

Real-time detection of anomalies in large-scale transient surveys

Daniel Muthukrishna,¹★ Michelle Lochner,² and Sara Webb³

¹*Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, United Kingdom*

²*African Institute for Mathematical Sciences, 6 Melrose Road, Muizenberg, 7945, South Africa*

³*Centre for Astrophysics and Supercomputing, Swinburne University of Technology, John St, Hawthorn VIC 3122, Australia*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

New large-scale transient surveys will observe millions of transient alerts each night, making standard approaches of visually identifying new and interesting transients unfeasible. We present a novel method of automatically detecting anomalies in real-time transient light curves. Using state-of-the-art deep recurrent neural networks with Long Short Term Memory (LSTM) units, we present one of the first methods designed to provide anomaly scores of photometric data as a function of time. We build six generative models of six common transient classes, trained on light curve simulations matching the observing properties of the Zwicky Transient Facility (ZTF) public survey. Our method is able to identify anomalous transients (such as the Kilonovae of the GW170817 event, and peculiar SNe Ia) within days of their explosion. We can obtain anomaly scores with respect to the six trained models. This method can also be used for classification with a Bayesian model selection. We demonstrate the effective performance of our method on simulations and real data from the live ZTF data stream.

Key words: methods: data analysis – methods: observational – techniques: photometric, virtual observatory tools – supernovae: general

1 INTRODUCTION

Astronomy is reaching an unprecedented phase of big data, observing more events than humans can possibly visually examine alone. Upcoming large scale surveys of the transient universe such as the Large Synoptic Survey Telescope (LSST) will observe two orders of magnitude more transient phenomena than any survey to date, (Ivezić et al. 2019). LSST is expected to receive over 10 million transient alerts each night, making it utterly unfeasible to visually examine each event. However, for a long time, discovery in astronomy has been driven by identifying anomalies in data sets. With so much data, the task of identifying anomalous and interesting objects for follow-up needs to be automated.

Anomaly detection (AD) is the task of finding outliers in a data-driven fashion, and the aim is to find outliers that are scientifically interesting, rather than random statistical fluctuations. Within astronomy, anomaly detection algorithms have been used to identify anomalous galaxy spectra (e.g. Baron & Poznanski 2017), problematic objects in photometric redshift estimation tasks (e.g. Hoyle et al. 2015), and

others. Most approaches to Anomaly detection have generally fallen into two different methods: clustering or subspace analysis. More recently, isolation forests and GANs have also been applied to anomaly detection. However, dealing with time-series data makes the standard machine learning approaches to AD unfeasible for real-time detection.

While efforts have been made to classify the real-time transient universe photometrically (Muthukrishna et al. 2019a; Möller & de Boissière 2019), not much has been put toward discovering new and interesting phenomena in real-time data streams. Advanced neural network architectures are non-feature-based approaches that have recently been shown to have several benefits such as low computational cost, and being robust against some of the biases that can afflict machine learning techniques that require “expert-designed” features (Aguirre et al. 2018; Charnock & Moss 2017; Moss 2018; Naul et al. 2018). The use of Artificial Neural Networks (ANN, McCulloch & Pitts 1943) and deep learning, in particular, has seen dramatic success in image classification, speech recognition, and computer vision, outperforming previous approaches in many benchmark challenges (Krizhevsky et al. 2012; Razavian et al. 2014; Szegedy et al. 2015).

★ E-mail: daniel.muthukrishna@ast.cam.ac.uk

In time-domain astronomy, deep learning has recently been used in a variety of classification problems including variable stars (Naul et al. 2018; Hinners et al. 2018), supernova spectra (Muthukrishna et al. 2019b), photometric supernovae (Charnock & Moss 2017; Moss 2018; Möller & de Boissière 2019; Pasquet et al. 2019), and sequences of transient images (Carrasco-Davis et al. 2018). A particular class of ANNs known as Recurrent Neural Networks (RNNs) are particularly suited to learning sequential information (e.g. time-series data, speech recognition, and natural language problems). While ANNs are often feed-forward (e.g. convolutional neural networks and multilayer perceptrons), where information passes through the layers once, RNNs allow for cycling of information through the layers. They are able to encode an internal representation of previous epochs in time-series data, which along with real-time data, can be used for classification.

A variant of RNNs known as Long Short Term Memory Networks (LSTMs, Hochreiter & Schmidhuber 1997) improve upon standard RNNs by being able to store long-term information, and have achieved state-of-the-art performance in several time-series applications. In particular, they revolutionized speech recognition, outperforming traditional models (Fernández et al. 2007; Hannun et al. 2014; Li & Wu 2015) and have very recently been used in the trigger word detection algorithms popularized by *Apple’s Siri*, *Microsoft’s Cortana*, *Google’s voice assistant*, and *Amazon’s Echo*. Naul et al. (2018) and Hinners et al. (2018) have had excellent success in variable star classification. Charnock & Moss (2017) applied the technique to supernova classification. They used supernova data from the SNPhotCC and fed the multiband photometric full lightcurves into their LSTM architecture to achieve high SNIa vs non-SNIa binary classification accuracies. Moss (2018) recently followed this up on the same data with a novel approach applying a new phased-LSTM (Neil et al. 2016) architecture. These approaches have the advantage over previous supernova photometric classifiers of not requiring computationally-expensive and user-defined (and hence, possibly biased) feature engineering processes.

In this paper, we apply state-of-the-art deep learning approaches to identify new, interesting and anomalous light-curve behaviour to real-time transient data-streams. In section 2, we describe the data used to train our deep neural network (DNN) architecture. In section 3, we detail our recurrent neural network (RNN) based regression approach to anomaly detection. In section 4, we evaluate the performance of our generative models on simulated and real data. And in section 5, we present the conclusions of our work.

2 DATA

2.1 Simulations

One of the key challenges with training algorithms for upcoming transient surveys is the lack of labelled samples that are appropriate for training. Moreover, even once a survey commences, it can take a significant amount of time to accumulate a well-labelled sample that is large enough to develop robust learning algorithms. To meet this difficulty for LSST, the PLAsTiCC collaboration has developed the infrastructure to simulate light curves of astrophysical sources

with realistic sampling and noise properties. This effort was one component of an open-access challenge to develop algorithms that classify astronomical transients. By adapting supernova analysis tools such as SNANA (Kessler et al. 2009) to process several models of astrophysical phenomena from leading experts, a range of new transient behavior included in the PLAsTiCC dataset. The challenge has recently been released to the public on Kaggle¹ (The PLAsTiCC team et al. 2018) along with the metric framework to evaluate submissions to the challenge (Malz et al. 2018). The PLAsTiCC models are the most comprehensive enumeration of the transient and variable sky available at present.

We use the PLAsTiCC transient class models and the simulation code developed in Kessler et al. (2019) to create a simulated dataset that is representative of the cadence and observing properties of the ongoing public “Mid Scale Innovations Program” (MSIP) survey at the ZTF (Bellm 2014). This allows us to compare the validity of the simulations with the live ZTF data stream, and apply our generative models to it as illustrated in section 4.

2.1.1 Zwicky Transient Facility

ZTF is the first of the new generation of optical synoptic survey telescopes and builds upon the infrastructure of the Palomar Transient Factory (PTF, Rau et al. 2009). It employs a 47 square degree field-of-view camera to scan more than 3750 square degrees an hour to a depth of 20.5 - 21 mag (Graham & Zwicky Transient Facility (ZTF) Project Team 2018). It is a precursor to the LSST and will be the first survey to produce one million alerts a night and to have a trillion row data archive. To prepare for this unprecedented data volume, we build an automated regressor for each transient class trained on a large simulated ZTF-like dataset that contains a labelled sample of transients.

We built simulations matching the observing properties of the ZTF using SNANA (Muthukrishna et al. 2018). SNANA simulates millions of light curves for each model, following a class-specific luminosity function prescription within the ZTF footprint. The sampling and noise properties of each observation on each light curve is set to reflect a random sequence from within the observing conditions library. The simulated light curves thus mimic the ZTF observing properties with a median cadence of 3 days in the *g* and *r* passbands. As ZTF had only been operating for four months when we constructed the observing conditions library, it is likely that our simulations are not fully representative of the survey. Nevertheless, this procedure is more realistic than simulating the observing conditions entirely, as we would have been forced to do if we had developed our work for LSST or *WFIRST*. We verified that the simulated light curves have similar properties to observed transient sources detected by ZTF that have been announced publicly. The dataset consists of a labelled set of 48029 simulated transients evenly distributed across a range of different classes. An example of a simulated light curve from each class is shown in Fig. 1.

The specific models used in the simulations derived from Kessler et al. (2019) are SNIa-norm: Guy et al. (2010);

¹ <https://www.kaggle.com>

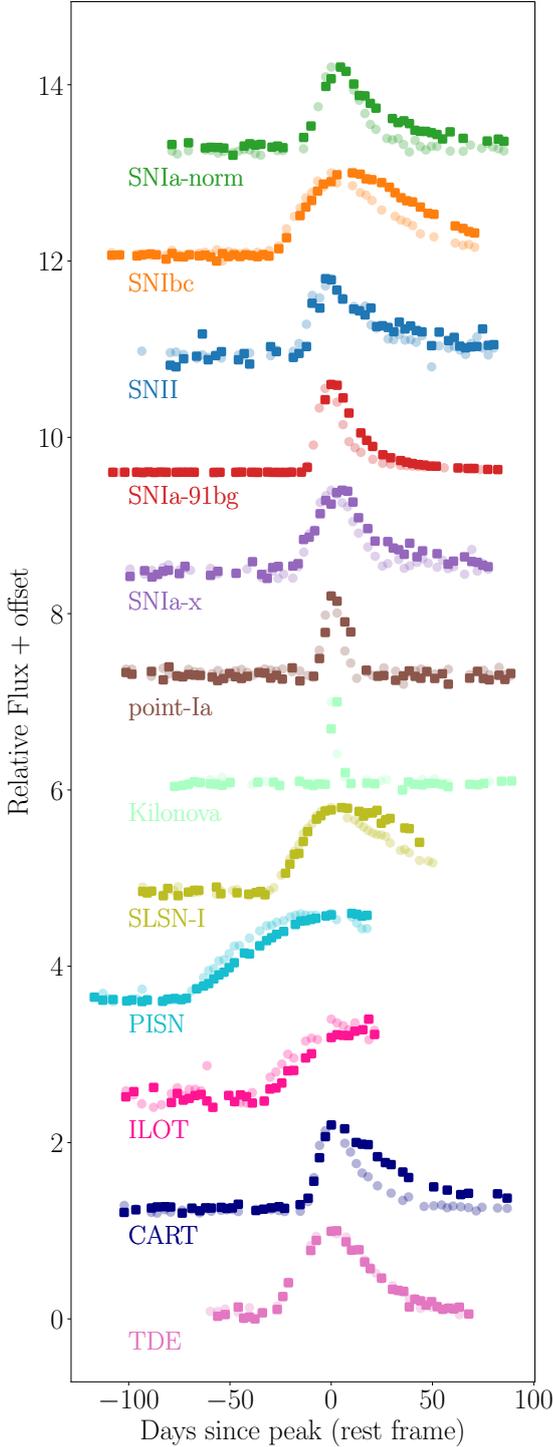


Figure 1. The light curves of one example transient from each of the 12 transient classes is plotted with an offset. We have only plotted transients with a high signal-to-noise and with a low simulated host redshift ($z < 0.2$) to facilitate comparison of light curve shape between the classes. The dark-coloured square markers plots the r band light curves of each transient, while the lighter-coloured circle markers are the g band light curves of each transient.

Kessler et al. (2013); Pierel et al. (2018), SNIbc: Kessler et al. (2010); Pierel et al. (2018); Guillochon et al. (2018); Villar et al. (2017), SNII: Kessler et al. (2010); Pierel et al. (2018); Guillochon et al. (2018); Villar et al. (2017), SNIa-91bg: (Galbany et al. in prep.), SNIa-x: Jha (2017), pointIa: Shen et al. (2010), Kilonovae: Kasen et al. (2017), SLSN (Super-luminous SNe): Guillochon et al. (2018); Nicholl et al. (2017); Kasen & Bildsten (2010), PISN (Pair Instability SNe): Guillochon et al. (2018); Villar et al. (2017); Kasen et al. (2011), ILOT (Intermediate Luminosity Transients): Guillochon et al. (2018); Villar et al. (2017), CART (Calcium-rich Transients): Guillochon et al. (2018); Villar et al. (2017); Kasliwal et al. (2012), TDE (Tidal Disruption Events): Guillochon et al. (2018); Mockler et al. (2019); Rees (1988).

Each simulated transient dataset consists of a time series of flux and flux error measurements in the g and r ZTF bands, along with sky position, Milky Way dust reddening, a host-galaxy redshift, and a photometric redshift. The models used in PLAsTiCC were extensively validated against real observations by several complementary techniques, as described by Narayan et al. (2019, in prep.). We split the total set of transients for each class into two parts: 80% for the *training set* and 20% for the *testing set*, respectively. The *training set* is used to train the generative model to predict the next photometric data given all data until that point in time, while the *testing set* is used to test the performance of the model.

2.1.2 Trigger for Issuing Alerts

The primary method used for detecting transient events is to subtract real-time or archival data from a new image to detect a change in observed flux. This is known as *difference imaging*, and has been shown to be effective, even in fields that are crowded or associated with highly non-uniform unresolved surface brightness (Tomaney & Crotts 1996; Bond et al. 2001). Most transient surveys, including ZTF, use this method, and ‘trigger’ a transient event when there is a detection in a difference image that exceeds a 5σ signal-to-noise (S/N) threshold. Throughout this work, we use *trigger* to identify this time of detection.

3 METHOD

3.1 Light curve preprocessing

Arguably, one of the most important aspects in an effective learning algorithm is the quality of the training set. In this section we discuss efforts to ensure that the data is processed in a uniform and systematic way before we train our DNN.

The light curves are measured in flux units, as is expected for the ZTF difference imaging pipeline. The simulations have a significant fraction of the observations being 5-10 sigma outliers. These outliers are intended to replicate the difference image analysis artifacts, telescope CCD deficiencies, and cosmic rays seen in observational data. We perform ‘sigma clipping’ to reject these outliers. We do this by rejecting photometric points with flux uncertainties that are more than 3σ from the mean uncertainty in each pass-band, and iteratively repeat this clipping 5 times. Next, we

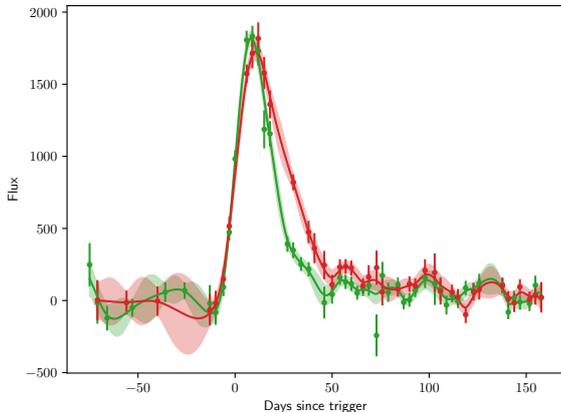


Figure 2. An example SNIa fit with Gaussian Process regression. The two light curves are fit with two GPs with a Matern-32, respectively, but the hyperparameters of each are shared by optimising a combined log-likelihood. The green and red represent the g and r passbands, respectively. The data is shown with uncertainties, the solid line is the mean of the GP samples, and the shaded region shows the $1 - \sigma$ variation of the GP samples.

correct the light curves for interstellar extinction using the reddening function of Fitzpatrick (1999). We assume an extinction law, $R_V = 3.1$, and use the central wavelength of each ZTF filter to de-redden each light curve listed as follows²:

$$g: 4767 \text{ \AA}, r: 6215 \text{ \AA}.$$

3.2 Gaussian process regression

Gaussian process (GP) regression has been shown to be effective for astronomical light curve modelling. Recently, Lochner et al. (2016) and Boone (2019) used GP regression as a core part of the modelling process of their light curve classification approaches. GP regression enables us to deal with missing data and to interpolate the light curves at regular intervals, while also incorporating uncertainty.

We use the `celerite` package (Foreman-Mackey et al. 2017) to fit each light curve with a GP. We use a Matern kernel, and share the same parameters for all passbands in a light curve by minimising the combined negative log-likelihood of the fit to each passband light curve. An example of our GP fit to a SNIa in the training set is illustrated in Figure 2.

We augment our training set by sampling the GP regressive fit of each transient 100 times, effectively increasing the size of our training set 100-fold.

Irregularly sampled time-series data is a common problem in machine learning, and is particularly prevalent in astronomical surveys where the intranight cadence choices and seasonal constraints lead to naturally arising temporal gaps. Therefore, interpolate from the gaussian process such that each light curve is sampled at 3-day intervals between

$-70 < t < 80$ days since trigger (or as far as the observations exist), to give a vector of length $n = 50$, where we set the values outside the data range to zero. We ensure that each light curve in a given passband is sampled on the same 3-day grid. The final input image for each transient is a matrix with each row composed of the imputed light curve fluxes for each passband and an additional row containing repeated values of the MW dust reddening. Hence, the input image is an $n \times (p + 1)$ matrix, where p is the number of passbands.

3.3 Deep Neural Network Architecture

Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been shown to achieve state-of-the-art performance in many benchmark time-series and sequential data applications (Bahdanau et al. 2014; Sutskever et al. 2014; Che et al. 2018). Its success in these applications is due to its ability to retain an internal memory of previous data, and hence capture long-term temporal dependencies of variable-length observations in sequential data. We extend this architecture to our case with a time-varying multi-channel (multiple passbands) input and a time-varying multi-class output.

The deep neural network (DNN) is illustrated in Fig. 3. We have developed the network with the high level Python API, Keras (Chollet et al. 2015), built on the recent highly efficient TensorFlow machine learning system (Abadi et al. 2016). We describe the architecture in detail here.

Input: The input is an $n \times (p + 1)$ matrix. However, as we are implementing a sequence regressor, we can consider the input at each time-step as being vector of length $(p + 1)$.

First LSTM Layer: Long Short Term Memory networks are an improved version of a standard RNN. They are able to capture long-term dependencies in time-varying data with parameters that control the information that should be remembered at each step along the light curve. We use the first LSTM layer to read the input sequence one time-step at a time and encode it into a higher-dimensional representation. We set-up this LSTM layer with 100 units such that the output is a vector of shape 1×100 .

Second LSTM Layer: The second LSTM layer is conditioned on the input sequence. It takes the output of the previous LSTM and generates an output sequence. Again, we use 100 units in the LSTM to maintain the $n \times 100$ output shape. We use uni-directional LSTMs that enable only information from previous time-steps to be encoded and passed onto future time-steps.

Batch Normalization: We then apply *Batch Normalization* (first introduced in Ioffe & Szegedy 2015) to each LSTM layer. This acts to improve and speed up the optimization while adding stability to the neural network and reducing overfitting. While training the DNN, the distribution of each layer’s inputs changes as the parameters of the previous layers change. To allow the parameter changes during training to be more stable, batch normalization scales the input. It does this by subtracting the mean of the inputs and then dividing it by the standard deviation.

Dropout: We also implement dropout regularization to each layer of the neural network to reduce overfitting during training. This is an important step that effectively ignores randomly selected neurons during training such that their

² We use the `extinction` code: <https://extinction.readthedocs.io>

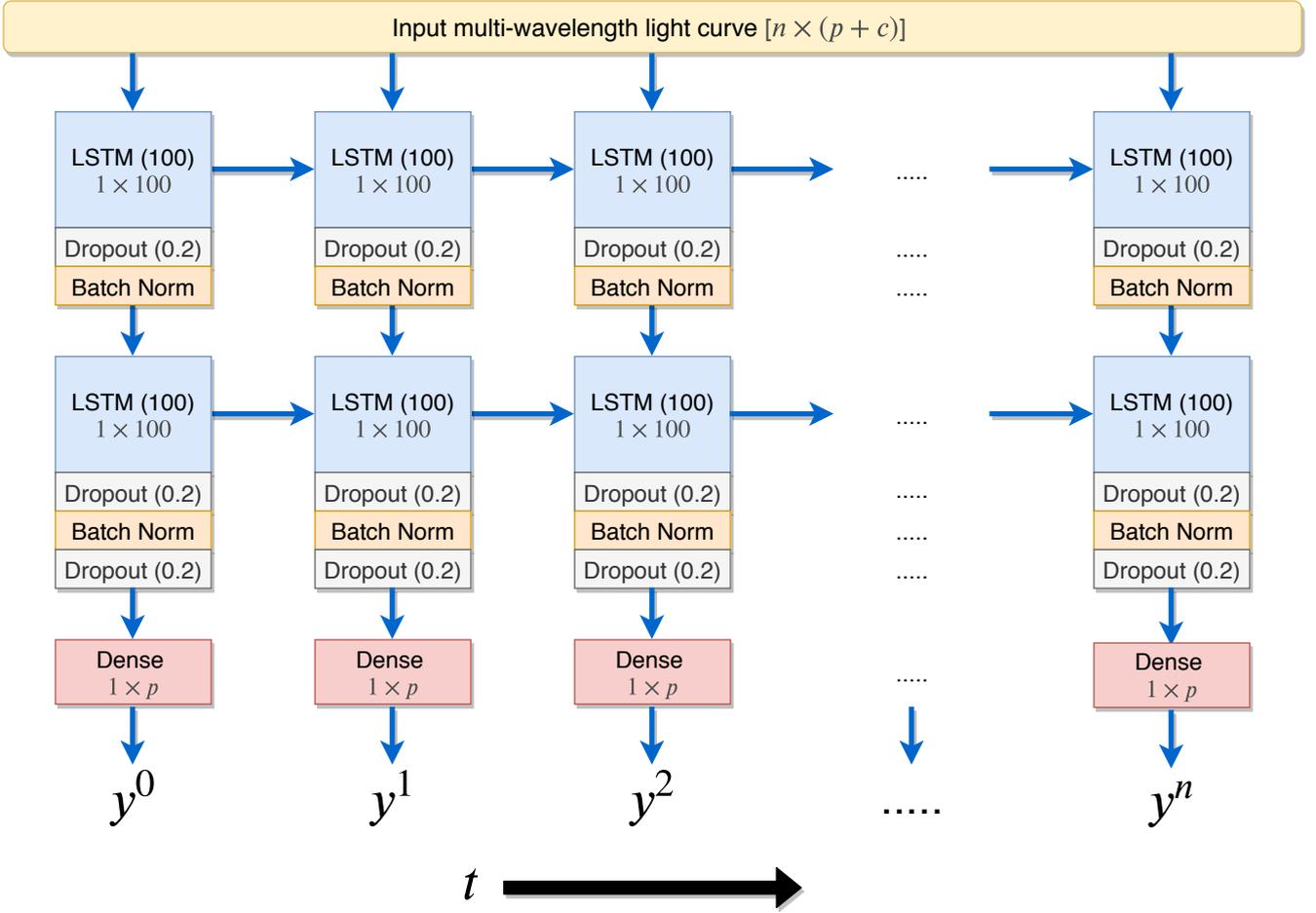


Figure 3. Schematic of the deep recurrent neural network architecture used in our method. Each column in the diagram is one of the n time steps of the processed light curve, while each row represents a different neural network layer. The grey text in each block states the shape of the output matrix of each layer in that block. The input image is composed of an $n \times (p + 1)$ matrix consisting of the light curve fluxes in each passband and Milky Way reddening. Two uni-directional LSTM layers of size 100 are used for encoding and decoding the input sequences, respectively. It is in these RNN layers that information about previous time-steps is encoded. Batch normalization is applied between each layer to normalize the network parameters and hence, speed the training process. To counter overfitting during training, we employ the dropout optimization technique (Srivastava et al. 2014) to the neurons in each of the LSTM and Batch Normalization layers, and set the dropout rate to 20%. Finally, a fully-connected (dense) layer is applied at each time-step. We wrap the final layer in Keras’ *Time Distributed* layer so that each time step is treated independently, and only uses information from the current and previous time-steps.

contribution to the network is temporarily removed. This process causes other neurons to more robustly handle the representation required to make predictions for the missing neurons, making the network less sensitive to the specific weights of any individual neuron. We set the dropout rate to 20% of the neurons present in the previous layer each time the Dropout block appears in the DNN in Fig. 3.

Dense Layer: A dense (or fully-connected) layer is the simplest type of neural network layer. It connects all 100 neurons at each time-step in the previous layer using equation 1. As we are interested in time-varying predictions, we wrap this Dense layer with a *Time-Distributed* layer, such that the dense layer is applied independently at each time-step, hence giving an output matrix of shape $n \times p$.

Neurons: The output of each neuron in a neural network layer can be expressed as the weighted sum of the connec-

tions to it from the previous layer:

$$\hat{y}_i = f\left(\sum_{j=1}^M W_{ij} x_j + b_i\right), \quad (1)$$

where x_j are the different inputs to each neuron from the previous layer, W_{ij} are the weights of the corresponding inputs, b_i is a bias that is added to shift the threshold of where inputs become significant, j is an integer running from 1 to the number of connected neurons in the previous layer (M), and i is an integer running from 1 to the number of neurons in the next layer. For the Dense layer, \mathbf{x} is simply the (1×100) matrix from the output of the LSTM and Batch Normalisation, \mathbf{y} is made up of the p output passband fluxes, j runs from 1 to p and i runs across the 100 input neurons from the LSTM. The matrix of weights and biases in the Dense

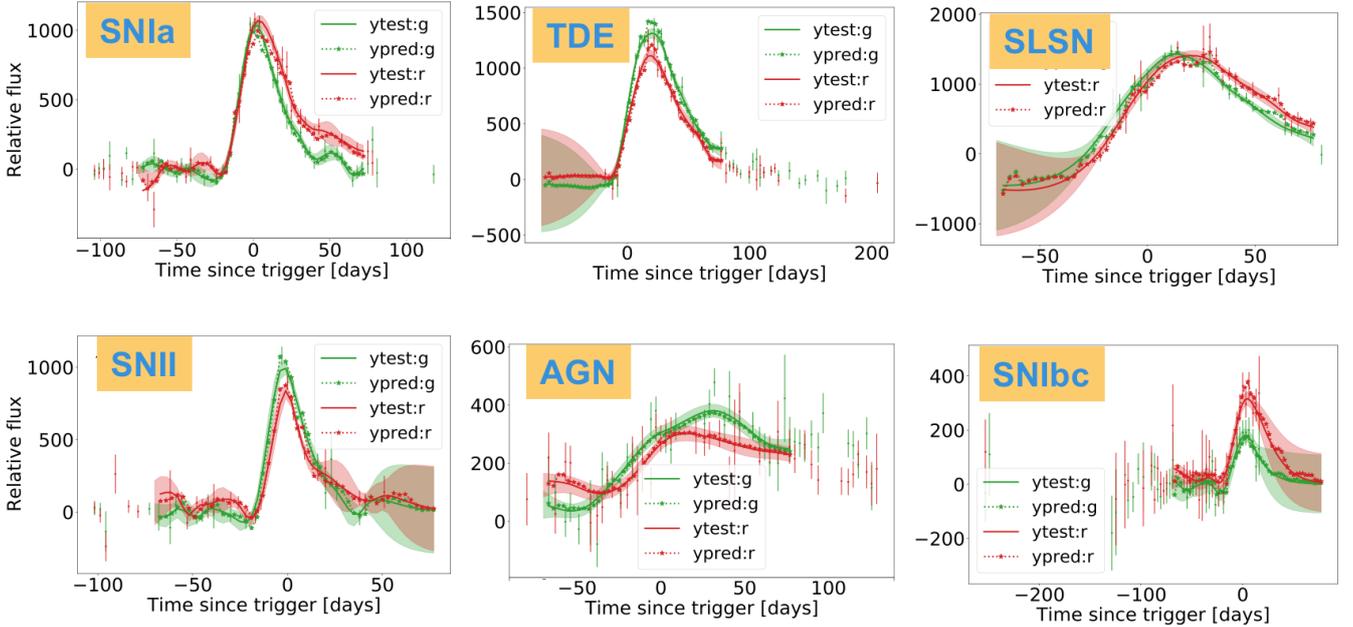


Figure 4. The performance of six generative models on example transients taken from the testing set of each, respectively. The observed data is shown with uncertainties in the g and r band. The GP fit is shown as the shaded regions, with the mean GP fit shown as the solid lines. The predictions from the deep neural network regressor is illustrated as the dashed line with star points. The predictions use all the data previous in time to each given data point. In each case, the predictions are very close to the GP fits, indicating that each of the regressive models are able to fit light curves in their testing set well.

layer and throughout the LSTM layers are some of the free parameters that are computed by **TensorFlow** during the training process.

Activation function: As with any neural network, each neuron applies an activation function $f(\cdot)$ to bring non-linearity to the network and hence help it to adapt to a variety of data. For feed-forward networks it is common to make use of Rectified Linear Units (ReLU, [Nair & Hinton 2010](#)) to activate neurons. However, the LSTM architecture uses sigmoid activation functions as it outputs a value between 0 and 1 and can either let no flow or complete flow of information from previous time-steps.

The several layers in the DNN create a model that has over one hundred thousand free parameters. As we feed in our training set in batches of 64 light curves at a time, the neural network updates and optimizes these parameters. While the size of the parameter space seems insurmountable, the **Adam** optimizer is able to compute individual adaptive learning rates for different parameters from estimates of the mean and variance of the gradients and has been shown to be extraordinarily effective at optimizing high-dimensional deep learning models.

With the often quoted ‘black box’ nature of machine learning, it is always a worry that the machine learning algorithms are learning traits that are specific to the training set but do not reflect the physical nature of the classes more generally. Ideally, we would like to ensure that the model we build both accurately captures regularities in the training data while simultaneously generalizing well to unseen data. Simplistic models may fail to capture important patterns in the data, while models that are too complex may

overfit random noise and capture spurious patterns that do not generalize outside the training set. While we implement regularization layers (dropout) to try to prevent overfitting, we also monitor the performance of the models on the training and testing sets during training. In particular, we ensure that we do not run the training over so many iterations that the difference between the values of the objective function evaluated on the training set and the testing set become significant.

4 RESULTS

In this section we detail the performance of our method on simulated ZTF light curves and on some real data from the live MSIP ZTF data stream. We trained six independent regressive models for SNe Ia, SNe II, SNe Ibc, SLSNe, TDEs, and AGN, respectively. Each training set consisted of approximately 8000 light curves. Each of the light curves were fit with Gaussian Process regression, and were sampled 100 times, increasing the training set 100-fold. We tested the performance of each model on approximately 2000 light curves.

In [Figure 4](#), we illustrate the performance of our six generative models on example light curves in the testing sets of each model. The data is shown with uncertainties, the shaded green and red regions show the $1-\sigma$ Gaussian process regression samples to the data, and the solid line shows the mean GP fit. The dashed line with star markers illustrate the deep neural network generative model predictions. In all six examples, the predictions are very close to the GP fits. Each prediction uses only the data in the previous time-steps.

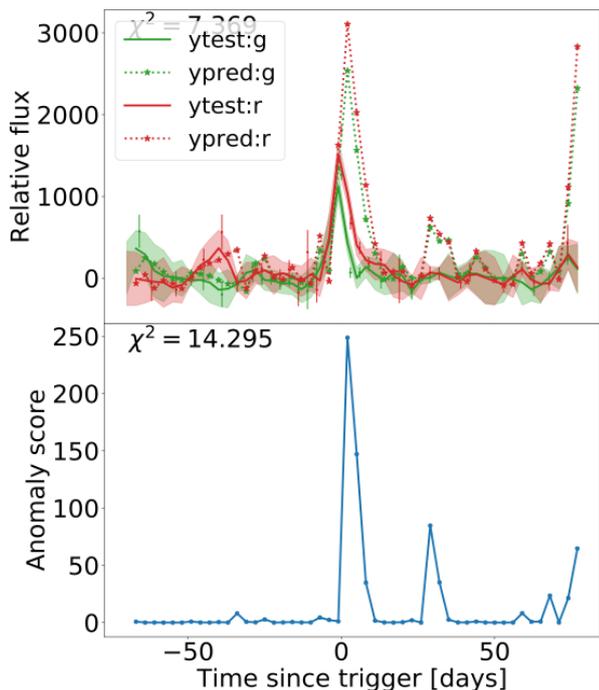


Figure 5. Using the SNIa generative model, we try to fit an example kilonova from our simulated data set. The top panel shows the GP mean as the solid line and the $1-\sigma$ samples as the shaded regions. The dotted line with star points are the DNN’s predictions. In the bottom panel, the χ^2 anomaly score metric is plotted as a function of time. The poor fit highlights that kilonovae are anomalous with respect to a model trained on SNe Ia.

We define a χ^2 metric to highlight the distance of the truth to the DNN predictions.

$$\chi^2 = \frac{(GP_{mean} - predictions)^2}{\sigma^2_{GP}} \quad (2)$$

This metric can be used as a real-time anomaly score. Higher values indicate that the regressive model was less able to fit the data given the training set, while lower scores indicate that the generative model was able to effectively fit the data. In Figure 5, we illustrate an example simulated kilonovae being fit with the SNIa regressive model. The top panel shows the fit, while bottom panel shows the χ^2 anomaly score as a function of time. The poor fit and high anomaly scores indicate that this transient is flagged as anomalous with respect to the SNIa model - showing a first-order success in our method. We note that most kilonovae in data set were similarly flagged as highly anomalous at a similar epoch.

To compare the anomaly scores of all transients in our data set against our trained generative models, we have fit every transient in the testing sets with each of the regressive models. Since we obtain anomaly scores as a function of time, we use the peak anomaly score of each transient, take the mean of this for each transient class, and display this in Figure 6.

The plot highlights the similarity of each trained class

to every other class, and acts as a similarity matrix for the shown transient classes. Higher numbers indicate classes that are more dissimilar, and lower number indicate classes that are more similar. The model trained on the SNIa, as expected, has the lowest score for SNIa, indicating that it can fit SNe Ia very well. Similarly, SNIa-x also have very low scores due to their similarity to normal SNe Ia. Core collapse SNe, SNII, SNIIn, and SNIbc are the next lowest, indicating that these are more similar to normal SNe Ia than the other classes. Kilonovae stand out as very anomalous for every trained model, indicating that the short lifetime and low luminosity of these classes cannot be well-fit with the generative models.

Similar trends can be seen in the other rows of the matrix. For each trained model, the lowest number corresponds to the same class, as expected, showing that each model can be well-fit on what it was trained on. The large numbers on the Ia-91bg class highlight potential problems with this class that requires further examination. The low numbers in the AGN row highlight that the AGN model can fit nearly any type of transient class (except for kilonovae). This is perhaps due to the intrinsic variability of AGN, illustrating that the DNN learned that there are many degenerate functions that can fit an AGN.

Overall, Figure 6 highlights some interesting similarities between transient classes, and confirms what may already be known about their general behaviour. It highlights the overall performance of method on the testing sets, and effectively shows that we are able to identify anomalies with this method.

After training our model on simulated ZTF-like data, we next tested it’s performance on real data from the public MSIP ZTF data stream. In Figure 7 and 8, we plot ZTF18acahuph and ZTF19aadnmgf, real spectroscopically confirmed peculiar SNIa and normal SNIa discovered by ZTF earlier this year, respectively. The top panel of Figure 7, shows the observed photometric data of the peculiar SNIa, while the bottom panel plots the anomaly scores for five of our generative models. We can see that the light curve is anomalous for all of the five models, but least anomalous for SNIa (green line). This illustrates the effectiveness of our method on real data. Peculiar SNIa should be interesting to many transient astronomers, and obtaining following them up, may teach us about their properties, physics, and progenitor system. The high anomaly scores could be used to prioritise follow-up. The fact that it is least anomalous for SNIa leads to the idea that perhaps this method can be used for classification.

In Figure 8, we plot a SNIa discovered by ZTF in the top left panel. The bottom left panel plots the anomaly scores as a function of time for five trained models. In the right panel, we define the likelihood,

$$\text{Likelihood} = \exp\left(-\frac{\chi^2}{2}\right). \quad (3)$$

While future work should aim to do a more thorough Bayesian model comparison, the likelihood measure can be viewed as a trivial method to classify light curves. The SNIa-norm model has the highest likelihood score, correctly indicating that the transient is most likely to be a SNIa and least likely to be a SLSN-I. The AGN model is the next most likely, but this is another example of the AGN model being

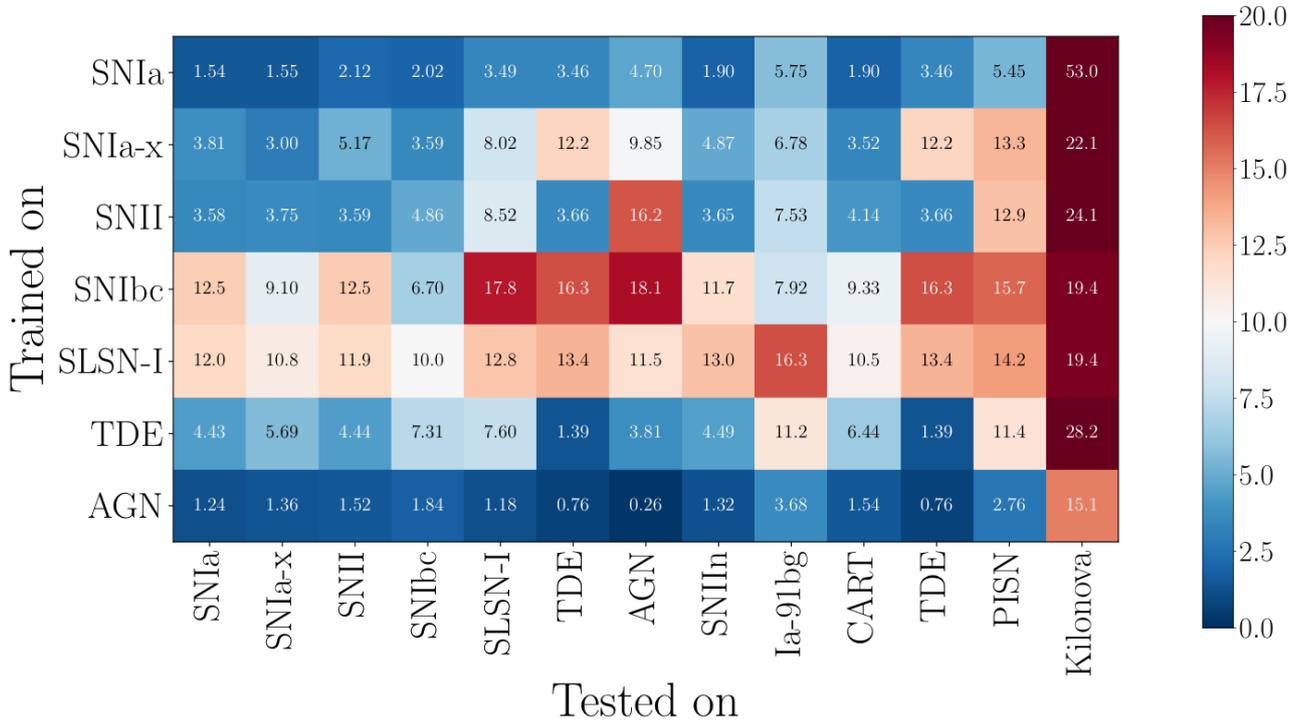


Figure 6. The matrix illustrates the similarity of different transient classes, with lower number being more similar, and higher numbers being less similar. The vertical axis shows seven trained generative models, and the horizontal axis are transients from a range of classes. Each transient in our dataset is fit with the seven generative models, and the peak anomaly score of each is recorded. The mean of the peak anomaly scores for each class are the numbers shown. Each trained model is most similar to transients from its own class, and most dissimilar to kilonovae.

able to fit nearly any transient due to its intrinsic variability as illustrated by the similarity matrix in Figure 6.

5 CONCLUSIONS

Existing and future wide-field optical surveys will probe new regimes in the time-domain, and find new astrophysical classes, while enabling a deeper understanding of presently rare classes. In addition, correlating these sources with alerts from gravitational wave, high-energy particle, and neutrino observatories will enable new breakthroughs in multimessenger astrophysics. However, the alert-rate from these surveys far outstrips the follow-up capacity of the entire astronomical community combined. Realising the promise of these wide-field surveys requires that we characterize sources from sparse early-time data, in order to select the most interesting objects for more detailed analysis.

Standard supervised learning approaches are unable to deal with the scope for new discovery offered by the wealth of data from upcoming surveys. Anomaly detection algorithms enable an opportunity to automatically flag unusual and interesting transients for further follow-up. We have detailed the development of a real-time anomaly detection framework for identifying unusual transients in large-scale transient surveys. Our deep recurrent neural network is well-suited for the millions of alerts that ongoing and upcoming wide-field surveys such as ZTF and LSST will produce. It allows us to

identify anomalies as a function of time, and we have demonstrated its performances on both ZTF-like simulations and real ZTF light curves from the public MSIP survey.

ACKNOWLEDGEMENTS

We would like to thank the organisers of the 2019 Kavli Summer Program in Astrophysics hosted at the University of California, Santa Cruz, without which this collaboration and work would not have been possible. The program was funded by the Kavli Foundation, The National Science Foundation, UC Santa Cruz, and the Simons Foundation.

REFERENCES

- Abadi M., et al., 2016, in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI’16. USENIX Association, Berkeley, CA, USA, pp 265–283
- Aguirre C., Pichara K., Becker I., 2018, preprint, ([arXiv:1810.09440](https://arxiv.org/abs/1810.09440))
- Bahdanau D., Cho K., Bengio Y., 2014, preprint, ([arXiv:1409.0473](https://arxiv.org/abs/1409.0473))
- Baron D., Poznanski D., 2017, *MNRAS*, **465**, 4530
- Bellm E., 2014, in Wozniak P. R., Graham M. J., Mahabal A. A., Seaman R., eds, The Third Hot-wiring the Transient Universe Workshop. pp 27–33 ([arXiv:1410.8185](https://arxiv.org/abs/1410.8185))
- Bond I. A., et al., 2001, *MNRAS*, **327**, 868

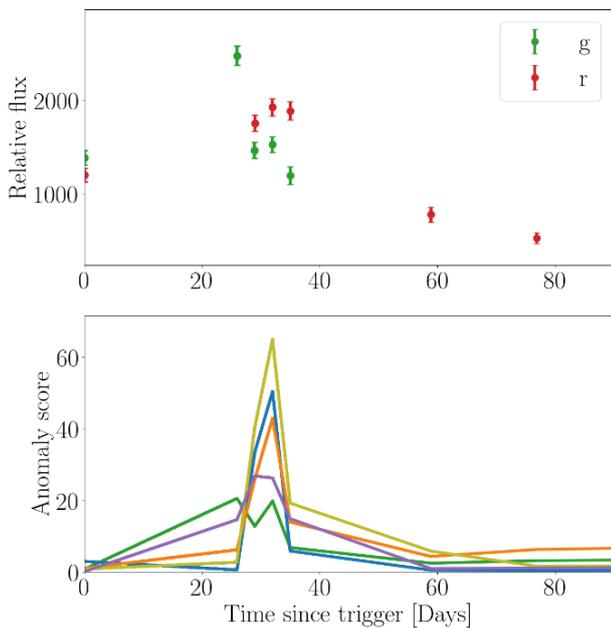


Figure 7. ZTF18acahuph (SN2018hfu), a spectroscopically confirmed SNIa-pec discovered by ZTF is plotted in the top panel. Five generative models, SNIa (green), SNIi (blue), SNIbc (orange), SLSN-I (yellow), AGN (purple), are fit to the light curve, and the anomaly scores as a function of time of each are plotted in the bottom panel. The transient is anomalous to all models, correctly identifying that it is an anomalous object, and is least anomalous to SNIa, indicating that it is most similar to the SNIa class.

Boone K., 2019, arXiv e-prints, p. [arXiv:1907.04690](https://arxiv.org/abs/1907.04690)
 Carrasco-Davis R., et al., 2018, arXiv e-prints, p. [arXiv:1807.03869](https://arxiv.org/abs/1807.03869)
 Charnock T., Moss A., 2017, *ApJ*, **837**, L28
 Che Z., Purushotham S., Cho K., Sontag D., Liu Y., 2018, *Scientific Reports*, **8**, 6085
 Chollet F., et al., 2015, Keras, <https://keras.io>
 Fernández S., Graves A., Schmidhuber J., 2007, in Proceedings of the 17th International Conference on Artificial Neural Networks. ICANN'07. Springer-Verlag, Berlin, Heidelberg, pp 220–229
 Fitzpatrick E. L., 1999, *PASP*, **111**, 63
 Foreman-Mackey D., Agol E., Angus R., Ambikasaran S., 2017, ArXiv
 Graham M., Zwicky Transient Facility (ZTF) Project Team 2018, in American Astronomical Society Meeting Abstracts #231. p. 354.16
 Guillochon J., Nicholl M., Villar V. A., Mockler B., Narayan G., Mandel K. S., Berger E., Williams P. K. G., 2018, *apjs*, **236**, 6
 Guy J., et al., 2010, *aap*, **523**, A7
 Hannun A., et al., 2014, preprint, ([arXiv:1412.5567](https://arxiv.org/abs/1412.5567))
 Hinnert T. A., Tat K., Thorp R., 2018, *AJ*, **156**, 7
 Hochreiter S., Schmidhuber J., 1997, *Neural Comput.*, **9**, 1735
 Hoyle B., Rau M. M., Paech K., Bonnett C., Seitz S., Weller J., 2015, *MNRAS*, **452**, 4183
 Ioffe S., Szegedy C., 2015, in Proceedings of the 32nd International Conference on Machine Learning - Volume 37. ICML'15. JMLR.org, pp 448–456
 Ivezic Ž., et al., 2019, *ApJ*, **873**, 111

Jha S. W., 2017, Type Iax Supernovae. p. 375, [doi:10.1007/978-3-319-21846-5_42](https://doi.org/10.1007/978-3-319-21846-5_42)
 Kasen D., Bildsten L., 2010, *apj*, **717**, 245
 Kasen D., Woosley S. E., Heger A., 2011, *apj*, **734**, 102
 Kasen D., Metzger B., Barnes J., Quataert E., Ramirez-Ruiz E., 2017, *at*, **551**, 80
 Kasliwal M. M., et al., 2012, *apj*, **755**, 161
 Kessler R., et al., 2009, *PASP*, **121**, 1028
 Kessler R., et al., 2010, *pasj*, **122**, 1415
 Kessler R., et al., 2013, *apj*, **764**, 48
 Kessler R., et al., 2019, arXiv e-prints, p. [arXiv:1903.11756](https://arxiv.org/abs/1903.11756)
 Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., USA, pp 1097–1105
 Li X., Wu X., 2015, in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. ICASSP'15. IEEE
 Lochner M., McEwen J. D., Peiris H. V., Lahav O., Winter M. K., 2016, *ApJS*, **225**, 31
 Malz A., et al., 2018, preprint, ([arXiv:1809.11145](https://arxiv.org/abs/1809.11145))
 McCulloch W. S., Pitts W., 1943, *The bulletin of mathematical biophysics*, **5**, 115
 Mockler B., Guillochon J., Ramirez-Ruiz E., 2019, *apj*, **872**, 151
 Möller A., de Boissière T., 2019, arXiv e-prints, p. [arXiv:1901.06384](https://arxiv.org/abs/1901.06384)
 Moss A., 2018, preprint, ([arXiv:1810.06441](https://arxiv.org/abs/1810.06441))
 Muthukrishna D., Narayan G., Mandel K., 2018, *PASP*, Special Edition, in preparation
 Muthukrishna D., Narayan G., Mandel K., 2019a, in preparation
 Muthukrishna D., Parkinson D., Tucker B., 2019b, arXiv e-prints, p. [arXiv:1903.02557](https://arxiv.org/abs/1903.02557)
 Nair V., Hinton G. E., 2010, in Proceedings of the 27th International Conference on Machine Learning. ICML'10. Omnipress, USA, pp 807–814
 Narayan G., et al., 2019, in preparation
 Naul B., Bloom J. S., Pérez F., van der Walt S., 2018, *Nature Astronomy*, **2**, 151
 Neil D., Pfeiffer M., Liu S.-C., 2016, in Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16. Curran Associates Inc., USA, pp 3889–3897
 Nicholl M., Guillochon J., Berger E., 2017, *apj*, **850**, 55
 Pasquet J., Pasquet J., Chaumont M., Fouchez D., 2019, arXiv e-prints, p. [arXiv:1901.01298](https://arxiv.org/abs/1901.01298)
 Pierel J. D. R., et al., 2018, *pasj*, **130**, 114504
 Rau A., et al., 2009, *PASP*, **121**, 1334
 Razavian A. S., Azizpour H., Sullivan J., Carlsson S., 2014, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW '14. IEEE Computer Society, Washington, DC, USA, pp 512–519, [doi:10.1109/CVPRW.2014.131](https://doi.org/10.1109/CVPRW.2014.131)
 Rees M. J., 1988, *at*, **333**, 523
 Shen K. J., Kasen D., Weinberg N. N., Bildsten L., Scannapieco E., 2010, *ApJ*, **715**, 767
 Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, *J. Mach. Learn. Res.*, **15**, 1929
 Sutskever I., Vinyals O., Le Q. V., 2014, in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14. MIT Press, Cambridge, MA,

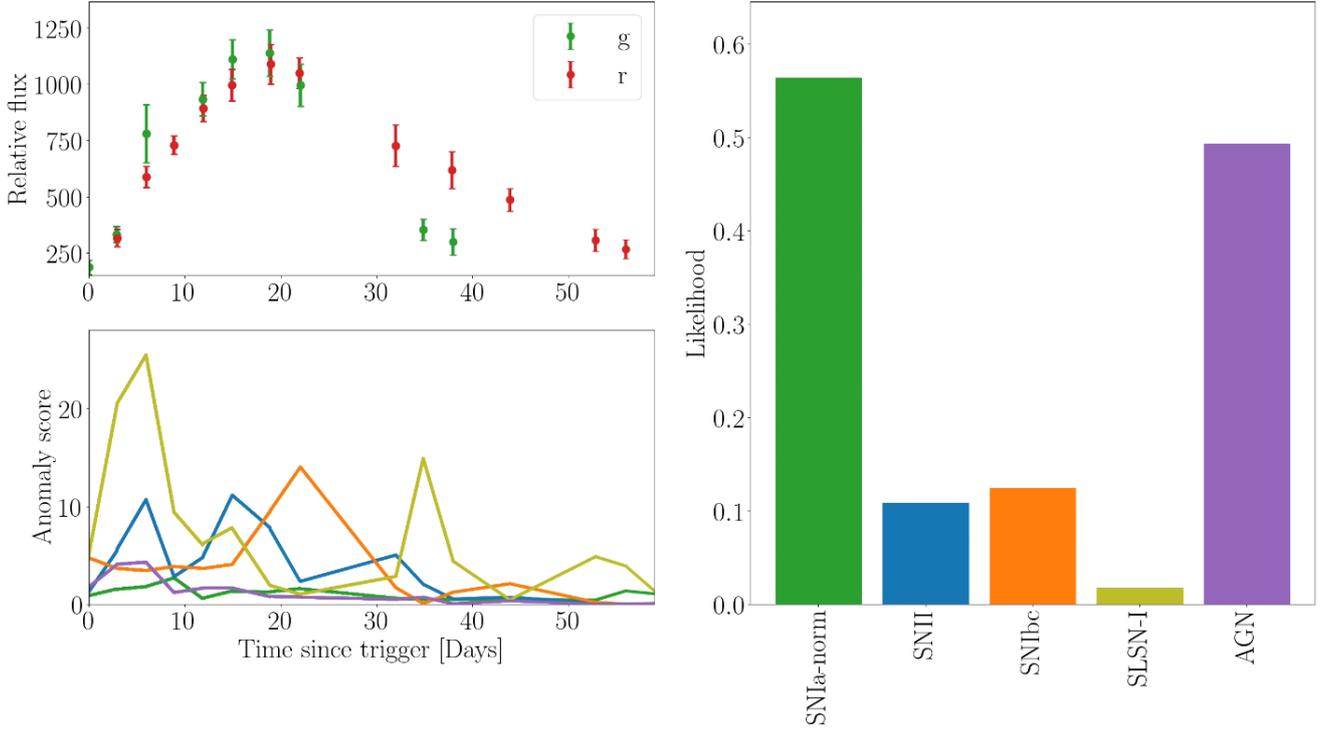


Figure 8. ZTF19aadnmgf, a spectroscopically confirmed normal SNIa discovered by ZTF is plotted in the top-left panel. Five generative models are fit to the light curve, and the anomaly scores as a function of time of each are plotted in the bottom-left panel. The likelihood of each model is plotted on the right, and illustrates that the transient is most likely a SNIa. The high score to the AGN class again indicates that the variability of AGN mean that the DNN trained on AGN can fit nearly any transient (as further illustrated by Figure 6).

USA, pp 3104–3112

Szegedy C., et al., 2015, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1–9, [doi:10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594)

The PLAsTiCC team et al., 2018, preprint, ([arXiv:1810.00001](https://arxiv.org/abs/1810.00001))

Tomaney A. B., Crofts A. P. S., 1996, *AJ*, **112**, 2872

Villar V. A., Berger E., Metzger B. D., Guillochon J., 2017, *apj*, **849**, 70

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.