to access the individual measurements. Finally, the data structure that we obtain once we apply our feature function set over our input data set has the format shown in Table B.6.

We have here a column for each one of our feature functions. This table may contain `null` values if the corresponding feature function cannot be applied to the light curve of that candidate. The **target** column contains the target values for classification.

Table B.6. Feature table format.

id	f_1	f_2	...	f_m	target
A	f_1(A)	f_2(A)	...	f_1(A)	target(A)
B	f_1(B)	f_2(B)	...	f_1(B)	target(B)

This can be an integer number representing one of the possible classes, or a Boolean value in the case of a binary classification task. This data structure is already in a format that is accepted by most machine learning models, as it can be easily used as an input matrix and target vector.

For `SNGuess` we opted for internally using a dictionary format for representation of light curve data. This is very easy to derive from the AVRO format used by ZTF to distribute data via Apache Kafka streams. Then, after extracting features from the light curves, we use the feature table format to represent the light curves. All of the statistical analysis and machine learning model training, evaluation and testing is done over feature data in this format.

Appendix C: Feature definition

cut_pp

Number of duplicate detections in the alert. By duplicate we mean one or more detections that have a time stamp value (floating point precision), in Julian date, that is the same as the time stamp of another detection in the alert. When a group of detections with the same time stamp is detected, the one with the highest real/bogus score is saved and the rest of them are labeled as duplicates and removed.

jd_det

Time stamp (floating point, Julian date) of the first (oldest) detection in the alert.

jd_last

Time stamp (floating point, Julian date) of the most recent detection in the alert.

ndet

Total number of detections in the alert.

mag_det

Brightness, in apparent magnitude, of the first (oldest) detection in the alert.

mag_last

Brightness, in magnitude, of the most recent detection in the alert.

t_lc

Time span (in Julian date, floating point) between the oldest and the most recent detection in the alert.

{proptype}_med

{proptype} can be rb, drb, distnr, magnr, classtar, sgscore1, distpsnr1, sgscore2, distpsnr2, neargaia, or maggaia.

Median values of the properties of a particular detection across all detections in the alert. These properties are mostly candidate-related and most of the time they do not vary between detections.

bool_pure

Indicates absence of upper limits after the first detection.

t_predetect

Time span between the latest upper limit prior to the first detection, and the first detection in the alert.

bool_peaked

Indicates if the light curve of the alert has reached peak brightness.

jd_max

Time stamp of the peak brightness detection (when bool_peaked is true).

mag_peak

Peak brightness, in magnitude, of the alert (when bool_peaked is true).

bool_rising

Indicates whether the light curve of the alert is rising in brightness.

bool_norise

Indicates whether there are no significant differences (within an error margin) between the peak magnitude of the alert and the detection magnitudes.

bool_hasgaps

Indicates whether there is a significant time gap (30 days) between consecutive detections in the alert.

slope_rise_{g,r}

When bool_norise is false, the slope of the rising part of the light curve of the alert.

slope_fall_{g,r}

Slope of the decline part of the light curve of the alert, when this decline has taken place over less than 30 days.

col_{det,last,peak}

Color (magnitude difference between bands) at first, last, and peak detections.