Real-time Detection of Anomalies in Multivariate Time Series of Astronomical Data

Daniel Muthukrishna
Massachusetts Institute of Technology
Cambridge, MA, USA
danmuth@mit.edu

Kaisey S. Mandel
University of Cambridge
Cambridge, United Kingdom
kmandel@ast.cam.ac.uk

Michelle Lochner
Department of Physics and Astronomy, University of the Western Cape*
and South African Radio Astronomy Observatory (SARAO)†
mlochner@uwc.ac.za

Sara Webb
Swinburne University of Technology
Melbourne, VIC, Australia
swebb@swin.edu.au

Gautham Narayan
University of Illinois at Urbana-Champaign
Urbana, IL, USA
gsn@illinois.edu

Abstract

Astronomical transients are stellar objects that become temporarily brighter on various timescales and have led to some of the most significant discoveries in cosmology and astronomy. Some of these transients are the explosive deaths of stars known as supernovae while others are rare, exotic, or entirely new kinds of exciting stellar explosions. New astronomical sky surveys are observing unprecedented numbers of multi-wavelength transients, making standard approaches of visually identifying new and interesting transients infeasible. To meet this demand, we present two novel methods that aim to quickly and automatically detect anomalous transient light curves in real-time. Both methods are based on the simple idea that if the light curves from a known population of transients can be accurately modelled, any deviations from model predictions are likely anomalies. The first approach is a probabilistic neural network built using Temporal Convolutional Networks (TCNs) and the second is an interpretable Bayesian parametric model of a transient. We show that the flexibility of neural networks, the attribute that makes them such a powerful tool for many regression tasks, is what makes them less suitable for anomaly detection when compared with our parametric model.

1 Introduction

Upcoming sky-surveys of the time-varying universe such as the Vera Rubin Observatory Legacy Survey of Space and Time (LSST) will observe over 10 million transient alerts each night: two orders of magnitude more than any survey to date [9]. These new transient surveys will probe entirely new regimes in the time-varying universe, likely observing completely new classes of astronomical objects. For a long time, discovery in astronomy has been driven by serendipity [14] and by identifying anomalies in datasets, which has primarily been achieved by visual examination

*Bellville, Cape Town, 7535, South Africa
†2 Fir Street, Observatory, Cape Town, 7925, South Africa

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

and follow-up observations. However, with the large influx of data, it is infeasible to visually examine or follow up any significant fraction of transient candidates. To this end, identifying anomalous objects and prioritising which of the millions of alerts are most suitable for followup observations is a challenge that needs to be automated. In this paper, we develop a novel framework for identifying anomalous transients in real-time.

Most previous efforts to automate the identification of transients require the full phase coverage of the transient. While retrospective classification after the full light curve of an event has been observed is useful, it also limits the scientific questions that can be answered about these events, many of which exhibit interesting physics at early times. There has been very little focus at identifying transients in real-time. Obtaining detailed followup observations shortly after a transient's explosion provides insights into the object's physical mechanism. While the mechanism of some transients are reasonably well-understood, the central engine of various exotic classes are poorly understood [e.g. 4].

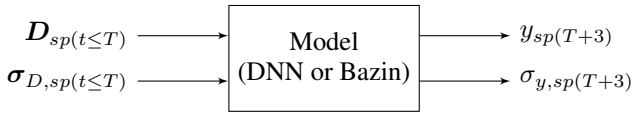
2 Method

To evaluate our method, we used the dataset described in Muthukrishna et al. [17] that augmented real transients [12] using the `SNANA` software [11] to match the observing properties of the Zwicky Transient Facility (ZTF) [3]. We simulated 10,000 transients from three common transient classes (SNIa, SNII, SNIbc) and eight rare transient classes (Kilonova, SLSN-I, TDE, CART, PISN, ILOT, AGN, uLens-BSR). Each simulated transient consists of a time-series of flux and flux uncertainty measurements in the g (green wavelength range) and r (red wavelength range) passband filters, known as a *light curve*.

Our methods for anomaly detection involve first developing an autoregressive sequence model of a transient class, and then using the model's ability to predict future fluxes as an anomaly score (or conversely, a goodness of fit score). We develop two methods for regressing over a transient. The first is a probabilistic deep neural network (DNN) approach using Temporal Convolutional Networks (TCNs), and the second is a Bayesian parametric approach using the flexible Bazin function [2].

Each model aims to do real-time detection, and is hence causal, using only past values to predict future values. Specifically, our model is a function that predicts future fluxes in a time-series as well as the uncertainty of that prediction; it then compares the prediction with the observed data to obtain an anomaly score.

In the following two subsections (§2.1, §2.2), we describe our two approaches of developing a function that maps the observed fluxes for a transient s at passband p up to time T onto flux predictions $y_{sp(T+3)}$ and predictive uncertainties $\sigma_{y,sp(T+3)}$ three days after a given set of observations:



$$\boldsymbol{X}_{spt} = [D_{spt}, \sigma_{D,spt}] \quad (1)$$

$$\boldsymbol{Y}_{spt} = [y_{spt}, \sigma_{y,spt}] \quad (2)$$

The DNN approach builds a neural network that effectively performs regression over past data in order to predict the flux 3 days in the future. On the other hand, the Bazin approach performs regression over time to predict the flux at any time. We then feed in partial light curves into the Bazin model and infer a prediction 3 days after given data to obtain anomaly scores comparable with the DNN. We choose to predict the flux 3 days in the future to approximately match the 3-day cadence between ZTF observations.

2.1 Probabilistic Neural Network

The DNN is an autoregressive mapping function that aims to map an input multi-passband light curve matrix, $\boldsymbol{X}_{s(t\leq T)}$, for transient s up to a time T, onto an output multi-passband flux vector at the next time-step $T+3$,

$$\boldsymbol{Y}^{w}_{s(T+3)} = \boldsymbol{f}_T(\boldsymbol{X}_{s(t\leq T)}; \boldsymbol{w}) \quad (3)$$

where $\boldsymbol{w}$ are the parameters (i.e. weights and biases) of the network. We define $\boldsymbol{X}_{s(t\leq T)}$ as the matrix $\boldsymbol{X}_s$ but up to a time T in each of its N_p passbands. The model output prediction $\boldsymbol{Y}^{w}_{s(T+3)}$ is a $1 \times 2N_p$ vector consisting of the predicted mean flux $\tilde{y}_{sp(T+3)}(\boldsymbol{w})$ and intrinsic uncertainty $\tilde{\sigma}_{\mathrm{int},sp(T+3)}(\boldsymbol{w})$

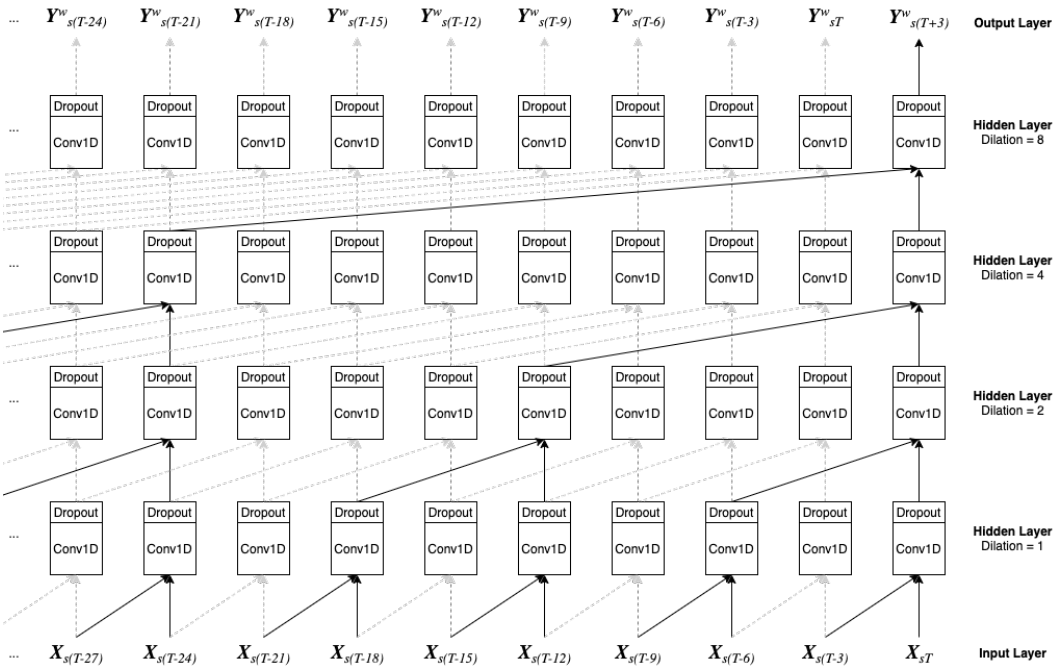


Figure 1: Temporal Convolutional Neural Network architecture used in this work. Each column in the diagram is a subsequent time-step from left to right. The bottom row is the input light curve (from equation 1) where each input is a vector of the observed flux and flux uncertainty in all passbands for a transient s at a time t. The input fluxes and uncertainties of two adjacent time-steps are passed into a residual block consisting of a 1D convolutional neural network layer (Conv1D) with dropout. While not shown in the figure, the residual block also contains a second Conv1D layer with dropout. The outputs of these are then convolved with the outputs from some previous time-steps in the above hidden layers as shown in the diagram, until the final Output Layer is the predicted light curve at the following time-step (from equation 2). The solid arrows show how the prediction $\boldsymbol{Y}^{w}_{s(T+3)}$ is made, and the gray dashed arrows show the neural network layers that lead to all other predictions. The network is causal, whereby new predictions only use information from previous time-steps in the light curve. We set the dropout rate to 20% for all layers in the network. We build this model using the `Keras` and `TensorFlow Probability` libraries after adapting the TCN model from Bai et al. [1] and their code in `https://github.com/philipperemy/keras-tcn`.

in the g and r passbands at the next time-step for a particular set of network weights. The model $\boldsymbol{f}_T(\boldsymbol{X}_{s(t\leq T)}; \boldsymbol{w})$ in equation 3 is represented by the complex DNN architecture illustrated in Fig. 1.

To characterise the intrinsic uncertainty of our network's ability to represent a light curve, we build a probabilistic neural network. Our DNN parameterizes a Normal distribution and outputs a predictive mean $\tilde{y}_{spt}(\boldsymbol{w})$ and standard deviation $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ for a particular set of network weights w. The predictions $\tilde{y}_{spt}(\boldsymbol{w})$ and $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ are components of the vector $\boldsymbol{Y}^{w}_{spt}$ from equation 3. We include the learned uncertainty $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$ because we know that our DNN model is not a perfect representation of a light curve, and even if we had no measurement error and had an infinite training set, there would still be some discrepancy between our DNN predictions and the observed light curves.

A Bayesian neural network enables us to also quantify the uncertainty in our model's predictions of the outputs, $\tilde{y}_{spt}(\boldsymbol{w})$ and $\tilde{\sigma}_{\mathrm{int},spt}(\boldsymbol{w})$. We approximate a Bayesian network using an approach called Monte Carlo (MC) dropout sampling (see Gal & Ghahramani [6]). We use MC dropout with our probabilistic neural network to estimate the predictive uncertainty. We do this by collecting the results of stochastic forward passes through the network as approximate posterior draws of $\boldsymbol{Y}^{w}_{spt}$, and use the mean and standard deviation of these draws as our marginal predictive mean y_{spt} and predictive uncertainty $\sigma_{y,spt}$. We define the model loss function as a log-likelihood comparing the flux predictions and observations including model and data uncertainties as described in A.1.

2.2 Parametric Bayesian Bazin function

We have also built a Bayesian model of each transient light curve based on the widely used phenomenological Bazin function from Bazin et al. [2],

$$f_{spt}(\boldsymbol{\theta}) = A\frac{e^{-(t-t_0)/\tau_{\rm fall}}}{1+e^{-(t-t_0)/\tau_{\rm rise}}} + B \tag{4}$$

where $\boldsymbol{\theta} = [A, B, t_0, \tau_{\rm fall}, \tau_{\rm rise}, \sigma_{\rm int}]$ are the free parameters of the model. We then model the latent flux of a transient s in passband p with the Bazin function plus an additional intrinsic error $\epsilon_{\rm int}(t)$ to account for discrepancies between the model and the observed light curve,

$$F_{spt}(\boldsymbol{\theta}) = f_{spt}(\boldsymbol{\theta}) + A\epsilon_{\rm int}(t), \tag{5}$$

where $\epsilon_{\rm int}(t) \sim \mathcal{N}(0, \sigma^2_{\rm int})$ is a zero-mean Gaussian random variable with intrinsic variance $\sigma^2_{\rm int}$. Next, we derive a generative model of the observed flux, $D_{spt} = F_{spt}(\boldsymbol{\theta}) + \epsilon_{D,spt}$, where we assume that the measurement error $\epsilon_{D_{spt}} \sim \mathcal{N}(0, \sigma^2_{D,spt})$ is a zero-mean Gaussian random variable with variance $\sigma^2_{D,spt}$.

To model a partial light curve from -70 days before the first detection (*trigger*) up to time T, we write the likelihood function as follows,

$$\mathcal{P}(\boldsymbol{D}_{sp(t\leq T)}|\boldsymbol{t}, \boldsymbol{\theta}) = \prod_{t=-70}^{T} \mathcal{N}\left(D_{spt} \mid f_{spt}(\boldsymbol{\theta}), A^2\sigma^2_{\rm int} + \sigma^2_{D,spt}\right) \tag{6}$$

We define a Bayesian model to fit each transient light curve in a particular passband after choosing a Gaussian prior $\mathcal{P}(\boldsymbol{\theta})$ based on the population of transients, $\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp}, \boldsymbol{t}) \propto \mathcal{P}(\boldsymbol{D}_{sp}|\boldsymbol{t}, \boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta})$. In practice, for anomaly detection, we need to make predictions of D at new times T. This requires that we evaluate the predictive distribution defined by

$$\mathcal{P}(F_{sp(T+3)}|\boldsymbol{D}_{sp(t\leq T)}, \boldsymbol{t}) = \int \mathcal{P}(F_{sp(T+3)}|\boldsymbol{\theta}, \boldsymbol{t})\mathcal{P}(\boldsymbol{\theta}|\boldsymbol{D}_{sp(t\leq T)}, \boldsymbol{t})d\boldsymbol{\theta}. \tag{7}$$

The integral on the RHS cannot be computed analytically, and so we approximate it by sampling. We draw sample parameters of the posterior (second term in the integrand) and compute the flux predictions for each set of parameters (first term in the integrand) with equation 5. The LHS of equation 7 is the sampled probability density function and we estimate its marginal predictive mean and uncertainty as the sample mean and standard deviation of the fluxes computed from the posterior draws.

3 Results and Discussion

We trained three autoregressive DNN models, one for each common transient class: SNIa, SNII, SNIbc. And similarly, for the Bazin function, we defined a prior distribution for each of these common classes based on the population of transients in each training set. Each training set consisted of $\sim$8000 light curves and we validated the performance of the models on $\sim$2000 light curves from each transient class. We then applied our models to the transients from all other classes to identify how well the rare transient classes were identified as anomalous.

We quantify anomaly scores using a χ^2 metric to compute the discrepancy between the observed flux at time t and the predictions of a model based on previous data. This χ^2 is weighted by the total variance including the predictive uncertainty and measurement error. A large χ^2 occurs when the model does not predict observations well and can thus be indicative of an anomaly. As illustrated in Figure 2, the Bazin model is significantly better at identifying anomalous classes than the DNN models. In Appendix A.2, we highlight that the poor performance of the DNN compared to the Bazin model is because it is too flexible at predicting light curves; and after being trained on one class, it is still able to accurately predict fluxes in a different class of transients. The DNN model is actually better at predicting the future fluxes of transients within a trained class, but is also able to predict the future fluxes of transients from different classes well. While this flexibility allows for good flux predictions, it is not good for anomaly detection.

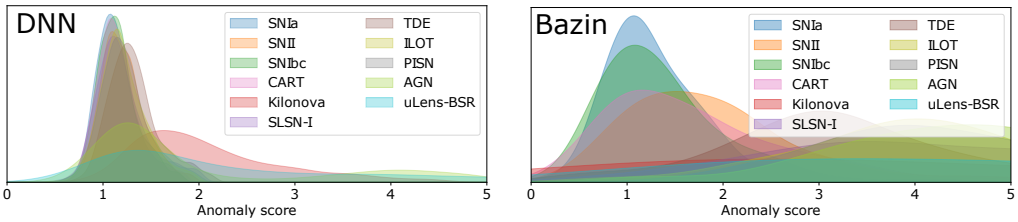


Figure 2: Anomaly score distribution recorded over the full light curve for the models that were trained on SNIa light curves and tested on the transient populations of ten different classes. Classes that are dissimilar to SNIa have higher anomaly scores, while similar classes have lower anomaly score distributions. The Bazin plot (right) shows a larger separation of the distributions of the anomalous classes from the common transient classes (SNIa, SNII, SNIbc) than the DNN (left), indicating that it is better at identifying anomalous classes. We refer the reader to Muthukrishna et al. [18] for a more comprehensive comparison and analysis of the two methods.

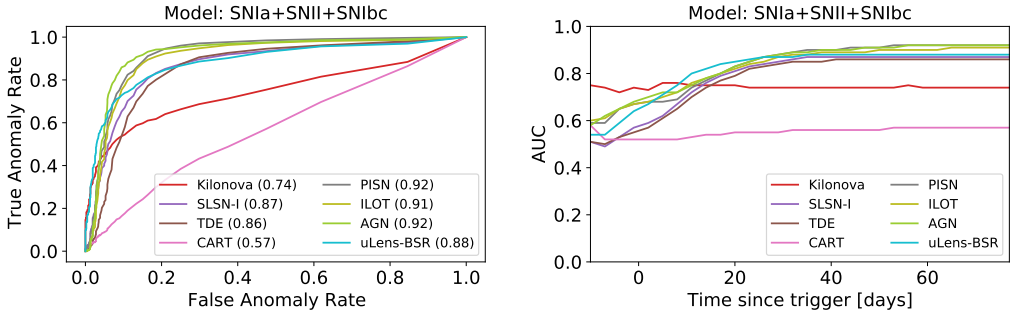


Figure 3: *Left panel:* The Receiver Operating Characteristic (ROC) curve plotting the True Anomaly Rate against the False Anomaly Rate for different threshold anomaly scores. We aggregated the results of the Bazin models trained on each of the common transients (SNIa, SNII and SNIbc) and show their combined performance on eight anomalous classes denoted in the legend. The area under the curves (AUCs) are shown in brackets in the legend. We use the anomaly scores over the full light curves to make these ROC curves. *Right panel:* We aggregate the ROC curves at all times by plotting the AUC scores as the light curves evolve, illustrating that our ability to identify anomalies improves with time. The model performs very well at distinguishing all classes except for CARTs which are known to be difficult to distinguish from common SNe based only on the light curves [17].

Our methods allow us to identify anomalies as a function of time, and in Figure 3, we show that the Bazin model is able to have high True Anomaly Rates and low False Anomaly Rates that improve over the lifetime of a transient, reaching AUC scores well above 0.8 for most rare classes. The DNN model, on the other hand, achieves an average AUC score of only 0.66 across the rare classes. In future work, we hope to apply our method on real-time ZTF observations to gauge our success at identifying real anomalous transients. In practice, applying an anomaly detection framework in conjunction with a transient classifier will provide more valuable information on whether a newly discovered transient is interesting enough for further follow-up observations. An issue with this work, is that there has been no distinction between anomalies and *interesting* anomalies. We expect that most bogus transient alerts will already be removed by *real-bogus* cuts [e.g. 5], however, our approach may still flag unusual transient phenomena that don't align with our trained transient classes but are uninteresting to most astronomers. To deal with this, future work should consider Active Learning frameworks that use methods such as *Human-in-the-loop learning* that specifically target what users define as interesting phenomena (recent work by Ishida et al. [8], Lochner & Bassett [15] have begun working on Active Learning for anomaly detection).

Overall, this paper presents a novel and effective method of identifying anomalous transients in real-time that is fast enough to be easily scaleable to surveys as large as LSST. Anomaly detection coupled with other classification approaches enables astronomers to prioritise follow-up candidates. This work and other recent approaches to transient anomaly detection [e.g. 16, 18, 20–23] are going to be critical for discovery in the new era of large-scale astronomical surveys.

Acknowledgments and Disclosure of Funding

ML acknowledges support from South African Radio Astronomy Observatory and the National Research Foundation (NRF) towards this research. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF. KSM acknowledges funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement No. 101002652). This project has been made possible through the ASTROSTAT-II collaboration, enabled by the Horizon 2020, EU Grant Agreement No. 873089.

References

[1] Bai S., Zico Kolter J., Koltun V., 2018, arXiv e-prints, p. arXiv:1803.01271

[2] Bazin G., et al., 2009, Astronomy and Astrophysics, 499, 653

[3] Bellm E. C., et al., 2019, Publications of the Astronomical Society of the Pacific, 131, 018002

[4] Coppejans D. L., et al., 2020, Astrophysical Journal, 895, L23

[5] Duev D. A., et al., 2019, Monthly Notices of the Royal Astronomical Society, 489, 3582

[6] Gal Y., Ghahramani Z., 2015a, arXiv e-prints, p. arXiv:1506.02142

[7] Gal Y., Ghahramani Z., 2015b, arXiv e-prints, p. arXiv:1506.02157

[8] Ishida E. E. O., et al., 2019, arXiv e-prints, p. arXiv:1909.13260

[9] Ivezić Ž., et al., 2019, Astrophysical Journal, 873, 111

[10] Jamal S., Bloom J. S., 2020, Astrophysical Journals, 250, 30

[11] Kessler R., et al., 2009, Publications of the Astronomical Society of the Pacific, 121, 1028

[12] Kessler R., et al., 2019, Publications of the Astronomical Society of the Pacific, 131, 094501

[13] Kingma D. P., Ba J., 2015, in Proceedings of the 3rd International Conference on Learning Representations. ICLR 2015

[14] Lang K. R., 2010, Science, 327, 39

[15] Lochner M., Bassett B. A., 2020, arXiv e-prints, p. arXiv:2010.11202

[16] Malanchev K. L., et al., 2021, Monthly Notices of the Royal Astronomical Society, 502, 5147

[17] Muthukrishna D., Narayan G., Mandel K. S., Biswas R., Hložek R., 2019, Publications of the Astronomical Society of the Pacific, 131, 118002

[18] Muthukrishna D., Mandel K. S., Lochner M., Webb S., Narayan G., 2021, arXiv e-prints, p. arXiv:2111.00036

[19] Naul B., Bloom J. S., Pérez F., van der Walt S., 2018, Nature Astronomy, 2, 151

[20] Pruzhinskaya M. V., Malanchev K. L., Kornilov M. V., Ishida E. E. O., Mondon F., Volnova A. A., Korolev V. S., 2019, Monthly Notices of the Royal Astronomical Society, 489, 3591

[21] Soraisam M. D., et al., 2020, Astrophysical Journal, 892, 112

[22] Villar V. A., Cranmer M., Berger E., Contardo G., Ho S., Hosseinzadeh G., Yao-Yu Lin J., 2021, arXiv e-prints, p. arXiv:2103.12102

[23] Webb S., et al., 2020, Monthly Notices of the Royal Astronomical Society, 498, 3077

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] This can be shared upon request to the authors.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] Data available under the Creative Commons License.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 DNN Model loss function

Before defining the loss function, we first develop a generative model of the latent flux of a transient s in passband p 3 days in the future at time $T+3$. We aim to model the underlying latent flux with the neural network as follows,

$$F_{sp(T+3)}(\boldsymbol{w}) = \tilde{y}_{sp(T+3)}(\boldsymbol{w}) + \epsilon_{\text{int},sp(T+3)}(\boldsymbol{w}), \tag{8}$$

where the error $\epsilon_{\text{int},sp(T+3)}(\boldsymbol{w}) \sim \mathcal{N}\left(0, \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})\right)$ is a zero-mean Gaussian random variable with variance $\tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})$. Thus, we write the predictive distribution of the latent flux as follows,

$$\mathcal{P}(\boldsymbol{F}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)}, \boldsymbol{w}) = \prod_{p=1}^{N_p} \mathcal{N}\left(F_{sp(T+3)}(\boldsymbol{w}) \mid \tilde{y}_{sp(T+3)}(\boldsymbol{w}), \tilde{\sigma}^2_{\text{int},sp(T+3)}(\boldsymbol{w})\right). \tag{9}$$

Next, a generative model of the observed flux is derived by adding a measurement error to the latent flux as follows,

$$D_{sp(T+3)} = F_{sp(T+3)}(\boldsymbol{w}) + \epsilon_{D,sp(T+3)}, \tag{10}$$

where we assume that the measurement error $\epsilon_{D,sp(T+3)} \sim \mathcal{N}(0, \sigma^2_{D,sp(T+3)})$ is a zero-mean Gaussian random variable with variance $\sigma^2_{D,sp(T+3)}$.

Typically, researchers will not use the uncertainty in the data within the loss function (e.g. Jamal & Bloom [10], Villar et al. [22], however, work by Naul et al. [19] included data uncertainty without model uncertainty). In this work, we construct our loss function to include both predictive uncertainties $\tilde{\sigma}_{\mathrm{int},sp(T+3)}(\boldsymbol{w})$ and flux uncertainties $\sigma_{D,sp(T+3)}$. Given equations 8 and 10, we write the likelihood function of the probabilistic DNN as follows,

$$\begin{aligned}\mathcal{P}(\boldsymbol{D}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w}) &= \prod_{p=1}^{N_p}\mathcal{N}\left(D_{sp(T+3)} \mid \tilde{y}_{sp(T+3)}(\boldsymbol{w}), \tilde{\sigma}^2_{\mathrm{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)}\right) \\ &= \prod_{p=1}^{N_p}\left(2\pi(\tilde{\sigma}^2_{\mathrm{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)})\right)^{-0.5} \\ &\quad \times \exp\left(-0.5\frac{(\tilde{y}_{sp(T+3)}(\boldsymbol{w}) - D_{sp(T+3)})^2}{\tilde{\sigma}^2_{\mathrm{int},sp(T+3)}(\boldsymbol{w}) + \sigma^2_{D,sp(T+3)}}\right).\end{aligned} \tag{11}$$

Following Gal & Ghahramani [6], we define the prior over the weights as a zero-mean Normal distribution,

$$\mathcal{P}(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid 0, \boldsymbol{I}/l^2), \tag{12}$$

where $\boldsymbol{I}$ is the identity matrix and l is the prior length-scale that regularises how large the weights can be. The posterior over the weights is given by the product of the prior distribution and the likelihood function over all N_s transients at all N_t time-steps,

$$\mathcal{P}(\boldsymbol{w}|\boldsymbol{X}) \propto \mathcal{P}(\boldsymbol{w})\prod_{s=1}^{N_s}\prod_{T=-70}^{80}\mathcal{P}(\boldsymbol{D}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w}), \tag{13}$$

where we ignore the Bayesian evidence as a scaling constant that is unnecessary for this work. We would ideally like to sample the negative log posterior while training our DNN, and so we derive the log prior from equation 12 as $\log\mathcal{P}(\boldsymbol{w}) = \mathrm{constant} - l^2||\boldsymbol{w}||_2^2/2$. We can ignore the additive constant not necessary for our optimisation and follow Gal & Ghahramani [6] to implement the log prior by including an L_2 regularisation term $\lambda||\boldsymbol{w}||_2^2$ weighted by some weight decay that averages over the number of transients N_s and time-steps N_t,

$$\lambda = \frac{l^2(1-d)}{2N_sN_t}, \tag{14}$$

where d is the dropout rate (set to 0.2 in this work), and we set $l = 0.2$ consistent with work by Gal & Ghahramani [6][3]. Here, we have included the $(1-d)$ term to account for the dropout regularisation used in our work. With this term, the L_2 regularisation term $\lambda||\boldsymbol{w}||_2^2$ matches the log prior, $\log\mathcal{P}(\boldsymbol{w})$, averaged over the number of transients and time-steps. However, we point out that our λ slightly differs from equation 18 of Gal & Ghahramani [7] because we use the negative log-likelihood instead of a squared loss as the cost function of our DNN. Furthermore, we also add the caveat that because of this difference of loss functions and our inclusion of a probabilistic neural network that outputs a predictive mean and standard deviation instead of a point estimate, it is not clear that the demonstration of MC dropout as an approximation to Bayesian neural networks in Gal & Ghahramani [6] necessarily holds true in our work. Future machine learning research should check the validity of MC dropout as a Bayesian approximation in a broader range of neural network architectures.

Since we use dropout regularisation, we define a dropout objective function over all time-steps and over all transients that we aim to minimise while training the neural network model as follows,

$$\mathrm{obj}(\boldsymbol{w}) = \sum_{s=1}^{N_s}\sum_{T=-70}^{80}\left[-\log\mathcal{P}(\boldsymbol{D}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w}) + \lambda||\boldsymbol{w}||_2^2\right] \tag{15}$$

where we sum the log-likelihood and L_2 regularisation term over all N_t time-steps (between -70 and 80 days) and N_s transients in the training set. To train the DNN and determine optimal values of its parameters $\hat{\boldsymbol{w}}$, we minimise the dropout objective function with the sophisticated and commonly used `Adam` gradient descent optimiser [13].

To make predictions, we evaluate the predictive distribution of the latent flux defined as follows,

$$\mathcal{P}(\boldsymbol{F}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)}) = \int\mathcal{P}(\boldsymbol{F}_{s(T+3)}|\boldsymbol{X}_{s(t\leq T)},\boldsymbol{w})\mathcal{P}(\boldsymbol{w}|\boldsymbol{X})d\boldsymbol{w}, \tag{16}$$

where we are marginalising over the weights of the network by integrating the product of the predictive distribution of the latent flux given the network weights (first term in the integrand and defined in equation 9)

[3] See Section 4.2 of Gal & Ghahramani [7] for a detailed explanation of this prior and `https://github.com/yaringal/DropoutUncertaintyExps/blob/master/net/net.py` for an example implementation of this L_2 regularisation by Yarin Gal.

and the posterior distribution over the network weights (second term in the integrand and defined in equation 13). The integral is intractable, and so we approximate it by using Monte Carlo dropout at inference time to sample the posterior distribution, as described in [6]. We draw 100 samples from the posterior $\mathcal{P}(\boldsymbol{w}|\boldsymbol{X})$ by running 100 forward passes of the neural network for a given input. Each run of the neural network outputs both a mean $\tilde{y}_{sp(T+3)}(\boldsymbol{w}_{\rm draw})$ and standard deviation $\tilde{\sigma}_{{\rm int},sp(T+3)}(\boldsymbol{w}_{\rm draw})$ because of our probabilistic neural network architecture. To include the variance of each draw in the marginal predictive uncertainty $\sigma_{y,sp(T+3)}$, we compute $F_{sp(T+3)}(\boldsymbol{w}_{\rm draw}) \sim \mathcal{N}\left(\tilde{y}_{sp(T+3)}(\boldsymbol{w}_{\rm draw}), \tilde{\sigma}^2_{{\rm int},sp(T+3)}(\boldsymbol{w}_{\rm draw})\right)$. We estimate the marginal predictive mean and uncertainty as the sample mean and standard deviation of the 100 values of $F_{sp(T+3)}(\boldsymbol{w}_{\rm draw})$ taken from the 100 forward passes of the neural network, respectively:

$$y_{sp(T+3)} = \frac{1}{100}\sum_{\rm draw=1}^{100} F_{sp(T+3)}(\boldsymbol{w}_{\rm draw}), \tag{17}$$

$$\sigma_{y,sp(T+3)} = \sqrt{\frac{1}{100}\sum_{\rm draw=1}^{100}\left(F_{sp(T+3)}(\boldsymbol{w}_{\rm draw}) - y_{sp(T+3)}\right)^2}. \tag{18}$$

These outputs are used to compute the anomaly scores.

A.2 Comparison of DNN and Bazin predictive power

We performed the following analysis to evaluate why the DNN model was less effective at identifying anomalies than the Bazin model. To compare the models on an even scale, we defined the Measurement-Uncertainty-Scaled Prediction Error as follows,

$$\mathrm{MUSPE}_{spt} = \frac{(y_{spt} - D_{spt})}{\sigma_{D,spt}}. \tag{19}$$

To analyse why the SNIa Bazin model is better than than the DNN at identifying anomalies, we plot the prediction errors for the SNIa models applied to each of the other transient classes for Bazin and DNN in Figures 4 and 5, respectively. We expect that the SNIa models should predict the SNIa light curves best, and indeed, we see that these blue lines for the SNIa have nearly the best prediction error distributions. In Figure 4, the prediction errors are significantly worse for the more anomalous classes (SLSNe, TDEs, PISNe, ILOTs) with deviations ranging up to 5 sigma. However, the prediction errors for these classes in the DNN are much smaller, not much more than 1 sigma deviations. This indicates that the Bazin model is much worse at predicting the observations from these anomalous classes than the DNN, and hence is better at identifying them as anomalies despite being better able to predict observations from common transients. We conclude that the flexibility of the DNN makes it excellent at predicting future fluxes for any transient despite only being trained on SNIa. This is great if the task was prediction, but it is a poor choice for anomaly detection in this framework.

Bazin

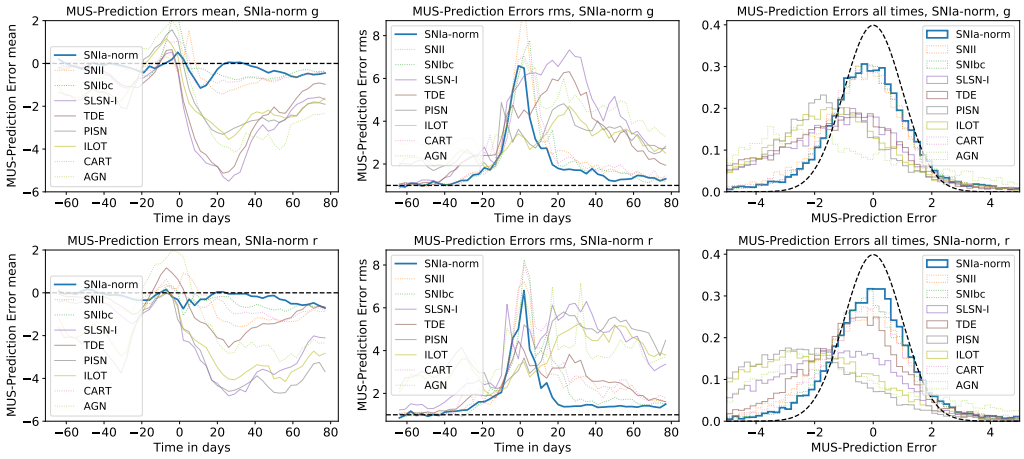


Figure 4: Distribution of the Measurement-Uncertainty-Scaled Prediction Errors (MUSPE) for the Bazin SNIa model at different times. We apply the Bazin SNIa model to each other class' datasets and compute Equation 19 at each time-step. We have not plotted the kilonova or uLens-BSR classes here because they show significant deviations, indicating that the SNIa model is very bad at predicting these classes, and hence easily identifies them as anomalies. We show the mean and root mean square (rms) for each prediction error distribution at each time-step in the first three panels, with the g band shown in the top row of panels, and the r band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. We plot a unit Gaussian as a black dashed line to help guide the eye.

DNN

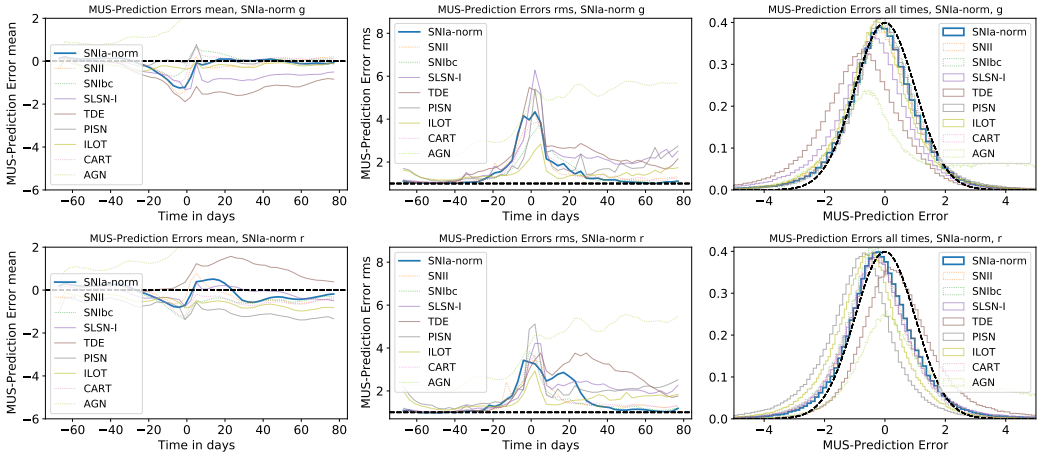


Figure 5: Distribution of the Measurement-Uncertainty-Scaled Prediction Errors (MUSPE) for the DNN SNIa model at different times. We apply the DNN SNIa model to each other class' datasets and compute Equation 19 at each time-step. We have not plotted the kilonova or uLens-BSR classes here because they show significant deviations, indicating that the SNIa model is very bad at predicting these classes, and hence easily identifies them as anomalies. We show the mean and root mean square (rms) for each prediction error distribution at each time-step in the first three panels, with the g band shown in the top row of panels, and the r band shown in the bottom row of panels. In the last column, we plot the distribution of scaled errors across all times. We plot a unit Gaussian as a black dashed line to help guide the eye.