Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **509,** 1269–1290 (2022) https://doi.org/10.1093/mnras/stab3043
Advance Access publication 2021 October 23

Light-curve fingerprints: an automated approach to the extraction of X-ray variability patterns with feature aggregation – an example application to GRS 1915+105

Jakub K. Orwat-Kapola ,[1]★ Antony J. Bird ,[1] Adam B. Hill ,[1,2] Diego Altamirano[1] and Daniela Huppenkothen[3]

[1] School of Physics and Astronomy, University of Southampton, Southampton, Hampshire SO17 1BJ, UK
[2] HAL24K Labs, Herikerbergweg 292, NL-1101 CT Amsterdam, the Netherlands
[3] SRON Netherlands Institute for Space Research, Sorbonnelaan 3, NL-3584 CA Utrecht, the Netherlands

Accepted 2021 October 8. Received 2021 October 5; in original form 2021 March 12

ABSTRACT

Time series data mining is an important field of research in the era of ‘Big Data’. Next generation astronomical surveys will generate data at unprecedented rates, creating the need for automated methods of data analysis. We propose a method of light-curve characterization that employs a pipeline consisting of a neural network with a long-short term memory variational autoencoder architecture and a Gaussian mixture model. The pipeline performs extraction and aggregation of features from light-curve segments into feature vectors of fixed length that we refer to as light-curve ‘fingerprints’. This representation can be readily used as input of down-stream machine learning algorithms. We demonstrate the proposed method on a data set of Rossi X-ray Timing Explorer observations of the Galactic black hole X-ray binary GRS 1915+105, which was chosen because of its observed complex X-ray variability. We find that the proposed method can generate a representation that characterizes the observations and reflects the presence of distinct classes of GRS 1915+105 X-ray flux variability. We find that this representation can be used to perform efficient classification of light curves. We also present how the representation can be used to quantify the similarity of different light curves, highlighting the problem of the popular classification system of GRS 1915+105 observations, which does not account for intermediate class behaviour.

Key words: methods: data analysis – X-rays: binaries.

1 INTRODUCTION

1.1 Machine learning for astronomical time series

Automated approaches to data analysis are becoming increasingly relevant as we are entering the era of big data in and outside of astronomy. Industrial applications include, for example, analysis of smart utility metre data and city traffic data. The ability to identify appliance energy usage patterns can provide actionable insights to utility providers (Singh & Yassine 2018), as smart metres are being installed in millions of houses. Prediction of the rate of traffic flow and smart route planning can aid in the management of congestion in intelligent transportation systems (Zhu et al. 2019). Within the field of astronomy, future and ongoing surveys, like those conducted by Vera C. Rubin Observatory [previously known as the Large Synoptic Survey Telescope (LSST)] (Ivezic et al. 2019) and Zwicky Transient Facility (ZTF; Bellm 2014), will produce terabytes of data at unprecedented rates, and manual analysis of this volume of data by human experts is impossible. In order to make sense of this data, analysts need methods that can extract descriptive variables from individual data observations (i.e. perform feature engineering). Resulting variables can then be used in machine learning pipelines to compare observations and perform tasks such as classification, outlier detection, or clustering.

Feature engineering often requires domain specific expertise from the analyst, who needs to identify descriptors containing relevant information about the observations in the data set (for example, Richards et al. 2011). Automated feature extraction is an alternative to manual feature engineering, and requires less specific domain knowledge. Automated feature extraction often involves methods that encode data into a more abstract, low-dimensional, latent representation. The resulting latent representation contains compressed information about the input data, and reflects the similarities and differences between observations. Machine learning algorithms can leverage this information to perform a variety of tasks. Several methods of automated feature extraction for light-curve data (time series data describing intensity of an astronomical object) have been studied in the past, for example, Kohonen self-organizing maps (Armstrong et al. 2015) and pattern dictionary learning (Pieringer et al. 2019). Mackenzie, Pichara & Protopapas (2016) extracted light-curve segments and clustered them to find common patterns of variability in variable star candidates, creating a representation compatible with machine learning classifiers. Similarly, Valenzuela & Pichara (2018) used a sliding window method to extract light-curve segments and classified the light curves based on the presence

★ E-mail: j.k.orwat-kapola@soton.ac.uk

of characteristic patterns. Naul et al. (2018) extracted features of phase-folded light curves using an autoencoding recurrent neural network (RNN) and used a random forest classifier on observations of variable sources from All Sky Automated Survey Catalog, Lincoln Near-Earth Asteroid Research survey, and Massive Compact Halo Object Project, achieving classification accuracy of well over 90 per cent.

End-to-end architectures have also been used for a variety of tasks. Those approaches use a single model that implicitly learns a low-dimensional representation of data to perform classification, inference etc. For example, Charnock & Moss (2017) used a bidirectional RNN to process and classify light curves of the supernovae photometric classification challenge data set and achieved impressive results. Mahabal et al. (2017) transformed the light curves of Catalina Real-Time Transient Survey to two-dimensional images and used a convolutional neural network (CNN) to classify them. Shallue & Vanderburg (2018) trained a deep CNN to detect exoplanet transits in folded light curves of the Kepler mission, and were able to discern plausible planet signals from false positives 98.8 per cent of the time. Becker et al. (2020) trained an RNN to classify variable star light curves, achieving results that were competitive with results of a random forest classifier trained on features generated by a library for time series feature extraction feature analysis for time series. They grouped each light curve with a sliding window, reducing the sequence length, which allowed the RNN to process long sequences. More recently, Zhang & Bloom (2021) demonstrated state-of-the-art classification performance using novel Cyclic-Permutation Invariant Neural Networks that are invariant to phase shifts of phase-folded light curves.

In this work, we contribute to the toolbox of automated characterization methods for light-curve data, as we explore the use of neural networks and Gaussian mixture models (GMMs) for descriptive feature generation and clustering. We construct a data representation called light-curve 'fingerprints' that can be used in downstream classification, outlier detection, and clustering tasks. We demonstrate that such methods can be useful in the analysis of the evolution of a particular source of interest.

1.2 GRS 1915+105

In this work, we analyse the complete data set of GRS 1915+105 observations captured by the Proportional Counter Array on-board of the Rossi X-ray Timing Explorer (RXTE/PCA; Bradt, Rothschild & Swank 1993; Glasser, Odell & Seufert 1994) between 1996 and 2011.

GRS 1915+105 is a Galactic black hole X-ray binary system discovered in 1992 (Castro-Tirado et al. 1994), which shows an extraordinary complexity of X-ray flux variability. It was the only known source to exhibit such intricate patterns of behaviour, until the discovery of black hole candidate IGR J17091-3624 (Kuulkers et al. 2003; Capitanio et al. 2006), which shares some of the same characteristics (Altamirano et al. 2011).

In an attempt to demonstrate that the complex behaviour of GRS 1915+105 is controlled by just a few simple variables, Belloni et al. (2000) constructed a classification system, which assigned observations of the source to one of 12 classes. Classification was based on the presence of qualitative patterns in light curves and colour–colour diagrams of source observations. Work that followed had expanded the classification system to the total of 14 classes (Klein-Wolt et al. 2002; Hannikainen et al. 2003, 2005). This classification system is hereafter referred to as the 'Belloni et al. system'. Fig. 1 shows an example of GRS 1915+105 light curve for each one of the 14 classes. Some classes show steady flux without any structured variability, other show periodic flares, dips or different types of periodic and aperiodic variability. There are both interclass and intraclass variations in the amplitude of flux variability.

Highly complex patterns of light-curve variability of GRS 1915+105 are commonly interpreted as being caused by a partial or complete disappearance of an observable innermost region of the accretion disc. Disappearance of the disc, in turn, is caused by thermal-viscous instability of the inner region of the disc (Belloni et al. 1997a,b). X-ray variability patterns corresponding to the 14 classes of source behaviour can repeat almost identically months and years apart. This suggests that the instability of the disc triggers a very specific and reproducible response (Belloni 2001).

GRS 1915+105 was the first discovered stellar mass source producing highly relativistic jets (Mirabel & Rodriguezt 1994). In this regard, it sparked great research interest, as it offered the possibility of studying coupled inflow–outflow processes of an accreting black hole, which unlike more massive active galactic nuclei, evolves in observable time-scales (Fender & Belloni 2004). Plasma ejections of GRS 1915+105 in the form of jets have also been associated with the instability of the accretion disc (Belloni et al. 2000; Nayakshin, Rappaport & Melia 2000; Fender & Belloni 2004), which supports the notion of disc–jet coupling.

Furthermore, Naik et al. (2002) found that periods when the innermost region of the accretion disc is not observable (which are associated with variability class χ), are preferentially followed by classes showing periodic bursts, i.e. classes ρ and α. Therefore, following the changes of variability classes can improve our understanding of the evolution of the source over longer time-scales, and it is an important method of probing the accretion dynamics, as pointed out by Huppenkothen et al. (2017).

Huppenkothen et al. (2017) conducted the first study of the whole set of GRS 1915+105 observations from RXTE/PCA using machine learning. They characterized the observations of the source and classified them according to the Belloni et al. system, using machine learning classification methods. They used features derived from power spectra, time series features extracted with an autoregressive model, and hardness ratio features derived from energy spectra.

1.3 Summary of this work

In this work, we perform classification of GRS 1915+105 observations using exclusively time series features derived from light-curve data in an unsupervised manner, using a neural network. We choose not to use energy and power spectral features, found in the works of Belloni et al. (2000) and Huppenkothen et al. (2017), hence making our method more generalizable to other data sets, sources, and energy bands. In principle, any type of univariate time series data can be analysed using this method.

In order to perform machine learning on light-curve data, they need to be represented with vectors of constant length. It is a common approach to segment light curves to create time series of constant length prior to feature extraction. However, it is not obvious how to aggregate features of a light curve from a variable number of segments. We propose the 'fingerprint' representation as a method of aggregating segment-wise time series features into a constant-length vector that describes a whole light curve of any length, in a manner that is interpretable by machine learning algorithms.

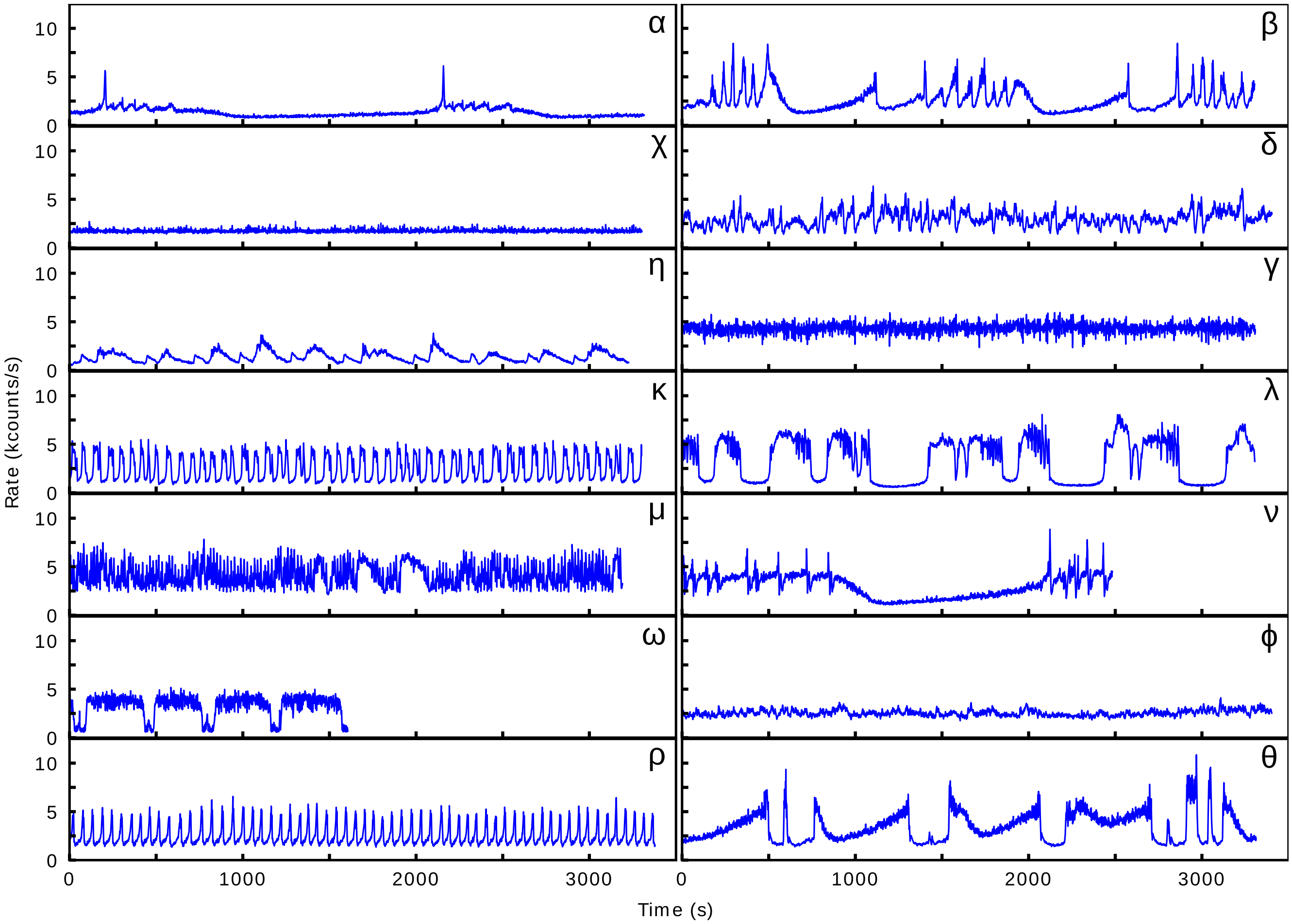

Figure 1. Examples of GRS 1915+105 X-ray light curves. Shown light curves have been classified according to the Belloni et al. system. Classifications are shown in the upper-right corner of each sub-figure. We use the curated set of classifications published by Huppenkothen et al. (2017). Light curves have 1 s cadence.

We train a neural network that performs dimensionality reduction of light-curve segments, producing time series feature vectors of individual segments. We then perform clustering of features of individual segments and use the number of segments assigned to each cluster to construct the 'fingerprint' representation, taking advantage of the fact that segments showing similar patterns of light-curve variability tend to cluster together. We show that the 'fingerprint' representation can be used to quantify the similarity of light curves and to perform machine learning classification.

This paper contains the following parts: Section 2 describes the data preparation process, which starts with RXTE/PCA observations and produces a data set of light-curve segments, a suitable input for a neural network. Section 3 provides details about the neural network and describes the process of dimensionality reduction of the data set, which results in the encoded data representation (time series features). Section 4 describes the process of cluster analysis of the encoded representation. Section 4.1 describes how GMMs are used to identify the set of light-curve patterns from the encoded data representation. Section 4.2 shows how the set of light-curve patterns is used to construct the observation 'fingerprints'. Section 4.3 shows how 'fingerprints' can be used to assign light curves to classes of the Belloni et al. system, and Section 4.4 demonstrates how the 'fingerprints' can be leveraged to refine the classification system in a data-driven way. Section 5 summarizes the main results, discusses limitations of the presented approach, and lists some ideas for further research.

2 DATA PREPARATION

We retrieve all available RXTE/PCA observations of GRS 1915+105 in Standard-1 format (0.125 s resolution light curves that combine all energy channels over the range of 2–60 keV).[1] Extraction is limited to the most reliable counting array number 2 (PCU2). This yields 1776 light curves, which are subsequently re-binned. We generate two data sets: one where binning is performed at 1 s resolution, and another where binning is performed at 4 s resolution. Two data sets are generated because we fix the input size of the neural network to 128 data points due to graphical processing unit (GPU) hardware constraints (see Appendix B for more detail) therefore the time resolution of the light curves is the main parameter that influences the amount and type of information that is provided to the network. On the one hand, short time resolution data can resolve fast variability structures of the X-ray source, but it cannot capture longer variability structures in a light-curve segment of fixed size. On the other hand,

[1] https://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl

Table 1. Parameters of the two data sets of light-curve segment.

Parameter	4 s data set	1 s data set
Cadence (s)	4	1
Segment length (s)	512	128
Stride length (s)	8	10
# Segments	468 202	474 471

longer time resolution data can capture more context within a light-curve segment, but any fast variability structures are smoothed or lost. Hence, we choose to train two separate models on the two data sets and compare them in order to explore the effect of changing data resolution.

Similar to Huppenkothen et al. (2017), we perform light-curve segmentation in order to produce a data set of segments of standard length. Only the good time intervals from each observation are segmented, and the interruption periods of missing data are skipped. We choose the segment size of 128 data points, resulting in segment length of 512 s for data with 4 s resolution. A moving window segmentation is performed with a stride of two data points (8 s), yielding a set of 468 202 segments derived from 1738 observations that satisfy the segmentation criteria. This data set of light-curve segments is hereafter referred to as '4s data set'. The same set of 1738 observations, binned to 1 s resolution, is segmented with a stride length of 10 data points, yielding 474 471 segments. This data set of light-curve segments is hereafter referred to as '1 s data set'. Stride size is adjusted in order to make the number of segments as close as possible to the 4 s data set. Table 1 lists parameters of the two data sets.

The time duration captured by the segments is not sufficient to contain the longest cycles of flux variability produced by the source. For example, some observations of class α show intervals of quiescence that last ~1000 s, and are interlaced by periods of flaring that last ~500 s. However, the main goal of our study is not to classify individual light-curve segments, but rather to construct a new, data-driven system of segment templates, which create classifiable observation signatures when grouped together with other segments from the same light curve. See Section 4.3 for an example of a classification experiment that illustrates the use of observation signatures (fingerprints).

A small stride size is selected in order to maximize the number of light-curve segments available for neural network training. It also ensures that the model is exposed to the full range of phase shift of light-curve patterns (Huppenkothen et al. 2017).

Light-curve segments are independently standardized; count rate values of each segment are mean centred and scaled to units of standard deviation. Segments are standardized in order to decouple their shape and intensity information. Many of the patterns observed in the variability of GRS 1915+105 repeat at various mean count rate levels. Therefore, standardization of the segments is likely to cause segments showing similar shape patterns to align in the latent feature space extracted from the data. We also allow for the possibility that some of the shapes could be shared by several classes of variability, as described by Belloni et al. (2000), but at different intensities, and standardization can make it easier to draw links between those cases.

The resulting data set of light-curve segments, together with corresponding count rate uncertainty values (needed to calculate the reconstruction error, see equation 1), is used to train a neural network. The process is described in Section 3.

Original count rate levels of the light-curve segments are also important in the data analysis, so intensity information is added to the final feature set in the form of four descriptive statistics; mean, standard deviation, skewness, and kurtosis, which are calculated from the distribution of count rate of each segment. These statistics make up one of the two sets of features used in cluster analysis in Section 4, and this set of four intensity features is hereafter referred to as 'intensity features of light-curve segments' (IFoS).

3 FEATURE EXTRACTION WITH A NEURAL NETWORK

In order to represent the light-curve data using a small number of variables, and hence allow us to analyse that data space in a relatively small number of dimensions, we perform dimensionality reduction using a neural network. This process aims to extract a small set of 'hidden' variables, which encode information about the structured variability of the light curves. Various methods of modelling of time series data have been employed in numerous fields of research (Längkvist, Karlsson & Loutfi 2014; Hyndman, Wang & Laptev 2015; Benkabou, Benabdeslem & Canitia 2018; Ismail Fawaz et al. 2019).

In order to address the problem of dimensionality reduction of light-curve data, we propose a variational autoencoder (VAE) with long-short term memory (LSTM) cells within its encoder and decoder blocks. VAE is a type of probabilistic neural network model first proposed by Kingma & Welling (2014). As mentioned, it uses two blocks of neurons; an encoder, which maps input data on to a small set of distributions (commonly referred to as continuous latent variables), and a decoder, which samples from those distributions and maps them back to the input data space, hence reconstructing observations of input data. VAE is therefore trained to effectively perform compression and decompression of input data. The compression process requires the construction of a meaningful latent space of considerably smaller dimensionality than the input. Resulting latent variables are the compressed representation of data and can be leveraged in the data analysis process.

Autoencoders have been used in the past for the purpose of feature extraction from astronomical data (for example, Naul et al. 2018). Here, we use a variational variant of an autoencoder because it introduces a Gaussian prior that applies a regularization effect on the produced latent space. Non-regularized autoencoders can map similar inputs to very different latent variables (Hinton 2013) therefore regularization of the latent space is required for the downstream clustering and merging of clusters discussed in Section 4.

LSTM cells found in the encoder and decoder blocks of the proposed architecture are a type of RNN, first proposed by Hochreiter & Urgen Schmidhuber (1997). RNNs learn from sequential data, and their use has been researched extensively for the processing of text, handwriting, speech, and sound (and references therein Yu et al. 2019).

Cells of an RNN process sequential input one data point at a time. At every iteration, the state of the cell is fed as input of the next iteration, which allows the network to learn from consecutive points of the sequence and extract temporal patterns. RNNs are able to make predictions based on the immediate context of the processed data point, but they tend to quickly forget information that is not frequently reinforced, which means that they struggle to capture long term patterns. LSTM cells address this issue through the introduction of so-called 'gates' which consist of non-linear functions that control the state of the LSTM cell and protect the relevant information over long time-scales.

3.1 Training of the neural network

Both data sets are subdivided into training, validation, and testing subsets. In order to ensure that the subsets are independent, segments derived from the same observation are only included in one of the subsets. In order to ensure completeness of the subsets, observations that have Belloni et al. system classifications available (Huppenkothen et al. 2017), are assigned to the subsets in a random, stratified fashion. At least one observation of each class is assigned to each subset, and the remaining observations are assigned to training, validation, and testing subsets according to the split ratio of 7/1/2. Since only two observations of η class are available at this stage, both observations are assigned to the training subset. Observations without Belloni et al. system classifications are randomly distributed between the three subsets, while accounting for the fact that observations contain variable number of count rate data points. The resulting training sets contain ~70 per cent of total data points, validation sets contain ~10 per cent of data points, and testing sets contain ~20 per cent of data points.

Two LSTM–VAE neural networks with identical architecture are trained to compress light-curve segments of 128 data points into 20 continuous latent variables, one network for each data set. Data from the training set is used to adjust the parameters of the networks, and the validation set is used to measure whether the adjustment improved the networks' ability to process previously unseen examples of data. Testing set is kept aside until all training and fine-tuning of models is finished. We perform dimensionality reduction on both data sets of standardized light-curve segments, which are generated using the method described in Section 2.

The networks are built using KERAS, an open-source neural network library (Chollet 2015) with TENSORFLOW backend, and PYTHON 3 programming language.[2] More details about network architecture can be found in Appendix A.

Output of the network consists of the reconstruction of an input light-curve segment. Performance of the network is quantified using a loss function, which depends on the reconstruction error and a regularization term. The loss function is a sum of a mean square error weighted by the data uncertainty (equation 1) and Kullback–Leibler divergence (equation 2). The weighted mean square error is defined as

$$\mathrm{WMSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \hat{x_i}}{\sigma_i}\right)^2, \qquad (1)$$

where N is the number of processed data points, x_i is the value of a data point i in the input light-curve segment, $\hat{x_i}$ is the value of data point i in the reconstructed sequence, and σ_i is the uncertainty of the value of i in the input light-curve sequence. Uncertainty values are scaled with the same value of standard deviation as the count rate values of light-curve segments.

Kullback–Leibler divergence is a regularization term, which prevents the distribution of the latent variables from substantially deviating from the standard normal distribution. This term is used to ensure that the latent space is continuous and meaningful. Kullback–Leibler divergence is defined as

$$D_{\mathrm{KL}} = -\frac{1}{2Z}\sum_{j=1}^{Z}\left(\log\left(\hat{\sigma_j}^2\right) - \hat{\mu_j}^2 - \hat{\sigma_j}^2 + 1\right), \qquad (2)$$

where Z is the number of latent variables, $\hat{\sigma_j}^2$ is the variance of latent variable j, and $\hat{\mu_j}$ is the mean of latent variable j.

The networks are trained to minimize the value of the loss function. Details about the training process of the neural networks can be found in Appendix B. Examples of light-curve segments from 1s data set, together with their LSTM–VAE reconstructions are shown in Fig. 2. Examples for segments from 4 s data set are shown in Appendix C.

3.2 Light-curve feature extraction

The network infers mean and variance of the 20 continuous latent variables from the data (i.e. outputs of `Latent_mean` and `Latent_log_variance` layers of the network correspond to the mean and variance parameters, see Appendix A for more details). The latent variables are a compressed representation of the network's input. For the purpose of further analysis, we do not sample from latent distributions, but take the means of latent distributions, which are representative of the position of each light-curve segment within the latent space. The resulting set of latent variables is hereafter referred to as 'shape features of light-curve segments' (SFoS). Hence, the shape information of each light-curve segment is represented by 20 SFoS values.

In order to assess how well SFoS describe the shape of light-curve segments, we perform reconstruction of several segments. Figs 2 and C1 show reconstructions of light-curve segments of each class of the Belloni et al. system. This set of segments demonstrates how the LSTM–VAE responds to a range of different light-curve patterns found in the data sets. The model is able to reproduce the gross features of each segment, but it often does not account for fast count rate changes, which results in reconstructions that are significantly smoother than the input segments. This means that SFoS likely do not account for differences between segments lacking structured variability, where the major difference lies in the root-mean-square (RMS) deviation from the mean. Reconstructions of those segments would differ only in terms of the shape of random noise. For example, see segments of class ϕ and χ in Fig. C1.

In order to remedy this limitation, IFoS (containing information about count rate mean, standard deviation, skewness, and kurtosis of each segment) are used in the cluster analysis stage (Section 4) together with SFoS. Segments with indistinguishable shapes and dissimilar RMS can be distinguished based on their standard deviation values.

Fig. 3 shows the distribution of IFoS values for segments of ϕ, χ, and γ classes from the 1 s data set. As expected, segments of class γ generally have larger mean and standard deviation values than the other two classes, which indicates that IFoS would allow for segments of class γ to be distinguished from classes ϕ and χ, even in segments where the characteristic 'dip' of class γ is not observed.

Furthermore, projections of SFoS and IFoS in Fig. 4 show that IFoS could be used to distinguish classes ϕ, χ, and γ much more reliably than SFoS alone. The projection was produced using uniform manifold approximation and projection (UMAP; McInnes, Healy & Melville 2018) algorithm, which aims to preserve the global structure of the high-dimensional data. In the projection of SFoS, data classified as ϕ, χ, and γ occupied the same region of UMAP space (in the central cluster of the top sub-figure), indicating that those classes are mostly indistinguishable in the SFoS space. Data classified as ρ tends to occupy separate regions of the SFoS space. Class ρ is included in the projection to demonstrate that data associated with

[2] Relevant code for data pre-processing and model training will be available at https://github.com/jorwatkapola/autoencoders-GRS-1915.

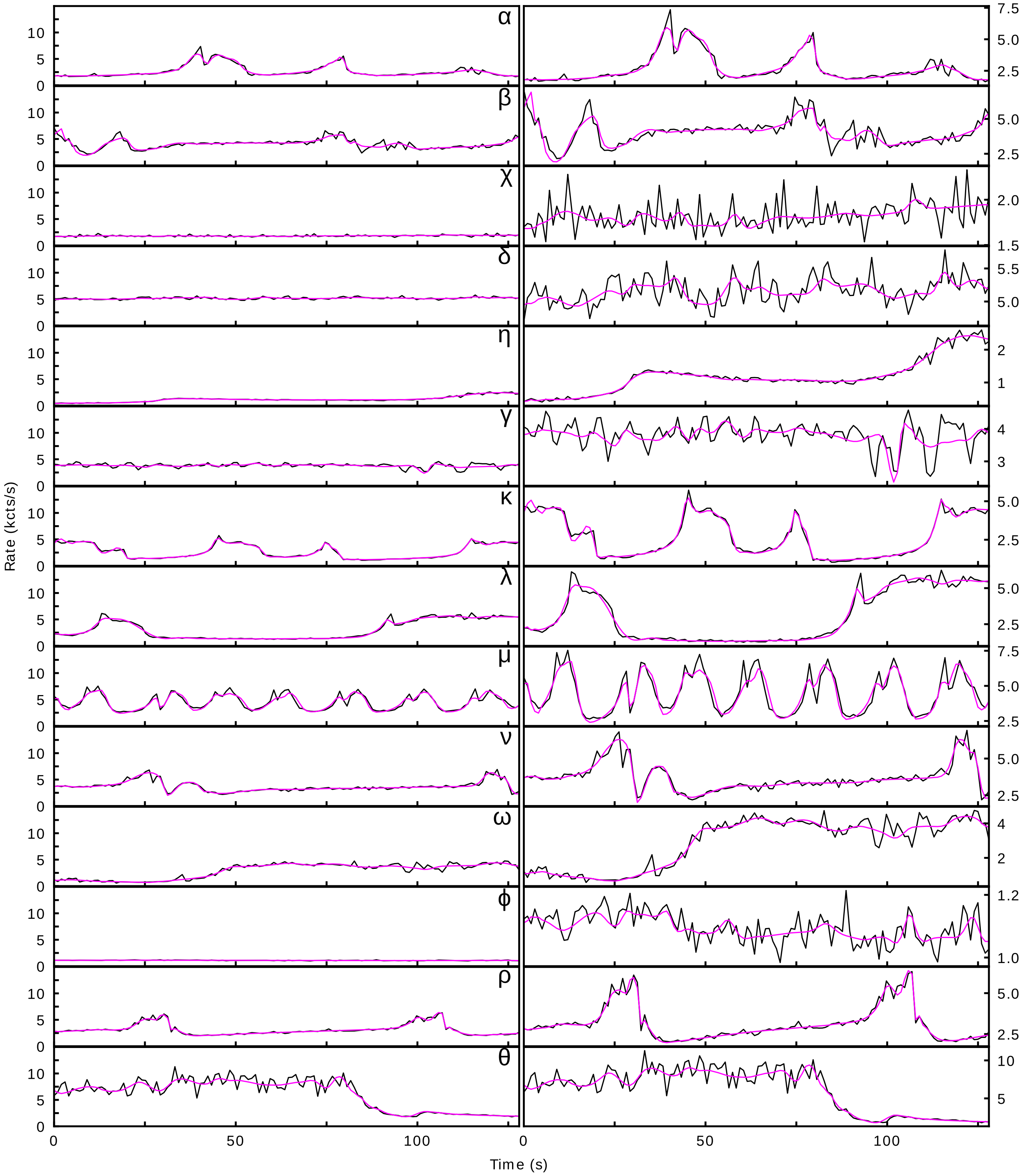

Figure 2. Examples of light-curve segments from 1 s data set. Segment reconstruction output of the LSTM–VAE is shown in magenta. Segments originate from observations that had been classified according to the Belloni et al. system. We use the curated set of classifications published by Huppenkothen et al. (2017). Both columns shows the same segments, but in the right column the range of the vertical axis is dynamic. All segments come from the testing subset of 1 s data set, with exception of class η, which was included only in the training subset.

characteristic light-curve shapes tends to be distinguishable in terms of its position in the SFoS space. IFoS projection uses the same subset of light-curve segment data. Classes still show significant overlap in the IFoS projection, but intensity features can clearly provide meaningful information about the differences between classes ϕ, χ, and γ.

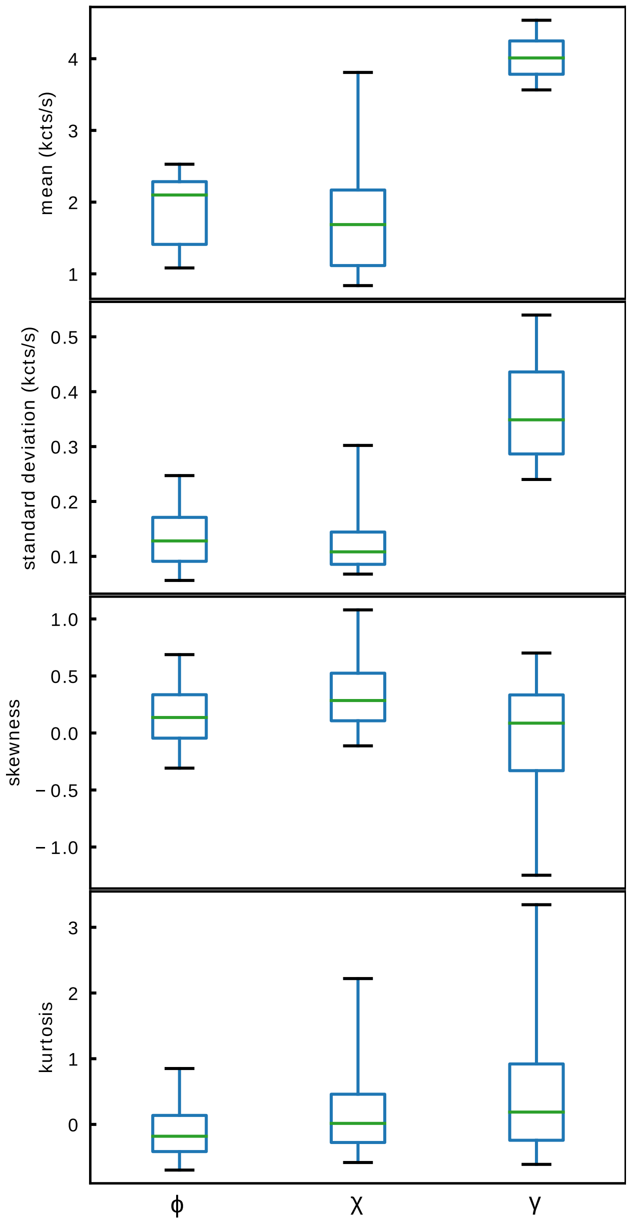

Figure 3. Distributions of the IFoS values from segments assigned to classes ϕ, χ, and γ of the Belloni et al. system. Classifications come from the set curated by Huppenkothen et al. (2017). Four IFoS are the mean, standard deviation, skewness, and kurtosis of count rate values of each segment. Box extends from the first to third quartile, and the green line shows the median. Whiskers extend from the box to the 5th and 95th percentiles.

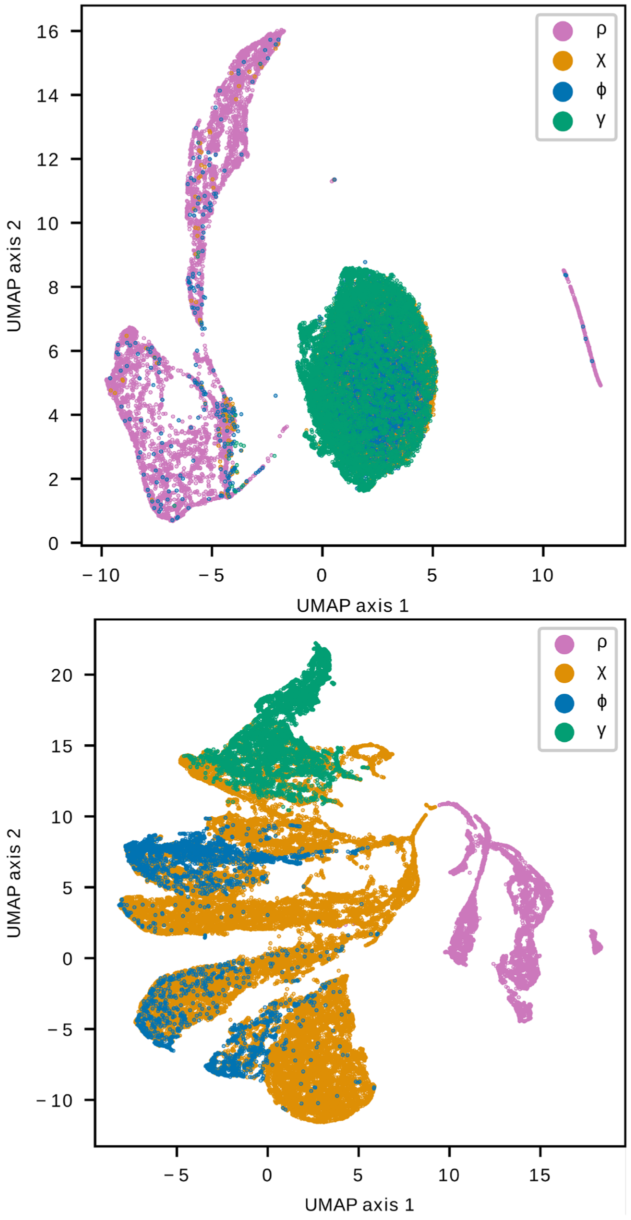

Figure 4. UMAP projection of SFoS (top) and IFoS (bottom) of light-curve segments from the 1 s data set that have Belloni at al. system classifications of ϕ, χ, γ or ,ρ. Each point represents a light-curve segment, and colour-coding shows their classification according to the Belloni et al. system.

4 CLUSTER ANALYSIS OF GENERATED FEATURES

4.1 Identifying the set of light-curve patterns

Sections 2 and 3 describe the process of feature engineering of four IFoS and 20 SFoS, which encode the shape and intensity information about light-curve segments. This set of 24 features is hereafter collectively referred to as ‘shape and intensity features of light-curve segments’ (SIFoS). We generate two separate sets of SIFoS, one for the 1 s data set, and one for the 4 s data set.

In order to find the exhaustive set of light-curve pattern templates that have been produced by GRS 1915+105, we perform clustering of the data in the 24-dimensional space of SIFoS. Clustering is performed with an implementation of GMM included in the machine learning library SCIKIT-LEARN (Pedregosa et al. 2011). The algorithm approximates the probability distribution of the data using a set of multidimensional Gaussian components (see Appendix D for more details).

We explored the possibility of using density-based clustering algorithms DBSCAN (Ester et al. 1996) and OPTICS (Ankerst et al. 1999), and we find that it is difficult to fine-tune the hyper-parameters of those algorithms in a way that would not lead to a large proportion of the data being rejected as noise. We find that GMM offers a much more straightforward method of assigning the data to clusters. Therefore, we focus on the discussion of the application of GMM in the remainder of this text.

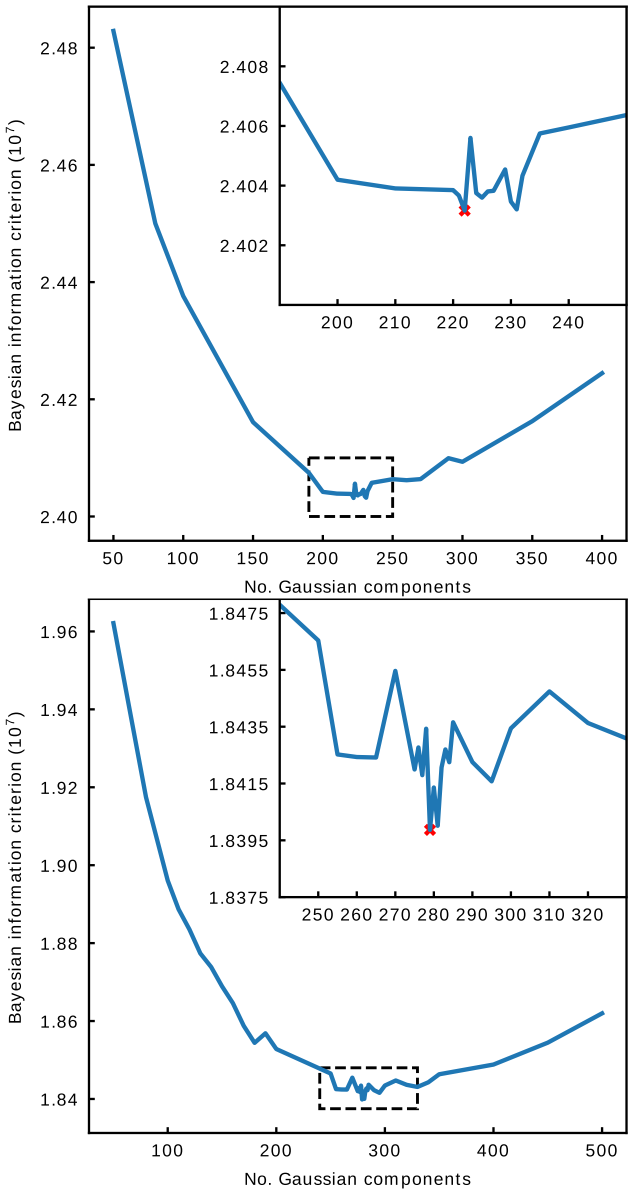

Figure 5. Bayesian information criterion of GMM as a function of the number of Gaussian components. We performed grid searches for the 1 s data set (top) and the 4 s data set (bottom). Figure insets are zoomed into the regions indicated by the black-dotted line boxes, which contain the global minima. Minima were found at 222 Gaussian components for 1s data set and at 279 Gaussian components for 4s data set. Minima are marked by the red crosses.

We choose to use GMM to approximate the shape of latent manifold in the SIFoS space, with the intention of merging of the Gaussian components that show significant overlap. This way we use combinations of Gaussian components to account for the presence of any extended, curved, non-Gaussian structures in the latent space. We are assuming that those structures correspond to particular light-curve patterns, and for each observation of the source, the relative amount of time the source spends showing those patterns allows us to determine the class of that observation.

We perform a hyper-parameter grid search to find the optimal number of Gaussian components for the GMMs in SIFoS space. Fig. 5 shows values of the Bayesian information criterion (BIC) of those models as a function of the number of Gaussian components. BIC is a criterion commonly used to select the best model from a set of models fit to the same data set. The number of free parameters is one of the terms of BIC, so it penalizes overly complex models. Model which produces the minimum BIC achieves the compromise between complexity and likelihood of the data (see equation 3):

$$\mathrm{BIC} = k \cdot \log(N) - 2 \cdot \mathcal{L}, \tag{3}$$

where k is the number of free parameters of the model, N is the number of samples of data, and $\mathcal{L}$ is the log-likelihood of the data.

Grid searches indicate that the global BIC minimum for 1 s data set is produced by a models with 222 Gaussian components, and for 4 s data set with 279 components. We accept those numbers as the optimal numbers of components for GMM, however, stochastic nature of the algorithm could cause the numbers to change slightly if the grid-search was to be repeated. The set of clusters resulting from the assignment of light-curve segment data points to one of the Gaussian components of GMM is hereafter referred to as ‘Gaussian clusters’.

Light-curve segments showing similar type of variability patterns and count rate distribution are expected to produce similar values of SFoS and IFoS. Consequently, segments showing similar patterns are separated by smaller distances within the SIFoS feature space. Therefore, it is expected that Gaussian clusters contain homogeneous subsets of light-curve segments, which are more similar to each other than to segments found in other Gaussian clusters.

Qualitative inspection of populations of the 222 Gaussian clusters of the 1 s data set revealed that it is the case indeed. A total of 42 Gaussian clusters contain recurrent flares, characteristic of light-curve classes ρ and ν (see Fig. 2 for examples). Table 2 shows a breakdown of the population of the clusters in terms of the classes of Belloni et al. system, and it shows that the Gaussian clusters containing recurrent flares are indeed dominated by classes ρ and ν. A total of 69 Gaussian clusters contain segments with low RMS and no obvious patterns of variability, and their population consist mostly of classes χ and ϕ. Further, 15 Gaussian clusters contain segments showing similar type of behaviour, but at higher average count rates, and the population of those clusters consist mostly of classes χ, γ, and θ. Virtually every one of the remaining Gaussian clusters has some characteristic pattern; irregular flares, dips, negative or positive gradient, flaring followed by quiescence and vice versa. Few Gaussian clusters contain segments whose common patterns of variability are not immediately apparent upon visual inspection of a small random sample of segments, which can indicate that the number of Gaussian components of the GMM is too small.

In order to find the minimal, exhaustive set of light-curve variability patterns, segments showing the same characteristic patterns should arguably all belong to one cluster. Gaussian clusters produced with the 222 and 279 component GMMs contain apparent degeneracies. The presence of very similar Gaussian clusters is caused by the limitation of the GMM, which approximates the probability density of the data set using multivariate Gaussian components. A single component cannot spread over a curved data manifold; only multiple components can approximate the curvature, as a series of locally flat sections. Light-curve segments that show a similar type of variability pattern can vary in more than one SIFoS due to their non-linear interaction within the neural network model. Therefore, segments which, for example, vary in the frequency and amplitude of a similar type of pattern, can follow a curved manifold, and hence end up in separate Gaussian clusters. We address that in Section 4.3.

Fig. 6 shows a two-dimensional UMAP projection of mean positions of GMM Gaussian components that are fit to the 1 s data

Table 2. Percentage distribution of light-curve segments across the classes of Belloni et al. system for each type of Gaussian clusters identified in Fig. 6. Classes showing similar patterns tend to be clustered together, which is an indication of the meaningfulness of the SIFoS feature space.

	α	β	χ	δ	η	γ	κ	λ	μ	ν	ω	ϕ	ρ	θ
Recurrent flare	1.2	1.0	0.0	0.0	0.0	0.0	0.9	0.0	0.1	5.9	0.0	0.0	90.9	0.0
Mid-random	0.1	2.4	46.9	0.8	0.0	27.5	0.3	0.1	0.0	0.9	3.9	1.0	0.0	16.1
Low random	2.5	2.5	80.4	0.0	1.4	0.0	0.0	0.4	0.0	1.3	0.0	11.0	0.0	0.4
-ve gradient	15.5	19.9	0.1	12.2	14.4	0.1	0.0	3.4	0.2	2.8	0.1	4.8	0.0	26.7
Long flare	0.5	9.7	0.0	2.8	0.0	0.0	82.3	1.1	1.1	0.0	0.0	0.1	2.3	0.1
Other	4.1	11.3	22.6	8.6	2.4	8.2	7.3	4.0	7.4	3.5	0.9	1.9	3.4	14.5

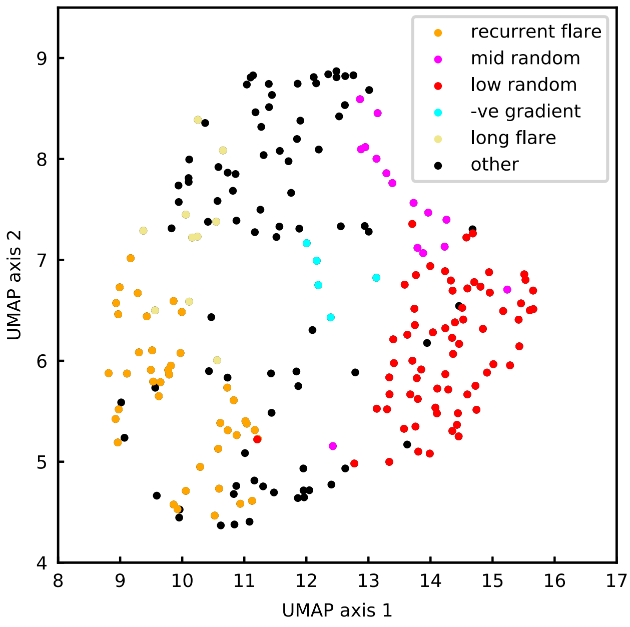

Figure 6. UMAP projection of the 222 means of Gaussian components of the GMM fit to 1 s data set. UMAP reduced dimensionality from 24 to 2. Some points are colour coded, and common colours indicate Gaussian clusters containing light-curve segments that show common characteristic variability patterns. As mentioned in the text, 'recurrent flare' stands for behaviour characteristic of light-curve classes ρ and ν, 'mid-random' and 'low random' stands for the lack of apparent structured variability (only random noise) at low and medium-high mean count rates, '-ve gradient' stands for slow decrease of mean count rate, whereas 'long flare' stands for irregular, long periods of flaring characteristic of classes κ and λ. All of the remaining Gaussian clusters are labelled as 'other'. The population of each type of cluster in terms of the classes of Belloni et al. system is shown in Table 2.

set. Some of the points are colour coded to indicate the characteristic patterns of light-curve segments found in the corresponding Gaussian clusters. Colour coding is produced by manual inspection of data found in Gaussian clusters, which is not a part of the proposed method of unsupervised data analysis. The purpose of this visualization is to show that Gaussian clusters sharing the characteristic patterns of behaviour tend to occupy similar regions of the SIFoS space.

4.2 Relating the set of light-curve patterns to the Belloni et al. system

In order to transform the two sets of clusters into a sets of observation features, for each observation we count the number of light-curve segments assigned to each of the Gaussian components. This results in 222 values per observation in 1 s data set and 279 values per observation in 4 s data set. For each observation we divide the counts by the sum of all counts for that observation. Feature vectors are independently normalized in order to reduce the impact of variance in the total number of segments extracted from each observation. This variance is caused by the fact that observations vary in total duration and the number of data gaps. Since feature vectors are normalized, they only contain information about the relative abundance of light-curve patterns within the corresponding observation. We refer to such 222-vectors and 279-vectors as observation 'fingerprints' because they allow identification of distinct classes of light-curve variability.

In order to showcase the usefulness of 'fingerprint' representation of data, Fig. 7 shows the subset of 1 s data set that has been human-labelled according to the Belloni et al. system, in terms of the Gaussian clusters described in Section 4. Fig. 7 shows combined 'fingerprints' for each class of observation ('fingerprints' of observations of the same class were summed to produce the combined 'fingerprints'). Rows of this heat map correspond to the 14 classes of observations, and columns correspond to the 222 Gaussian clusters of light-curve segments. Particular cells of the heat map reflect the relative abundance of segments of a particular class that have been assigned to a particular Gaussian cluster. Heat map was normalized row-wise, which means that high values indicate the Gaussian clusters which are most closely associated with a particular light-curve class. This in turn indicates that light-curve patterns are most abundant in light curves of a particular class.

The distinct appearance of the rows of the heat map indicates that the representation of light curves in terms of Gaussian clusters allows us to distinguish observations of different classes of the Belloni et al. system. This indicates that the light-curve representation that employs the set of patterns could serve as a viable feature space for supervised classification algorithms.

4.3 Classification of light curves using 'fingerprint' representation

In order to test the usefulness of the Gaussian cluster representation in light-curve classification, we train random forest classifiers (Breiman 2001) to assign observations to the Belloni et al. system based on their 'fingerprints' (note: we choose to disregard the subdivision of the χ class because the main distinguishing feature of those sub-classes is the position in the colour–colour diagram). We perform classification with the random forest classifier implementation included in the machine learning library SCIKIT-LEARN (Pedregosa et al. 2011; see Appendix E for more details about the algorithm). We train separate classifiers for 1 s data set and 4 s data set.

In order to address the issue of degeneracy of light-curve patterns described in Section 4.1, we merge Gaussian clusters separated by small distances. We use the Mahalanobis distance metric between the mean positions of Gaussian components. Mahalanobis distance is a distance metric used to measure the distance between a point and

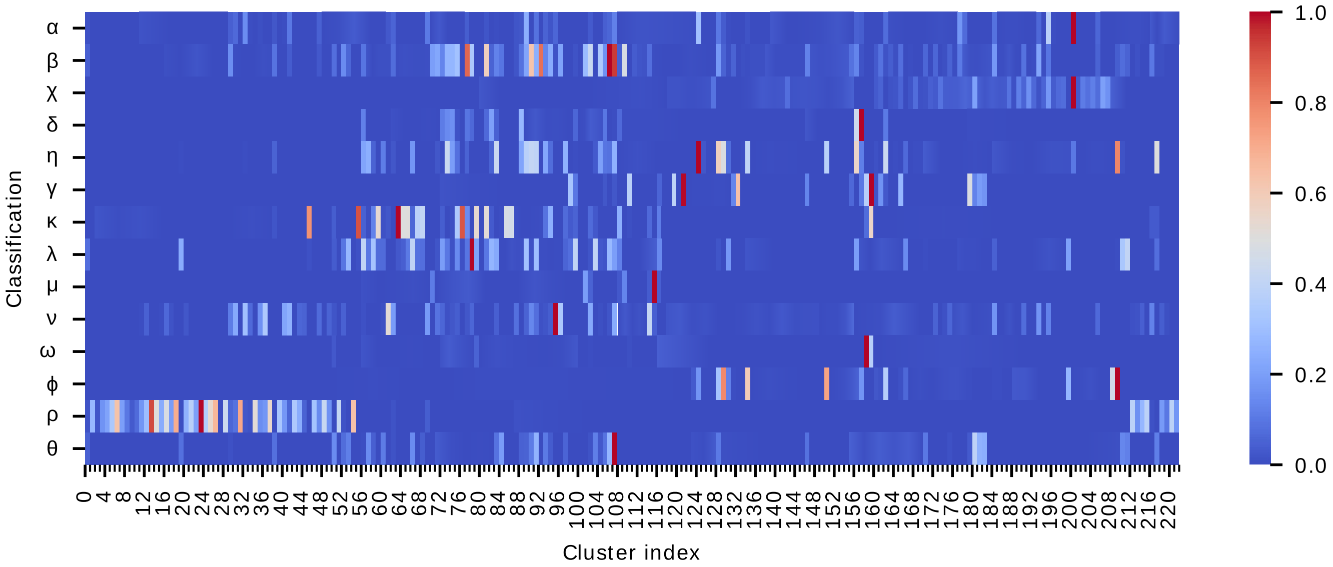

Figure 7. Heat map showing the distribution of light-curve segments across the set of 222 Gaussian clusters that were fit to the classified subset of 1 s data set. Heat map was normalized row-wise. Colour indicates the relative abundance of light-curve segments in the corresponding cluster. Clusters are ordered based on their proximity in the SIFoS space; this was determined using a hierarchical (single linkage) clustering algorithm. Observations' class labels come from Huppenkothen et al. (2017).

a distribution, while scaling the distance using the variance of that distribution. Mahalanobis distance is defined as

$$D_\mathrm{M} = \sqrt{(\boldsymbol{u} - \boldsymbol{v})\mathbf{V}^{-1}(\boldsymbol{u} - \boldsymbol{v})^T}, \tag{4}$$

where u and v are vectors whose separation is calculated, and V is the co-variance matrix.

The distance threshold between means of Gaussian components that are merged is one of the hyper-parameters included in the grid-search during the training of random forest classifiers. Gaussian components are allowed to merge if the distance calculated for both co-variance matrices is smaller than the threshold hyper-parameter.

We use the training and validation data subsets (described in Section 3.1) for the training of random forest classifiers. We find that the label of observation 10258-01-10-00 from Huppenkothen et al. (2017) disagrees with preceding literature (Klein-Wolt et al. 2002; Belloni & Altamirano 2013) therefore we change the label from λ to μ prior to classifier training.

Due to the limited number of labelled observations, we do not perform n-fold cross-validation, but instead we train the random forest classifier on 137 observations randomly sampled in a stratified manner from the data set of 159 observations that were used for training and validation of the neural network, and use the remaining 22 for validation. We repeat this process 100 times for each combination of hyper-parameters and find the mean of performance scores resulting from the 100 validation trials. In order to account for the class imbalance, the classifiers automatically adjust weights of each training sample to be inversely proportional to class frequencies. Table 3 lists hyper-parameters included in the grid-search, and Appendix E contains more details about the training of random forest classifier.

We use F1 and accuracy scores to measure the performance of classifiers. Both scores can take values in the range between 0 and 1, the higher the better. Accuracy is the proportion of correct classifications out of all predicted classifications. F1 score is the harmonic mean of recall and precision of classifications of a single class:

Table 3. List of hyper-parameters included in the grid-search of classification experiment described in Section 4.3. Hyper-parameters criterion and max_depth belong to the random forest classifier. The possible values of criterion are 'gini' and 'entropy', which stand for Gini impurity and information gain, respectively (see Appendix E for more details), and max_depth controls the depth of decision trees of the random forest. 'Merge distance' refers to the Mahalanobis distance threshold used in the process of merging Gaussian clusters. We test 100 values evenly spaced between 1.5 and 5. Hyper-parameter producing the largest F1 values for the two data sets are also listed.

Hyperparameter	Possible values	1 s data set	4 s data set
Criterion	'Gini', 'entropy'	'Gini'	'Gini'
Max_depth	None, 5, 10, 15, 25	5	15
Merge distance	Between 1.5 and 5	3.34	2.84

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{5}$$

where recall is the proportion of true positives out of the sum of all positives and precision is the proportion of true positives out of the sum of true positives and false negatives.

Reported F1 scores are weighted averages across the 14 classes. The scores are weighted by the number of observations of the corresponding class in order to account for the class imbalance. In general, weighted F1 is a more reliable performance indicator because accuracy can be easily biased when observations of one class significantly outnumber observations of other classes.

We find that the highest average validation performance scores for the 1 s data set are weighted F1 of 0.814 ± 0.065 and accuracy of 0.854 ± 0.054, while the highest average validation scores for the 4 s data set are 0.760 ± 0.068 and 0.810 ± 0.056 (reported uncertainty values are equal to one standard deviation calculated

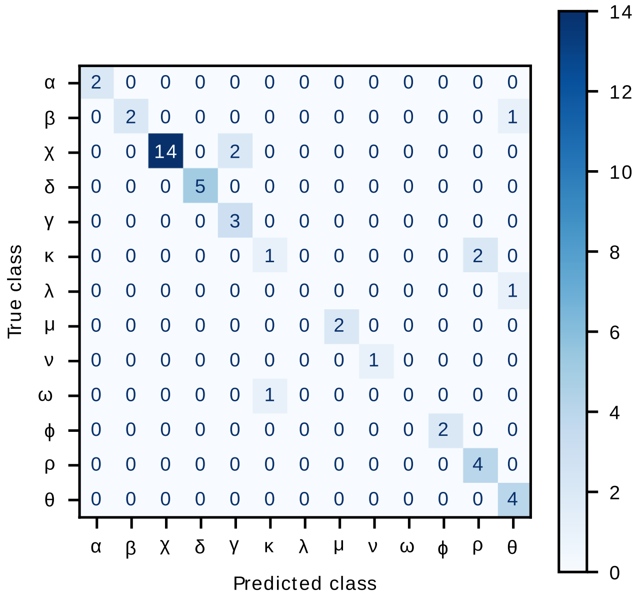

Figure 8. Confusion matrix showing classification results produced by the random forest classifier with optimal set of hyper-parameters for 47 testing set observations. The matrix shows results for the lowest weighted F1 score out of the one thousand random initiations of the classifier. Weighted F1 and accuracy of this initiation are 0.834 and 0.851, respectively. The mean weighted F1 and accuracy performance scores across the 1000 initiations are 0.878 ± 0.027 and 0.894 ± 0.027, respectively.

from the performance scores of the 100 validation trials). Therefore, we conclude that features derived from 1 s data set perform better in the task of light-curve classification. Hyper-parameters producing the highest average validation scores are listed in Table 3 for both data sets.

A random forest classifier with the optimal set of hyper-parameters is trained on the observations of the training subset and tested on the testing subset of 1 s data set, containing 47 observations. Random initiation of the algorithm is a significant cause of variance in the model performance therefore training and testing is repeated 1000 times. Mean of weighted F1 and accuracy performance scores of those classifications are 0.878 ± 0.027 and 0.894 ± 0.027, respectively. It should be noted that reported uncertainty values account only for variance caused by changes to the initial random state of the classifier.

Fig. 8 shows the classification results with the lowest performance scores out of the set of 1000 testing trials. We present the results of the test that achieved the lowest scores because they reveal the most information about the low-confidence classifications. It appears that our classifications disagree the most with the human-assigned labels for classes that have few observations available. Therefore, it is likely that classification performance could improve given a larger amount of training data. Since the number of labelled observations is very small, the variance of test results is likely to be high. In order to test this, for each combination of hyper-parameters shown in Table 3, we perform 100 tests where 69 test observations are randomly sampled in a stratified manner from the entire data set of 206 labelled observations (combined training, validation, and test data subsets), and the remaining observations are used for training. Appendix F shows the aggregated classification results of the 100 tests of the classifier with optimal hyper-parameters. Those results seem to agree with the results shown in Fig. 8. The mean of weighted F1 and accuracy performance scores of those classifications are 0.877 ± 0.031 and 0.896 ± 0.026, respectively. The results also show that classes which have only one training observation available (λ, ν, and ω) tend to disagree with the human-assigned labels the most frequently.

From the test results presented in Fig. 8, we examine each observation whose classification disagrees with the human-assigned labels. Observations of class β get assigned to other classes most likely because of the complex behaviour of its light curves. Some of the patterns of class β can be seen in light curves of other classes (Belloni et al. 2000). Observation with ID 40703-01-35-01 belonging to class β (Klein-Wolt et al. 2002) is assigned to class θ. This observation contains many periods of missing data, and the good time intervals contain dips similar to the class θ. The observation also contains W-shaped intervals that resemble those characteristic of class θ. Furthermore, the classifier predicts that β is the s most probable classification of this observation [top three predictions are θ (37.1 per cent), β (30.4 per cent), and δ (6.1 per cent)].

Two observations of class χ, 10408-01-42-00 and 40703-01-20-03, are assigned to class γ. Both observations show significantly higher count rate and RMS then an average χ observation, which is a possible cause of their classification. Class χ is the s most probable class prediction for both observations; top three predictions for 10408-01-42-00 are γ (21.8 per cent), χ (21.8 per cent), and ϕ (14.8 per cent), whilst for 40703-01-20-03 they are γ (21.9 per cent), χ (21.9 per cent), and ϕ (14.7 per cent).

Two observations of class κ, 40703-01-24-00 and 40703-01-25-00, are assigned to class ρ. Both observations show fairly regular, sharp flares, similar to those characteristic to class ρ, however, they are noticeably wider than the canonical ρ flares (see Fig. 9 for the light curves of the two observations). Other than the similarity to ρ light curves, another factor influencing the classification of these observations is likely the heterogeneity of the light curve behaviour of observations labelled as κ by Klein-Wolt et al. (2002). More details about the ambiguity of κ classifications is provided in Section 4.4. Furthermore, class κ is the s most probable class prediction for both observations. Top three predictions for 40703-01-24-00 are ρ (16.0 per cent), κ (15.5 per cent), and χ (14.0 per cent), whilst for 40703-01-25-00 they are ρ (15.7 per cent), κ (15.3 per cent), and χ (13.4 per cent).

Observation 20402-01-36-01 belongs to class λ and is assigned to class θ by the random forest classifier. This observation shows behaviour that is very characteristic of class λ; it shows periods of flaring that alternate with low, quiet periods (see λ light curve in Fig. 1 for an example). The likely cause of this class assignment is the fact that only one λ observation is included in the training data subset. The classifier predicts that λ is the s most probable classification; top three predictions are θ (30.1 per cent), λ (20.8 per cent), and κ (13.2 per cent).

Finally, observation 40703-01-29-01 belonging to class ω is assigned to class κ by the classifier. This observation shows steady flux with no structured variability except a singe W-shaped dip, which is a very typical ω behaviour (see Fig. 9 for the light curve). The classifier predicts that ω is the second most probable classification for this observation [top three predictions are κ (29.9 per cent), ω (21.0 per cent), and γ (16.8 per cent)]. One cause contributing to this classification is the fact that only one labelled observation of class ω is available in the training data subset. However, another major cause becomes apparent upon inspection of ω and κ observations. Many of the observations labelled as κ by Klein-Wolt et al. (2002) show behaviour that very strongly resembles class ω. We discuss this issue in more detail in Section 4.4.

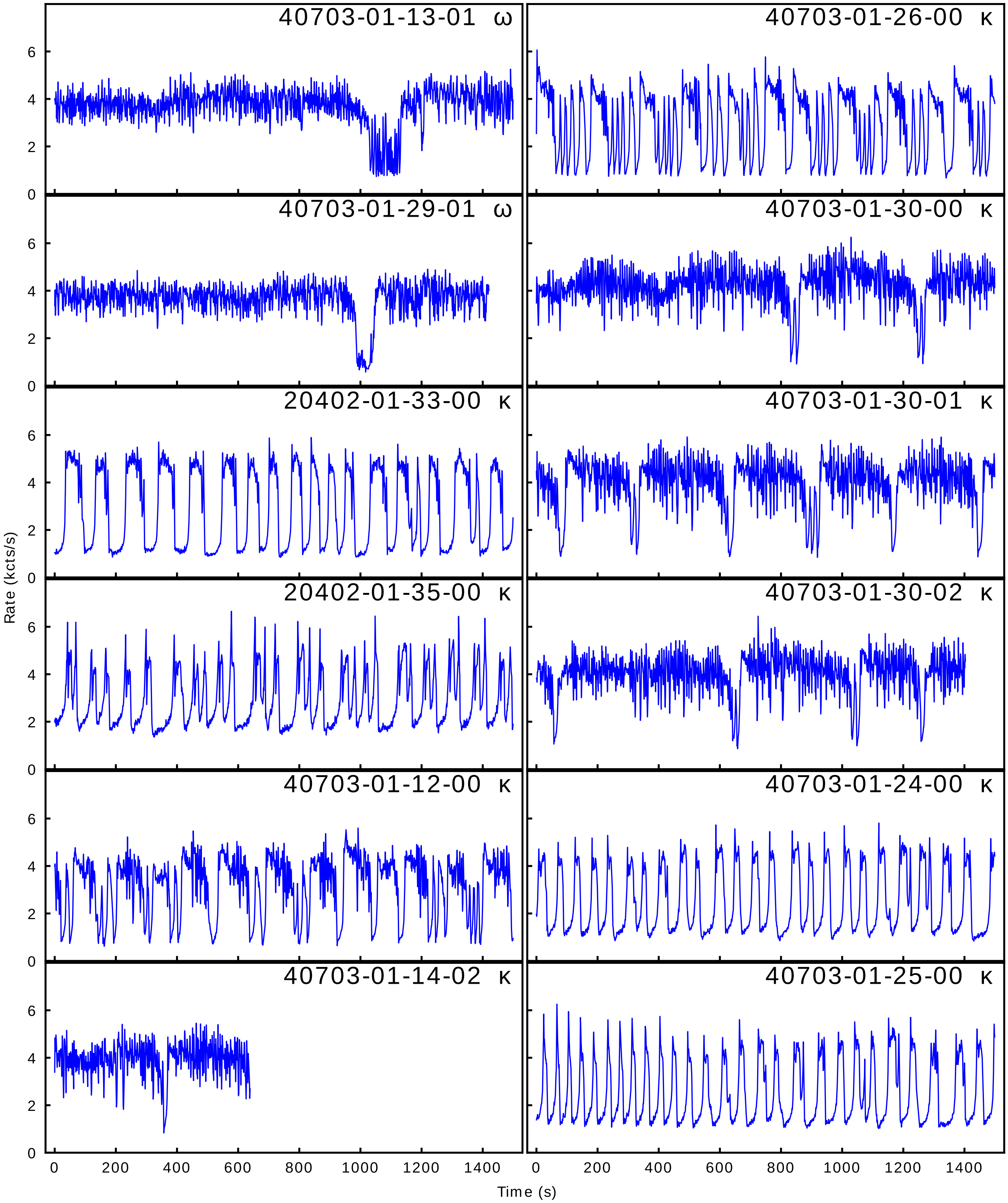

Figure 9. Light curves of ω and κ observations discussed in Section 4.4. Each subfigure contains first 1500 s of the light curve or as much as is available in case of shorter observations. Each subfigure contains the observation ID and classification from Huppenkothen et al. (2017). Fig. 12 contains light curves that require individual time axes.

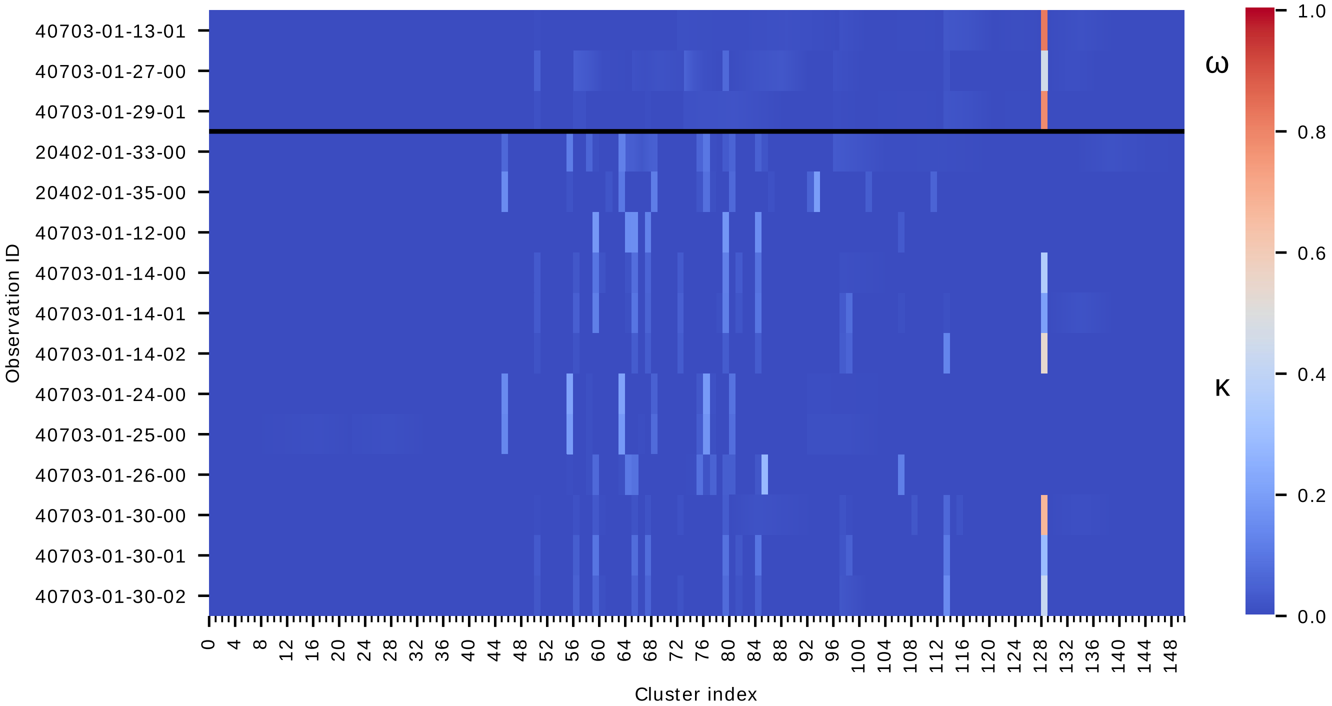

Figure 10. ‘Fingerprint’ representation of ω (above the black line) and κ (below the black line) observations from Huppenkothen et al. (2017). Clusters are merged using the method described in Section 4.3, using the optimal Mahalanobis distance threshold of 3.34. Colour indicates the relative abundance of light-curve segments in the corresponding cluster.

4.4 Data-driven review of ω and κ classifications

Fig. 10 shows the ‘fingerprint’ representation of all ω and κ observations from Huppenkothen et al. (2017). There clearly exist at least two groups of κ observations. Six of the observations (40703-01-14-00/01/02, 40703-01-30-00/01/02) are much more similar to ω observations than the other κ observations. In order to assess the similarity of presented observations, we perform hierarchical clustering of observations in the ‘fingerprint’ space. We use the hierarchical clustering algorithm included in the SCIPY package (Virtanen et al. 2020; see Appendix G for more details about the algorithm).

Fig. 11 shows a dendogram resulting from the clustering of ‘fingerprints’ of ω and κ observations (see Figs 9 and 12 for their light curves). In the green branch of the dendogram, observations classified as ω by Klein-Wolt et al. (2002) are clustered with the six κ observations mentioned above. These observations show semiregular dips with one or more re-flares. Observations 40703-01-13-01/29-01/30-00 clustered in the lower green sub-branch show lower frequency of dipping than the other observations, making them more alike to the canonical ω behaviour. Observations 40703-01-30-02/14-02/27-00 show a slightly higher dipping frequency, whilst observations 40703-01-30-01/14-00/14-01 show the highest frequency of dips in the green branch of the dendogram. Similarity between ‘fingerprints’ of κ observations 40703-01-14-00/01/02, 40703-01-30-00/01/02, and canonical ω observations is the result of the similarity between the light curves of those observations. Based on this similarity, we suggest that those κ observations should be viewed as examples of an intermediate κ/ω state, with a major ω component.

Observations in the orange branch of the dendogram show characteristic quasi-periodic and aperiodic flares of class κ. However, observations 40703-01-12-00/26-00 show a mix of flares with large and small width as opposed to 40703-01-24-00/25-00 and 20402-01-33-00/35-00, show a more consistent flare profile. Moreover, the presence of flares with greater width suggests an intermediate κ/ω state with a major κ component. Furthermore, Belloni & Altamirano (2013) indicate that observations 40703-01-12-00/26-00 belong to class ω, whilst both Pahari & Pal (2010) and Belloni & Altamirano (2013) indicate that observations 40703-01-14-00/30-00 belong to class ω, which shows that ambiguity of κ and ω classification exists in the literature.

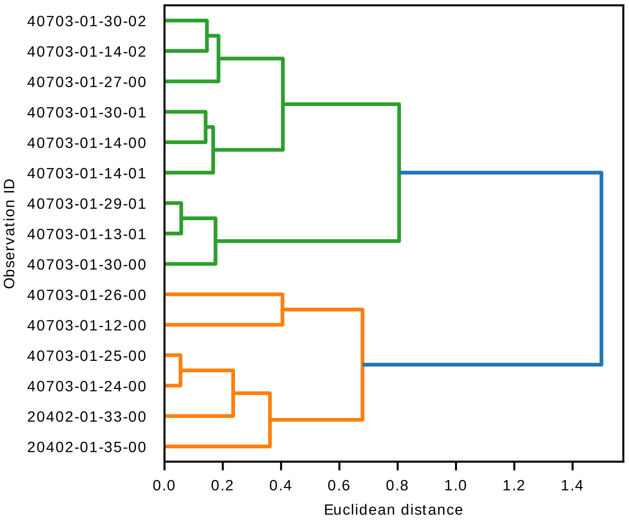

Figure 11. Dendogram resulting from the hierarchical clustering of ω and κ observations based on their ‘fingerprints’ shown in Fig. 10. Hierarchical clustering algorithm uses the ward method and euclidean metric. Splitting of branches of the dendogram at smaller values of Euclidean distance indicates that the observations in corresponding leaf nodes are more closely related.

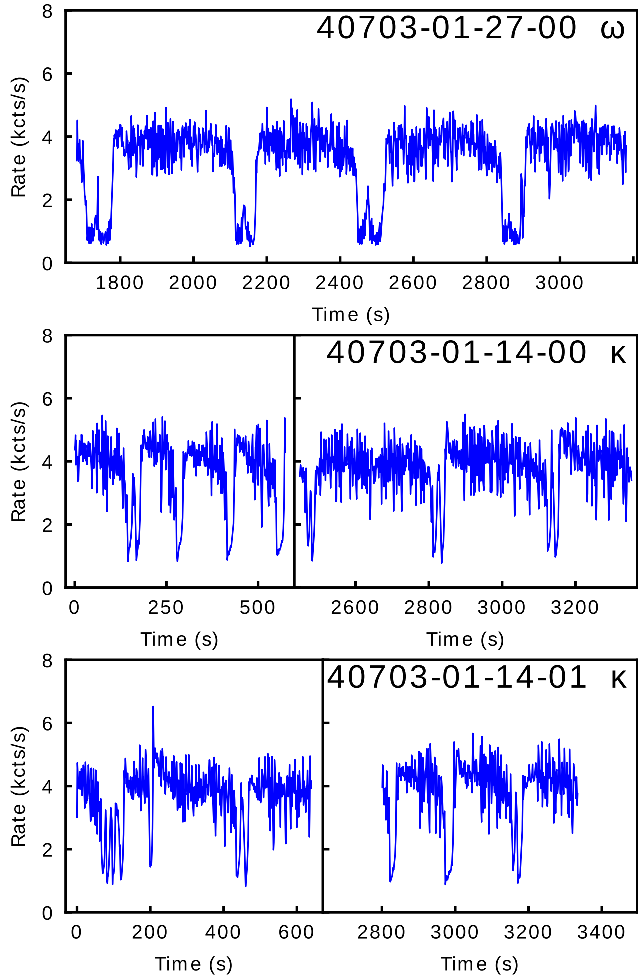

Figure 12. Light curves of ω and κ observations that are mentioned in Section 4.3. Top subfigure contains only the second, longer good time interval of the observation. Middle and bottom subfigures contain two good time intervals of their respective observations, and the data gaps are removed. Each subfigure contains 1500 s of the light curve or as much as is available in case of shorter observations. Each subfigure shows the observation ID and classification from Huppenkothen et al. (2017). Fig. 9 contains light curves of remaining ω and κ observations.

Table 4 shows a chronological list of observations captured in the periods when the source was showing κ and ω behaviour, along with the classifications from Klein-Wolt et al. (2002), Pahari & Pal (2010), and Belloni & Altamirano (2013). Observations of class κ and ω were observed in close succession, which supports the notion of intermediate states between those classes. Based on the classification of observations in the two periods shown in Table 4, it seems that the source tends to transition between classes in the order $\kappa \rightarrow \omega \rightarrow \kappa$.

However, clustering results of the ‘fingerprint’ representation should be interpreted with caution due to the geometry of the data sampling space (see Section 5 for more details).

Table 4. Chronological list of RXTE/PCA observations of GRS 1915+105 during periods 51288-51306 MJD and 51394-51432 MJD. Classifications from Klein-Wolt et al. (2002) (Class. K), Pahari & Pal (2010) (Class. P) and Belloni & Altamirano (2013) (Class. B) are provided.

Observation ID	Date (MJD)	Class. K	Class. P	Class. B
40703-01-12-00[a]	51288	κ	ω	ω
40115-01-01-00	51288	–	–	–
40403-01-07-00	51291	ω	ω	ω
40703-01-13-00	51299	γ	–	–
40703-01-13-01	51299	ω	ω	ω
40115-01-02-00	51299	–	–	–
40703-01-14-00[a]	51306	κ	ω	–
40703-01-14-01	51306	κ	–	–
40703-01-14-02	51306	κ	–	–
40703-01-24-00	51394	κ	–	–
40703-01-25-00	51399	κ	–	–
40115-01-05-00	51406	–	–	–
40703-01-26-00[a]	51407	κ	–	ω
40703-01-27-00	51413	ω	–	–
40703-01-27-01	51413	γ/ω	ω	–
40703-01-28-00	51418	ω	–	ω
40703-01-28-01	51418	–	–	–
40703-01-28-02	51418	ω	ω	ω
40115-01-06-00/01	51423	–	–	–
40703-01-29-00	51426	ω	ω	ω
40703-01-29-01	51426	ω	–	ω
40703-01-29-02	51426	γ/ω	–	ω
40703-01-30-00[a]	51432	κ	ω	–
40703-01-30-01	51432	κ	–	–
40703-01-30-02	51432	κ	–	–
40703-01-30-03	51432	γ/κ	–	–

Note.[a]Observations whose classifications seem to be inconsistent.

4.5 Classification of 1024 s segments

We conduct an additional classification experiment with the goal of making it as comparable as possible to the work of Huppenkothen et al. (2017). The purpose of our paper is primarily to introduce an unsupervised method of light-curve data characterization, whilst the primary objective of Huppenkothen et al. (2017) was to study GRS 1915+105, however, both papers use machine learning methods to perform classification on this source, so in spite of the differences, we believe that this experiment is informative.

We segment all the available RXTE/PCA observations of GRS 1915+105 in Standard-1 format to 1024 s segments and use the stride length of 256 s, yielding the total of 11 028 segments, 2141 of which have human-assigned labels. We split this data between the training, validation and testing data subsets in the ratio of approximately 50:25:25, ensuring that segments created from a single observation are not split between data subsets. Data are split in a stratified manner; at least one observation of each class is randomly assigned to each subset, and the remaining observations are distributed according to the ratio of 50:25:25. This results in 982, 592 and 566 segments in the training, validation, and testing subsets, respectively. Further 6573 and 2314 segments of observations with no human-assigned labels are also assigned to the training and validation sets, respectively.

In order to make the segments compatible with the LSTM–VAE network discussed in Section 3, we further subdivide the 1024 s segments into 64 non-overlapping segments of 16 s (which at the cadence of 0.125 s comprise 128 data points, as required by the network). We train the network using the training subset and evaluate the performance of the network after each epoch of training using the validation subset of data. We use the Adam optimizer with the `clipvalue` argument set to 0.5 to train the network. Training is stopped after 269 epochs, when the validation loss stopped improving for 50 consecutive epochs. We use the resulting network to encode the light-curve segments and create 20 SFoS features.

We fit a GMM with 250 components to the SIFoS features of the training and validation subsets of data. We then perform a grid search

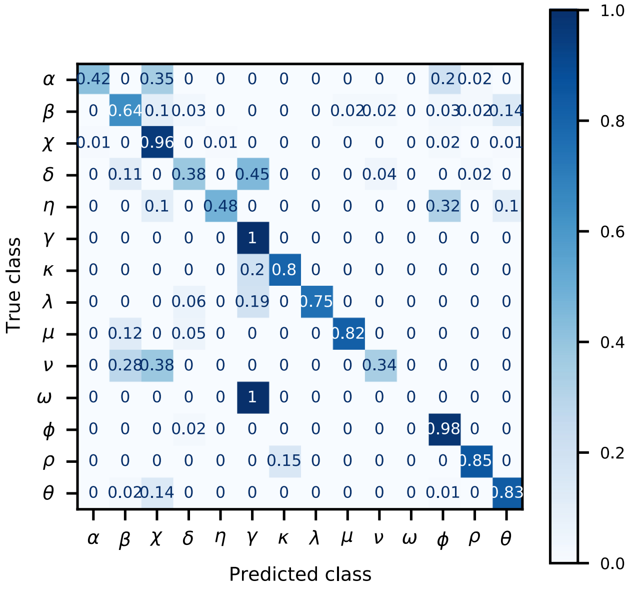

Figure 13. A row-wise normalized confusion matrix showing classification results for light-curve segments of 1024 s. The numbers in each row are divided by the sum of numbers in that row.

of hyper-parameters listed in Table 3, where for each combination of hyper-parameters we train a random forest classifier on the training set and we evaluate the prediction on the validation set. We find that the highest weighted F1 and accuracy performance scores of 0.834 and 0.851, respectively, are produced using the merge distance of 3.798, criterion of ‘entropy’, and max_depth of 5.

We test the classifier with optimal hyper-parameters on combined validation and testing data subsets, resulting in weighted F1 and accuracy performance scores of 0.811 and 0.822, respectively. The accuracy of this classification is ca. 10 percentage points lower than that of Huppenkothen et al. (2017). Their method is simpler and more accurate, but it requires a larger variety of data features, i.e. power-spectral and hardness ratio features. Therefore, depending on the available features, it might be the preferable method for the task of classification of 1024 s segments. Including those features in our method could result in a performance improvement, which is a possible avenue of future investigation.

The possible avenues for improvement of both methods include the training on a larger amount of labelled data, as well as the reduction of ambiguity in the labelled data. Ambiguity is caused by the fact that the classes of the Belloni et al. system are not fully exhaustive and mutually exclusive. This is discussed further in Section 5.

Fig. 13 shows the confusion matrix of the predicted labels against the human-assigned labels. Even though most of the segments are classified in agreement with the human-assigned labels, the drop in performance is significant when compared to the other experiments shown in this paper (Section 4.3 and Appendix F). In particular, all of the ω segments are classified as γ, which is most likely caused by the noise in the data. The data with cadence of 0.125 s contain a significant amount of random noise, and the segmentation of data to the length of only 16 s followed by standardization, results in segments that are seemingly featureless, with occasional peaks and troughs beyond the noise level. The characteristic dips of class ω were not preserved in this representation of data. In conclusion, re-binning of data to cadence longer than 0.125 s prior to LSTM–VAE training is likely to be beneficial for the performance of our method. Additionally, when light curves are segmented to 1024 s prior to classification, we cannot exploit the main advantage of the ‘fingerprint’ representation, which aggregates features of long light curves. Light curves of classes whose cycles of variability evolve over longer time than 1024 s can yield segments that appear significantly different, depending on the phase of a variability cycle.

5 CONCLUSIONS

We introduce a data-driven method of light-curve feature extraction and test the utility of resulting features by conducting a set of supervised multiclass classification experiments, using a set of human-labelled observations. Light-curve classification of data in 1 s resolution resulting in a mean weighted F1 score of 0.878 suggests that the proposed method of unsupervised feature extraction is capable of producing features that represent light-curve data in a meaningful way.

In regard to potential future work on the data set of GRS 1915+105 X-ray light curves, a supervised approach could be taken to study the link between the observed light-curve patterns and underlying physics. Huppenkothen et al. (2017) used a classification scheme of GRS 1915+105 states into stochastic and chaotic processes (Harikrishnan, Misra & Ambika 2011) in order to leverage their machine learning classification algorithm in the study of the long term evolution of accretion properties of the source, and similar studies could be complemented using our method of feature extraction.

Another possible application of the method is the unsupervised exploration of the data space. The shape of the latent data manifold encodes information about modes of activity of the source and its evolution between them. The analysis of correlation between SIFoS and various power-spectral and energy-spectral features could be performed in order to study the links between latent variables, the occurrence of different variability patterns, and the physically interpretable parameters. For example, broad-band power spectral shape could help to link variability patterns with the geometry of the emitting regions (Heil, Uttley & Klein-Wolt 2015), presence of high-frequency quasi-periodic oscillations could link the patterns with the orbital motion of matter in the accretion disc (see Motta 2016, for a review), and radio flux could link them with the emission of relativistic jets (Belloni, Motta & Muñoz-Darias 2011).

Work presented in this paper shows that dimensionality reduction of the data set, followed by clustering of observations in this reduced space could be a way to derive a set of classes of source behaviour, which avoids the biases of human characterization and annotation of data. Furthermore, we find that Gaussian component merging based on the Mahalanobis distance between them can help to reduce the problem of cluster degeneracy caused by the limitations of GMM, which requires multiple components to follow curved data manifolds.

The Belloni et al. system of classification discussed in this work is not comprehensive, and to some degree, it is arbitrary. As Belloni et al. (2000) point out, their classification system was not intended to exhaustively list mutually exclusive modes of behaviour. This creates a problem for any classification effort that is based on this system. A smaller number of classes could be chosen, because some classes show behaviour that arguably lies on the same continuum. One example of such continuum could be followed by classes λ and κ, which show similar behaviour at slightly different time-scales (Belloni et al. 2000). Furthermore, our work shows that classes κ and ω can show very similar light-curve behaviour and possibly lie on a continuum as well. A larger number of classes could just as well be required, as it is possible that there exist additional patterns of behaviour that have not been characterized yet [similar to classes ω and ξ (a.k.a. η) that were added to the Belloni et al. system later than the original 12]. Transitions between classes are observed, so ambiguity in classification of observations cannot be avoided.

Taking these considerations into account, it is very difficult to ensure the accuracy of classification for large sets of unknown data, which cannot be standardized to adhere to the assumed classification system. Performance of any supervised classification algorithm greatly depends on the definition of the classes of observations, and any ambiguity is going to affect this performance. Therefore, clustering of the data set can help in deriving a data-driven set of observation classes, which helps to avoid the biases of human characterization and annotation of data.

However, further work in this direction will need to address the problem of clustering of compositional data. The 'fingerprint' representation uses vectors of fixed size, whose values sum up to a positive constant. This is a type of compositional data, and as such it is constrained to the geometry of a simplex (Aitchison & Egozcue 2005), which means that results of clustering of raw compositional data are often unreliable. Clustering methods applied to compositional data should take into account the need of prior transformation of the data into an unbounded, Euclidean space. This issue is an open area of research, and it is further complicated by the presence of a large number of zero values in the 'fingerprint' compositional data (Aitchison et al. 2000; Martín-Fernández, Barceló-Vidal & Pawlowsky-Glahn 2003). Study of the appropriate transformation methods for compositional data goes beyond the scope of our work, therefore results of 'fingerprint' clustering should be interpreted with caution.

Since the proposed feature extraction method is easily generalizable to different types of time series data, there exists a range of possible applications for the proposed feature extraction pipeline. A similar type of analysis is possible for sources other than GRS 1915+105, and in principle any energy band of light curves.

Fingerprint representation could be used to test whether the phenomenology observed in GRS 1915+105 is also present in other sources. For example, similarity in the variability of IGR J17091-3624 (Court et al. 2017), and the Rapid Burster, MXB 1730-335 (Bagnoli & In't Zand 2015) could point to accretion physics that are independent of the nature of the accretor.

Pattnaik et al. (2021) attempted machine learning classification of compact objects in low-mass X-ray binaries based on energy-spectral features, and found that fairly accurate classification was possible. The presence of a black hole or neutron star in the binary system can have a significant impact on the physical interpretation of the observed phenomenology. Classification of compact objects using the fingerprint representation of X-ray variability patterns is a possible subject of future work. However, care must be taken to select sources in the hard state, which show structured X-ray variability.

Derivation of fingerprints that represent the set of light-curve patterns observed in a fixed amount of time could be the basis of a live monitoring system, which would alert the user about changes in the behaviour of the source. This could involve classifying observations using a known system of classes, but it could also involve the task of outlier detection, where position of an observation would be tracked within the encoded feature space. Observations producing feature vectors that fall in sparse regions of the feature space would indicate an anomaly.

The main requirement of the proposed feature extraction method is that the light curves must be evenly sampled in time. This requirement is satisfied by data similar in nature to the pointed observations of RXTE/PCA. This includes data captured by the X-ray Telescope aboard the Neil Gehrels *Swift* Observatory (*Swift*; Burrows et al. 2005) or X-ray Timing Instrument aboard the Neutron Star Interior Composition Explorer (Gendreau et al. 2016), etc. High-speed optical telescopes ULTRA-fast CAMera (ULTRACAM; Dhillon et al. 2007) and High PERformance CAMera (HiPERCAM; Dhillon et al. 2016) also produce light curves that are evenly sampled over the time of an exposure, and could be analysed using the method we propose. However, the amount of data produced by ULTRACAM and HiPERCAM is not large enough to justify the use of machine learning analysis. The optical timing camera (Castro et al. 2019) will produce a larger amount of data of similar nature, and the proposed method of feature extraction could be appropriate for their analysis.

The proposed method could be used to characterize long-term light curves captured by all-sky X-ray surveys, like those performed with the Gas Slit Camera aboard MAXI (Matsuoka et al. 2009), the Burst Alert Telescope aboard *Swift* Gehrels et al. (2004) or the All Sky Monitor aboard RXTE, provided that light curves have regular time bins. Interpolation could also be performed if needed.

Pursiainen et al. (2020) interpolated ca. 30 000 light curves of the Dark Energy Survey Supernova Programme using Gaussian Processes, and increased the cadence from the average of 7 d to a constant 0.5 d cadence (see also Wiseman et al. 2020). Light curves generated by several ground-based surveys could be made compatible with our feature extraction method using such interpolation techniques. Those surveys include LSST, ZTF, and Asteroid Terrestrial-impact Last Alert System (Heinze et al. 2018). Optimization of the type and size of kernel used in Gaussian Processes interpolation would need to be performed by the user prior to feature extraction, and the choice of parameters would depend on the nature of data and the scientific goal.

In addition to the requirement of having even sampling in time, light curves should not have many gaps that cannot be interpolated over, in order to be suitable for analysis using the proposed feature extraction method. During the light-curve segmentation stage, segments are extracted using a moving window method, and segments that straddle over data gaps are discarded. Therefore, light curves must have few gaps to allow for a choice of segment size that is large enough to encompass time-scales that are relevant for the variability of the analysed data.

There are several limitations to this work. Fig. 3 reveals that classes ϕ and χ are similar in terms of IFoS. The only significant differences are (1) that mean count rate values of segments of class χ span a larger range than the segments of class ϕ and (2) that segments of class χ can have more positive outliers in terms of kurtosis. Fig. 4 shows that the combination of IFoS contains enough information to distinguish many cases of those classes, but significant areas of overlap still exist. The two classes are best distinguished based on the hardness of their colour–colour diagrams (Belloni et al. 2000). Since both classes of observations show no structured X-ray variability patterns, and their IFoS distributions largely overlap, it should be noted that classification attempts based on features presented in this work could disagree with classification of Belloni et al. (2000). The proposed method is only able to capture time series patterns therefore it would not be able to differentiate between observations that differ only in terms of energy spectra. Nevertheless, users of the method could choose to supplement SIFoS with additional features, dependent on their individual use case.

Another set of limitations involves data pre-processing. The rebinning of light-curve data to 1 and 4 s resolution was a compromise between computational tractability and descriptiveness of GRS 1915+105 variability. Even though the shortest time-scale of significant X-ray flux changes in this source is ca. 5 s (Nayakshin et al. 2000), binning inevitably led to a loss of some of the fast variability information. Furthermore, the choice of light-curve segment length was influenced by computational constrains, but also informed by the previous work and knowledge of the time-scales of light-curve patterns in GRS 1915+105. There is a fairly large degree of tolerance for the choice of these parameters, but some knowledge of the time-scales of interest was required.

We did not fine-tune the method to study any particular subset of variability classes within the source, however, users interested in a more specialized study of a particular scientific problem could choose to analyse light curves binned to smaller time bins to improve the modelling of very narrow features. For example, Massaro et al. (2020) modelled the light curves of GRS 1915+105 using a system of ordinary differential equation, and successfully reproduced the sharp peaks present at the beginning of the κ class bursts. Reconstruction of light-curve segments binned to 1 s resolution results in some smearing of such fine features (see Fig. 2), so the study of this particular feature would benefit from higher time resolution. However, the light curves of GRS 1915+105 contain a significant amount of noise, and increasing the temporal resolution of light curves will result in the increased amount of noise in the training data set, which could affect the ability of the model to identify patterns within the data.

For cases where relevant time-scales are difficult to predict, multiple ‘fingerprints’ could be derived from data in a range of temporal resolutions, and subsequently concatenated to create a single feature vector. However, such an approach would increase the noise in feature vectors, so further work is required to test this.

Our main assumption was that light-curve segments exhibiting similar type of variability patterns and count rate distribution had similar values of SFoS and IFoS. Segments with more similar values of SIFoS were consequently separated by smaller distances within the SIFoS feature space. Therefore, it was expected that Gaussian clusters contained homogeneous subsets of light-curve segments, which were more similar to each other than to segments found in other Gaussian clusters. Inspection of Gaussian cluster populations revealed that this assumption was justified to a large degree, however, fitting a GMM with a larger number of Gaussian components would provide a greater precision of density estimation of data in the feature space, and hence reduce the risk of non-homogeneity of created clusters. On the other hand, such an approach would likely aggravate the issue of pattern degeneracy in the resulting set of light-curve patterns, which would potentially lead to more complex ‘fingerprints’ and ultimately to a more noisy feature set.

It should be noted that GMM is not inherently a clustering algorithm but a density estimation algorithm. Following preliminary experiments with alternative clustering algorithms we find the approach of merging Gaussian components of GMM to be the most straightforward method of clustering for this data set. However, it is possible that the issue of cluster degeneracy could be circumvented if a better-suited clustering method was found. This is an important area of future work.

Merging of clusters based on the Mahalanobis distance can help to reduce the number of degenerate features, but in this work we rely on the classified subset of data to find the optimal distance threshold. In cases where data classification is not possible, unsupervised methods of evaluating clustering performance could be used instead, for example, Silhouette Coefficient, Calinski-Harabasz Index, and Davies-Bouldin Index included in the SCIKIT-LEARN library (Pedregosa et al. 2011), however, computational cost of such an approach might be higher.

ACKNOWLEDGEMENTS

This research has made use of data obtained through the High Energy Astrophysics Science Archive Research Center Online Service, provided by the NASA/Goddard Space Flight Center.

We thank Tomaso Belloni for sharing his work on GRS 1915+105. We also thank Kevin Alabarta, Mehtap Ozbey Arabaci, John Paice, Philip Wiseman, Miika Pursiainen, and the data scientists of HAL24K Labs for useful discussion and advice. Diego Altamirano and Adam B. Hill acknowledge support from the Royal Society.

DATA AVAILABILITY

The data underlying this article are publicly available in the High Energy Astrophysics Science Archive Research Center (HEASARC) at https://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl, and in the repository of the article of Huppenkothen et al. (2017) at https://github.com/dhuppenkothen/BlackHoleML.

REFERENCES

Aitchison J., Barceló-Vidal C., Martín-Fernández J. A., Pawlowsky-Glahn V., 2000, Math. Geol., 32, 271
Aitchison J., Egozcue J. J., 2005, Math. Geol., 37, 829
Altamirano D. et al., 2011, ApJ, L742
Ankerst M., Breunig M. M., Kriegel H.-P., Sander J., 1999, ACM SIGMOD Record, 28, 49
Armstrong D. J. et al., 2015, MNRAS, 456, 2260
Bagnoli T., In’t Zand J. J., 2015, MNRAS, 450, L52
Becker I., Pichara K., Catelan M., Protopapas P., Aguirre C., Nikzat F., 2020, MNRAS, 493, 2981
Bellm E. C., 2014, in The Third Hot-wiring the Transient Universe Workshop. p. 27, http://www.slac.stanford.edu/econf/C131113.1/
Belloni T. M., Altamirano D., 2013, MNRAS, 432, 10
Belloni T. M., Motta S. E., Muñoz-Darias T., 2011, Bull. Astron. Soc. India, 39, 409
Belloni T., 2001, in The Neutron Star-Black Hole Connection, vol. 567. Kluwer, Dordrecht, p. 295
Belloni T., Klein-Wolt M., Mendez M., van der Klis M., van Paradijs J., 2000, A&A, 355, 271
Belloni T., Méndez M., King A. R., Van Der Klis M., Van Paradijs J., 1997a, ApJ, 479, 145
Belloni T., Méndez M., King A. R., Van Der Klis M., Van Paradijs J., 1997b, ApJ, 488, 109
Benkabou S.-E., Benabdeslem K., Canitia B., 2018, Knowl. Inf. Syst., 54, 463
Bradt H. V. , Rothschild R. E., Swank J. H., 1993, Astronomy and Astrophysics Supplement Series, 97, 355
Breiman L., 2001, Machine Learning, 45, 5
Burrows D. N. et al., 2005, Space Sci. Rev., 120, 165
Capitanio F. et al., 2006, ApJ, 643, 376
Castro A. et al., 2019, Rev. Mex. Astron. Astrofis., 55, 363
Castro-Tirado A. J., Brandt S., Lund N., Lapshov I., Sunyaev R. A., Shlyapnikov A. A., Guziy S., Pavlenko E. P., 1994, ApJS, 92, 469
Charnock T., Moss A., 2017, ApJ, 837, L28
Chetlur S., Woolley C., Vandermersch P., Cohen J., Tran J., Catanzaro B., Shelhamer E., 2014, cuDNN: Efficient Primitives for Deep Learning, preprint (arXiv:1410.0759)
Chollet F., 2015, Keras, https://keras.io
Court J. M., Altamirano D., Pereyra M., Boon C. M., Yamaoka K., Belloni T., Wijnands R., Pahari M., 2017, MNRAS, 468, 4748
Dempster A. P., Laird N. M., Rubin D. B., 1977, J. R. Stat. Soc. B, 39, 1
Dhillon V. S. et al., 2007, MNRAS, 378, 825
Dhillon V. S. et al., 2016, in Evans C. J., Simard L., Takami H., eds, Ground-based and Airborne Instrumentation for Astronomy VI, 99080Y. SPIE, p. 251
Ester M., Kriegel H.-P., Sander J., Xu X., 1996, in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. AAAI Press, Portland, Oregon, United States, p. 226
Fender R., Belloni T., 2004, ARA&A, 42, 317
Gehrels N. et al., 2004, ApJ, 611, 1005

Gendreau K. C. et al., 2016, in Proc. SPIE Conf. Ser. Vol. 99051, Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray. SPIE, Bellingham, p. 99051H
Glasser C. A., Odell C. E., Seufert S. E., 1994, IEEE Trans. Nucl. Sci., 41, 1343
Hannikainen D. C. et al., 2003, A&A, 411, 415
Hannikainen D. C. et al., 2005, A&A, 435, 995
Harikrishnan K. P., Misra R., Ambika G., 2011, Res. Astron. Astrophys, 11, 71
Heil L. M., Uttley P., Klein-Wolt M., 2015, MNRAS, 448, 3348
Heinze A. N. et al., 2018, The Astronomical Journal, 156, 241
Hinton G., 2013, Non-linear dimensionality reduction. https://www.cs.toronto.edu/ hinton/csc2535/notes/lec11new.pdf
Hochreiter S., Urgen Schmidhuber J., 1997, Neural Comput., 9, 1735
Huppenkothen D., Heil L. M., Hogg D. W., Mueller A., 2017, MNRAS, 466, 2364
Hyndman R. J., Wang E., Laptev N., 2015, in 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, Atlantic City, NJ, USA, p. 1616
Ismail Fawaz H., Forestier G., Weber J., Idoumghar L., Muller P. A., 2019, Data Mining Knowl. Discovery, 33, 917
Ivezic Z. et al., 2019, ApJ, 873, 111
Kingma D. P., Ba J., 2015, in Bengio Y., LeCun Y., eds, 3rd International Conference on Learning Representations. San Diego, CA, USA
Kingma D. P., Welling M., 2014, in Bengio Y., LeCun Y., eds, 2nd International Conference on Learning Representations, Conference Track Proceedings. Banff, AB, Canada
Klein-Wolt M., Fender R. P., Pooley G. G., Belloni T., Migliari S., Morgan E. H., van der Klis M., 2002, MNRAS, 331, 745
Kuulkers E., Lutovinov A., Parmar A., Capitanio F., Mowlavi N., Hermsen W., 2003, Astron. Telegram, 149, 1
Längkvist M., Karlsson L., Loutfi A., 2014, Pattern Recognit. Lett., 42, 11
Mackenzie C., Pichara K., Protopapas P., 2016, ApJ, 820, 138
Mahabal A., Sheth K., Gieseke F., Pai A., Djorgovski S. G., Drake A., Graham M., 2017, in 2017 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, Honolulu, HI, USA, p. 1
Martín-Fernández J. A., Barceló-Vidal C., Pawlowsky-Glahn V., 2003, Math. Geol., 35, 253
Massaro E., Capitanio F., Feroci M., Mineo T., Ardito A., Ricciardi P., 2020, MNRAS, 496, 1697
Matsuoka M. et al., 2009, PASJ, 61, 999
McInnes L., Healy J., Saul N., Großberger L., 2018, Journal of Open Source Software, 3, 861
Mirabel I. F., Rodriguezt L. F., 1994, Nature, 371, 46
Motta S. E., 2016, Astron. Nachr., 337, 398
Naik S., Agrawal P. P. C., Rao P. A. R., Paul B., 2002, MNRAS, 330, 487
Naul B., Bloom J. S., Pérez F., Van Der Walt S., 2018, Nat. Astron., 2, 151
Nayakshin S., Rappaport S., Melia F., 2000, ApJ, 535, 798
Pahari M., Pal S., 2010, MNRAS, 409, 903
Pattnaik R., Sharma K., Alabarta K., Altamirano D., Chakraborty M., Kembhavi A., Méndez M., Orwat-Kapola J. K., 2021, MNRAS, 501, 3457
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Pieringer C., Pichara K., Catelán M., Protopapas P., 2019, MNRAS, 484, 3071
Pursiainen M. et al., 2020, MNRAS, 494, 5576
Richards J. W. et al., 2011, ApJ, 733, 10
Rokach L., Maimon O., 2005, in The Data Mining and Knowledge Discovery Handbook. Springer, Berlin, p. 321
Shallue C. J., Vanderburg A., 2018, AJ, 155, 94
Singh S., Yassine A., 2018, Energies, 11, 1
Valenzuela L., Pichara K., 2018, MNRAS, 474, 3259
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Wiseman P. et al., 2020, MNRAS, 498, 2575
Yu Y., Si X., Hu C., Zhang J., 2019, Neural Comput, 31, 1235
Zhang K., Bloom J. S., 2021, MNRAS, 505, 515
Zhu L., Yu F. R., Wang Y., Ning B., Tang T., 2019, IEEE Transactions on Intelligent Transportation Systems, 20, 383

APPENDIX A: ARCHITECTURE OF LSTM–VAE NETWORK

Fig. A1 shows a visualization of the proposed LSTM–VAE architecture. The purpose of each layer is as listed below:

(i) `Encoder_input` creates an instance of a tensor with dimensions of the model input, i.e. a sequence of 128 values.

(ii) `Encoder_LSTM` is a layer of CuDNNLSTM cells. CUDA Deep Neural Network (CuDNN) was developed by NVIDIA (Chetlur et al. 2014). The library accelerates training of neural networks using a GPU. This layer consists of 1024 such LSTM cells, which are not interconnected, but perform recurrent computation on the input sequence, one data point at a time. Output from every point of the sequence is stored within the state of the cell and used as input of the next computation of the sequence. Output of this layer consists of the final state of the cells, produced after the entire sequence has been processed. LSTM cells are trained to extract informative variables from the data through the process of backpropagation of errors. Increasing the number of LSTM cells tends to improve network's reconstruction loss (see below for the definition of reconstruction loss), and the number of 1024 is selected due to the GPU memory size constrain.

(iii) `Latent_mean` and `Latent_log_variance` are two separate layers of fully interconnected neurons. Their purpose is to perform the dimensionality reduction of the 1024 variables extracted by the `Encoder_LSTM` layer. `Latent_mean` and `Latent_log_variance` each output 20 values (which is the dimensionality of the latent space). First set of 20 values is used as the mean of the continuous latent variables, whereas the other set encodes their spread. In other words, `Encoder_LSTM`, `Latent_mean`,

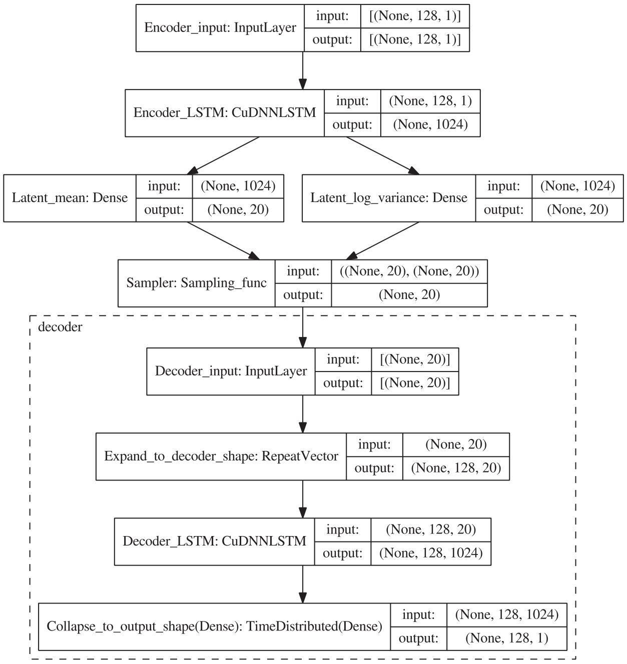

Figure A1. Architecture of the proposed LSTM–VAE model. Figure was generated using the KERAS utility `plot_model`. Left-most cell of each block contains the label assigned to corresponding instance of a KERAS object, followed by class of KERAS model layer (except from `Sampling_func` that is a custom function; see Appendix A). Right-most cells contain shapes of input and output tensors of the objects. Shapes are presented using the convention followed by KERAS.

and Latent_log_variance make up the Encoder block of the VAE, which maps the network input to the mean and (log) variance vectors. Increasing the number of latent variables tends to improve network's reconstruction loss, but it also increases the dimensionality of the latent space and the computational complexity of downstream processes. We choose the number of latent variables based on the results of preliminary experiments, where increasing the number tended to improve the reconstruction quality, and the number of 20 reached the compromise between the reconstruction performance and the complexity of the latent space.

(iv) Sampler generates random numbers from normal distributions whose parameters are set to the values of latent variable mean and variance. It is required in order to allow for deterministic treatment of the inherently probabilistic network during training (the so-called 're-parametrization trick'; Kingma & Welling 2014).

(v) Decoder_input initializes the input tensor of the Decoder block of the model.

(vi) Expand_to_decoder_shape replicates the values of latent variables to create sequences of the same length as the initial light-curve sequences. In other words, each LSTM cell of the Decoder block receives values of the 20 latent variables at each iteration of the sequential computation.

(vii) Decoder_LSTM layer is the Decoder counterpart of Encoder_LSTM layer of the Encoder. It also performs recurrent computation on the sequential input, but rather than processing a single sequence of variable values, it processes 20 sequences of constant values. This layer has 1024 cells, each producing a sequence of cell states from each iteration of recurrent computation.

(viii) Collapse_to_output_shape is a fully connected layer which combines the 1024 sequences from the LSTM layer into a single sequence of 128 data points. Output of this layer is the reconstruction of an input light-curve segment.

APPENDIX B: TRAINING AND FINE TUNING OF LSTM–VAE NETWORKS

We train networks using the KERAS implementation of Adam optimisation algorithm (Kingma & Ba 2015), and fine-tune them using the stochastic gradient descend (SGD) algorithm. The clipvalue argument is set to 0.5 for both optimizers, which prevents numerical errors due to exploding gradients. Training is performed with the batch size of 1024 (number of light-curve segments propagated through the network simultaneously) for various numbers of epochs (i.e. complete passes through the training set). Training is terminated when the validation loss value does not improve for 50 consecutive epochs. See Table B1 for full training history, and Fig. B1 for loss curves resulting from the training. Results suggest that satisfactory training results could be achieved using Adam optimizer only because improvements caused by SGD with decaying learning rate are marginal.

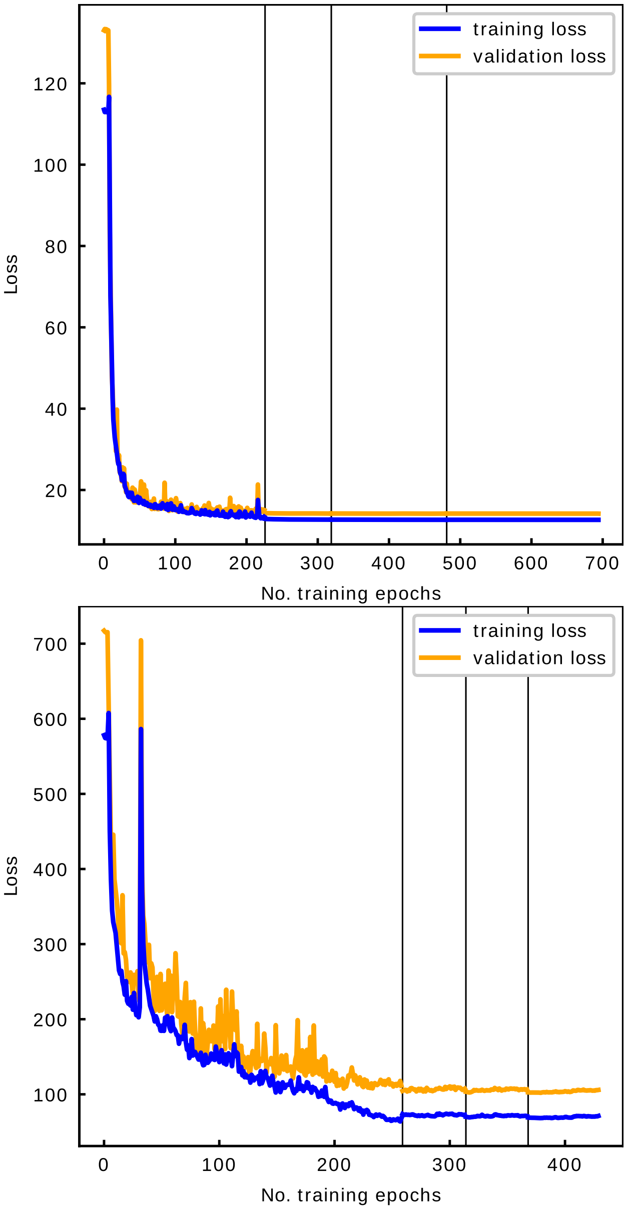

Figure B1. Loss curves resulting from the training of LSTM–VAE models on 1 s data set (top) and on the 4 s data set (bottom). The black vertical lines indicate that no improvement in validation loss in the last 50 consecutive epochs and change of the optimizer. See Table B1 for full training history.

Table B1. Summary of the LSTM–VAE training and fine tuning. Validation loss stopped improving after the quoted number of training epochs. Training was stopped after 50 consecutive epochs without validation loss improvement. Best validation loss is shown.

Optimizer (rate)	# Epochs	Loss	Data set
Adam (Default)	176	14.44	1 s
SGD (3×10^{-4})	43	14.20	1 s
SGD (1.5×10^{-4})	112	14.17	1 s
SGD (7.5×10^{-5})	164	14.16	1 s
Adam (Default)	209	107.63	4 s
SGD (3×10^{-4})	5	103.67	4 s
SGD (1.5×10^{-4})	4	102.55	4 s
SGD (7.5×10^{-5})	12	102.02	4 s

We use an Nvidia Geforce Titan Xp GPU for network training. One epoch of training takes ∼266 s of computation, resulting in the total of ∼36.6 and ∼17.0 h of training for 1 and 4s data sets, respectively, which could be reduced to ∼13.0 and ∼15.4 h, respectively, if fine-tuning with SGD optimizer was not performed.

The amount of GPU memory (12 GB for Titan Xp) sets a limit on the number of parameters of the network for a given size of data batch. The number of parameters depends on the size of propagated tensors, which in turn depends on the sizes of layers of the model (see shapes of the KERAS objects shown in Fig. A1). In order to optimize the reconstruction performance within the constraints of the GPU memory limit, we maximize the number of LSTM cells, which increases the computational power of the network. Increasing the number of LSTM cells relative to the number of data points in the input layer (i.e. the segment length) tends to improve reconstruction performance. However, a decrease in the number of input data points means that less information about the context of variability patterns is given to the network, which shifts the focus of the analysis to short term patterns. The size of the training data batch could be reduced to allow for an increased number of parameters of the network, leading to improved performance at the expense of increased training time.

APPENDIX C: LSTM–VAE RECONSTRUCTIONS OF 4S DATA SET SEGMENTS

See Fig. C1 for example reconstructions of light-curve segments from 4 s data set.

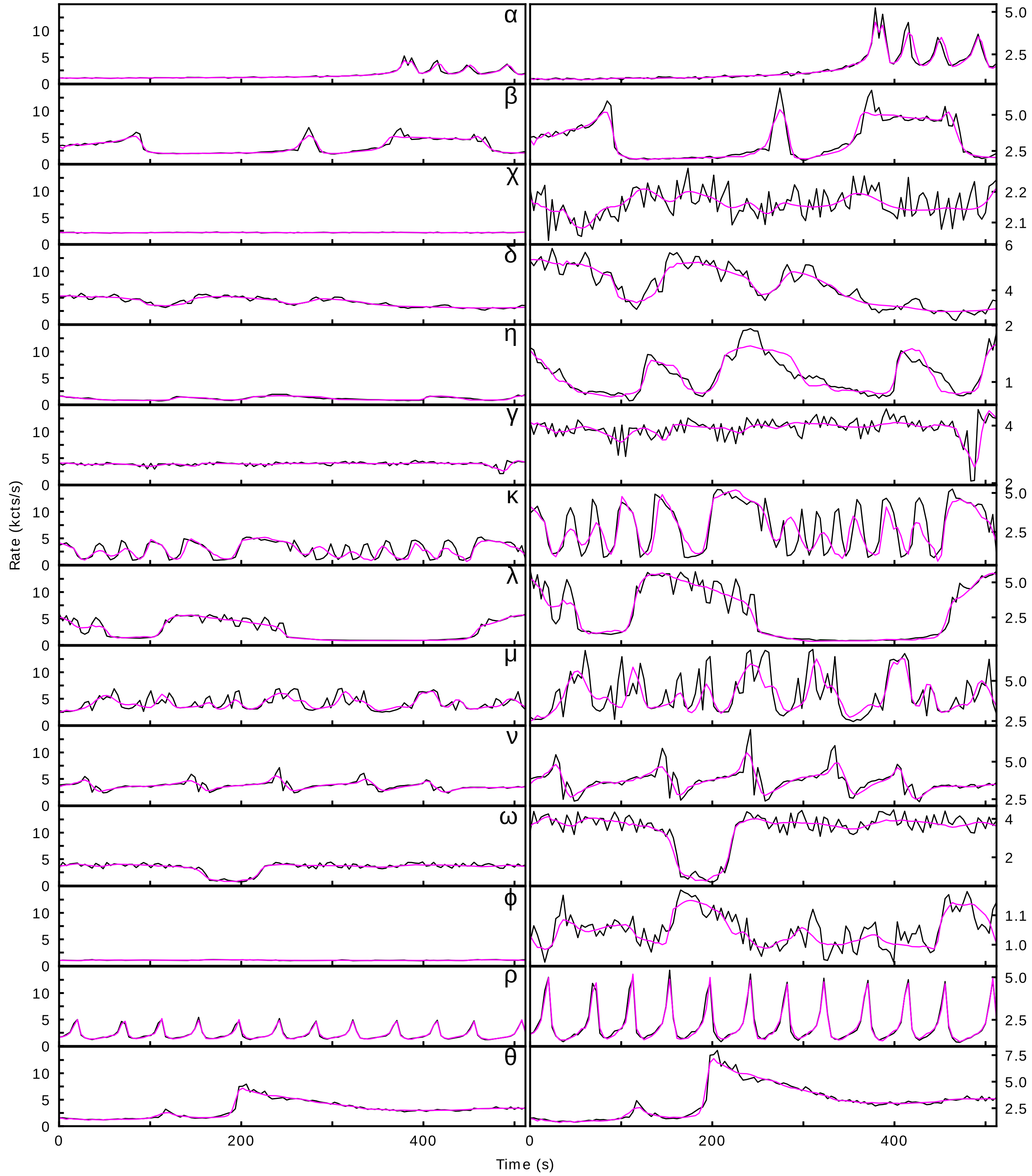

Figure C1. Examples of light-curve segments and their LSTM–VAE reconstructions – the counterpart of Fig. 2 for 4 s data set.

APPENDIX D: GAUSSIAN MIXTURE MODEL

GMM uses the expectation-maximization (Dempster, Laird & Rubin 1977) algorithm to approximate the probability distribution of the data using a set of multidimensional Gaussian components. The number of components is a hyper-parameter set by the user, and the mean position of each component is initiated randomly. Position and co-variance matrix of each component are then iteratively optimized to maximize the likelihood of the data under the model. GMM is a 'soft' clustering method; likelihood value of each data point is calculated for each Gaussian component, and the data points are assigned to components which give the largest likelihood.

The grid search for the optimal number of components of the GMM is the most computationally expensive procedure described in this paper, barring the training of neural networks. The convergence time for this algorithm is highly dependent on the initial random state, but as an approximate point of reference, we inform that the fitting of the 222 component model to 1s data set and 279 component model to 4 s data set require 1.96 and 2.91 CPU hours, respectively, using a CPU with clock speed of 3.40 GHz. Additionally, the fitting of 400 component models to 1 and 4 s data sets require 3.42 and 3.80 CPU hours, respectively.

We use the default set of GMM hyper-parameters as implemented in the SCIKIT-LEARN library (Pedregosa et al. 2011), with the exception of the number of components. We therefore fit full covariance matrices for each one of the Gaussian components. The computation time could be reduced at the expense of fit quality if different type of covariance matrix was chosen. Convergence threshold and the number of iterations of the expectation-maximization algorithm are other tunable parameters.

APPENDIX E: RANDOM FOREST CLASSIFIER

The random forest classifier (Breiman 2001) uses an ensemble of decision trees to perform classification, and it averages the results across the trees to find the final classification. Each decision tree is trained on a sub-sample of observations, picked at random from the training data set using bootstrapping. Additionally, only a subset of features is used to train each decision tree, which is another source of randomness impacting the construction of the classifier. This randomness allows random forest to reduce overfitting and the overall variance of the classifier, which leads to improved classification performance.

The SCIKIT-LEARN (Pedregosa et al. 2011) implementation of the random forest classifier controls the number of decision trees in the random forest ensemble using the n_estimators hyper-parameter. Increasing the number of trees tends to reduce the variance of predictions, as the forest converges on the answer, but at the expense of increased computation time. Therefore, we keep this number constant for all the grid-search classifiers, and set the hyper-parameter to 1000 for the final classifier used on the testing data subset.

Hyper-parameters min_samples_split and min_samples_leaf control the construction process of the decision trees, and we keep the values constant at 2 and 1, respectively. These hyper-parameters control the splitting of the trees, and increasing the values would not allow for the classes with just a single observation to be separated from other classes. We set the class_weight hyper-parameter to 'balanced', which sets the weights of the classes to be inversely proportional to the class frequency in order to account for the imbalance in the number of observations of each class.

Hyper-parameter criterion is used to choose between the functions that control the quality of a split in a tree, and the possible values of criterion are 'gini' and 'entropy', which stand for Gini impurity and information gain, respectively. Gini Index impurity criterion is defined as

$$\mathrm{I}_G = 1 - \sum_{k=1}^{K} p_k^2, \tag{E1}$$

where K is the number of classes and p_k is the proportion of observations of class k in a set. Information gain is the difference between the entropy of a set before and after a split. Entropy is defined as

$$\mathrm{E} = -\sum_{k=1}^{K} p_k \log(p_k), \tag{E2}$$

where K is the number of classes and p_k is the proportion of observations of class k in a set.

Any hyper-parameters of the random forest classifier which are not mentioned in the text were set to default values.

APPENDIX F: CLASSIFICATION TEST WITH ALL AVAILABLE DATA

Fig. F1 shows a summary of the classification results for the 100 tests where 69 test observations are randomly sampled in a stratified manner from the entire data set of 206 labelled observations (combined training, validation, and test data subsets), and the remaining observations are used for training (as described in Section 4.3). The mean of weighted F1 and accuracy performance scores of those tests are 0.877 ± 0.031 and 0.896 ± 0.026, respectively. Observations of the majority of the classes are classified in agreement

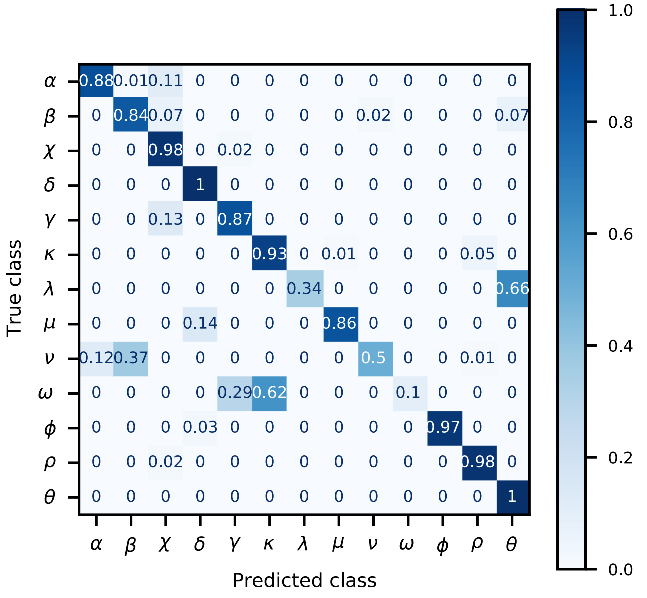

Figure F1. A row-wise normalized confusion matrix showing classification results of 100 test trials where 69 test observations are randomly sampled in a stratified manner from all of the available 206 labelled observations, as described in Section 4.3. Rows of the matrix correspond to human-assigned classes (assumed true class), columns correspond to the classifications predicted by the classifier, and the values in the matrix represent the number of observations. The numbers in each row are divided by the sum of numbers in that row.

with the human-assigned classifications >80 per cent of the time, however, observations of classes λ, ν, and ω are usually assigned to other classes. These classes are affected by very small sample size (each has only one training observation), which limits the statistical significance of their classification results.

APPENDIX G: HIERARCHICAL CLUSTERING

Hierarchical (agglomerative) clustering algorithm creates a hierarchy of clustering solutions through recursive merging of pairs of clusters. The merging algorithm starts with the value of the distance threshold set to zero, resulting in the assignment of each data point to singleton clusters. The threshold is then increased, and pairs of nearby clusters are merged when the distance between them is less than the threshold. The algorithm finishes when all data points belong to a single cluster, or when a threshold set by the user is reached (see Rokach & Maimon 2005 for more details).

The hierarchical clustering algorithm can use a variety of distance metrics and different methods of calculating distance between clusters. We use the Euclidean metric and 'ward' method, which uses the Ward variance minimization algorithm and creates clusters of fairly regular sizes. The choice of clustering method affects the resulting cluster structure, for example 'single' defines the distance between two clusters as the distance between the nearest data points of the two clusters, and it tends to build clusters with very uneven sizes, whereas 'complete' method uses the distance between two furthest data points, and it tends to create more even, compact clusters.

This paper has been typeset from a TEX/LATEX file prepared by the author.