

Transient classifiers for Fink

Benchmarks for LSST

B. M. O. Fraga^{1, *}, C. R. Bom^{1,2}, A. Santos¹, E. Russeil³, M. Leoni⁴, J. Peloton⁴,
E. E. O. Ishida³, A. Möller^{5,6}, and S. Blondin^{7,8}

¹ Centro Brasileiro de Pesquisas Físicas, Rua Dr. Xavier Sigaud 150, Rio de Janeiro, Brazil

² Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Rodovia Márcio Covas, lote J2, Itaguaí, Brazil

³ Université Clermont-Auvergne, CNRS, LPCA, 63000 Clermont-Ferrand, France

⁴ Université Paris-Saclay, CNRS/IN2P3, IJCLab, 15 rue Georges Clemenceau, 91405 Orsay, France

⁵ Centre for Astrophysics and Supercomputing, Swinburne University of Technology, Mail Number H29, PO Box 218, 31122 Hawthorn, VIC, Australia

⁶ ARC Centre of Excellence for Gravitational Wave Discovery (OzGrav), John St, Hawthorn, VIC 3122, Australia

⁷ Aix Marseille Univ, CNRS, CNES, LAM, Marseille, France

⁸ European Southern Observatory, Karl-Schwarzschild-Straße 2, Garching 85748, Germany

Received 15 April 2024 / Accepted 10 November 2024

ABSTRACT

Context. The upcoming Legacy Survey of Space and Time (LSST) at the Vera C. Rubin Observatory is expected to detect a few million transients per night, which will generate a live alert stream during the entire ten years of the survey. This stream will be distributed via community brokers whose task is to select subsets of the stream and direct them to scientific communities. Given the volume and complexity of the anticipated data, machine learning (ML) algorithms will be paramount for this task.

Aims. We present the infrastructure tests and classification methods developed within the FINK broker in preparation for LSST. This work aims to provide detailed information regarding the underlying assumptions and methods behind each classifier and enable users to make informed follow-up decisions from FINK photometric classifications.

Methods. Using simulated data from the Extended LSST Astronomical Time-series Classification Challenge (ELAsTiCC), we showcase the performance of binary and multi-class ML classifiers available in FINK. These include tree-based classifiers coupled with tailored feature extraction strategies as well as deep learning algorithms. Moreover, we introduce the CBPF (Centro Brasileiro de Pesquisas Físicas) Alert Transient Search (CATS), a deep learning architecture specifically designed for this task.

Results. Our results show that FINK classifiers are able to handle the extra complexity that is expected from LSST data. CATS achieved $\geq 93\%$ precision for all classes except ‘long’ (for which it achieved $\sim 83\%$), while our best performing binary classifier achieves $\geq 98\%$ precision and $\geq 99\%$ completeness when classifying the periodic class.

Conclusions. ELAsTiCC was an important milestone in preparing the FINK infrastructure to deal with LSST-like data. Our results demonstrate that FINK classifiers are well prepared for the arrival of the new stream, but this work also highlights that transitioning from the current infrastructures to *Rubin* will require significant adaptation of the currently available tools. This work was the first step in the right direction.

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

1. Introduction

The advent of large-scale sky surveys has forced astronomy to enter the era of big data, with current experiments already producing data sets that challenge traditional analysis techniques (Hilbe et al. 2014). In this context, machine learning (ML) methods are almost unavoidable (e.g. Bamford et al. 2009; Baron 2019; Bom et al. 2022). For time-domain astronomy, the ability to quickly process data and obtain meaningful results has become critical due to current and upcoming projects such as the Zwicky Transient Facility (ZTF; Bellm et al. 2019) and the Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019), respectively. As will be the case for LSST, the ZTF project employs a difference imaging analysis pipeline that streams to community brokers, in the form of alerts, every detection above a given signal-to-noise threshold.

Brokers are subsequently tasked with filtering and analysing the data in detail, selecting the most promising objects for different science cases, and redirecting them to different research communities. FINK (Möller et al. 2021) is one of the official LSST brokers that has been selected to receive the raw alert stream from the beginning of LSST’s operations¹, which are expected to start in 2025. In the meantime, broker systems are operating and being tested with alerts from ZTF. FINK ingests and processes the stream, making use of several different science modules that contain cross-matching capabilities, ML classifiers, and user-specified filters (for details on FINK see Möller et al. 2021, and references therein).

The experience accumulated in the past few years in FINK with ZTF has been paramount for the design, development, and

¹ The other selected brokers are ALERCE (Förster et al. 2021), AMPEL (Nordin et al. 2019), ANTARES (Matheson et al. 2021), Babamul, LASAIR (Smith et al. 2019), and Pitt–Google.

* Corresponding author; bernardo@cbpf.br

fine-tuning of the broker services according to the needs of different scientific communities. Beyond the scientific results already reported (see, e.g., Aivazyan et al. 2022; Kuhn et al. 2023; Carry et al. 2024; Karpov & Peloton 2022), this partnership has enabled the development of a series of tools specifically designed to deal with the alert stream (Leoni et al. 2022; Biswas et al. 2023a; Russeil et al. 2022; Allam et al. 2023; Le Montagner et al. 2023; Biswas et al. 2023b; Karpov & Peloton 2023; Moller & Main de Boissiere 2022). Nevertheless, given the volume and complexity of the expected data, restructuring algorithms to transition from ZTF to LSST is a non-trivial task.

Fortunately, there have been similar situations in the past from which important lessons were learned. The astronomical transient community has a long-standing tradition of preparing for the arrival of data from a new survey by producing detailed light curve simulations and hosting data challenges. The Supernova Photometric Classification Challenge (SNPCC; Kessler et al. 2010) consisted of simulated data representing three supernova classes (Ia, Ibc, and II) whose light curve properties and rate were built to mimic the then upcoming Dark Energy Survey² (DES; Dark Energy Survey Collaboration et al. 2016). The challenge was open to professional astronomers, and received answers from ten different research groups. Despite not identifying a single classifier as being significantly better than the others and even when considering the undeniable differences between the generated simulations and the data finally observed by DES, the challenge was of crucial importance to prepare the necessary tools for the upcoming telescope.

This successful experience led to the development of a second simulated data set, this time in preparation for the arrival of LSST. The Photometric LSST Astronomical Time-series Classification Challenge³ (PLAsTiCC, Hložek et al. 2023) was proposed to the general public and received more than 1000 submissions. The data set enclosed 14 different transient classes in the training sample and 15 in the target sample⁴. Many algorithms were able to achieve similar results, while the numerically best ranked solution used data augmentation and boosted decision trees (Boone 2019). Beyond the official results presented within the time frame of both competitions, the most enduring legacy of both challenges has been the data sets they produced, which were subsequently used for the development of numerous methods and tools (e.g. Alves et al. 2022; Qu & Sako 2022; Malz et al. 2024; Allam et al. 2023, to cite a few).

All of these efforts constitute the astronomical branch of ML exploring ‘transfer learning’ (Pan & Yang 2010; Eriksen et al. 2020), a strategy commonly employed when labels are rare or expensive. It consists of training ML models in one domain (simulations) and using it to leverage information from the target one (real data). The most recent sample of photometrically classified supernova Ia from DES (Möller et al. 2024) was identified following this approach (Möller & de Boissière 2019).

Nevertheless, in all the previous attempts, simulations were generated in the format of full static light curves. The task of on-the-fly classification of nightly detected transient candidates had yet to be approached. The Extended LSST Astronomical Time-Series Classification Challenge⁵ (ELAsTiCC; Knop & ELAsTiCC Team 2023), developed by the LSST Dark Energy

Science Collaboration (DESC) was created to allow for a concrete assessment of the performance of different stages of the communication pipeline. This includes simulating an alert stream to be received by brokers, the ingestion and analysis of the alerts using ML-based classifications by the broker teams, and reporting back such scores to DESC. Recently, one of the instances of this data set was used by Cabrera-Vives et al. (2024) to test a multi-class classifier using transformers and by Khakpash et al. (2024) in the study of stripped-envelope supernovae. In this work, we use the first version of the ELAsTiCC data set, streamed between September 2022 and January 2023⁵ in alert format, to stress test the performance of the entire FINK infrastructure in an LSST-like data scenario.

This paper is organised as follows: In Sect. 2, we present the data set, its properties, and our chosen experiment design. Sect. 3 presents with more details how FINK works, showing the preparations for LSST. Section 4 presents the metrics we used to evaluate the classifiers. Sect. 5 gives more details about the construction and training of ML models, while Sect. 6 presents an evaluation of these models on a blind test set. We show how some classifiers can be combined to improve stand-alone performances in Sect. 7, and Sects. 8 and 9 contain discussions and our conclusions, respectively.

2. The ELAsTiCC data set

The Extended LSST Astronomical Time-series Classification Challenge (Knop & ELAsTiCC Team 2023) was designed to test broker systems and classification algorithms when applied to a state-of-the-art data set that mostly resembles LSST alerts. It emerged from the experiences accumulated during two previous challenges, SNPCC (Kessler et al. 2010) and PLAsTiCC⁶ (Hložek et al. 2023), and it is lead by LSST DESC. Its first objective was to test the brokers infrastructure capability of ingesting and processing a real-time alert stream. The second goal was to enable the evaluation of ML classification algorithms.

In the first instance of the challenge (hereafter ELAsTiCCv1), two different simulated data sets were provided: a static set of full light curves (one light curve per astrophysical source) was made available in 18 May 2022 and an alert stream corresponding to three years of LSST operations was streamed between September 2022 and January 2023. Both data sets were simulated using SuperNova ANALysis package (SNANA; Kessler et al. 2009) and contained 19 classes divided into five broad categories⁷ (SN-like, periodic, non-periodic, long, and fast). Light curves, comprising detections and forced photometry in the LSST broad-band filters $\{u, g, r, i, z, Y\}$ were provided (for details on how the simulation was generated we refer to the project website⁵). In order to isolate the performance of our classifiers, avoid issues due to the data generating process and circumvent the need to transform full light curves into alerts, we chose to use here only the streamed data set⁸. In this format, each alert holds the photometric information obtained at a given day, as well as the previous photometric history of that point in space. Thus, at each new light curve point, a new alert is generated that differs in one point from previous alerts from that same source.

² <https://www.darkenergysurvey.org/>

³ <https://www.kaggle.com/c/PLAsTiCC-2018>

⁴ The extra class was intended to test the resilience of algorithms to anomalous observations.

⁵ https://portal.nersc.gov/cfs/lsst/DESC_TD_PUBLIC/ELASTICC/

⁶ <https://www.kaggle.com/c/PLAsTiCC-2018>

⁷ Full taxonomy can be found at <https://github.com/LSSTDESC/elasticc/blob/main/taxonomy/taxonomy.ipynb>

⁸ A second version, ELAsTiCCv2, was released in mid-2023, with updates in photometric redshift model and cadence. The second version was not used in this work.

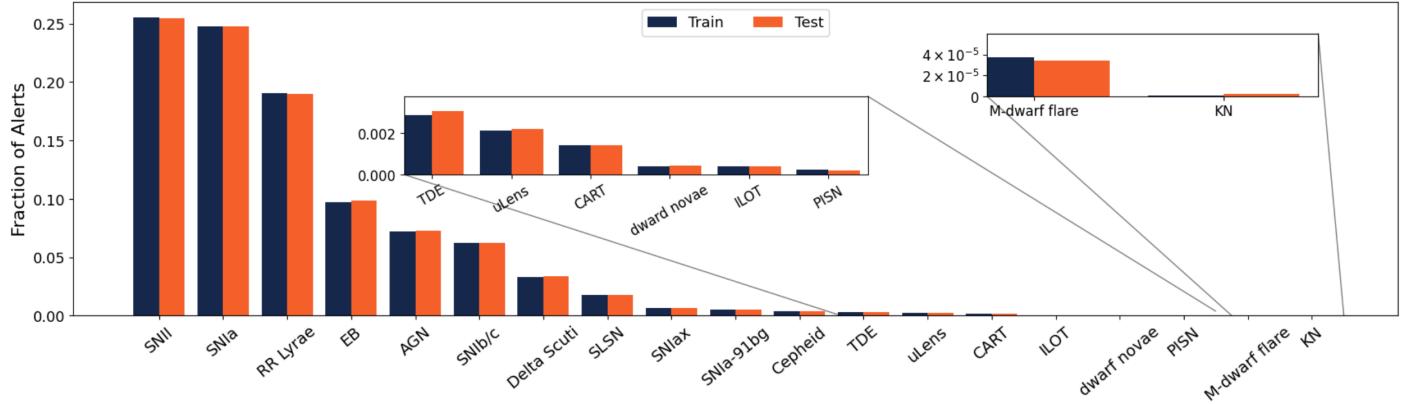


Fig. 1. ELAsTiCC class distribution for our training (dark blue) and test (orange) sets.

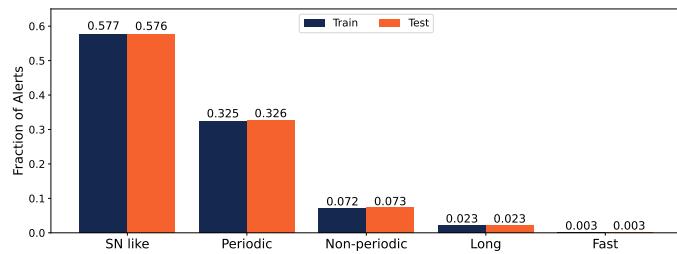


Fig. 2. ELAsTiCC broad class distribution for our training (dark blue) and test (orange) sets.

We selected one third of all unique objects as training sample for all our algorithms (1 417 375 distinct objects corresponding to 17 214 758 alerts). The remaining objects were used as a test set to evaluate the performance of our classifiers (2 874 008 distinct objects corresponding to 34 891 855 alerts).

The class distributions for the alerts in our train and test sets are shown in Figs. 1 (unique classes) and 2 (broad classes). We note that in our experiment design, the population distributions between classes are similar in both samples, with supernova types Ia (SNIa) and II (SNII) comprising almost half of the alerts, and the fast class being the least represented one. Overall, the distributions for the unique and broad classes are extremely imbalanced, which could pose a problem for multi-class classifiers.

Each alert package included both, light curve data (`mjd`, `fluxcal`, `fluxcal_err`, band-pass filter: `filtername`) as well as object metadata, comprised of properties such as position, Milky Way extinction and estimated photometric redshift, among others. In Fig. 3 we show the distributions for two of these properties for the training and test set: the Milky Way extinction (`mwebv`, left) and photometric host galaxy redshift (`hostgal_zphot`, right). Approximately 91% of the objects in the training and test sets have a photometric redshift available. The distributions are similar, with both photometric redshift distributions displaying a double peaked structure.

Fig. 4 shows the distribution of number of detection points per alert with and without forced photometry (left), as well as the global number of detection points in each bandpass (right), for the training (top) and test (bottom) sets. It can be seen from the left panel that the distribution of light curve sizes considering only the detections is strongly peaked around 10 detections, dropping heavily after that, with very few alerts having more than 50 measurements in both the training and test sets.

Including forced photometry, the peak is still present around 10 detections, although less pronounced; the decrease is also less steep, as expected. Overall, the distribution is similar for both sets, with the majority of alerts having less than 50 points, even including forced photometry.

The distribution of detections per passband (right column of Fig. 4) is similar for the training and test sets: the redder the band, the larger the maximum number of detections. The exception to this is the z band, which do not appear in longer light curves. This is especially important for classifiers that rely on colours or use only specific passbands. Nevertheless, this feature is a direct consequence of the chosen survey strategy, and it is reasonable to expect that the real data will also hold differences in number of detections on each band. Thus, it is paramount to assess the robustness of classifiers in this scenario.

3. Fink infrastructure

Since 2019, FINK has been processing alert data from the ZTF public stream. Not only the ZTF data rate is lower than the one intended for LSST, but the schema and content of the alert packets from each stream are also different. Therefore, the ELAsTiCC challenge was an opportunity to show that the Fink architecture can scale in terms of data volume and that it is possible adapt the current classifiers, or create new ones, to a new schema with different incoming information.

FINK operates in real time on large-scale computing infrastructures. For the ZTF processing, the incoming data stream is provided by the ZTF Alert Distribution System (ZADS; Patterson et al. 2018) that runs a Kafka instance on the DigitalOcean cloud infrastructure (USA). FINK is currently deployed on the VirtualData⁹ cloud infrastructure (France), and makes use of distributed computing to split the incoming stream of alerts into smaller chunks of data to be analysed independently over many machines in parallel. For LSST, FINK will be deployed at CC-IN2P3 (Centre de Calcul de l’Institut National de Physique Nucléaire et de Physique des Particules), which will have a local copy of LSST data that can be efficiently exploited for internal cross-match needs.

What is deployed for ZTF typical rates (of the order of 200 000 alerts per night) can be easily scaled to ELAsTiCC rates (of the order of 1 000 000 alerts in a few hours every night) by adding more machines. In practice for the challenge, we use a total of 33 cores for all operations (listening to the

⁹ <https://virtualdata.fr/>

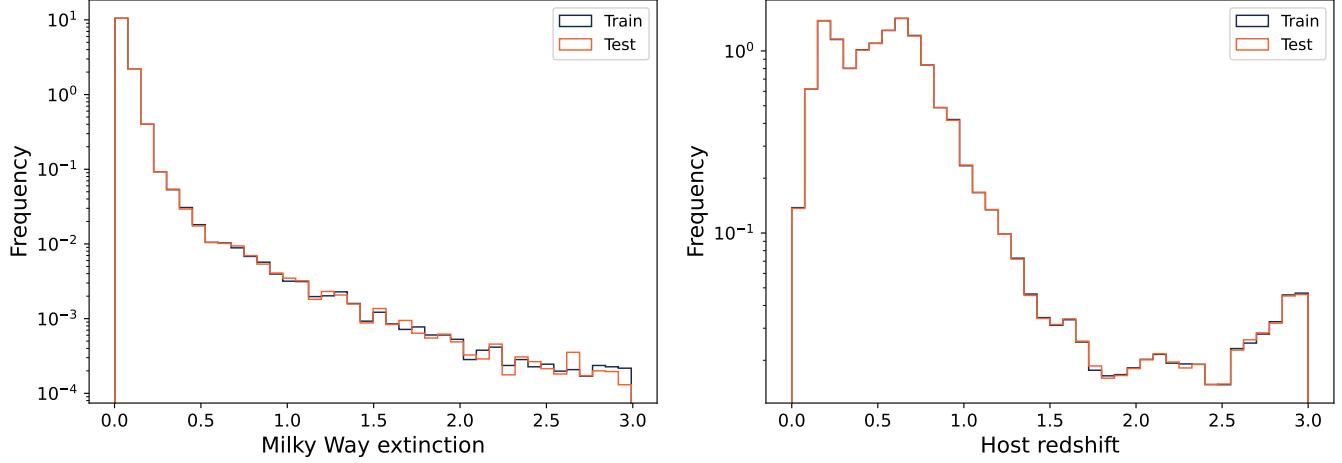


Fig. 3. Distribution of milky way extinction (left) and host galaxy photometric redshift (right) for the training (dark blue) and test (orange) sets.

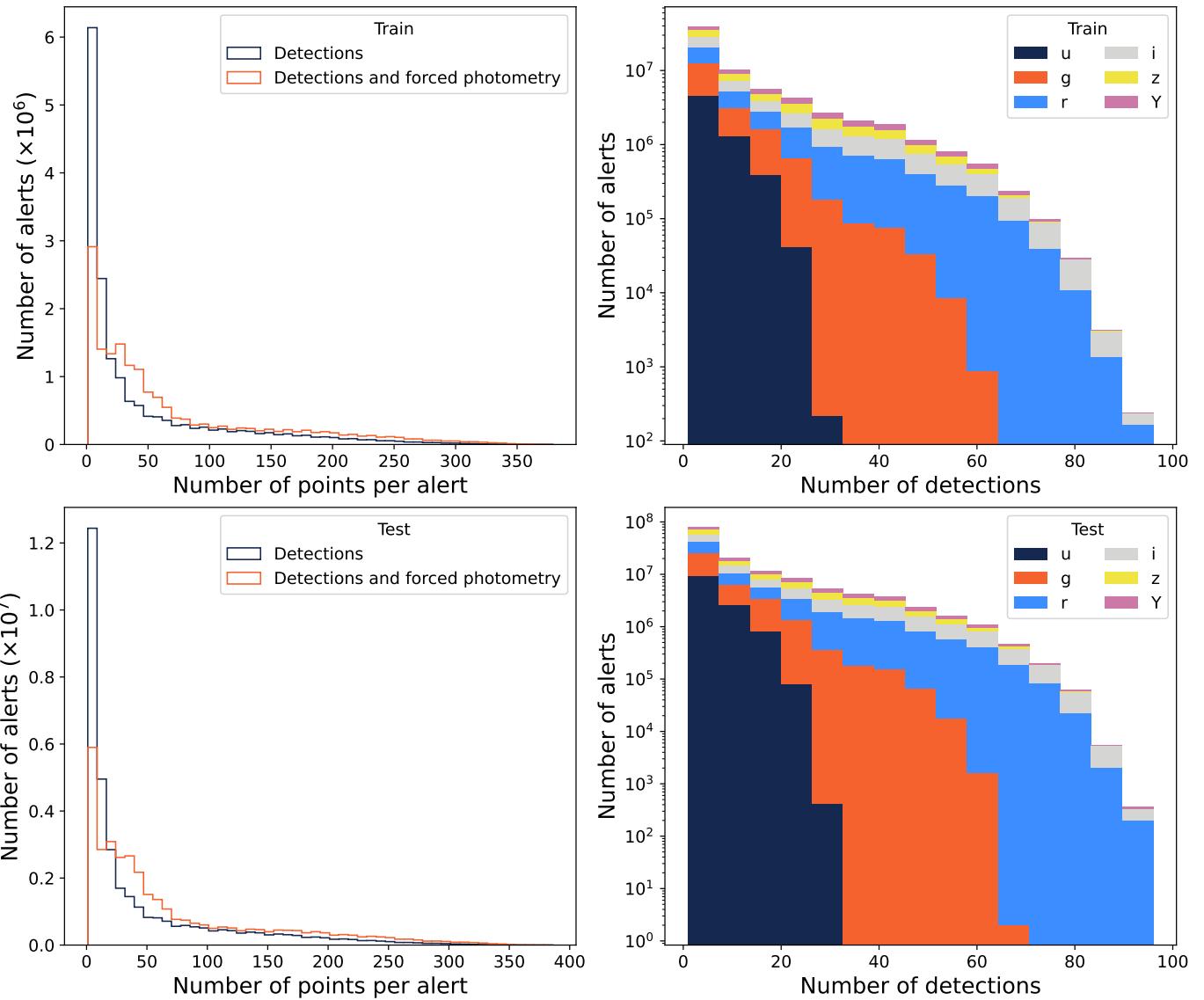


Fig. 4. Distribution of light curve length in the training (top) and test (bottom) samples. On the left, the total number of points including only detections (blue) and with forced photometry (orange). On the right, the distribution of detections per filter.

incoming stream, processing it and sending back results). Moreover, the processing was done on the same platform, alongside the real-time processing of ZTF alerts.

For the ELAsTiCC challenge, FINK operations consisted in three steps, done in real time: (1) decoding the incoming alert stream, (2) applying all classifiers on every alert, and (3) redistributing all enriched alert packets to the DESC team in the USA. The first and last steps are under the responsibility of the FINK engineering team, while the second part involves classifiers provided by the community of users. The development of science models within FINK is completely driven by the user community¹⁰, who is responsible for development and validation of its outputs. Each team responsible for a classifier typically provides a pre-trained model, as well as the snippet of code necessary to run the inference on one alert¹¹, and the FINK engineering team integrates it into FINK for streaming processing at scale.

Given the large volume of data, we developed a new service for the community during the challenge, the FINK Data Transfer service¹². The ELAsTiCC training set was made available via this service, which enables easy distribution of large volumes of data for many decentralised users. It also allows users to select any observing nights, apply selection cuts based on alerts content, define the content of the output, and stream data directly anywhere. Since the start of the challenge, more than one billion alerts have been streamed via this service.

For the experiments described in this work, nine classifiers were deployed (see Sect. 5). Some of the classifiers used in the challenge are also classifiers used to process the ZTF alert stream. The differences in data rate, schema and the data itself (available filter bands, cadence and magnitude limit, among others) between the ZTF stream and the simulated ELAsTiCC data made the transition less easy than we had originally anticipated. From this point of view, the lessons learned from the ELAsTiCC challenge were paramount in preparing for the arrival of the LSST alert stream.

Throughout the classifier design phase and the challenge itself, we monitor classifier performance in terms of throughput (alerts processed/second/core) and memory usage (MB/core). LSST will impose stringent requirements on throughput (we expect a continuous flow of 10 000 alerts per exposure, i.e. every 30 seconds), while our computing infrastructure imposes constraints on memory usage (cores with 2 GB RAM each). As an example, while processing in real-time the first version of the challenge (using 24 cores in parallel), 82.2% of alerts were classified in less than 30 seconds (the time between when an alert enters FINK and when it is fully classified by the nine classifiers), 90.5% in less than a minute, and over 99.9% in less than 10 minutes. Delays larger than expected are partly due to processing (classifier versions and performance have evolved over time), but the high values are mainly explained by human intervention in the computing infrastructure, which interrupted operations while we were processing live data. These interventions are not expected during normal LSST operations. Regular measurements of Fink operations performance (profiling) are analysed to check that requirements are met.

¹⁰ Users can propose science modules following instructions at <https://fink-broker.org/joining/>

¹¹ The FINK engineer team provides examples to manipulate alert data, and all code and models can be found online at <https://github.com/astrolabsoftware/fink-science>

¹² <https://fink-portal.org/download>

4. Metrics

In a classification task, several metrics can be used to assess the performance of the classifier, such as the receiver operating characteristic (ROC) and precision-recall curves and the confusion matrix. These are built from the precision (P), recall (R, also called the true positive rate, or TPR), and false positive rate (FPR), which in a binary classification are defined as

$$\begin{aligned} P &= \frac{TP}{TP + FP}, \\ R &= \frac{TP}{TP + FN} \\ FPR &= \frac{FP}{FP + TN}, \end{aligned} \quad (1)$$

with TP(N) the number of true positives (negatives) and FP(N) the number of false positives (negatives). Precision can be understood as the purity of the predictions, while Recall is its completeness or efficiency, and the FPR is the ratio of wrongly classified objects of the negative class (also known as the false alarm rate). The output of a binary classifier is a the probability of a light curve belonging to the class of interest. Thus, the quantities on Eq. (1) will depend on a chosen probability threshold. By varying this threshold, one can obtain curves for recall versus FPR (ROC) and precision versus recall. The area under these curves (AUC) can be used as a metric to assess the performance of the classifier, where a perfect classifier would have an AUC of 1, and an AUC of 0.5 for the ROC corresponds to a random classifier.

The binary case can be straightforwardly extended to the multi-class case by using a one versus all approach, wherein the problem is split into a binary classification case per class, gathering every class except one as the negative class. In this case, there will be a ROC and a precision-recall curve for every single class.

Another popular metric for classification task is the confusion matrix, a square matrix showing the number (or percentage) of objects classified in every combination true/predicted class. Both metrics are used to access the efficiency of classifiers described in this work.

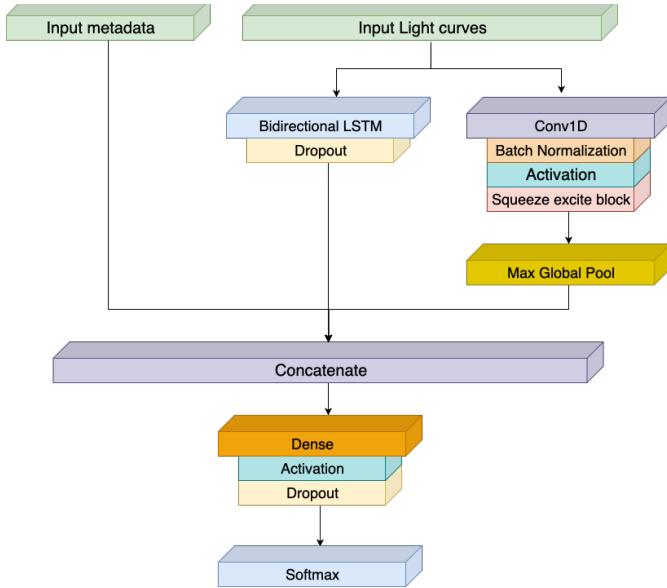
5. Classifiers in Fink

For this work, we used 4 algorithms for classification. These are meant to represent the two possible applications of the broker infrastructure: broad classifiers using deep learning (DL), represented by CATS (Sect. 5.1) and SuperNNova (Sect. 5.2), and binary classifiers using feature extraction and tree-based algorithms, represented by the Early SNIa (Sect. 5.3) and the SLSN (Sect. 5.4) classifiers.

The class specific classifiers (Sects. 5.3 and 5.4) represent the most common scenarios in which a team interested in a specific class can profit from the broker infrastructure. As an example, we used a tool that incorporates physical assumptions about thermodynamic behaviour of the transient (Russeil et al. 2024b) in combination with traditional ML algorithms. The same infrastructure can be used to incorporate other requests coming from science teams.

5.1. CATS

Recurrent neural networks (RNNs) are a class of models adapted to work with sequential data. They do so by constructing hidden

**Fig. 5.** Illustration of the CATS architecture.

states that carry information from the previous part of the sequence (Rumelhart & McClelland 1987; Schmidt 2019). One of the main problems found in training RNNs was the vanishing gradients: When the input sequence was long, the successive derivatives during backpropagation tended to erase the gradient (Bengio et al. 1994), which then cause later time steps to be ‘disconnected’ from earlier ones (i.e. a low memory capacity). Long-short term memory units (LSTM; Hochreiter & Schmidhuber 1997) were designed to avoid the vanishing gradient problem by keeping not only a hidden state but also a memory state across all time steps. By using gates, the network can learn what information needs to be kept, removed, or inserted to the memory vector, increasing the RNN’s memory capacity.

The CBPF Alert Transient Search (CATS) was built by starting with a base network very similar to the Multivariate LSTM Fully Convolutional Network (MLSTM-FCN; Karim et al. 2019), using squeeze and excitation blocks; we use bidirectional LSTM layers and a series of fully connected layers before the output, adding a dropout layer after each of those. This architecture was shown to perform very well in several different time series tasks¹³, and the base architecture is shown in Fig 5. We then performed a hyperparameter search using keras-tuner, searching for the best configuration of number of convolutional, LSTM and Dense blocks, convolutional filters, LSTM and Dense layers’ units and activation functions. This search was done using a subset of 10% of the unique objects in our training sample to speed up the process. The final architecture consisted of one convolutional block with 32 filters, two bidirectional LSTM layers with 400 and 500 units respectively, and two fully connected layers with 64 and 96 units. All convolutional and fully connected layers are followed by a Rectified Linear Unit (ReLU) activation.

Our inputs were the normalised flux and errors, (where both are normalised per light curve), and the difference in days from the first available light curve data point for that transient and the current alert detection, using forced photometry when available. To that, the filter was added as an integer in the range [1, 6] corresponding to the LSST passbands, $[u, g, r, i, z, Y]$. All inputs were right-padded to match the longest light curve, so that the shape of

Table 1. CATS results for the validation sample across all folds.

	ROC AUC	PR AUC	Precision	Recall
SN-like	0.99 (0.0001)	0.99 (0.0002)	0.97 (0.0022)	0.99 (0.0013)
Fast	0.99 (0.0013)	0.80 (0.047)	0.85 (0.092)	0.72 (0.017)
Long	0.96 (0.0012)	0.68 (0.0041)	0.82 (0.032)	0.47 (0.041)
Periodic	1.00 (0.00001)	1.00 (0.00003)	1.00 (0.0002)	1.00 (0.0001)
Non- Periodic	1.00 (0.00003)	1.00 (0.0005)	0.97 (0.003)	0.96 (0.005)

Notes. Reported are the means with standard deviations in parentheses. ROC AUC and PR AUC are the area under the curve for the ROC and precision-recall curves, respectively, while the values for the precision and recall are taken from the confusion matrix averaged over all folds.

the input is (395, 4) (top-right box in Fig. 5). To this, we concatenated the Milky Way extinction (`mwext`), host galaxy redshift (`hostgal_zphot`), and the redshift of the transient (`z_final`), when available, plus their errors (Input metadata) before passing the result to the fully connected layers. Using these features gave the best results when testing with a smaller subset of the data, slightly better than using only extinction and redshift of the host.

We performed a K-fold cross-validation process to assess the robustness of our model. It consists of splitting the data into k groups (folds), and performing k iterations of training; at each iteration, one of the folds is used for validation, while the other $k - 1$ are used for training.

We used five folds; the unique object identifier (`diaObjectID`) was the parameter used to split the data, thus making every light curve of a single object either part of the training or validation set, with no contamination between them.

The model was trained for 15 epochs at each fold, with the NAdam optimizer (Dozat 2016), on a cluster with eight NVIDIA RTX A6000. At each fold, the model where the validation loss was lowest was selected to perform the predictions. The architecture and training were implemented using TensorFlow 2 (Abadi et al. 2015).

Table 1 shows the metrics derived from the validation sets of the k-fold. The model performed satisfactorily, having a mean ROC AUC above 0.95 for all classes, and a mean precision-recall AUC above 0.8 except for the Long class, with little variance between the folds. Although the mean Precision for the Long class is approximately 80%, the Recall is less than 50%, due to half of the Long alerts being classified as SN-like events. The major type behind this confusion are Super luminous Supernovae (SLSN), which could resemble some types of supernovae when looking at short enough light curves. On the other hand, the 20% rate of false positives of the Long class is dominated by SNII, which is also the dominant subclass; however, contamination by SNIa (the second most dominant subclass) is negligible. SNII with an extended plateau can be very similar to SLSNe, while SNIa are generally brighter and exhibit a characteristic feature in redder passbands, thus rendering their classification less controversial. Despite such caveats, results shown in Table 1 confirm that CATS will be able to provide reliable classification results even under challenging data scenarios.

¹³ See <https://paperswithcode.com/task/time-series-classification> for benchmarks on several different time series tasks.

5.2. SuperNNova

SUPERNNNOVA (SNN; Möller & de Boissière 2019) is a DL light curve classification framework based on RNNs. SNN makes use of fluxes over different band-passes and their measurement uncertainties over time for classification of time-domain candidates in different classes. Additional information such as host-galaxy redshifts and Milky Way extinction and their errors can be included to improve performance.

SUPERNNNOVA includes different classification algorithms, such as LSTM RNNs and two approximations for Bayesian Neural Networks (BNNs). Here, we only use the LSTM architecture, which was also used for classification of type Ia SNe (SNe Ia) in the Dark Energy Survey (Möller et al. 2022, 2024). For work on BNNs in the context of *Rubin* see Möller & Main de Boissière (2022).

We used the LSTM RNN to process the photometric time series and produce a sequence of hidden states. The sequence is condensed to fixed length through mean pooling. Finally, a linear projection layer was applied to obtain an N -dimensional vector, where N is the number of classes. A softmax function was used to obtain probabilities for an input to belong to a given class. During training, we randomly truncated light curves to improve the robustness of the classifier with partial light curves. SNN is trained to optimize accuracy of balanced classes.

We grouped observations in each passband within a given night. If a given filter is not observed, we assigned it a special value to indicate that it is missing. To deal with irregular time sampling, we added a delta time feature to indicate how much has elapsed since the last observation. We used the default configuration of SNN with the normalisation as in (Möller et al. 2022) and added redshift and Milky Way extinction as additional features for this work.

Classification probabilities from SNN can be used to select a sample by performing a threshold cut or by weighting the contribution of candidates by their classification score (Vincenzi et al. 2023, 2024; DES Collaboration 2024). In this work we evaluated only the selection of samples using a probability cut set to $p > 0.5$ or in multi-class classification, the largest probability for all classes.

We trained binary and broad class models. For the binary classification, we balanced the training set light curve numbers between the target class and other classes (randomly sampled). For the broad classifier we did also a balanced training set. However, the fast, long and non-periodic classes have considerably smaller numbers than the SN and periodic classes as shown in Fig. 2; the balanced training set for broad classification was ≈ 2000 events per class.

We split the data set in 80% for training, 10% for validation and 10% for performance evaluation. In Table 2, we show the performance metrics for the different models obtained for this test set.

For the broad model, we find that the lowest accuracy classes are SNe and Long events. This may be due to the time-range of the light curve provided for classification as some Long events such as PISN and SLSNe during a reduced time range may resemble shorter time-scale SNe. We also find small confusion between Fast and Periodic transients that may be due to the same effect.

For the binary models, we also find that the model targeting Long events has lower accuracies than the other binary classification models. This may be due to the small data set used for training, which is composed of only thousands of light curves.

Table 2. SUPERNNNOVA results for a blind test sample.

Class	Accuracy	ROC AUC	Precision	Recall
SN-like	95.58	0.9890	96.19	93.38
Fast	98.15	0.9958	99.98	96.3
Long	84.58	0.9270	83.85	87.76
Periodic	99.93	0.9999	99.98	99.9
Non-Periodic	99.4	0.9999	99.98	99.07
Broad	88.52	–	68.15	80.0

Notes. Results obtained on complete light curves using an independent test set from the training sample. All rows except the last one show the metrics for a binary target versus other types.

As expected, small and unbalanced training sets impact the performance of this framework, which was built for accurate classification with large and representative training sets. Further discussion on the performance of SUPERNNNOVA with respect to training set size can be found in Möller & de Boissière (2019).

5.3. Early supernova Ia

Supernovae Ia (SNIa) were first used as standard candles in cosmological analysis in the end of the 20th century, when they provided the first evidence of the Universe's current accelerated expansion (Riess et al. 1998; Perlmutter et al. 1999). Since then, large efforts have been devoted to the compilation of large SNIa samples, in the hope they can help unravel details about the behaviour of dark energy (e.g. Aleo et al. 2023; Möller et al. 2024).

Despite the undeniable impact large scale sky surveys can imprint on SN cosmology results, such potential is strongly dependent on our ability to distinguish SNIa from other types of SN-candidates (see, e.g., Ishida 2019, and references therein). In the context of real data, labelling is an extremely expensive process and ideally we would like to discover such transients early enough so they are still sufficiently bright to allow spectroscopic classification.

For ZTF processing, FINK has an early supernova Ia classifier (hereafter, EarlySNIa) based on independent feature extraction for each of the two ZTF passbands and a random forest classifier enhanced by active learning (for a complete description see Leoni et al. 2022). Its goal is to identify SNIa before or at peak, to optimally allow spectroscopic follow-up resources to be allocated. The module has been successfully reporting EarlySNIa candidates to the Transient Name Server (TNS) since November/2020. However, in the context of LSST, with 3 times more passbands and a considerable sparser cadence, the module required significant modifications.

In order to allow classification with a lower number of points per filter and, at the same time, take into account colour information, we implemented the RAINBOW multi-band feature extraction method proposed by Russeil et al. (2024b) to comply with the characteristics of the new data set. A parametric model was simultaneously fitted to the light curves in all available passbands, and the best-fit parameter values were used as features, thus given as input to the random forest classifier (Ho 1995). Assuming the transient can be approximated by a black-body, the framework combines temperature and bolometric light curve models to construct a 2D continuous surface. This approach

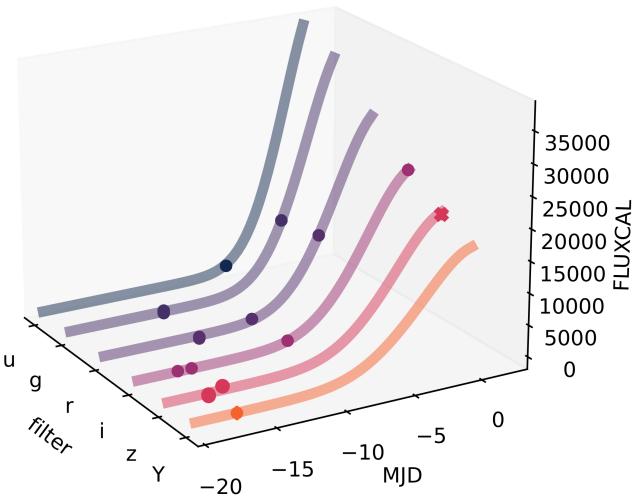


Fig. 6. Example of light curve fit for an early SNIa performed with RAINBOW. The estimated light curve behaviour in each filter (full lines) was found by using the photometric history within `alertId = 244687224069` (circles). The most recent observation within that alert (cross) was added subsequently to illustrate the agreement between the estimation and measured value in the z -band.

enables early description even when the number of observations in each filter is significantly limited (for more details see [Russeil et al. 2024b](#)).

The preprocessing for each alert included (i) averaging all observations within the same night; (ii) removing any intra-night flux measurements lower than -10 ($\text{FLUXCAL} > -10$); (iii) requiring a minimum of seven points per object, in any filter, including forced photometry; and (iv) ensuring that intra-night flux measurements are consistently increasing within at least two passbands. Thus, considering that only rising alerts survived such selection cuts, we described the bolometric evolution of our light curves with a logistic function of the form

$$f(t) = \frac{\text{amplitude}}{1 + \exp\left(-\frac{t-t_0}{\text{rise_time}}\right)}, \quad (2)$$

where `rise_time` is the characteristic time of rise, `amplitude` is the amplitude, and t_0 describes a reference time that corresponds to the time at half of the rising light curve. The temperature evolution was described with a falling logistic function of the form

$$T(t) = \text{Tmin} + \frac{\text{delta_T}}{1 + \exp\left(\frac{t-t_0}{\text{k_sig}}\right)}, \quad (3)$$

where `delta_T` is the full amplitude of temperature, `Tmin` denotes the minimum temperature reached, `k_sig` describes a characteristic timescale, and t_0 is a reference time parameter that corresponds to the time at half of the slope. We note that t_0 from the bolometric and temperature descriptions is used as a single common parameter whose role is to anchor functional behaviour, but it holds no physical meaning without a reference point, and thus it was not included in our final parameter set. Beyond these, RAINBOW also returns a measurement of the quality of the fit (`reduced_chi2`) and the maximum measured flux (`1c_max`). Figure 6 illustrates the capability of the method in extrapolating the behaviour of a rising light curve even when the number of rising points is sparse. In this figure, the model (full lines) was fit using only the history within the alert (circles). The most

recent observation (cross) was added subsequently as a way to compare the measurement with the prediction.

As a result, each alert is represented by seven values. To this we added the mean signal-to-noise ratio ($\text{FLUXCAL}/\text{FLUXCALERR}$); the number of points in all passbands before intra-night smoothing¹⁴ (`nobs`); separation between the host and the transient (`hostgal_snsep`) and the host photometric redshift (`host_photoz`). Thus resulting in 11 parameters per alert.

A total of 5 334 911 alerts (791 779 objects) survived selection cuts, which corresponds to $\approx 31\%$ of the alerts in the full training sample described in Sect. 2. Among them, SNIa ($\approx 34\%$), SNII ($\approx 28\%$), RR Lyrae ($\approx 11\%$), Ibc ($\approx 8\%$) and AGN ($\approx 8\%$) were the most frequent classes. From these, we selected a sample of 205 228 alerts (30 000 objects) for training a supervised ML model. In this sample, the distribution of the major classes was unchanged. We note that we report populations of individual classes to better illustrate the composition of the sample, but in reality, we trained a binary classifier whose positive class corresponded to approximately a third of the full sample. The remaining 5 131 798 alerts (761 779 objects) were used for validating the classifier results. To avoid an information leak, we made sure to place all alerts from the same object either in the training or in the validation sample.

We trained a random forest model using a scikit-learn ([Pedregosa et al. 2011](#)) implementation, using 50 trees, maximum depth of 15 and set the minimum number of alerts per leaf to be 0.01% of the training size. In the validation sample, this resulted in $P = 0.70/R = 0.70$, considering a probability threshold of 0.5.

5.4. Superluminous supernovae

Superluminous supernovae (SLSN) are SNe whose peak optical luminosity exceeds -21 mag (see, e.g., [Moriya et al. 2018](#) for a review). Their rise times can vary between ~ 20 days to more than 100 days for some events, but their post-maximum decline rates are either consistent with ^{56}Co decay (at least initially), or significantly faster. This suggests that some SLSN can have a possible thermonuclear origin, which given the mass required can only be explained by population III star origin. At such mass, a pair production mechanism triggers and results in an instability that eventually leads to the collapse of the core, hence pair-instability SNe (PISN). Currently, there have been no direct confirmation of their existence, although good candidates have been reported ([Moriya et al. 2022](#); [Pruzhinskaya et al. 2022](#)). The discovery and characterisation of a PISN would significantly impact our understanding of the connection between chemical evolution and structure formation in the Universe ([LSST Science Collaboration 2009](#)).

Given that predicted light curves of SLSNe and PISNe can have a similar morphology, we use a common classifier for both and call it the SLSN classifier. Its core implementation is based on a feature extraction of normalised alerts followed by a random forest classification. For each filter we computed the following set of features: maximum and standard deviation of the flux; mean signal-to-noise ratio and number of points. We also added the following metadata information: right ascension (`ra`), declination (`dec`), host galaxy photometric redshift (`hostgal_zphot`), host galaxy photometric redshift error (`hostgal_zphoterr`) and distance between the host and the

¹⁴ This number was obtained by counting the number of observations and each alert, not the corresponding metadata column.

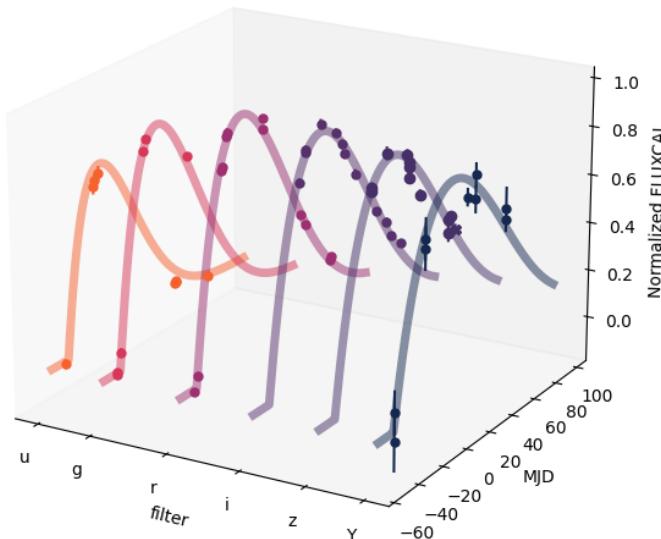


Fig. 7. Example of a RAINBOW fit on an SLSN light curve (diaSourceId = 35707054076) using Eq. (4) for the bolometric component and Eq. (3) for the temperature component. The flux is normalised to have a maximum at one.

transient (hostgal_snsep). In addition, parametric fits of the light curves are computed and best-fit values are used as parameters. Similarly to Sect. 5.3, we used the RAINBOW (Russeil et al. 2024b) framework to obtain a multi-passband description of the light curves. In this context, the bolometric flux is modelled using the following three-parameter function:

$$f(t) = \max(0, A(t - t_0) \times e^{-\frac{t-t_0}{t_{fall}}}), \quad (4)$$

which depends on amplitude (A), a time offset (t_0) and a characteristic time of decay (t_{fall}). We found this simple functional form to be effective in classifying both SLSN-I and PISN. This equation was obtained by applying the Multi-View Symbolic Regression algorithm (Russeil et al. 2024a) to real ZTF light curves from the SLSN candidate SNAD160¹⁵ (Pruzhinskaya et al. 2023). The max operation has been manually added to prevent negative fluxes. We chose to model the temperature using Eq. (3), which enables the description of cooling transients. An example of a SLSN fit is shown in Fig. 7. It highlights the good agreement of the blackbody approximation to SLSN observations. The fit is performed using LIGHT-CURVE¹⁶ python package, which internally uses iminuit minimising the negative log likelihood. The optimised parameters and the loss are included as features for the classifier. Thus we imposed that alerts contain at least seven observed points. In total, 26 features are extracted for each alert.

The classifier was based on a scikit-learn random forest algorithm trained using the active learning (AL) procedure proposed in Leoni et al. (2022). This strategy allows the classifier to focus on the relevant boundaries between SLSN and similar transients rather than simply learning the global distribution between very unbalanced classes. This procedure tends to favour purity over completeness, which is reasonable given the volume of alerts that will be produced by LSST. To proceed we randomly sampled 2 million alerts from the full training sample (Sect. 2), which were further equally divided in training and validation samples. We queried 6 alerts at a time for 3500 loops (further queries

resulted in overfitting, leading to performance decrease) for a final training sample composed of 21 100 alerts. We set the maximum depth of trees and minimum number of alerts per leaf are set to be 15 and 0.01% of the training size, respectively. The model achieved 90.8% purity and 52.4% completeness in the validation sample.

6. Performance of classifiers

In this section we evaluate the different classifiers available in FINK using the test sample composed of all alerts from objects not in the training set of ELAsTiCCv1. In Sect. 7 we show how combining different classifiers can boost classification results.

Figure 8 shows the confusion matrix for each classifier; each cell in the confusion matrix displays the results normalised by the predictions on top (bold), and by the true values at the bottom (italics). Besides these overall metrics, we analyse in Fig. 9 how metrics change with the number of detections and the true host redshift (ZHELIO from the truth table). This allows us to assess the ability of classifiers to generate unbiased samples, and their capabilities as a tool to select follow-up sources close to the time of detection.

Machine Learning metrics such as ROC and precision-recall curve can be found in Appendix A.

6.1. CATS broad classification

Figure 8a shows the metrics for the CATS broad classifier. Similar to the cross-validation results presented in Sect. 5.1, the model performs excellently with the SN-like, Periodic and Non-Periodic classes, having AUCs above 0.95 for all of them. However, again similar to the earlier results, it had some issues with the Long and Short classes: more than half of the Long alerts were classified as SN-like, and about 30% of the Fast alerts were classified as Periodic, lowering the recall for both classes when compared to the other three. However, the purity for the Fast class is still above 90%, while for the Long class the value is 83%, significantly lower than for all others.

In Fig. 9a, it can be seen that CATS is able to correctly classify samples with less than ten detections, obtaining a precision of over 80% for every class except Long, and nearly 100% for SN-like, Periodic, and Non-periodic events. CATS is able to successfully classify these objects very early, an important quality when considering the schedule of follow-up observations. Also, Precision remains nearly constant as the number of detections grow for all classes (except for the Long class), showing the robustness of the model.

Precision for the Long alerts increase as the number of detections grow since the model starts to better distinguish SN-like from SLSN alerts as the features from longer light curves begin to show. As expected from the results in Fig. 8a, recall for both the Fast and Long classes increase as the number of detections grow, while SN-like and Periodic alerts have near-constant recall for all light curve lengths. Interestingly, very short Non-periodic light curves are classified as SN-like by the model, producing a low recall for Non-periodic alerts with less than five detections.

When looking at the metrics as a function of redshift, the precision for SN-like alerts drop as the simulated redshift increases: this is due to SLSN alerts being classified as SN-like. The opposite is seen for the Long class, where low redshift alerts are classified as SN-like. Recall, on the other hand, remains almost constant for the full redshift range.

¹⁵ <https://ztf.snad.space/dr17/view/821207100004043>

¹⁶ <https://github.com/light-curve/light-curve-python>

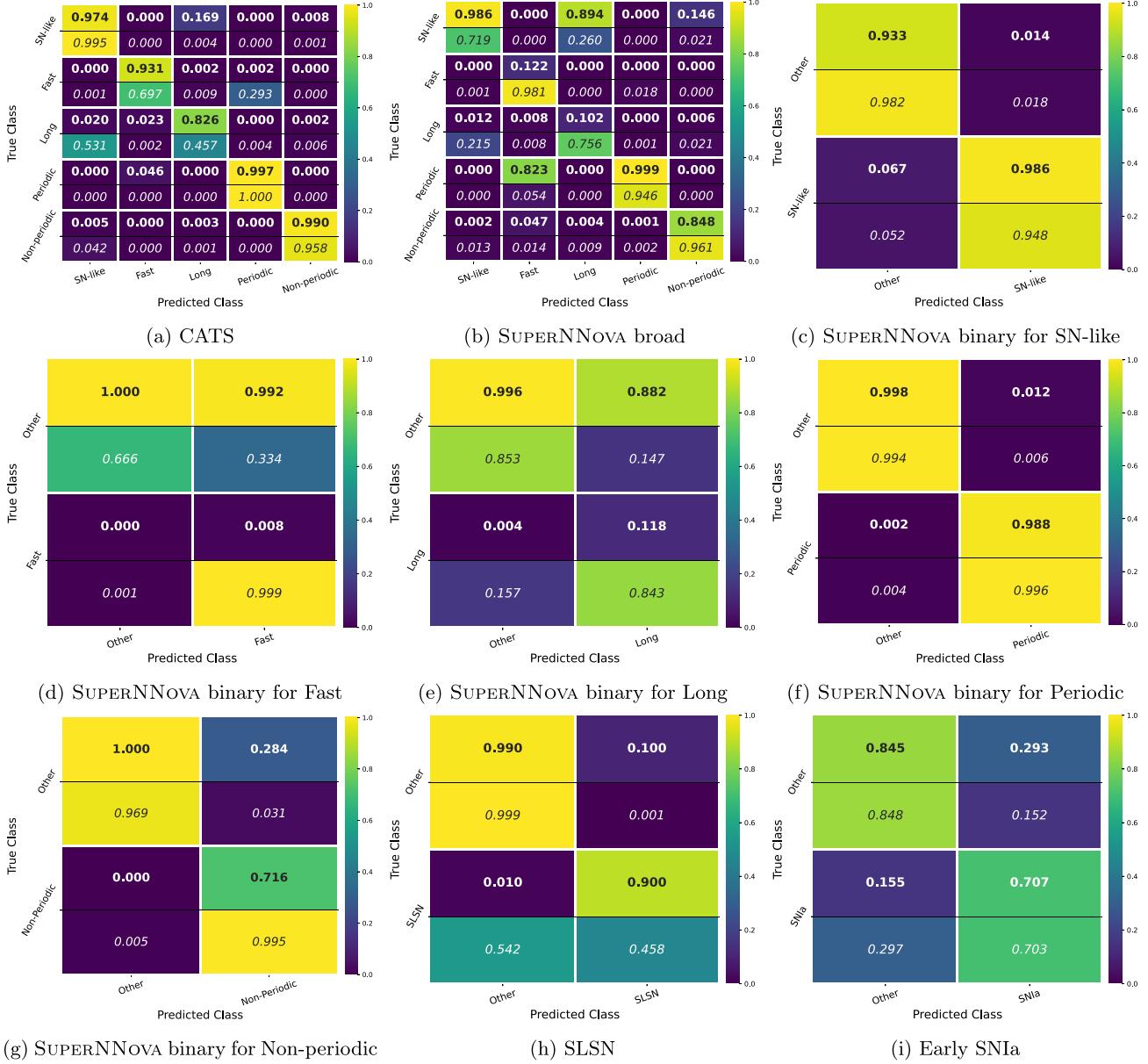


Fig. 8. Confusion matrices for each classifier in the test test. For each cell in the confusion matrix, the top row (in bold) was normalised to the predictions and the bottom row (in italics) to the true values so that the main diagonal shows the precision in bold, on top, and recall in italics, at the bottom.

6.2. SUPERNNova as a broad classifier

We show the performance metrics and confusion matrix in Fig. 8b for SNN broad classifier. As found when validating the model, both the SN and Long classes have large classification confusion. In Fig. 9b we see this confusion is reduced with more detections as the classifier disentangles Long and SNe light curves with more precision.

As with the CATS classifier, the Fast class is where the results differ the most between the training and test metrics with a similar trend of confusion between Fast and Periodic light curves.

We highlight, that the loss used to train SUPERNNova is optimised for classification with large and representative training sets. In Möller & de Boissière (2019), we have shown that $\approx 10^5$ light curves per class for training are necessary to achieve its top performance. To improve performance with ELASTICC,

the loss could be modified to allow the usage of non-balanced training sets and/or larger training sets could be constructed with additional simulations or augmentation techniques. We leave this task for future work.

6.3. SUPERNNova binary classifiers

We find better performance for the SN-like classification with the binary classifier (Fig. 8c) than with the broad one in particular for recall. This suggests, as expected, that the increase of the training set for the target is extremely important for our algorithm. We also find an improvement for classes with smaller training sets such as the Long class (Fig. 8e), although the model still has troubles with this class. The Fast class remains the hardest to classify with this algorithm (see Fig. 8d).

The evolution of metrics as a function of number of detections and redshift for the SNN binary and broad classifiers is

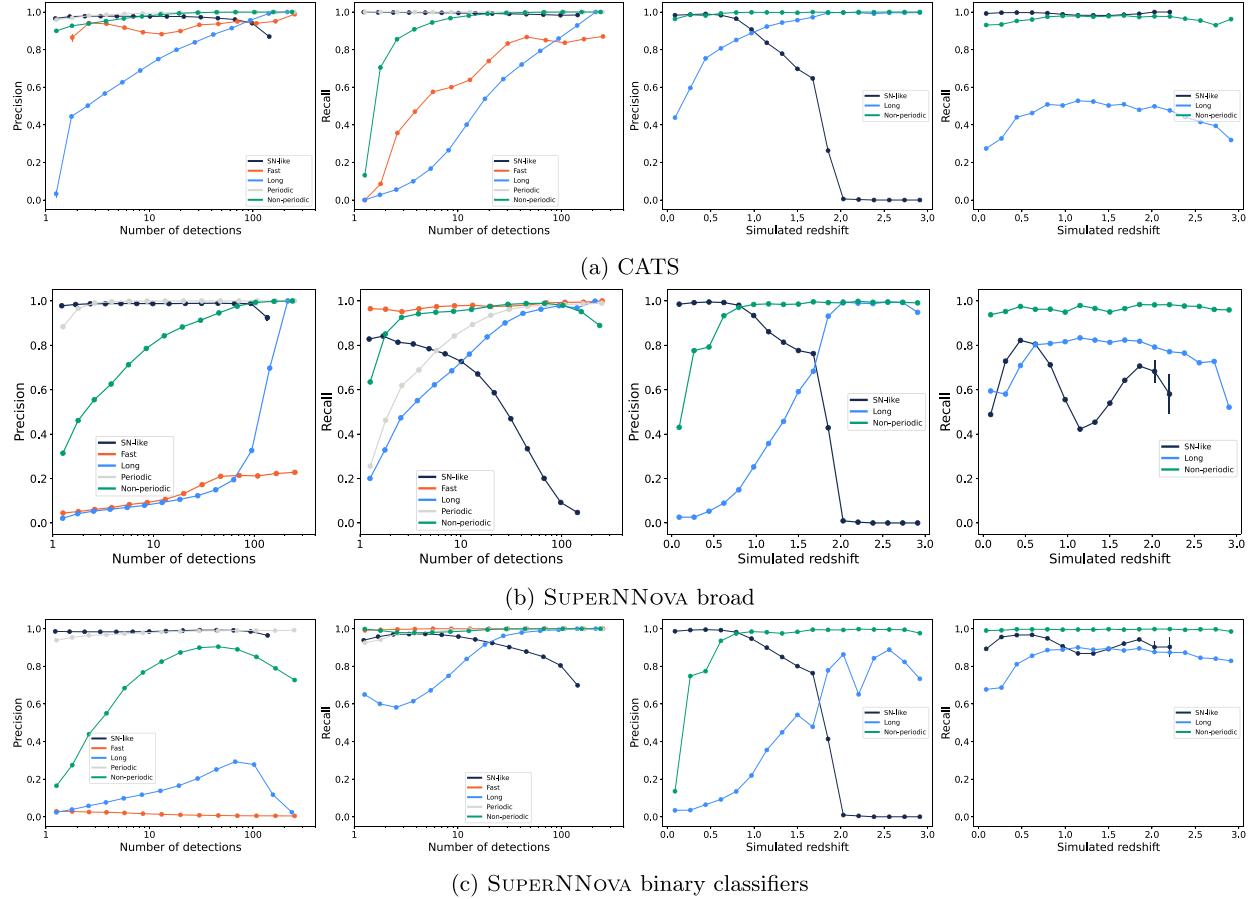


Fig. 9. Evolution of precision and recall as a function of the number of detections (left two panels) and host galaxy redshift (right two panels). Fast and periodic alerts have no redshift available and thus have only the first two panels.

shown in Figs. 9c and 9b. The same trend is found for both the broad and binary classifiers, where the precision for the Non-periodic and Long classes increase as more detections are available. Periodic and SN-like classification show high performance with low number of detections. Thus, we expected a good performance in the classification of early light curves for these classes. This is an important feature when scheduling follow-up observations, as explored for *Rubin* SNe Ia in Möller et al. (2024).

We highlight that the binary classifiers presented in this work were all trained in a similar manner without tuning SNN to the different targets. Depending on the science goal, data curation, loss and algorithm hyper-parameters could be adjusted to improve its performance. For example, to improve Fast transients classification we could train with shorter light curves for all classes, an augmented training set or a modified loss to tackle small training sets. As shown in Möller et al. (2022) for SNe Ia, an adequate light curve time span selection for a given goal to reduce non-transient detections improves performance.

6.4. Early supernova Ia

The performance of the EarlySNIa classifier is reported in Fig. 8i. One caveat we should keep in mind is that the module is only interested in classifying rising light curves. Thus, several alerts are eliminated by selection cuts, never being classified at all. Results presented here correspond to alerts that survived the feature selection (Sect. 5.3). Among these, the module was

able to achieve ~ 0.7 precision and recall. Meaning the precision was maintained while recall increased slightly when compared to results from the validation sample. This matches the results found during training and represents $\sim 10\%$ decrease in comparison to those reported in Leoni et al. (2022), which reports results from the current EarlySNIa module within FINK, applied to the ZTF stream. Considering the significant increase in volume and complexity present in the ELAsTiCC data set, we consider this a resilient and promising result. Once LSST start running, an active learning procedure similar to the one described in Leoni et al. (2022) can be employed in real time, thus improving results found in this work. The most common contaminants are: SN II ($\sim 18\%$) and SNIbc ($\sim 9\%$). All other classes correspond to the remaining 3%, with SNIa-91bg and SNIax comprising $\sim 1\%$ each.

Figure 9e shows how classification results evolve with the number of detections and simulated redshift. We note that precision already starts higher than 0.6 for seven observed data points (the minimum requirement) and peaks around 20 photometric points, while recall remains almost stable even with more detections. The sample identified as EarlySNIa by the algorithm is highly skewed towards small light curves, with $\sim 75\%$ of them having ten detections or less. Moreover, since we are working only with rising behaviours, it is reasonable to expect that light curves with lower number of points will dominate the results¹⁷.

¹⁷ The long duration alerts are attributed to long history of forced photometry or variable sources with a significant burst in the last months of the survey. This also correlates with the steep decline in precision.

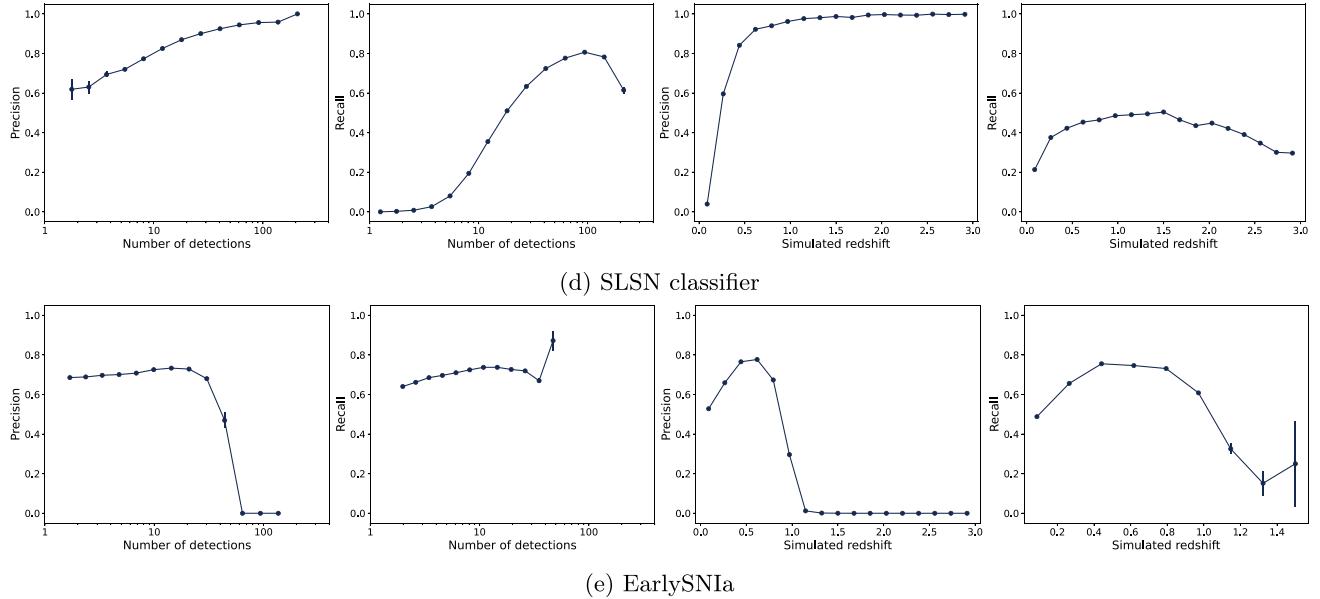


Fig. 9. Continued.

In this figure, we can also observe that classification results peak for redshift around 0.5 and degrades after that. This is expected since the SN is most likely to be discovered at or after maximum when at high redshifts.

6.5. Superluminous supernovae

Figure 8h displays the confusion matrix of the binary SLSN classifier. It provides an excellent purity of 90.0% and a recall of 45.8%. In the context of the very large ELAsTiCC data set, we favour this high precision low recall asymmetry as it would still result in almost 200K SLSN alerts being classified with high confidence. However, as illustrated by Fig. 9d, the recall is highly dependent on the number of detections of the light curves. While most SLSN are missed when only a few data points are available, the recall increases to $\sim 50\%$ when 20 observations are available and up to $\sim 80\%$ for 100 detections. Purity-wise, the performance of the model linearly increases with the number of detections, ranging from $\sim 60\%$ to almost 100%. We observe that the classifier presents a clear under performance in precision for objects with redshift lower than 0.5. Indeed, SLSN are intrinsically bright objects that are more frequently found at high redshift (corresponding large observation volumes). Therefore, a large majority of the sample has a redshift above 0.5 (with a maximum around $z = 1$). It results in a challenging learning task at low redshift, which impacts the final performances. Outside this range, the classifier is conservative and therefore very reliable on positive answers.

Although the purity of the classifier is very high, a study of the contamination reveals that almost two thirds of the alerts wrongly classified as SLSN are SNII. This result can be explained by the presence of SNIIn within the ELAsTiCC data set. This subtype of SNII, resulting from the interaction of the SNII with the circumstellar medium, is particularly bright and long-lasting. The most extreme ones even stand at the border between the SNII and the SLSN class, potentially forming a continuum between the two classes Moriya et al. (2018). Hence, the classifier contamination is to be expected and is particularly hard to reduce.

Finally, a feature importance study has been conducted in order to characterize the decision process. The most important column for classification is the error on the fit, highlighting that the parametric model chosen is suited to describe SLSN events. The second and third most important features are parameters from the model, respectively the amplitude (A) and the minimum temperature (T_{min}). It once again demonstrates that the fit is key in the separation of the parameter space. The high relevance of the temperature parameter in particular indicates that the RAINBOW framework enables the computation of informative features to distinguish transient events.

7. Combining classifiers

Given the wealth of developed classifiers targeting different science cases, combining some of them could provide better results. In this session we investigate the effectiveness of considering ensembles of two classifiers built from intrinsically different algorithms in order to boost classification results.

7.1. Broad classifier as a first step

We investigate the possibility of building a hierarchical classifier, where a broad classifier is initially applied to remove a large part of the contaminants, passing its results to a binary, more specific, model, possibly resulting in a more pure sample. We explore here using CATS before the SLSN classifier. Since the EarlySNIa binary classifier has the majority of contaminants within the SN-like broad class, a previous broad classifier is not expected to impact their results.

We take only the alerts that CATS classified as Long to assess the performance of the SLSN classifier. Results are shown in Fig. 10, where we can see improvement in both, precision (from 0.9 to 0.953) and recall (0.458 to 0.682), together with both AUCs (from 0.887/0.687 to 0.907/0.955 values for ROC/PR). Albeit small, this improvement could help produce more reliable follow-up catalogues with very few contaminants. The fraction of false negatives increased, simply due to the sample to be classified by the SLSN binary classifier being smaller.

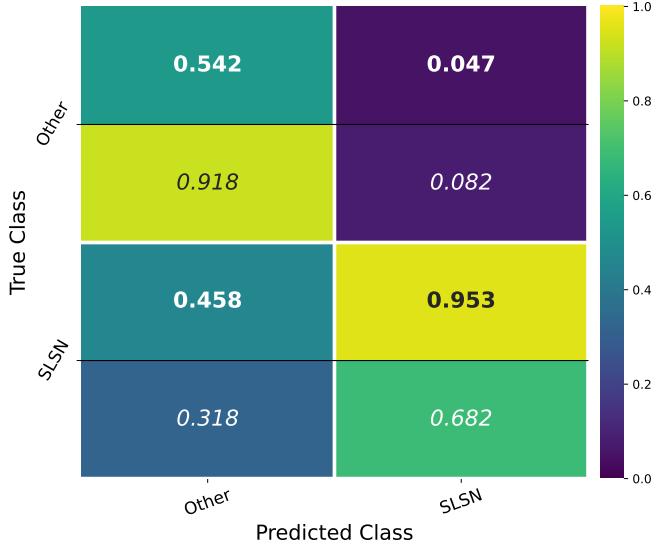


Fig. 10. Performance metrics for the SLSN classifier applied to the sample of alerts classified as ‘long’ by CATS.

The total completeness from this sample is approximately 35%, a small decrease when compared to the binary classifier alone. The fraction of correctly identified PISNe drops by approximately the same amount; a still significant amount of PISNe were identified by this hierarchical method, an important result considering the rarity of the class.

This comes mainly from missing SLSNe, while the fraction of misclassified PISNe lowers by approximately 10%; an important result since PISNe are one of the least represented objects in the test set.

7.2. SUPERNNOVA binary into a broad classifier

Given that SNN has one binary classifier for each broad class, we can combine them to create a multi-class classifier, potentially outperforming the broad SUPERNNOVA model. In order to obtain ROC and precision-recall curves, we join together all probabilities for all five binary broad classifiers and apply a softmax function to the probability vector, similar to what a standard DL multi-class classifier does. The predicted class is considered to be the one with highest probability. We note that there are a few light curves (approximately 0.09%) that have all binary probabilities less than 0.5; that is, the models did not classify it as being in any of the broad classes. These alerts are mainly from the SN-like class, with SNII as the majority. We nevertheless assigned the broad class with the highest probability to them since it will not impact the results.

Results are presented in Fig. 11 where we see improvements over the broad SUPERNNOVA classifier mainly for the SN-like and Long classes, while for the Periodic and Non-periodic the results are similar. The Fast class is the exception, where the combined binary results are worse than the broad SNN and the binary SNN; the reason is that for a lot of Fast alerts, the binary classifiers for Periodic and Non-periodic give larger probabilities than the Fast one. So although in a Fast versus others scenario they would be correctly classified, when combining all binary classifiers they end up being classified as either Periodic or Non-periodic, making the results worse. Since SUPERNNOVA requires balanced samples for training, the limitation on the number of alerts is applied on each class separately, instead of all classes being limited by the least represented. The model now is

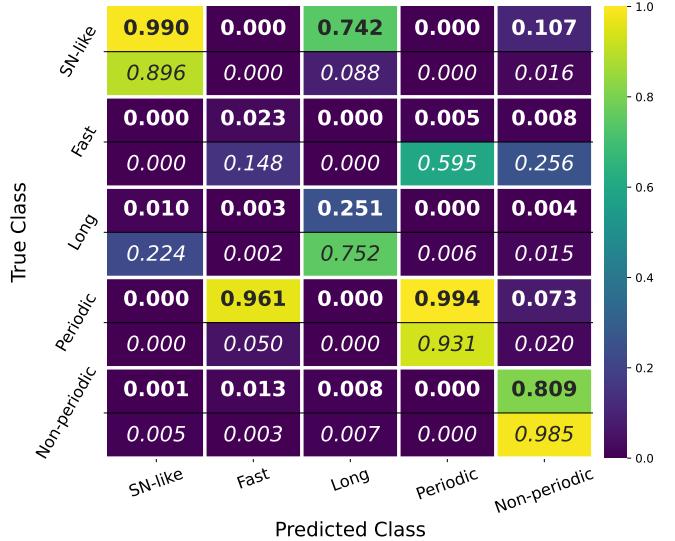


Fig. 11. Performance metrics for the combined SUPERNNOVA binary classifiers.

presented with more variety during training, and the probability of the correct class are improved.

7.3. Combining classifiers for purity

Given the large volume of data LSST is expected to deliver on a daily basis, being able to build extremely pure samples out of tens of millions of light curves is extremely important, especially for follow-up purposes. Some of the models presented in Sect. 5 target the same class (or broad class); therefore, one way to improve the purity is to combine classifiers by only considering alerts that have been assigned the same class when presented to intrinsically different classifiers.

We analyse how our broad classification may be improved using CATS and the SUPERNNOVA binary classifiers: we only consider alerts for which the probability of the binary SNN classifier corresponding to the class predicted by CATS is larger than 0.5. This happens for approximately 96% of the alerts in the test set. We show the confusion matrix over this sample in Fig. 12, where it can be seen that the precision improved for all classes when compared to CATS and all SUPERNNOVA classifiers. The recall is lower compared to SNN for the long and fast classes (due to CATS).

8. Discussion

Our results show that all classifiers have mostly satisfactory performance for their classes of interest, with very few exceptions. Despite the difference in models and training methods in the classifiers, a few common trends emerged.

All broad classifiers have trouble identifying the fast and long classes, for both the validation and test sets. CATS and SUPERNNOVA (both the broad and the combined binary classifiers) have different issues, with the former having higher purity and the latter higher completeness. Furthermore, looking at Figs. 8a, 8b, and 11, both CATS and SUPERNNOVA mix up the same classes: Long with SN-like, Fast with Periodic. This suggests that the similarity of the classes is intrinsic to the data set.

The Periodic broad class is very well characterised in the data, with all multi-class classifiers achieving almost perfect purity and completeness. Indeed, the light curves for variable

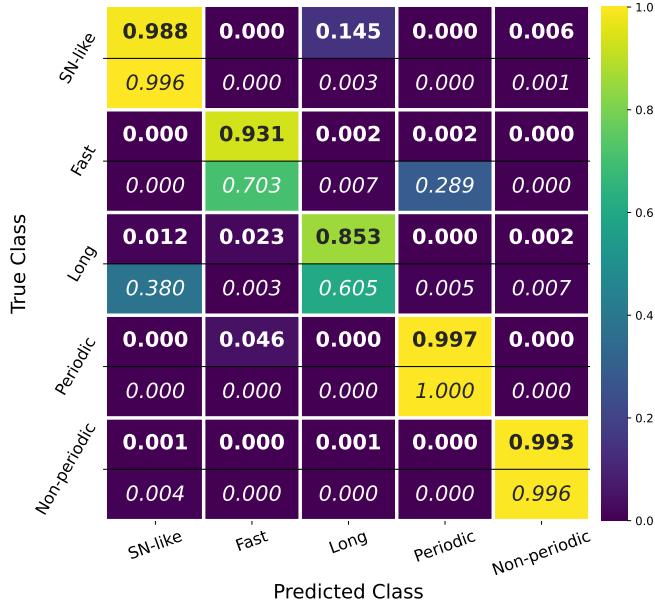


Fig. 12. Confusion matrix for the alerts where CATS and each SUPERNNOVA binary classifier agree (see text for details).

stars show a very distinctive periodicity, making them easy to identify among the other classes.

CATS can be used as a first step in a hierarchical classification scheme, in which it sends to binary classifiers only the alerts belonging to their respective broad class. When applied to the SLSN classifier, it shows a small increase in purity, even though both classifiers were not meant to work together in the first place. The in-sample completeness shows significant increase when compared to SLSN by itself.

The false positives of the EarlySNIa classifier are all within the SN-like broad class. This in turn means that using a broad classifier as a first step will not help improve its results.

SUPERNNOVA binary classifiers have better recall than the broad classifier for most classes, and an overall improvement particularly for the Long and SN-like classes. Given the current configuration of SNN, binary classifiers leverage better small training sets. Additionally, binary classifiers can be tuned to the target goal and improve performance with simple actions such as light curve length selection and hyper-parameter variation.

SUPERNNOVA binary classifiers can be combined into one multi-class model, with better (or similar, depending on the class) performance than its broad classifier. SNN is lightweight, and thus it is feasible to do this without impacting LSST processing. A similar approach could be used by other broad classifiers, although care must be taken that the model is lightweight enough so that the extra computational cost does not affect the time necessary to process the alerts.

Samples with close to perfect purity can be built by requiring two or more classifiers to match their predictions. Combining CATS and all SUPERNNOVA binaries gives slightly better purity for all classes when compared to each individual model, with all classes except the Long reaching close to 100% purity.

9. Conclusions

For a few years now, broker teams have been successfully working with the ZTF alert stream and communication protocols as a test bench for what is to be expected for LSST. This experience has been extremely successful and has allowed for the

development of an entire broker ecosystem, along with a diverse and interdisciplinary community who supports it. Nevertheless, as valuable as this experience has been, it is also crucial to prepare the infrastructure for the important and challenging differences between the data delivered by the two experiments.

ELAsTiCC is a kind reminder that beyond hardware and data format, ML models and broker infrastructure will need to change significantly in order to fulfill expectations that will rise with the arrival of LSST. This includes the design of algorithms themselves, protocols for massive data transfers between geographically disconnected science teams, and experiment design for proper evaluation and optimisation of trained models to allow for the processing of millions of alerts per night. The analysis presented here describes the strategies developed by the FINK team to address these issues.

We introduced CATS, a deep learning classifier built especially to work with LSST data for broad classification, and it has shown great performance. Other classifiers, including SUPERNNOVA and tree-based models, were adapted from their current use on ZTF data. The adapted models performed well in their respective classification tasks, delivering pure and/or complete sub-samples. Moreover, we have also shown examples of how different algorithms can be used to build an ensemble classifier whose results outperform those from individual algorithms.

Nevertheless, it is important to keep in mind that FINK operates in a framework where each classifier (science module) is developed independently by different science teams. Processing is centralised by the FINK infrastructure, but model development is geographically and scientifically distributed. This means that each team has a different scientific goal in mind when developing their own classifier.

Given nine different classifiers (plus the combined ones) working at different tasks and showing different strong points, it is paramount to clarify the requirements for each science case so that the user can make an informed decision when applying the models presented here. CATS, for example, can produce pure samples for every class, but SUPERNNOVA (both multi-class and binary classifiers) have a better completeness for the fast and long classes and a similar purity for the SN-like and periodic classes. In the future, it will be important to tailor models and/or combine different ones to achieve the top performance for a given science goal.

We also call attention to details that should be kept in mind when interpreting the classification results stated here. First, we are bound by the diversity and complexity of the initial data set. The real data will certainly contain a number of intermediate objects whose properties lie in between classes, thus affecting these results. Moreover, the choice of training sample is of utmost importance in ML problems regarding both their sample sizes and representativeness. Compared to its predecessor, PLAsTiCC, the objective of ELAsTiCC was the classification of alerts (i.e. partial views of complete light curves). Faced with the two different possibilities for a training set, we chose the streamed alerts since they most resemble the test set and results are more easily transferable to what is expected from LSST.

Moreover, it is crucial to recognise that training an ML model implies access to the data and to the necessary hardware to process it. During this challenge, a new service was designed by FINK so that each team could access the curated data necessary for the training. This service was able to serve millions of alerts regularly to various teams, where the training of the models was largely performed on commodity hardware. Despite the undoubtedly usefulness of the data transfer service, irrespective of

the volume of data to be transferred, we note that at the LSST scale, training models will require user teams to access dedicated hardware accelerators hosted on large data centres. This is an area where FINK is actively planning on providing and thus enabling a service for the community to train models at scale.

Considering the practical observational application of the classification results, early identification is paramount for the optimisation of follow-up resources and a major task of alert brokers. We have shown that most of our ML algorithms are capable of obtaining high-precision classification with less than 20 points. As more observations are added, the models generally give more accurate results. For the SN-like and fast classes, this performance increase is only valid with detection time spans related to their variability.

Finally, keeping in mind the intrinsic differences between simulated and real data, results presented here can be used to calibrate expectations regarding the output of FINK science modules in the first stages of LSST operations. Throughout the ten years of the survey, classifications will certainly evolve and present even better numerical results.

In summary, we have shown that FINK is able to process ELAsTiCC alerts in a fast and efficient manner and provide informative ML classification scores for a variety of science cases. This proves the adequateness of the FINK infrastructure and ML algorithms to process the big data volumes of Rubin LSST and, consequently, to contribute to the realisation of its scientific potential.

Acknowledgements. We thank Robert Knop, Gautham Narayan, Rick Kessler and all others involved in the development of ELAsTiCC for enabling this work. We thank Konstantin Malanchev for adapting the `light_curve` feature extraction package (Malanchev et al. 2021, <https://github.com/light-curve/light-curve-python>) for use within the EarlySNIA and SLSN classifier. 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. This is a result from the 2022 Fink Hackathon, 19–26 November 2022, Grimentz, Switzerland. This work received funding from CNRS MITI Événements Rares – 2022, under project number 226696, Finding the first generation of stars with LSST (Fink). A.M. is supported by the ARC Discovery Early Career Researcher Award (DECRA) project number DE230100055. Part of this work was supported by the ARC Centres of Excellence OzGrav and OzGrav 2 project numbers CE170100004 and CE230100016. This work was supported by the ‘Programme National de Physique Stellaire’ (PNPS) of CNRS/INSU cofunded by CEA and CNES. CRB acknowledges the financial support from CNPq (316072/2021-4) and from FAPERJ (grants 201.456/2022 and 210.330/2022) and the FINEP contract 01.22.0505.00 (ref. 1891/22). B.M.O.F., C.R.B. and A.S. acknowledge the LITCOMP/COTEC/CBPF multi-GPU development team for all the support in the artificial intelligence infrastructure and Sci-Mind’s High-Performance multi-GPU system.

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from tensorflow.org

Aivazyan, V., Almualla, M., Antier, S., et al. 2022, *MNRAS*, **515**, 6007

Aleo, P. D., Malanchev, K., Sharief, S., et al. 2023, *ApJS*, **266**, 9

Allam, Tarek, J., Peloton, J., & McEwen, J. D. 2023, arXiv e-prints [arXiv:2303.08951]

Alves, C. S., Peiris, H. V., Lochner, M., et al. 2022, *ApJS*, **258**, 23

Bamford, S. P., Nichol, R. C., Baldry, I. K., et al. 2009, *MNRAS*, **393**, 1324

Baron, D. 2019, arXiv e-prints [arXiv:1904.07248]

Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, *PASP*, **131**, 018002

Bengio, Y., Simard, P., & Frasconi, P. 1994, *IEEE Trans. Neural Netw.*, **5**, 157

Biswas, B., Ishida, E. E. O., Peloton, J., et al. 2023a, *A&A*, **677**, A77

Biswas, B., Lao, J., Aubourg, E., et al. 2023b, arXiv e-prints [arXiv:2311.04845]

Bom, C. R., Fraga, B. M. O., Dias, L. O., et al. 2022, *MNRAS*, **515**, 5121

Boone, K. 2019, *AJ*, **158**, 257

Cabrera-Vives, G., Moreno-Cartagena, D., Astorga, N., et al. 2024, *A&A*, **689**, A289

Carry, B., Peloton, J., Le Montagner, R., Mahlke, M., & Berthier, J. 2024, *A&A*, **687**, A38

Dark Energy Survey Collaboration, Abbott, T., Abdalla, F. B., et al. 2016, *MNRAS*, **460**, 1270

DES Collaboration, Abbott, T. M. C., Acevedo, M., et al. 2024, *ApJ*, **973**, L14

Dozat, T. 2016, in Proceedings of the 4th International Conference on Learning Representations, 1

Eriksen, M., Alarcon, A., Cabayol, L., et al. 2020, *MNRAS*, **497**, 4565

Förster, F., Cabrera-Vives, G., Castillo-Navarrete, E., et al. 2021, *AJ*, **161**, 242

Hilbe, J. M., Riggs, J., Wandelt, B. D., et al. 2014, *Significance*, **11**, 48

Hložek, R., Malz, A. I., Ponder, K. A., et al. 2023, *ApJS*, **267**, 25

Ho, T. K. 1995, *Proc. 3rd Int. Conf. Doc. Anal. Recog.*, 1, 278

Hochreiter, S., & Schmidhuber, J. 1997, *Neural Comput.*, **9**, 1735

Ishida, E. E. O. 2019, *Nat. Astron.*, **3**, 680

Ivezic, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, **873**, 111

Karim, F., Majumdar, S., Darabi, H., & Harford, S. 2019, *Neural Netw.*, **116**, 237

Karpov, S., & Peloton, J. 2022, arXiv e-prints [arXiv:2202.05719]

Karpov, S., & Peloton, J. 2023, *Contrib. Astron. Observ. Skal. Pleso*, **53**, 69

Kessler, R., Bernstein, J. P., Cinabro, D., et al. 2009, *PASP*, **121**, 1028

Kessler, R., Bassett, B., Belov, P., et al. 2010, *PASP*, **122**, 1415

Khakpash, S., Bianco, F. B., Modjaz, M., et al. 2024, *ApJS*, submitted [arXiv:2405.01672]

Knop, R., & ELAsTiCC Team. 2023, *AAS Meeting Abstracts*, **55**, 117.02

Kuhn, M. A., Benjamin, R. A., Ishida, E. E. O., et al. 2023, *Res. Notes Am. Astron. Soc.*, **7**, 57

Le Montagner, R., Peloton, J., Carry, B., et al. 2023, *A&A*, **680**, A17

Leoni, M., Ishida, E. E. O., Peloton, J., & Möller, A. 2022, *A&A*, **663**, A13

LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, arXiv e-prints [arXiv:0912.0201]

Malanchev, K. L., Pruzhinskaya, M. V., Korolev, V. S., et al. 2021, *MNRAS*, **502**, 5147

Malz, A. I., Dai, M., Ponder, K. A., et al. 2024, *A&A*, in press, <https://doi.org/10.1051/0004-6361/202346891>

Matheson, T., Stubens, C., Wolf, N., et al. 2021, *AJ*, **161**, 107

Möller, A., & de Boissière, T. 2019, *MNRAS*, **491**, 4277

Möller, A., & Main de Boissière, T. 2022, in *Machine Learning for Astrophysics* (Berlin: Springer), 21

Möller, A., Peloton, J., Ishida, E. E. O., et al. 2021, *MNRAS*, **501**, 3272

Möller, A., Smith, M., Sako, M., et al. 2022, *MNRAS*, **514**, 5159

Möller, A., Wiseman, P., Smith, M., et al. 2024, *MNRAS*, **533**, 2073

Moriya, T. J., Sorokina, E. I., & Chevalier, R. A. 2018, *Space Sci. Rev.*, **214**, 59

Moriya, T. J., Inserra, C., Tanaka, M., et al. 2022, *A&A*, **666**, A157

Nordin, J., Brinnel, V., van Santen, J., et al. 2019, *A&A*, **631**, A147

Pan, S. J., & Yang, Q. 2010, *IEEE Trans. Knowledge Data Eng.*, **22**, 1345

Patterson, M. T., Bellm, E. C., Rusholme, B., et al. 2018, *PASP*, **131**, 018001

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *J. Mach. Learn. Res.*, **12**, 2825

Perlmutter, S., Aldering, G., Goldhaber, G., et al. 1999, *ApJ*, **517**, 565

Pruzhinskaya, M., Volnova, A., Kornilov, M., et al. 2022, *Res. Notes Am. Astron. Soc.*, **6**, 122

Pruzhinskaya, M. V., Ishida, E. E. O., Novinskaya, A. K., et al. 2023, *A&A*, **672**, A111

Qu, H., & Sako, M. 2022, *AJ*, **163**, 57

Riess, A. G., Filippenko, A. V., Challis, P., et al. 1998, *AJ*, **116**, 1009

Rumelhart, D. E., & McClelland, J. L. 1987, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Learning Internal Representations by Error Propagation (Cambridge: MIT Press), 318

Russeil, E., Ishida, E. E. O., Le Montagner, R., Peloton, J., & Möller, A. 2022, arXiv e-prints [arXiv:2211.10987]

Russeil, E., Olivetti de França, F., Malanchev, K., et al. 2024a, arXiv e-prints [arXiv:2402.04298]

Russeil, E., Malanchev, K. L., Aleo, P. D., et al. 2024b, *A&A*, **683**, A251

Schmidt, R. M. 2019, arXiv e-prints [arXiv:1912.05911]

Smith, K. W., Williams, R. D., Young, D. R., et al. 2019, *Res. Notes Am. Astron. Soc.*, **3**, 26

Vincenzi, M., Sullivan, M., Möller, A., et al. 2023, *MNRAS*, **518**, 1106

Vincenzi, M., Brout, D., Armstrong, P., et al. 2024, *ApJ*, **975**, 86

Appendix A: ROC and precision-recall curves

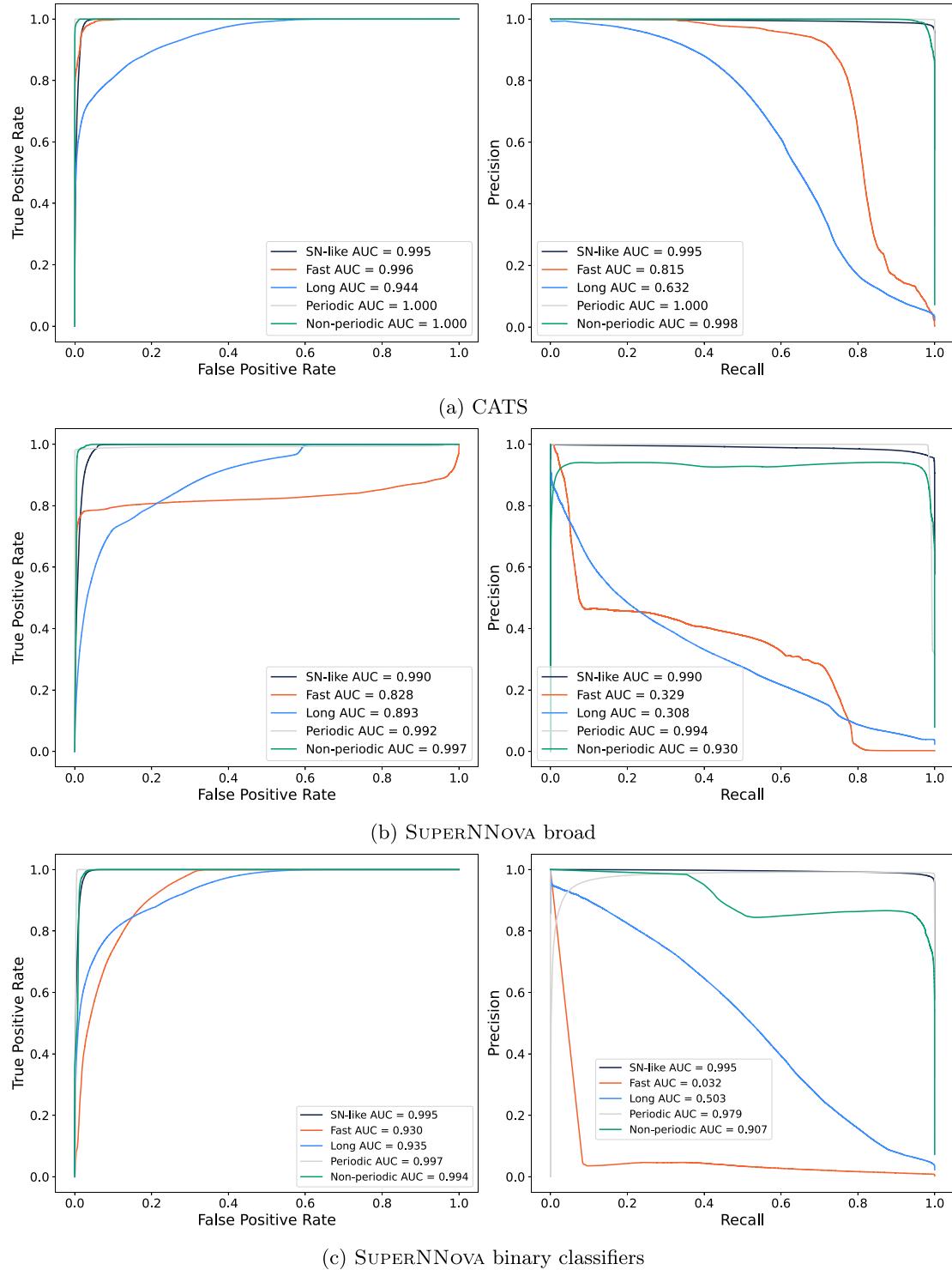


Fig. A.1: Receiver operating characteristic and precision-recall curves for all classifiers.

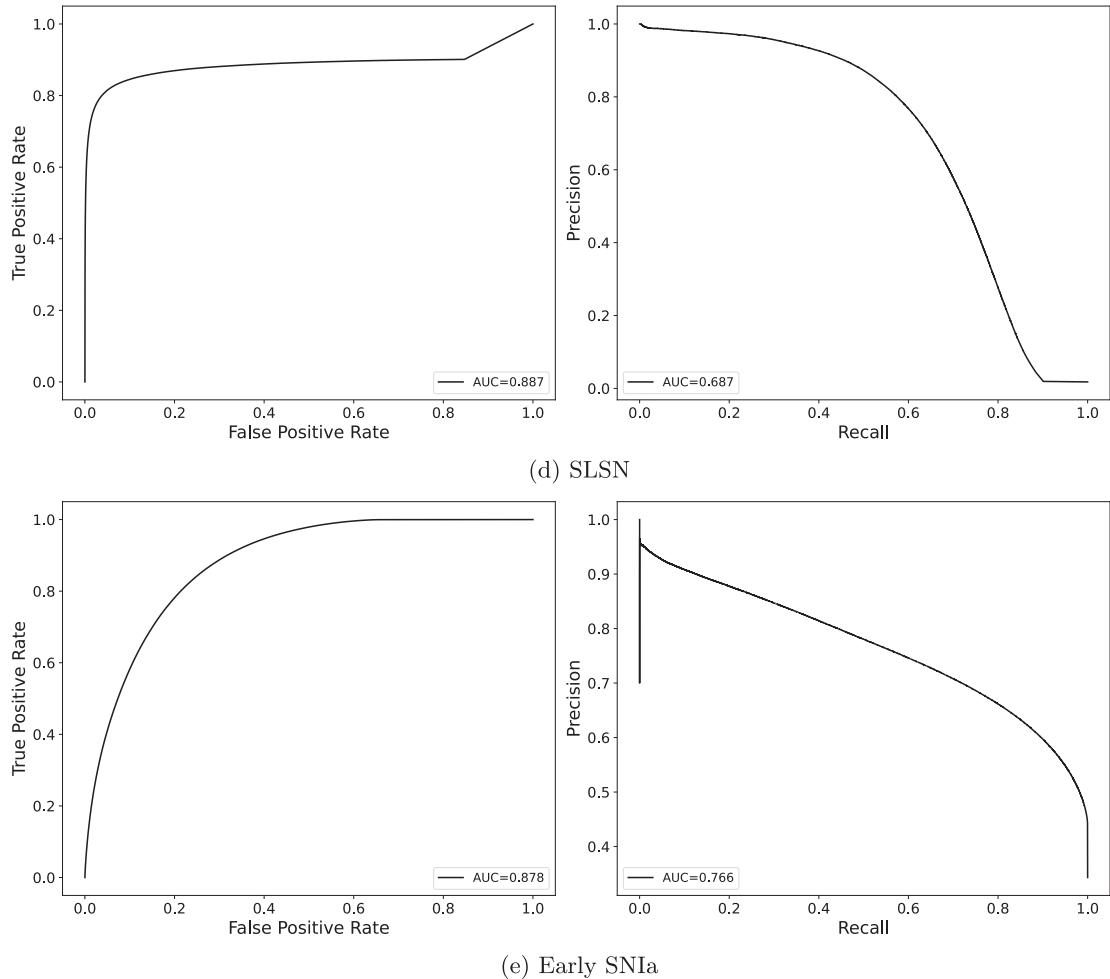


Fig. A.1: Continued.