BRIDGING THE SIM-TO-REAL GAP WITH A ROBUST DEEP LEARNING FRAMEWORK FOR EXTRACTING RARE TRANSIENT ANOMALIES IN COMPLEX STOCHASTIC TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting rare, morphologically diverse transient anomalies in data-scarce, highly stochastic time series remains a fundamental challenge in representation learning. Real-world temporal data often suffers from extreme class imbalance, variable sequence lengths, and significant domain shifts when relying on simulated training data. To address these challenges, we introduce a novel framework that leverages a Conditional Variational Autoencoder (CVAE) paired with a Power Spectral Density (PSD)-matched colored-noise padding strategy. This approach effectively bridges the simulation-to-real domain gap by synthesizing robust training representations that preserve underlying spectral characteristics. Furthermore, by treating temporal sequences as 1-dimensional images, our architecture employs a Convolutional Neural Network (CNN) to extract invariant features across multiple temporal windows via targeted trending and detrending mechanisms. We validate our framework on a highly complex real-world testbed: detecting anomalous X-ray time series events from Active Galactic Nuclei (AGN) driven by supermassive black holes. Our results demonstrate that this representation engineering approach significantly improves anomaly detection performance and generalization on highly stochastic datasets compared to standard sequence modeling baselines.

1 INTRODUCTION

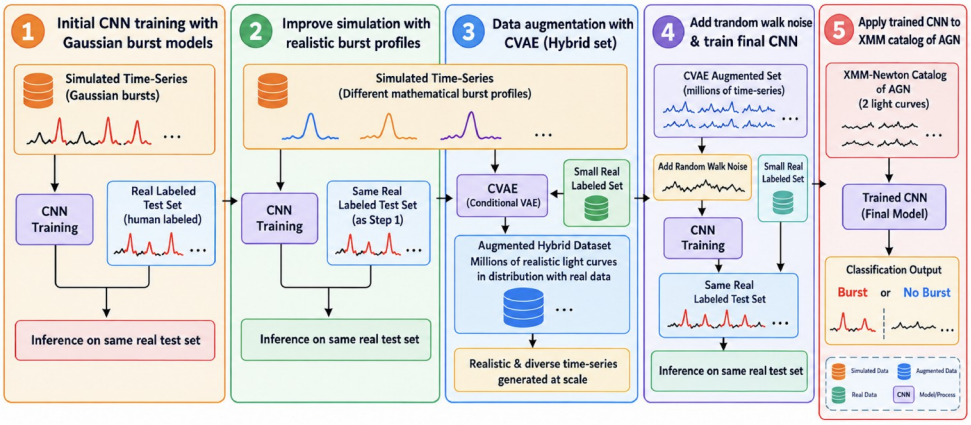
Time-series anomaly detection (Jia et al., 2025; Shaukat et al., 2021; Blázquez-García et al., 2021; Muthukrishna et al., 2022) is extremely challenging in instances where rare transient events are embedded in stochastic backgrounds and labeled anomalies are scarce. In such situations, anomalies may exhibit widely varying durations, amplitudes, and morphologies, while observations themselves may have different sequence lengths and signal strength. These properties limit conventional threshold-, template-, and parametric-model-based approaches and pose important challenges for real-world time-series anomaly detection (Shaukat et al., 2021; Kim et al., 2021; Liu et al., 2022; Pang et al., 2019). In this work we address the problem of detecting rare, morphologically diverse transient events in noisy/non-stationary, data-limited time series, without assuming any predefined analytical anomaly profile.

We study this problem using X-ray time-series signals of Active Galactic Nuclei (AGN), which provide a particularly demanding real-world benchmark. AGN X-ray emission originates primarily from a compact, hot corona surrounding an accreting supermassive black hole and exhibits strong stochastic variability (Uttley & McHardy, 2005; Burke et al., 2021; Laha et al., 2024). Occasionally, these time-series signals contain short transient bursts that may trace rapid coronal energy-release processes such as magnetic reconnection (Kammoun et al., 2024; Di Matteo, 1998). Their durations, energetics, and spectral evolution can therefore provide important constraints on how energy is pumped into, and dissipated in black-hole coronae. Systematically identifying such events, however, is difficult because the data simultaneously exhibit (a) few labeled anomalies (Hundman et al., 2018; Pang et al., 2019), (b) stochastic and non-stationary backgrounds (Richman & Moorman, 2000; Hong & Fan, 2016; Kim et al., 2021; Liu et al., 2022), (c) variable sequence lengths (Bag-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

nall et al., 2018; Tan et al., 2019) and (d) widely varying signal strength, and (e) rare events with heterogeneous morphologies.

Figure 1: The flowchart for pipeline of the work



We introduce a framework (see Figure 1) that combines power-spectral-density (PSD)-matched colored-noise padding, conditional variational autoencoder (CVAE)-based data augmentation (Torralba et al., 2024; Diederik & Max, 2019; Kingma & Welling, 2013), and convolutional neural networks (CNNs) (LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan & Zisserman, 2014) to address these challenges. PSD-matched padding standardizes variable-length observations while preserving their stochastic variability. On the other hand, CVAE augmentation expands the limited training distribution with diverse burst morphologies. The CNN then learns to distinguish time series having transient bursts from background variability without requiring a fixed burst template. We systematically benchmark this framework through increasingly realistic training strategies: Gaussian bursts, diverse analytical profiles, CVAE-augmented signals, and multi-scale representations and demonstrate a progressive improvement in the ability of the CNN to generalize to expert-labeled real events.

Importantly, our objective is candidate discovery rather than automated scientific classification. Distinguishing genuine astrophysical bursts from extreme stochastic fluctuations ultimately requires expert interpretation through follow-up timing and spectral analyses. The detector instead provides a scalable mechanism for identifying statistically interesting events from large archives and prioritizing them for expert investigation. We apply the resulting framework to ~ 1400 *XMM-Newton* AGN time-series signals.

2 DATASETS

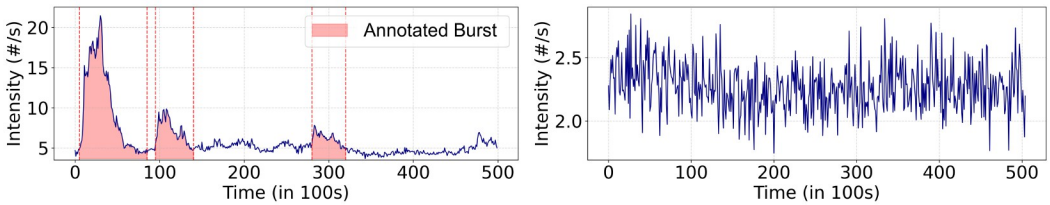


Figure 2: Two time-series from the WAX-XMM test set, with human annotated stochastic bursts and their positions (left) and without stochastic bursts (right).

We use *XMM-Newton* EPIC-pn (Strüder, L. et al., 2001) X-ray time-series because of their high signal-to-noise ratio and long, nearly uninterrupted exposures ($\sim$ 10–120 ks, where ks refers to kilo-seconds). These continuous observations are particularly suited for detecting short transient variability without observational gaps. We use three independent AGN samples:

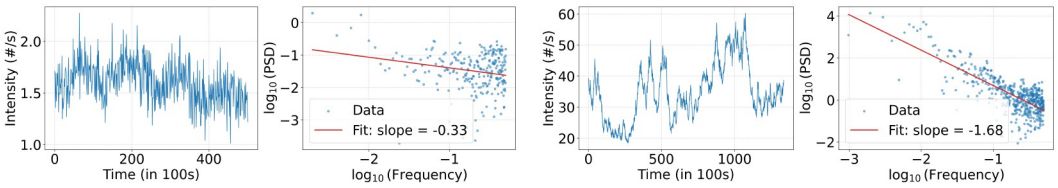


Figure 3: Two time-series signals from the WAX-XMM test set without stochastic bursts (left) and with stochastic bursts (right) and their corresponding power spectral density (PSD) plots.

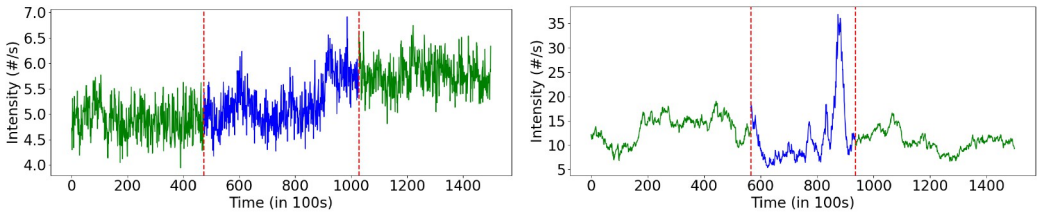


Figure 4: Two time-series from the WAX-XMM test set (without stochastic bursts on the left and with stochastic bursts on the right) which were extended using matched colored noise. Note that the green part of each time-series is pure noise generated based on the alpha value from the corresponding blue part of the signal.

(1) WAX-XMM (Laha et al., 2014): 117 time-series signals from 26 AGN, comprising 53 candidate-burst and 64 non-burst examples. This is our primary test set.

(2) BAT-AGN (Ricci et al., 2017): 152 time-series signals, equally divided into 76 candidate-burst and 76 non-burst examples, used for hybrid injection, CVAE augmentation, and CVAE augmented CNN training.

(3) 4XMM-DR14 Filtered Catalogue (Webb et al., 2020): ~ 1400 time-series signals used for the final large-scale candidate search.

The samples were cross-matched to remove overlapping sources and observations, ensuring independent training, testing, and application datasets. We use time-series signals generated by the standard *XMM-Newton* Pipeline Processing System (PPS) (Strüder, L. et al., 2001), uniformly binned at 100 s to retain short-timescale variability while maintaining sufficient signal-to-noise. Each time-series signals is represented by consecutive 100-s time bins and their corresponding X-ray count rates.

Candidate stochastic bursts lack a unique analytical profile and occur against strongly variable backgrounds. We therefore construct expert-labeled training and test sets through visual inspection. Candidate events are localized enhancements, typically reaching $\gtrsim 1.5$ times the preceding local flux and subsequently returning toward the local background. Importantly, the 1.5 factor is a labeling guideline, not a detection threshold; classification additionally depends on event morphology and the surrounding stochastic variability. These labels identify candidate bursts rather than establish their astrophysical origin. Distinguishing genuine events from extreme stochastic fluctuations requires follow-up timing and spectral analyses.

3 Approach and Methods

3.1 Power Spectral Density and PSD-matched colored-noise padding

We characterize the stochastic variability of each time-series signal through its power spectral density (PSD),

$$S(f) \propto f^{-\alpha}, \tag{1}$$

where α is the PSD slope. Candidate-burst time-series signals generally exhibit stronger low-frequency power and steeper PSDs (see Figure 3). We use the measured α of each time-series

signals to generate source-specific colored noise that approximately preserves its stochastic variability. See Figure 5 for a distribution of α for time-series signals with bursts (red) and without bursts (blue).

The observed time-series span $\sim$ 100–1400 time bins, while our CNN uses fixed-length inputs. Conventional padding with constants, repeated segments, or interpolation (Dang et al., 2024a; Bier et al., 2022; Bignoumba et al., 2024; Dang et al., 2024b) can alter the statistical and physical properties of the signal. We therefore introduce PSD-matched colored-noise padding to standardize sequence length while preserving the stochastic background which is physically compatible for that time-series signal. For each time-series, colored noise is generated from its measured PSD slope α and appended to the ends of the observation until a uniform length of 1500 time bins is reached. The original signal and class label remain unchanged. The procedure is summarized in Algorithm 1 and illustrated in Fig. 4.

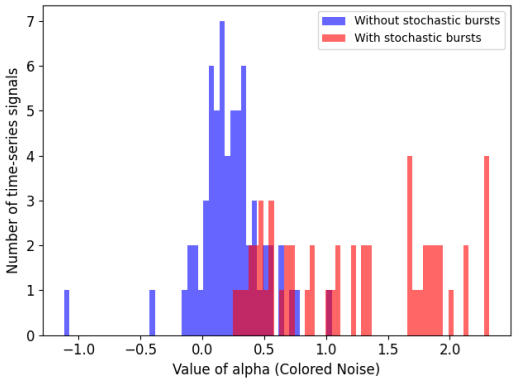


Figure 5: The distribution of alpha (Colored Noise) of the 117 time-series signals from the WAX test set

3.2 Simulations of X-ray AGN time-series signals with Gaussian Stochastic Bursts

Since we have only a few hundred real X-ray time-series signals to train and test, we take a phased sim-to-real approach to detect bursts in the time-series. We start off with basic simulated time-series for training and testing the CNN (George & Huerta, 2018; Gabbard et al., 2018; Kang et al., 2023).

The noise in an AGN X-ray time-series signal is not actually a noise, but a stochastically produced continuous emission from the high energy corona (Uttley & McHardy, 2001; Uttley et al., 2005; Nandra, 2001; Allevato et al., 2013; Kasliwal et al., 2017; O'Neill et al., 2005), which mimics random walk noise with varying amplitude (Vaughan et al., 2003; Lefkir et al., 2025). For our purpose of detecting the bursts, we refer to it as noise because it is the background emission on which the bursts appear (Timmer & Koenig, 1995; Vaughan, 2005; Uttley et al., 2005). To model the AGN background continuum, we superimpose three noise components: uniform noise ($\mathcal{U}[0.001, 0.1]$/median), zero-mean Gaussian noise ($\sigma = 0.1$), and random-walk noise (cumulative sum of $\mathcal{N}(0, 0.1)$ offset by 1). The random walk simulates the continuous, stochastically driven coronal emission characteristic of AGN time-series signals.

Algorithm 1: Extending a Time-series signal with Matched Colored Noise

Input: Original Signal, Target Length
Output: Extended Signal
if Target Length < Original signal length **then**
  raise ValueError
$\alpha \leftarrow$ `EstimateNoiseColor` (Signal)
calculate Extension lengths ($leftlength$, $rightlength$)
$leftnoise \leftarrow$ `gencolorednoise` (α, $leftlength$)
$rightnoise \leftarrow$ `gencolorednoise` (α, $rightlength$)
extract left and right signal edge windows
calculate mean, variance of both edge windows
update Generated noise signals $leftnoise$, $rightnoise$ with extracted mean, variance
concatenate Original Signal with updated noise signals
return Extended Signal

To model the stochastic bursts, we created a signal (on top of the base continuum emission) consisting of one or more Gaussian profiles (McLaughlin et al., 2024; McLaughlin, 2025) at random time stamps. The Gaussian profiles were tweaked using its two parameters: Normalization and σ, and selection of these values were randomized.

For the training set, we created an equal number of time-series signals with stochastic bursts (positive) and without stochastic bursts (negative) in batches. The chosen batch-size was 64, and each simulated time-series signal had a time-stamp size of 1500. Both time-series signals had all three

types of noise signals superimposed. We superimposed the Gaussian profiles on the noise for the positive time-series. (See Section A in the Supplementary Materials section for more details.)

3.3 Simulations of X-ray AGN time-series signals with Analytical Stochastic Bursts

Although the CNN model trained with the simulated time-series signals (with stochastic bursts modeled by Gaussian profiles) could correctly identify several time-series signals with stochastic bursts, it fails miserably when the stochastic burst profiles do not appear like symmetric Gaussians. The main types of real time-series signals that were not detected by the trained CNN models were identified and described below: **Oscillatory stochastic bursts** : containing stochastic bursts that have some periodicity in emission, and on top of the periodic fluctuations there are smaller emission profiles (Kasliwal et al., 2017; Moreno et al., 2018). **Asymmetric stochastic bursts** : having stochastic bursts that depict a steep rise and gradual fall, and hence are asymmetric (Stancik & Brauns, 2008). **Superimposed stochastic bursts** : consisting of two types: (a) where the smaller stochastic bursts sit on top of a larger emission profile, and (b) one stochastic burst has started before another has ended hence giving a complex shape (Terrell Jr, 1972). **Very Narrow or spiked stochastic bursts** : (Uttley & McHardy, 2001; Green et al., 1999). **Broad or Very Broad stochastic bursts** : when the entire stochastic burst is not captured within one observational window (Kelly et al., 2014). See Section D in Supplementary Materials for one example from each of these categories.

We chose typical mathematical functions that can best describe stochastic bursts that are misclassified the CNN trained model, which are described in detail below (Czerny & Lehto, 1997; Uttley & McHardy, 2001; Vaughan et al., 2003; Souza et al., 2026).

Asymmetric gaussian : A gaussian with two different sigmas, one for the rise and the other for the fall, this is able to model astrophysical stochastic bursts better which sometimes rise quickly and decay slowly, or vice-versa. We injected 2-5 such stochastic bursts in each positive time-series signal, with stochastic burst intensity multiplier 0.75, and width range 50 - 200.

Oscillatory Gaussian : A gaussian with a superimposed sine wave of lower intensity compared to that of the base symmetric gaussian, and much lower frequency (several orders of magnitude less) compared to the base symmetric gaussian. In each positive time-series signal, we injected a single such stochastic burst.

We chose two more modifications to the symmetric gaussian:

Very Broad Gaussian and **Very Narrow Gaussian** : A symmetric gaussian with a width range much larger than the original symmetric gaussian was chosen (200 - 600), and much lesser than the original symmetric gaussian (5 - 50) keeping the stochastic burst intensity multiplier fixed at 0.75. Each positive time-series had 2-5 of these.

We then used these four modifications to the symmetric gaussian stochastic burst (with stochastic burst intensity multiplier 0.75 and width range 50 - 200) in the simulated positive time-series signals to train the CNN in an effort to increase the recall of the CNN, i.e. the ability to identify the time-series signals with stochastic bursts correctly.

3.4 Data augmentation with Variational Auto-encoders

Since the training set operation regime is data-deprived, data augmentation is the next logical step.

Representation learning is known to map data to simple underlying distributions (Torralba et al., 2024). Autoencoder models can learn embeddings that can be decoded to reconstruct the input data. Variational Autoencoders (VAEs) turn an autoencoder into a proper generative model by regularizing the latent distribution, thereby compressing it into a Gaussian, and adding noise to the output of the encoder. This enables the sampling of the generative model and maximizes the likelihood of data in a formal probabilistic model (Torralba et al., 2024; Diederik & Max, 2019; Kingma & Welling, 2013).

A conditional VAE (cVAE) models the conditional probability distribution, hence allowing direct data generation by specifying a condition vector (e.g. a class label). The condition vector is concatenated directly with the input at the encoder and concatenated with the sampled latent vector at

the decoder. The targeted data are generated by sampling the latent vector, choosing a specific condition vector, concatenating them, and passing the result to the decoder to produce a sample of that specific target class (Sohn et al., 2015).

The overarching goal of using cVAEs is to create an in-distribution dataset similar to the real current test set time-series signals using which we can train the CNN models.

3.5 Training the CNN with CVAE-generated data

Once the CVAE's ability to generate time-series signals that are in-distribution to the real test set is established, we superimpose variable levels of Random Walk Noise into the CVAE generated time-series signals and then train the CNN with the these time-series signals.

For each type of CVAE model, there are four scenarios: (1) Training the CNN with univariate CVAE generated time-series signals only, (2) Training the CNN with multivariate CVAE generated time-series signals only, (3) Training the CNN with univariate hybrid (CVAE generated and real) time-series signals, and (4) Training the CNN with multivariate hybrid (CVAE generated and real) time-series signals. For each of these cases, the CVAE can be trained with simulated time-series signals only, or with hybrid (simulated and real) time-series signals. Hence, we effectively have eight scenarios.

We train every CNN 20 times for a given hyperparameter value and report the average metrics with error bars to account for the stochasticity of our randomly simulated and generated datasets and to demonstrate the reproducibility of our experiments.

3.6 Creating Multi-variate time-series signals: Original, Trending, De-trending, and Gradient.

We want to explore the variability at different time-scales. For smaller mass black holes the X-ray outbursts happen at shorter time scales and hence de-trending the time-series signal with a broader time-profile help us extract the narrow emission features. We also add the 'gradient' of the time-series signal for a few specific time-windows as a feature (which computes the rate of change of the low-frequency, emphasizing the presence of outbursts).

In Figure 6, all twelve features have been shown on a time-series signal with stochastic bursts from the WAX-XMM (real) test set - the original and five levels of trended (with window sizes $1/2$, $1/4$, $1/8$, $1/16$, and $1/12$ the length of the signal) in the first column (subfigures 1, 2A - 6A), and gradient on level two of trended, and five levels of detrended (corresponding to the five levels of trended with the same window sizes) in the second column (subfigures 7, 2B - 6B).

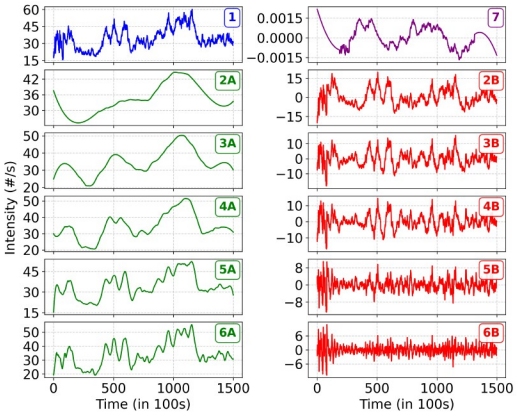


Figure 6: A time-series signal with stochastic bursts from the WAX-XMM test set with its twelve features - refer to Section 3.6 for descriptions

4 Results and Analysis

We trained the CNN model on the synthetic time-series signals described above with specific values of the parameters described below.

4.1 Variable Gaussian normalization and Width

We first evaluated CNN performance on synthetic signals featuring symmetric Gaussian bursts across varying intensity multipliers and width ranges (σ). Fixing the width range to 50–200 time

Algorithm 2: The cVAE augmented CNN training pipeline

Input: CNN model, cVAE model, hyperparameters of analytical stochastic bursts
Output: trained CNN model(s)

```
for range of training_volume, random_walk_noise multiplier values do
    train cVAE with simulated time-series signals
    train cVAE with hybrid (simulated + real) time-series signals
for cVAE trained on simulated, hybrid time-series signals do
    select cVAE with best generated time-series signals based on t-SNE and U-MAP plots
for univariate, multivariate representations do
    for range of random_walk_noise multiplier values do
        for best trained cVAE on simulated, hybrid time-series signals do
            train CNN with cVAE generated time-series signals
            train CNN with hybrid (cVAE generated + real) time-series signals
select CNNs based on performance on WAX test set with best F − 1, F − 2, F − 4 Scores, Recall
return selected trained CNN model(s)
```

bins, the model achieved optimal recall (0.652) at an intensity multiplier of 0.75 (row 1 of Table 1). At this threshold, burst amplitudes remain close to the background noise floor, forcing the network to learn subtle morphological discriminators rather than relying on high signal-to-noise ratios. Broadening the width range (10–300) slightly improved isolated Gaussian metrics. However, standardizing on 50–200 proved optimal as subsequent analytical extensions explicitly account for extreme broad and narrow profiles. (See Tables 1 and 2 in the Supplementary Materials Section for details.)

4.2 Analytical Stochastic Burst Profiles

To address false negatives arising from non-Gaussian morphologies, we incorporated four analytical burst variants into training: (1) **Asymmetric Gaussians** (rising width 5–150, falling width 150–300), (2) **Oscillatory Gaussians** (superimposed sine wave with amplitude $f/5$–$f/2$ and frequency $w/30000$–$w/8000$), (3) **Very Broad Gaussians** (width 200–600), and (4) **Very Narrow Spikes** (width 5–50). (See Tables 3 - 6 in the Supplementary Materials Section for details.)

Training a single CNN on a balanced mixture of all five burst categories improved overall test accuracy on WAX-XMM from 84.6% to 88.6% and boosted recall from 0.652 to 0.752 (See row 2 of Table 1), confirming that incorporating heterogeneous morphological priors significantly enhances real-world generalization.

4.3 Data Augmentation with CVAE

To bridge the simulation-to-real domain gap, we trained a Conditional VAE (CVAE) under two regime configurations: (1) purely simulated time-series signals, and (2) hybrid datasets combining simulated signals with labeled real AGN time-series signals (Figure 7).

We observed that the presence of random-walk noise in simulated training data confused the CVAE, causing noise fluctuations to mimic genuine stochastic bursts. To prevent latent space contamination, we removed random-walk noise during CVAE training and manually re-injected it post-generation prior to CNN downstream training.

Evaluations across parameter spaces revealed key trade-offs (See Table 7 in the Supplementary Materials section for details):

Random-Walk Noise Multiplier: Decreasing the noise multiplier from 0.3 down to 0 significantly improved class separation between burst and non-burst signals during CVAE generation.

Training Volume & Synthetic vs. Hybrid: Increasing training volume from 200k to 1.6M enhanced class separation. However, while purely synthetic CVAE models required very large training volumes to achieve separation, hybrid CVAE models achieved strong class separation at significantly

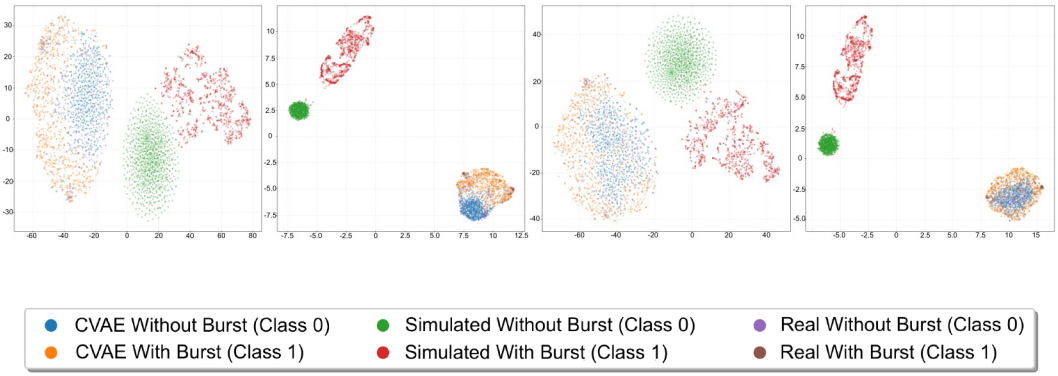


Figure 7: t-SNE and U-MAP visualizations of the CVAE generated time-series compared to their simulated and real counterparts when the CVAE was trained with a simulated dataset only (left) and when the CVAE was trained with a hybrid dataset (right); the legend for the colors on the plot (bottom) - each colored dot in the plots represents a single time-series signal.

Table 1: A comparative tabular summary of the model hyperparameters and performances on the WAX test set at different stages of the pipeline

Model Configuration	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
CNN + Gaussian Stochastic Bursts	0.8462 ± 0.0261	0.9838 ± 0.0252	0.6520 ± 0.0682	0.7816 ± 0.0459	0.6980 ± 0.0613	0.6648 ± 0.0664
CNN + Analytical Stochastic Bursts	0.8855 ± 0.0116	0.9742 ± 0.0115	0.7520 ± 0.0248	0.8486 ± 0.0170	0.7878 ± 0.0220	0.7622 ± 0.0241
CNN + Analytical Stochastic Bursts + CVAE Augmentation	0.7966 ± 0.0595	0.7068 ± 0.0697	0.9651 ± 0.0161	0.8138 ± 0.0444	0.8975 ± 0.0215	0.9440 ± 0.0141
CNN + Analytical Stochastic Bursts + CVAE Augmentation + Multi-Scale Representation	0.8201 ± 0.0455	0.7294 ± 0.0591	0.9745 ± 0.0225	0.8324 ± 0.0340	0.9116 ± 0.0159	0.9550 ± 0.0177

lower training volumes because real time-series signals injections helped the network learn authentic background distributions much faster.

4.4 Training the CNN with CVAE generated (and real) time-series signals

As shown in Table 1, augmenting the CNN training set with hybrid CVAE-generated signals dramatically improved test recall on WAX-XMM from 0.752 to 0.974. Incorporating multi-scale time-series representations (trending, detrending, and gradient channels) further refined performance, yielding the highest overall F_2 (0.912), and F_4 (0.955) scores.

While synthetic-only training yields high precision at the expense of recall, CVAE-augmented multi-scale training prioritizes recall. This trade-off is ideal for candidate discovery, where missing genuine astrophysical bursts carries a higher scientific cost than evaluating false positives.

4.5 The advantage of multivariate time-series signals

Tables 8 and 9 and Tables 12 and 13 from the Supplementary Materials Section show the results of training the CNN with CVAE generated time-series and with hybrid time-series when the CVAE was trained with simulated time-series and hybrid time-series respectively, and only univariate representations of the CVAE generated time-series were used to train the CNN. Comparing this to Tables 10 and 11 and Tables 14 and 15 shows the improvement in performance of the trained CNN when multivariate time-series were used to train the CNN keeping the other parameters fixed.

5 APPLYING THE TRAINED CNN MODEL ON 1400 AGN TIME-SERIES SIGNALS FROM *XMM-Newton*

We applied our best-performing pipeline (CNN trained with CVAE-augmented multi-scale representations) to evaluate 1,363 filtered time-series signals from the 4XMM-DR14 catalogue. Out of 1,363 candidate time-series signals, the 20 model ensemble identified 321 transient burst candidates ($\sim 23.5\%$) against 1,042 non-burst baseline signals ($\sim 76.5\%$), providing a prioritized sample for follow-up expert timing and spectral analysis. See Figures 3 an 4 in Supplementary Materials for a sample set of the time-series and their predictions.

6 DISCUSSION AND CONCLUSIONS

In this work, a pipeline based on CNNs and CVAEs trained on simulated and real time-series signals was developed to identify the presence of candidate stochastic bursts in X-ray AGN time-series signals. Initially, the CNN was trained with simulated time-series signals with gaussian stochastic bursts, and its performance was middling (poor recall). An improvement in performance (recall) was observed on employing analytical stochastic bursts in the simulated time-series signals. Training the CNN with a CVAE augmented dataset led to a further boost of the recall driven by the ability of the CVAE to generate realistic time-series signals (in-distribution to the real datasets), and incorporating multiscale representation achieved the best result (shown in row 4 of Table 1). The model ensemble with the best overall recall was then used to predict the presence of candidate stochastic bursts in the 4XMM-DR14 filtered catalogue sample.

Since the primary training mode of the pipeline is simulated data, it can be adapted to model anomalies of different sizes and shapes, occurring at different frequencies, and across different domains. X-ray AGN time-series are extremely noisy, and hence the method proposed has the potential to perform better in other domains with comparatively lesser background noise in the signals.

Beyond astrophysics, the proposed framework has potential applications wherever rare transient events must be detected within noisy and dynamically evolving temporal signals. Representative examples include early detection of spacecraft subsystem anomalies from telemetry streams, predictive maintenance through vibration monitoring of rotating machinery, seizure detection in electroencephalography (EEG), arrhythmia identification in electrocardiography (ECG), structural health monitoring of aircraft and bridges, earthquake precursor detection from seismic records, cyber-physical system monitoring, and financial anomaly detection in non-stationary markets. In each of these domains, the combination of limited labeled data, evolving backgrounds, heterogeneous anomaly morphology, and expert validation closely mirrors the challenges encountered in AGN X-ray variability analysis.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for literature review and search, creating the flowchart for the pipeline, and executing the research ideas into code (research ideation and execution). We have not used generative AI tools for drafting sections of the paper, generating synthetic datasets, and proving mathematical claims are not applicable to this work. We have reviewed all AI-assisted work. For example, we cross-verified LLM-generated literature reviews against Google scholar, and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

The link for the codebase can be found in the Supplementary Materials section. Public facing NASA datasets were used for the project and can be downloaded from the same codebase.

REFERENCES

V Allevato, Maurizio Paolillo, I Papadakis, and C Pinto. Measuring x-ray variability in faint/sparsely sampled active galactic nuclei. The Astrophysical Journal, 771(1):9, 2013.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. arXiv preprint arXiv:1811.00075, 2018.

Agnieszka Bier, Agnieszka Jastrzębska, and Paweł Olszewski. Variable-length multivariate time series classification using rocket: A case study of incident detection. IEEE Access, 10:95701–95715, 2022. doi: 10.1109/ACCESS.2022.3203523.

Nzamba Bignoumba, Sadok Ben Yahia, and Nedra Mellouli. Deep padding and alignment strategies for irregular multivariate clinical time series. Procedia Computer Science, 246:3275–3284, 2024. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2024.09.312. URL https://www.sciencedirect.com/science/article/pii/S1877050924023342. 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. ACM computing surveys (CSUR), 54(3):1–33, 2021.

Colin J Burke, Yue Shen, Omer Blaes, Charles F Gammie, Keith Horne, Yan-Fei Jiang, Xin Liu, Ian M McHardy, Christopher W Morgan, Simone Scaringi, et al. A characteristic optical variability time scale in astrophysical accretion disks. Science, 373(6556):789–792, 2021.

Bożena Czerny and Harry J Lehto. Testing the x-ray variability of active galactic nuclei with the non-linear prediction method. Monthly Notices of the Royal Astronomical Society, 285(2):365–373, 1997.

Yizhou Dang, Yuting Liu, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, and Jianzhe Zhao. Repeated padding for sequential recommendation. In Proceedings of the 18th ACM Conference on Recommender Systems, RecSys '24, pp. 497–506, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400705052. doi: 10.1145/3640457.3688110. URL https://doi.org/10.1145/3640457.3688110.

Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. Ticoserec: Augmenting data to uniform sequences by time intervals for effective recommendation. IEEE Trans. on Knowl. and Data Eng., 36(6):2686–2700, June 2024b. ISSN 1041-4347. doi: 10.1109/TKDE.2023.3324312. URL https://doi.org/10.1109/TKDE.2023.3324312.

T. Di Matteo. Magnetic reconnection: flares and coronal heating in active galactic nuclei. , 299(1): L15–l20, August 1998. doi: 10.1046/j.1365-8711.1998.01950.x.

P Kingma Diederik and Welling Max. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning, 12(4):307–392, 2019.

Hunter Gabbard, Michael Williams, Fergus Hayes, and Chris Messenger. Matching matched filtering with deep networks for gravitational-wave astronomy. Physical review letters, 120(14): 141103, 2018.

Daniel George and Eliu Antonio Huerta. Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data. Physics Letters B, 778:64–70, 2018.

AR Green, IM McHardy, and C Done. The discovery of non-linear x-ray variability in ngc 4051. Monthly Notices of the Royal Astronomical Society, 305(2):309–318, 1999.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. International Journal of Forecasting, 32(3):914–938, 2016.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 387–395, 2018.

Xudong Jia, Peng Xun, Wei Peng, Baokang Zhao, Haojie Li, and Chiran Shen. Deep anomaly detection for time series: A survey. Computer Science Review, 58:100787, 2025.

Elias Kammoun, Anne M Lohfink, Megan Masterson, Dan R Wilkins, Xiurui Zhao, Mislav Balokovic, Peter G Boorman, Riley Connors, Paolo Coppi, Andrew Fabian, et al. The high energy x-ray probe (hex-p): probing the physics of the x-ray corona in active galactic nuclei. Frontiers in Astronomy and Space Sciences, 10:1308056, 2024.

Zihan Kang, Yanxia Zhang, Jingyi Zhang, Changhua Li, Minzhi Kong, Yongheng Zhao, and Xue-Bing Wu. Periodic variable star classification with deep learning: Handling data imbalance in an ensemble augmentation way. Publications of the Astronomical Society of the Pacific, 135(1051): 094501, 2023.

Vishal P Kasliwal, Michael S Vogeley, and Gordon T Richards. Extracting information from agn variability. Monthly Notices of the Royal Astronomical Society, 470(3):3027–3048, 2017.

Brandon C Kelly, Andrew C Becker, Malgosia Sobolewska, Aneta Siemiginowska, and Phil Uttley. Flexible and scalable methods for quantifying stochastic variability in the era of massive time-domain astronomical data sets. The Astrophysical Journal, 788(1):33, 2014.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In International conference on learning representations, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

Sibasish Laha, Matteo Guainazzi, Gulab C. Dewangan, Susmita Chakravorty, and Ajit K. Kembhavi. Warm absorbers in X-rays (WAX), a comprehensive high-resolution grating spectral study of a sample of Seyfert galaxies - I. A global view and frequency of occurrence of warm absorbers. , 441(3):2613–2643, July 2014. doi: 10.1093/mnras/stu669.

Sibasish Laha, Claudio Ricci, John C. Mather, Ehud Behar, Luigi C. Gallo, Frederic Marin, Rostom Mbarek, and Amelia Hankla. X-ray properties of coronal emission in radio quiet active galactic nuclei, 2024. URL https://arxiv.org/abs/2412.11321.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Mehdy Lefkir, Simon Vaughan, Mike Goad, Daniela Huppenkothen, and Phil Uttley. The variability of active galaxies–i. broad-band noise x-ray power spectra from xmm–newton and swift. Monthly Notices of the Royal Astronomical Society, 544(4):3260–3279, 2025.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in neural information processing systems, 35: 9881–9893, 2022.

Summer McLaughlin. The systematic detection of AGN flares in the era of time domain astronomy. PhD thesis, University of Sheffield, 2025.

Summer AJ McLaughlin, James R Mullaney, and Stuart P Littlefair. Using gaussian processes to detect agn flares. Monthly Notices of the Royal Astronomical Society, 529(3):2877–2892, 2024.

Jackeline Moreno, Michael S Vogeley, and Gordon T Richards. Agn variability analysis handbook. arXiv preprint arXiv:1811.00154, 2018.

Daniel Muthukrishna, Kaisey S Mandel, Michelle Lochner, Sara Webb, and Gautham Narayan. Real-time detection of anomalies in large-scale transient surveys. Monthly Notices of the Royal Astronomical Society, 517(1):393–419, 2022.

K Nandra. X-ray variability of agn and correlations with spectral properties. Advances in Space Research, 28(2-3):295–306, 2001.

Paul M O'Neill, Kirpal Nandra, Iossif E Papadakis, and T Jane Turner. The relationship between x-ray variability amplitude and black hole mass in active galactic nuclei. Monthly Notices of the Royal Astronomical Society, 358(4):1405–1416, 2005.

Guansong Pang, Chunhua Shen, and Anton Van Den Hengel. Deep anomaly detection with deviation networks. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 353–362, 2019.

C. Ricci, B. Trakhtenbrot, M. J. Koss, Y. Ueda, I. Delvecchio, E. Treister, K. Schawinski, S. Paltani, K. Oh, I. Lamperti, S. Berney, P. Gandhi, K. Ichikawa, F. E. Bauer, L. C. Ho, D. Asmus, V. Beckmann, S. Soldi, M. Baloković, N. Gehrels, and C. B. Markwardt. BAT AGN Spectroscopic Survey. V. X-Ray Properties of the Swift/BAT 70-month AGN Catalog. , 233(2):17, December 2017. doi: 10.3847/1538-4365/aa96ad.

Joshua S Richman and J Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. American journal of physiology-heart and circulatory physiology, 2000.

Kamran Shaukat, Talha Mahboob Alam, Suhuai Luo, Shakir Shabbir, Ibrahim A Hameed, Jiaming Li, Syed Konain Abbas, and Umair Javed. A review of time-series anomaly detection techniques: A step to future perspectives. In Future of Information and Communication Conference, pp. 865–877. Springer, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. Advances in neural information processing systems, 28, 2015.

RAA Souza, E Hatziminaoglou, A de Pádua Santos, and T Stosic. Complexity and multifractal variability in multiband emission of seyfert agn. Monthly Notices of the Royal Astronomical Society, 549(2):stag916, 2026.

Aaron L Stancik and Eric B Brauns. A simple asymmetric lineshape for fitting infrared absorption spectra. Vibrational Spectroscopy, 47(1):66–69, 2008.

Strüder, L., Briel, U., Dennerl, K., Hartmann, R., Kendziorra, E., Meidinger, N., Pfeffermann, E., Reppin, C., Aschenbach, B., Bornemann, W., Bräuninger, H., Burkert, W., Elender, M., Freyberg, M., Haberl, F., Hartner, G., Heuschmann, F., Hippmann, H., Kastelic, E., Kemmer, S., Kettenring, G., Kink, W., Krause, N., Müller, S., Oppitz, A., Pietsch, W., Popp, M., Predehl, P., Read, A., Stephan, K. H., Stötter, D., Trümper, J., Holl, P., Kemmer, J., Soltau, H., Stötter, R., Weber, U., Weichert, U., von Zanthier, C., Carathanassis, D., Lutz, G., Richter, R. H., Solc, P., Böttcher, H., Kuster, M., Staubert, R., Abbey, A., Holland, A., Turner, M., Balasini, M., Bignami, G. F., La Palombara, N., Villa, G., Buttler, W., Gianini, F., Lainé, R., Lumb, D., and Dhez, P. The european photon imaging camera on xmm-newton: The pn-ccd camera *. AA, 365(1):L18–L26, 2001. doi: 10.1051/0004-6361:20000066. URL https://doi.org/10.1051/0004-6361:20000066.

Chang Wei Tan, François Petitjean, Eamonn Keogh, and Geoffrey I Webb. Time series classification for varying length series. arXiv preprint arXiv:1910.04341, 2019.

N James Terrell Jr. Shot-noise character of cygnus x-1 pulsations. Astrophysical Journal, vol. 174, p. L35, 174:L35, 1972.

Jens Timmer and Michel Koenig. On generating power law noise. Astronomy and Astrophysics, v. 300, p. 707, 300:707, 1995.

Antonio Torralba, Phillip Isola, and William T Freeman. Foundations of computer vision. MIT Press, 2024.

Phil Uttley, IM McHardy, and S Vaughan. Non-linear x-ray variability in x-ray binaries and active galaxies. Monthly Notices of the Royal Astronomical Society, 359(1):345–362, 2005.

Philip Uttley and Ian M McHardy. The flux-dependent amplitude of broadband noise variability in x-ray binaries and active galaxies. Monthly Notices of the Royal Astronomical Society, 323(2): L26–L30, 2001.

Philip Uttley and Ian M McHardy. X-ray variability of ngc 3227 and 5506 and the nature of active galactic nucleus 'states'. Monthly Notices of the Royal Astronomical Society, 363(2):586–596, 2005.

Simon Vaughan. A simple test for periodic signals in red noise. Astronomy & Astrophysics, 431 (1):391–403, 2005.

Simon Vaughan, R Edelson, RS Warwick, and P Uttley. On characterizing the variability properties of x-ray light curves from active galaxies. Monthly Notices of the Royal Astronomical Society, 345(4):1271–1284, 2003.

N. A. Webb, M. Coriat, I. Traulsen, J. Ballet, C. Motch, F. J. Carrera, F. Koliopanos, J. Authier, I. de la Calle, M. T. Ceballos, E. Colomo, D. Chuard, M. Freyberg, T. Garcia, M. Kolehmainen, G. Lamer, D. Lin, P. Maggi, L. Michel, C. G. Page, M. J. Page, J. V. Perea-Calderon, F.-X. Pineau, P. Rodriguez, S. R. Rosen, M. Santos Lleo, R. D. Saxton, A. Schwope, L. Tomás, M. G. Watson, and A. Zakardjian. The XMM-Newton serendipitous survey. IX. The fourth XMM-Newton serendipitous source catalogue. , 641:A136, September 2020. doi: 10.1051/0004-6361/201937353.

SUPPLEMENTARY MATERIALS

This supplementary document provides additional quantitative results and implementation subtleties that were omitted from the main text due to the 9-page limitation.

The codebase for this project can be found here: https://anonymous.4open.science/r/ANDE

Specifically, this document includes:

- Section A: A description of how the simulated data was synthesized and used to train the CNN
- Section B: A summary of the computational facilities used
- Section C: Complete tabular documentation of the experimental results of training the CNN with simulated time-series signals
- Section D: Misclassified Examples from the WAX-XMM test set
- Section E: Visualizing the time-series signals generated by the trained CVAE using t-SNE and U-MAP
- Section F: Complete tabular documentation of the experimental results when the CNN was trained using CVAE generated time-series signals.
- Section G: A sample set of time-series from the real world 4XMM-DR14 test set and the predictions made by the best model ensemble on them

A GENERATING BATCHES OF TRAINING TIME-SERIES SIGNALS

To generate batches of input data and the corresponding labels for training the model described above, a data generator function was employed. It takes in as input the batch size and the stochastic burst intensity multiplier, and generates two arrays, one containing m generated time-series signals where m is the batch size, and the other containing the corresponding label for all those time-series signals, for both the positive and negative cases. During training, in every batch, the data generator pumped 64 time-series signals (32 without stochastic bursts and 32 with stochastic bursts) into the training pipeline, and 4 time-series signals (2 without stochastic bursts and 2 with stochastic bursts) in the validation pipeline. Each epoch had 32 batches. Hence, in every epoch, during training, the CNN was fed 64*32=2048 time-series signals in the training pipeline, and 64*4=256 time-series signals in the validation pipeline. Before each of these time-series signals are added to the arrays, they are normalized by dividing by their median values, and then also quantile transformed with a rank based custom batch quantile transformer function. Each positive time-series signal contained 2 - 5 stochastic bursts, the number chosen randomly.

In the training phase, the early stopping and model checkpoint callbacks were used. In general, callbacks are useful for monitoring and modifying the behavior of the model during the training or inference phases. The early stopping callback was implemented by monitoring the validation loss with a patience of 75 epochs and a minimum delta of 0.001. Since the validation loss is expected to reduce, the mode is minimum and the intention is to restore the best weights from the best epoch after training stops. In the model checkpoint callback, the mode was set to minimum while monitoring the validation loss to ensure the model is saved to file only when the validation loss is the least (i.e. it is lesser than the previous least).

B USE OF COMPUTATIONAL FACILITIES

The facilities will be described in the de-anonymized camera-ready version.

C EXPERIMENTAL TABULATION AND EVALUATION OF THE SIM-TO-REAL PIPELINE

C.1 TRAINING WITH SYMMETRIC GAUSSIAN STOCHASTIC BURSTS

The tables below elaborate the set of experiments that were performed to train the CNN with simulated time-series signals where the positive time-series signals only contained symmetric gaussian profiles for the stochastic bursts.

The first table (Table 2) tabulates the results when varying the intensity multiplier of the stochastic bursts keeping the width range of the stochastic bursts fixed between 50 and 200, and the second table (Table 3) tabulates the results when varying the width range, keeping the intensity multiplier of the stochastic bursts fixed at 0.75 (the optimal value based on the performance metrics).

Even though models trained with time-series signals having certain width ranges for the symmetric gaussian perform better than that with a width range of 50 - 200, those width ranges were not chosen for training the final model (with all modifications to the symmetric gaussian combined) as the subsequent types of modifications (namely very broad gaussian stochastic bursts and very narrow gaussian stochastic bursts) covered those width ranges.

C.2 TRAINING WITH ASYMMETRIC GAUSSIAN STOCHASTIC BURSTS

Table 4 elaborates the set of experiments that were performed to train the CNN with simulated time-series signals where the positive time-series signals only contained asymmetric gaussian profiles for the stochastic bursts.

C.3 TRAINING WITH OSCILLATORY GAUSSIAN STOCHASTIC BURSTS

Table 5 elaborates on the set of experiments that were performed to train the CNN with simulated time-series signals, where the positive time-series signals only contained oscillatory gaussian profiles for the stochastic bursts.

C.4 TRAINING WITH VERY BROAD (SYMMETRIC) GAUSSIAN STOCHASTIC BURSTS

Table 6 elaborates on the set of experiments that were performed to train the CNN with simulated time-series signals where the positive time-series signals only contained very broad symmetric gaussian profiles for the stochastic bursts.

C.5 TRAINING WITH VERY NARROW (SYMMETRIC) GAUSSIAN STOCHASTIC BURSTS

Table 7 elaborates on the set of experiments that were performed to train the CNN with simulated time-series signals where the positive time-series signals only contained very narrow symmetric gaussian profiles for the stochastic bursts.

D MISCLASSIFIED EXAMPLES FROM THE TEST SET

In this section (Figure 8), some of the misclassified examples (time-series signals) from the WAX-XMM test set are shown when they were inferenced using the model with the best performance from the set of models trained on time-series signals where the positive signals contained only symmetric gaussian stochastic bursts.

Table 2: Performance of trained models when varying the stochastic burst intensity multiplier (i.e., Gaussian Normalization) of symmetric gaussian stochastic bursts keeping the width range fixed between 50 and 200

Stochastic burst Intensity Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
2.00	0.7872 ± 0.0181	0.9927 ± 0.0146	0.5060 ± 0.0448	0.6690 ± 0.0383	0.5605 ± 0.0436	0.5209 ± 0.0446
1.75	0.7876 ± 0.0144	0.9963 ± 0.0111	0.5050 ± 0.0340	0.6695 ± 0.0296	0.5600 ± 0.0333	0.5200 ± 0.0339
1.50	0.7919 ± 0.0541	0.9469 ± 0.2174	0.5150 ± 0.1280	0.6657 ± 0.1578	0.5661 ± 0.1382	0.5290 ± 0.1308
1.25	0.8073 ± 0.0241	0.9908 ± 0.0164	0.5550 ± 0.0632	0.7090 ± 0.0475	0.6075 ± 0.0591	0.5694 ± 0.0622
1.00	0.8261 ± 0.0289	0.9871 ± 0.0228	0.6020 ± 0.0737	0.7447 ± 0.0545	0.6516 ± 0.0684	0.6157 ± 0.0724
0.75	0.8462 ± 0.0261	0.9838 ± 0.0252	0.6520 ± 0.0682	0.7816 ± 0.0459	0.6980 ± 0.0613	0.6648 ± 0.0664
0.70	0.8449 ± 0.0252	0.9842 ± 0.0209	0.6480 ± 0.0608	0.7796 ± 0.0437	0.6947 ± 0.0557	0.6610 ± 0.0595
0.60	0.8662 ± 0.0277	0.9794 ± 0.0216	0.7030 ± 0.0725	0.8157 ± 0.0475	0.7438 ± 0.0646	0.7145 ± 0.0704
0.50	0.8662 ± 0.0277	0.9794 ± 0.0216	0.7030 ± 0.0725	0.8157 ± 0.0475	0.7438 ± 0.0646	0.7145 ± 0.0704
0.40	0.8650 ± 0.0264	0.9760 ± 0.0194	0.7020 ± 0.0672	0.8145 ± 0.0433	0.7427 ± 0.0593	0.7135 ± 0.0651
0.30	0.7466 ± 0.1268	0.8719 ± 0.1751	0.5240 ± 0.2209	0.6200 ± 0.1908	0.5549 ± 0.2109	0.5323 ± 0.2181
0.20	0.5500 ± 0.1458	0.4931 ± 0.2508	0.4300 ± 0.2539	0.4245 ± 0.2156	0.4227 ± 0.2321	0.4271 ± 0.2463
0.10	0.4880 ± 0.1352	0.4222 ± 0.1395	0.5160 ± 0.2364	0.4424 ± 0.1698	0.4792 ± 0.2016	0.5039 ± 0.2246

Table 3: Performance of trained models when varying the width range of symmetric gaussian stochastic bursts keeping the stochastic burst intensity multiplier (i.e., Gaussian Normalization) fixed at 0.75

Width Range	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
50 - 300	0.8427 ± 0.0188	0.9749 ± 0.0211	0.6500 ± 0.0550	0.7780 ± 0.0355	0.6955 ± 0.0491	0.6627 ± 0.0535
50 - 400	0.8389 ± 0.0306	0.9760 ± 0.0194	0.6400 ± 0.0805	0.7695 ± 0.0562	0.6857 ± 0.0734	0.6528 ± 0.0787
50 - 500	0.8363 ± 0.0268	0.9760 ± 0.0227	0.6340 ± 0.0730	0.7655 ± 0.0497	0.6804 ± 0.0661	0.6469 ± 0.0712
50 - 600	0.8295 ± 0.0279	0.9867 ± 0.0171	0.6100 ± 0.0703	0.7512 ± 0.0519	0.6592 ± 0.0653	0.6237 ± 0.0690
30 - 200	0.8496 ± 0.0996	0.9569 ± 0.1224	0.7260 ± 0.0858	0.8131 ± 0.0637	0.7543 ± 0.0536	0.7333 ± 0.0723
10 - 200	0.8782 ± 0.0218	0.9817 ± 0.0169	0.7290 ± 0.0523	0.8354 ± 0.0356	0.7680 ± 0.0470	0.7400 ± 0.0509
5 - 200	0.8662 ± 0.0687	0.9318 ± 0.2144	0.7010 ± 0.1641	0.7996 ± 0.1846	0.7373 ± 0.1716	0.7113 ± 0.1662

Table 4: Performance of trained models when varying the rising and falling width range of asymmetric gaussian stochastic bursts keeping the stochastic burst intensity multiplier fixed at 0.75

Rising Width	**Falling Width**	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
50 - 150	150 - 300	0.8355 ± 0.0219	0.9792 ± 0.0189	0.6290 ± 0.0549	0.7642 ± 0.0408	0.6767 ± 0.0510	0.6423 ± 0.0540
30 - 150	150 - 300	0.8491 ± 0.0199	0.9729 ± 0.0136	0.6660 ± 0.0510	0.7893 ± 0.0356	0.7102 ± 0.0465	0.6784 ± 0.0499
10 - 150	150 - 300	0.8534 ± 0.0231	0.9835 ± 0.0160	0.6690 ± 0.0608	0.7943 ± 0.0405	0.7138 ± 0.0546	0.6816 ± 0.0592
5 - 150	150 - 300	0.8615 ± 0.0126	0.9712 ± 0.0161	0.6970 ± 0.0305	0.8111 ± 0.0202	0.7385 ± 0.0271	0.7087 ± 0.0296
50 - 150	150 - 400	0.8415 ± 0.0276	0.9785 ± 0.0245	0.6440 ± 0.0677	0.7743 ± 0.0502	0.6902 ± 0.0627	0.6569 ± 0.0664
50 - 150	150 - 500	0.8436 ± 0.0289	0.9770 ± 0.0168	0.6500 ± 0.0731	0.7779 ± 0.0507	0.6954 ± 0.0664	0.6627 ± 0.0713
50 - 150	150 - 600	0.8308 ± 0.0205	0.9710 ± 0.0220	0.6230 ± 0.0483	0.7578 ± 0.0364	0.6706 ± 0.0449	0.6363 ± 0.0474

Table 5: Performance of trained models when varying the amplitude and frequency of the oscillatory part of the stochastic bursts keeping the width range fixed and stochastic burst intensity multiplier of the base symmetric gaussian part of the stochastic bursts fixed at 0.75 and 50 - 200 respectively.

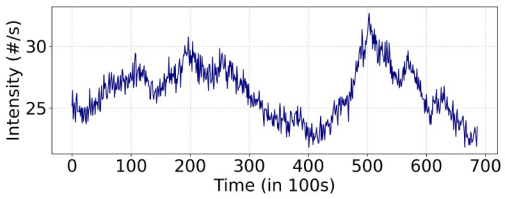
Intensity Range of Sinusoidal Superimposed Wave	**Frequency Range of Sinusoidal Superimposed Wave**	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
f/5 - f/2	w/30000 - w/10000	0.8496 ± 0.0975	0.9265 ± 0.1154	0.7500 ± 0.0637	0.8190 ± 0.0532	0.7725 ± 0.0240	0.7557 ± 0.0475
f/5 - f/3	w/30000 - w/10000	0.8585 ± 0.0661	0.9187 ± 0.2117	0.6930 ± 0.1598	0.7898 ± 0.1815	0.7287 ± 0.1677	0.7031 ± 0.1620
f/5 - f/4	w/30000 - w/10000	0.8389 ± 0.1156	0.8891 ± 0.2355	0.7150 ± 0.1748	0.7826 ± 0.1873	0.7369 ± 0.1700	0.7206 ± 0.1709
f/5 - f/2	w/30000 - w/8000	0.8718 ± 0.0085	0.9622 ± 0.0191	0.7290 ± 0.0184	0.8293 ± 0.0121	0.7660 ± 0.0158	0.7395 ± 0.0177
f/5 - f/2	w/30000 - w/6000	0.8538 ± 0.0648	0.9094 ± 0.2093	0.6890 ± 0.1583	0.7839 ± 0.1800	0.7240 ± 0.1663	0.6989 ± 0.1606
f/5 - f/2	w/50000 - w/10000	0.8590 ± 0.0667	0.9124 ± 0.2100	0.6990 ± 0.1618	0.7914 ± 0.1823	0.7332 ± 0.1693	0.7087 ± 0.1639
f/5 - f/2	w/50000 - w/6000	0.8551 ± 0.0657	0.9060 ± 0.2087	0.6950 ± 0.1605	0.7864 ± 0.1810	0.7289 ± 0.1680	0.7046 ± 0.1626

Table 6: Performance of trained models when varying the width range of the very broad (symmetric) gaussian stochastic bursts keeping the stochastic burst intensity multiplier (i.e., Gaussian Normalization) fixed at 0.75

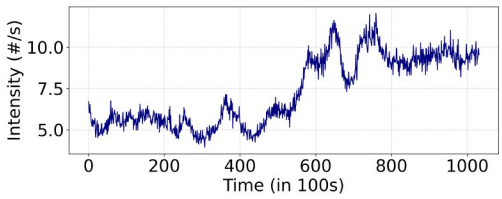
Width Range	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
200 - 300	0.7944 ± 0.0276	0.9801 ± 0.0263	0.5310 ± 0.0708	0.6854 ± 0.0578	0.5832 ± 0.0680	0.5453 ± 0.0702
200 - 400	0.7983 ± 0.0288	0.9731 ± 0.0277	0.5440 ± 0.0720	0.6947 ± 0.0575	0.5953 ± 0.0685	0.5581 ± 0.0712
200 - 500	0.7893 ± 0.0222	0.9788 ± 0.0255	0.5190 ± 0.0546	0.6763 ± 0.0452	0.5720 ± 0.0526	0.5335 ± 0.0542
200 - 600	0.7821 ± 0.0469	0.9664 ± 0.0320	0.5070 ± 0.1103	0.6577 ± 0.1062	0.5576 ± 0.1110	0.5209 ± 0.1107
200 - 700	0.7521 ± 0.0535	0.9475 ± 0.0932	0.4530 ± 0.1098	0.6024 ± 0.1094	0.5021 ± 0.1116	0.4664 ± 0.1104

Table 7: Performance of trained models when varying the width range of the very narrow (symmetric) gaussian stochastic bursts keeping the stochastic burst intensity multiplier (i.e., Gaussian Normalization) fixed at 0.75

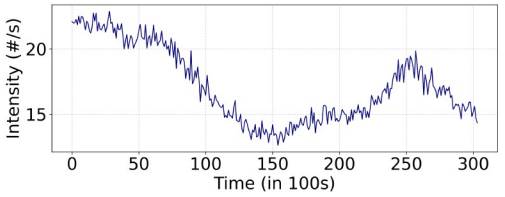
Width Range	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
40 - 50	0.8402 ± 0.0226	1.0000 ± 0.0000	0.6260 ± 0.0530	0.7687 ± 0.0400	0.6760 ± 0.0495	0.6399 ± 0.0521
30 - 50	0.8231 ± 0.0922	0.9700 ± 0.1246	0.6540 ± 0.0886	0.7672 ± 0.0480	0.6904 ± 0.0427	0.6634 ± 0.0717
20 - 50	0.8611 ± 0.0150	0.9859 ± 0.0189	0.6850 ± 0.0309	0.8079 ± 0.0233	0.7293 ± 0.0284	0.6975 ± 0.0302
10 - 50	0.8560 ± 0.0658	0.9364 ± 0.2154	0.6730 ± 0.1561	0.7829 ± 0.1803	0.7130 ± 0.1649	0.6843 ± 0.1586
5 - 50	0.8620 ± 0.0668	0.9315 ± 0.2143	0.6910 ± 0.1592	0.7933 ± 0.1823	0.7285 ± 0.1676	0.7016 ± 0.1616



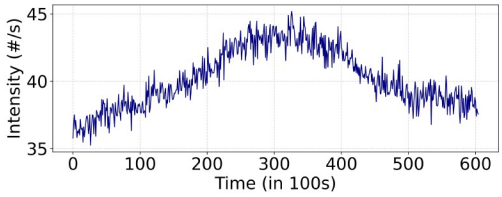
(a) An Asymmetric Stochastic Burst



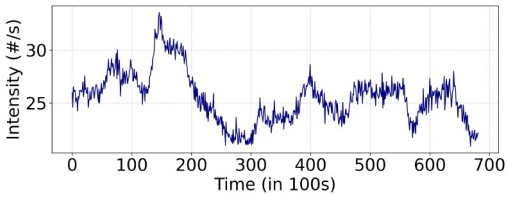
(b) A Superimposed Stochastic Burst



(c) A Very Narrow Stochastic Burst



(d) A Very Broad Stochastic Burst



(e) An Oscillatory Stochastic Burst

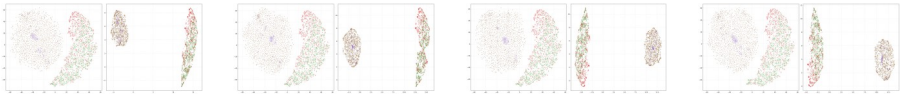
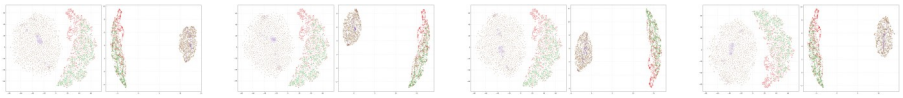
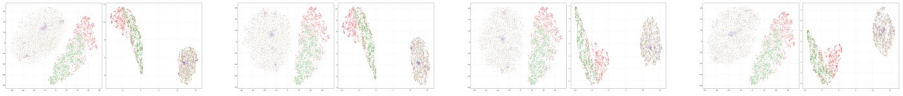
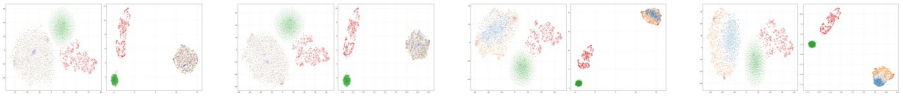
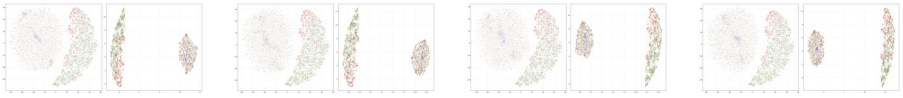
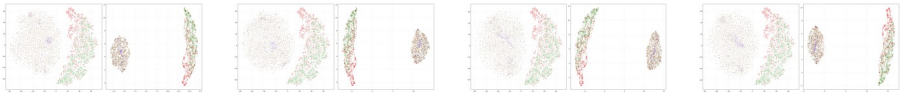
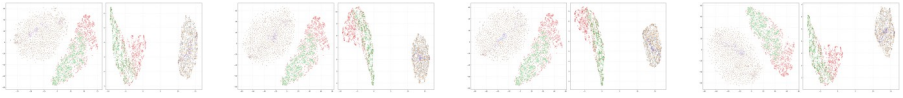
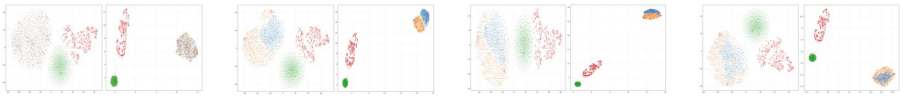
Figure 8: Five time-series signals from the WAX-XMM test set that was misclassified by a CNN trained only on time-series signals with symmetric gaussian burst profiles

E GENERATING TIME-SERIES SIGNALS FROM TRAINED CVAE

The table below summarizes the results of training the Conditional CVAE (Convolutional Variational Auto-Encoder) with simulated and hybrid (simulated and small number of real samples from the BAT-AGN sample set) time-series signals. The training set consisted of an equal volume of time-series signals with and without stochastic bursts, with the volume of time-series signals with stochastic bursts being distributed among the 5 types of gaussian stochastic bursts that were de-

scribed earlier. Each dimensionality reduction (t-SNE and U-MAP) plot corresponds to a specific value of the Random Walk Noise multiplier (for both the training set and the simulated time-series signals in the plots) and a specific value of the training volume used to train the CVAE.

Table 8: Table of CVAE Training Results with Dimensionality Reduction plots of Simulated, CVAE generated, and real time-series signals with and without stochastic bursts

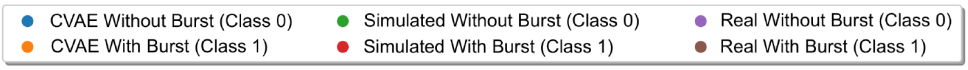
Type and Random Walk Noise Multiplier of training dataset	**Volume of training dataset**			
	200k	**400k**	**800k**	**1.6m**
Synthetic & 0.3				
Synthetic & 0.2				
Synthetic & 0.1				
Synthetic & 0.0				
Hybrid & 0.3				
Hybrid & 0.2				
Hybrid & 0.1				
Hybrid & 0.0				

F TRAINING THE CNN WITH CVAE GENERATED TIME-SERIES SIGNALS

In this section, the results of using the CNN trained on the CVAE generated time-series signals to predict the candidate stochastic variations in the WAX-XMM test dataset are tabulated.

The CVAE was trained in two ways: (1) with simulated time-series signals only, and (2) with both simulated time-series signals and real time-series signals (a small fraction of real time-series signals

Figure 9: The legend for Table 8



from the held out BAT-AGN test set was injected into every batch of the simulated time-series signals). For each of the above two cases, the CNN was trained in four ways: (1) with CVAE generated time-series signals only without multiscale representation (univariate features), (2) with CVAE generated time-series signals only with multiscale representation (multivariate features), (3) with hybrid (both CVAE generated time-series signals and real time-series signals) without multiscale representation (univariate features), and (4) with hybrid (both CVAE generated time-series signals and real time-series signals) with multiscale representation (multivariate features). In the multivariate cases, 12 features of every time-series signal (the original, five levels of trended, five levels of detrended, and one level of gradient on the trended) were used for both the training set, validation set, and the prediction set of the time-series signals for the CNN.

Table 9: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **simulated time-series signals only**, and the CNNs were trained on **CVAE generated time-series signals only** with **univariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.5483 ± 0.0049	0.1500 ± 0.3571	0.0038 ± 0.0096	0.0073 ± 0.0186	0.0047 ± 0.0119	0.0040 ± 0.0102
0.1	0.5538 ± 0.0088	0.5000 ± 0.5000	0.0151 ± 0.0194	0.0290 ± 0.0367	0.0187 ± 0.0239	0.0160 ± 0.0206
0.2	0.5782 ± 0.0130	0.9500 ± 0.2179	0.0689 ± 0.0288	0.1275 ± 0.0520	0.0844 ± 0.0350	0.0728 ± 0.0304
0.3	0.5641 ± 0.0285	0.8726 ± 0.3142	0.0981 ± 0.2088	0.1214 ± 0.1261	0.0994 ± 0.1656	0.0976 ± 0.1941
0.4	0.5615 ± 0.0112	0.7500 ± 0.4330	0.0321 ± 0.0247	0.0611 ± 0.0462	0.0396 ± 0.0303	0.0340 ± 0.0261
0.5	0.5573 ± 0.0253	0.8726 ± 0.3142	0.0830 ± 0.2112	0.0944 ± 0.1261	0.0811 ± 0.1677	0.0817 ± 0.1964
0.6	0.5714 ± 0.0095	0.9500 ± 0.2179	0.0538 ± 0.0209	0.1013 ± 0.0385	0.0662 ± 0.0256	0.0569 ± 0.0221
0.7	0.5756 ± 0.0311	0.9226 ± 0.2429	0.1236 ± 0.2032	0.1664 ± 0.1169	0.1303 ± 0.1589	0.1244 ± 0.1882
0.8	0.5842 ± 0.0172	0.9000 ± 0.3000	0.0821 ± 0.0379	0.1494 ± 0.0668	0.1001 ± 0.0458	0.0867 ± 0.0399
0.9	0.5872 ± 0.0118	0.9889 ± 0.0484	0.0906 ± 0.0277	0.1645 ± 0.0464	0.1104 ± 0.0331	0.0956 ± 0.0291
1.0	0.5880 ± 0.0110	0.9938 ± 0.0272	0.0915 ± 0.0255	0.1665 ± 0.0427	0.1116 ± 0.0304	0.0966 ± 0.0268

Table 10: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **simulated time-series signals only**, and the CNNs were trained on **hybrid (CVAE generated and real) time-series signals** with **univariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.6581 ± 0.1515	0.6477 ± 0.1757	0.7821 ± 0.1596	0.6807 ± 0.1084	0.7311 ± 0.1228	0.7653 ± 0.1464
0.1	0.6265 ± 0.1277	0.6733 ± 0.1954	0.5792 ± 0.1724	0.5788 ± 0.1248	0.5725 ± 0.1455	0.5763 ± 0.1633
0.2	0.6205 ± 0.1191	0.6281 ± 0.1662	0.6387 ± 0.1030	0.6089 ± 0.0716	0.6217 ± 0.0778	0.6329 ± 0.0941
0.3	0.6560 ± 0.1248	0.6812 ± 0.2403	0.6009 ± 0.2222	0.5982 ± 0.1572	0.5905 ± 0.1784	0.5964 ± 0.2063
0.4	0.6286 ± 0.1503	0.6296 ± 0.2432	0.6132 ± 0.1867	0.5944 ± 0.1621	0.5987 ± 0.1603	0.6078 ± 0.1761
0.5	0.6368 ± 0.1311	0.6596 ± 0.1984	0.6245 ± 0.1822	0.6074 ± 0.1185	0.6101 ± 0.1422	0.6191 ± 0.1682
0.6	0.6342 ± 0.1148	0.6289 ± 0.2218	0.5981 ± 0.2263	0.5727 ± 0.1900	0.5823 ± 0.2045	0.5926 ± 0.2188
0.7	0.6718 ± 0.1317	0.6882 ± 0.1663	0.6623 ± 0.0893	0.6553 ± 0.0791	0.6549 ± 0.0703	0.6594 ± 0.0817
0.8	0.6774 ± 0.1110	0.6404 ± 0.1985	0.6472 ± 0.1685	0.6342 ± 0.1651	0.6393 ± 0.1612	0.6444 ± 0.1653
0.9	0.6850 ± 0.1150	0.6949 ± 0.1573	0.6877 ± 0.1075	0.6697 ± 0.0744	0.6756 ± 0.0812	0.6834 ± 0.0983
1.0	0.5641 ± 0.1795	0.5535 ± 0.1556	0.7170 ± 0.0189	0.6145 ± 0.1077	0.6684 ± 0.0619	0.7013 ± 0.0329

Table 11: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **simulated time-series signals only**, and the CNNs were trained on **CVAE generated time-series signals only** with **multivariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.5551 ± 0.0168	0.4449 ± 0.4639	0.0321 ± 0.0594	0.0537 ± 0.0894	0.0381 ± 0.0683	0.0336 ± 0.0617
0.1	0.4966 ± 0.1241	0.0125 ± 0.0332	0.0123 ± 0.0344	0.0122 ± 0.0336	0.0122 ± 0.0340	0.0122 ± 0.0343
0.2	0.5342 ± 0.0917	0.4574 ± 0.4918	0.0264 ± 0.0432	0.0407 ± 0.0503	0.0299 ± 0.0435	0.0273 ± 0.0431
0.3	0.5538 ± 0.0100	0.4000 ± 0.4899	0.0151 ± 0.0220	0.0288 ± 0.0415	0.0186 ± 0.0271	0.0160 ± 0.0233
0.4	0.5402 ± 0.0630	0.6621 ± 0.4616	0.0245 ± 0.0247	0.0439 ± 0.0416	0.0296 ± 0.0290	0.0258 ± 0.0258
0.5	0.5470 ± 0.0616	0.8561 ± 0.3433	0.0340 ± 0.0264	0.0612 ± 0.0425	0.0411 ± 0.0305	0.0358 ± 0.0274
0.6	0.5778 ± 0.0122	1.0000 ± 0.0000	0.0679 ± 0.0269	0.1260 ± 0.0470	0.0833 ± 0.0325	0.0718 ± 0.0284
0.7	0.5902 ± 0.0106	1.0000 ± 0.0000	0.0953 ± 0.0235	0.1731 ± 0.0397	0.1162 ± 0.0281	0.1006 ± 0.0247
0.8	0.5991 ± 0.0114	1.0000 ± 0.0000	0.1151 ± 0.0252	0.2055 ± 0.0432	0.1397 ± 0.0303	0.1214 ± 0.0265
0.9	0.5987 ± 0.0092	1.0000 ± 0.0000	0.1142 ± 0.0202	0.2043 ± 0.0329	0.1386 ± 0.0240	0.1204 ± 0.0212
1.0	0.6026 ± 0.0128	1.0000 ± 0.0000	0.1226 ± 0.0283	0.2173 ± 0.0456	0.1485 ± 0.0334	0.1293 ± 0.0296

Table 12: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **simulated time-series signals only**, and the CNNs were trained on **hybrid (CVAE generated and real) time-series signals** with **multivariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.6466 ± 0.1547	0.6148 ± 0.1470	0.8274 ± 0.1056	0.6907 ± 0.0969	0.7614 ± 0.0809	0.8058 ± 0.0940
0.1	0.7051 ± 0.0685	0.7093 ± 0.0994	0.6349 ± 0.1585	0.6531 ± 0.0997	0.6389 ± 0.1321	0.6357 ± 0.1503
0.2	0.6632 ± 0.1198	0.6580 ± 0.1458	0.6736 ± 0.1376	0.6455 ± 0.1054	0.6579 ± 0.1158	0.6682 ± 0.1297
0.3	0.6551 ± 0.1379	0.6432 ± 0.1547	0.7170 ± 0.1541	0.6552 ± 0.1092	0.6863 ± 0.1285	0.7070 ± 0.1454
0.4	0.7103 ± 0.0840	0.7352 ± 0.1334	0.6538 ± 0.1723	0.6595 ± 0.1406	0.6526 ± 0.1546	0.6529 ± 0.1663
0.5	0.6983 ± 0.1321	0.6999 ± 0.1446	0.7094 ± 0.1234	0.6858 ± 0.0989	0.6952 ± 0.1031	0.7045 ± 0.1155
0.6	0.7962 ± 0.0738	0.8026 ± 0.0982	0.7594 ± 0.1285	0.7696 ± 0.0797	0.7609 ± 0.1042	0.7595 ± 0.1208
0.7	0.7427 ± 0.1038	0.7093 ± 0.1211	0.7943 ± 0.0943	0.7411 ± 0.0825	0.7698 ± 0.0802	0.7866 ± 0.0888
0.8	0.7239 ± 0.1738	0.7132 ± 0.1646	0.8085 ± 0.0765	0.7430 ± 0.1078	0.7757 ± 0.0740	0.7976 ± 0.0711
0.9	0.7500 ± 0.1098	0.7280 ± 0.1177	0.7783 ± 0.0907	0.7436 ± 0.0807	0.7615 ± 0.0770	0.7729 ± 0.0851
1.0	0.7799 ± 0.0666	0.7693 ± 0.1082	0.7736 ± 0.0718	0.7637 ± 0.0536	0.7677 ± 0.0533	0.7716 ± 0.0653

Table 13: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **hybrid (simulated and real) time-series signals**, and the CNNs were trained on **CVAE generated time-series signals only** with **univariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.7590 ± 0.0501	0.6858 ± 0.0620	0.8840 ± 0.0427	0.7701 ± 0.0392	0.8339 ± 0.0334	0.8685 ± 0.0387
0.1	0.7641 ± 0.0555	0.6948 ± 0.0746	0.8783 ± 0.0577	0.7728 ± 0.0480	0.8319 ± 0.0450	0.8640 ± 0.0527
0.2	0.8043 ± 0.0386	0.7553 ± 0.0632	0.8566 ± 0.0621	0.7992 ± 0.0343	0.8317 ± 0.0430	0.8490 ± 0.0558
0.3	0.8325 ± 0.0369	0.8572 ± 0.0868	0.7717 ± 0.0553	0.8075 ± 0.0350	0.7847 ± 0.0406	0.7753 ± 0.0507
0.4	0.8329 ± 0.0354	0.9007 ± 0.0551	0.7142 ± 0.0739	0.7931 ± 0.0492	0.7432 ± 0.0647	0.7224 ± 0.0713
0.5	0.8205 ± 0.0199	0.9608 ± 0.0391	0.6321 ± 0.0568	0.7600 ± 0.0352	0.6773 ± 0.0496	0.6447 ± 0.0548
0.6	0.7829 ± 0.0251	0.9727 ± 0.0314	0.5368 ± 0.0564	0.6896 ± 0.0492	0.5888 ± 0.0552	0.5511 ± 0.0562
0.7	0.7547 ± 0.0306	0.9875 ± 0.0232	0.4651 ± 0.0712	0.6289 ± 0.0634	0.5189 ± 0.0702	0.4797 ± 0.0711
0.8	0.7466 ± 0.0288	0.9948 ± 0.0166	0.4434 ± 0.0658	0.6103 ± 0.0621	0.4976 ± 0.0662	0.4581 ± 0.0661
0.9	0.7325 ± 0.0268	0.9929 ± 0.0175	0.4132 ± 0.0642	0.5801 ± 0.0642	0.4667 ± 0.0658	0.4276 ± 0.0648
1.0	0.7150 ± 0.0227	0.9937 ± 0.0203	0.3736 ± 0.0519	0.5407 ± 0.0550	0.4261 ± 0.0542	0.3876 ± 0.0526

Table 14: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **hybrid (simulated and real) time-series signals**, and the CNNs were trained on **hybrid (CVAE generated and real) time-series signals** with **univariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.7603 ± 0.0457	0.6652 ± 0.0493	0.9651 ± 0.0209	0.7862 ± 0.0320	0.8841 ± 0.0172	0.9397 ± 0.0171
0.1	0.7944 ± 0.0531	0.7077 ± 0.0712	0.9566 ± 0.0343	0.8105 ± 0.0393	0.8913 ± 0.0231	0.9362 ± 0.0286
0.2	0.7966 ± 0.0595	0.7068 ± 0.0697	0.9651 ± 0.0161	0.8138 ± 0.0444	0.8975 ± 0.0215	0.9440 ± 0.0141
0.3	0.8047 ± 0.0586	0.7215 ± 0.0768	0.9509 ± 0.0256	0.8178 ± 0.0455	0.8919 ± 0.0252	0.9326 ± 0.0229
0.4	0.8274 ± 0.0372	0.7524 ± 0.0477	0.9292 ± 0.0273	0.8306 ± 0.0323	0.8868 ± 0.0253	0.9163 ± 0.0258
0.5	0.8197 ± 0.0569	0.7609 ± 0.0707	0.8934 ± 0.0561	0.8193 ± 0.0496	0.8614 ± 0.0467	0.8836 ± 0.0522
0.6	0.8167 ± 0.0401	0.7743 ± 0.0739	0.8613 ± 0.0588	0.8109 ± 0.0316	0.8392 ± 0.0379	0.8545 ± 0.0519
0.7	0.8329 ± 0.0317	0.8252 ± 0.0599	0.8132 ± 0.0866	0.8138 ± 0.0400	0.8122 ± 0.0657	0.8128 ± 0.0803
0.8	0.7906 ± 0.0505	0.7963 ± 0.0882	0.7538 ± 0.1071	0.7644 ± 0.0506	0.7556 ± 0.0783	0.7539 ± 0.0981
0.9	0.7872 ± 0.0492	0.7957 ± 0.0631	0.7236 ± 0.1294	0.7496 ± 0.0767	0.7321 ± 0.1076	0.7259 ± 0.1229
1.0	0.7585 ± 0.0565	0.7799 ± 0.0776	0.6821 ± 0.1415	0.7141 ± 0.0726	0.6914 ± 0.1102	0.6843 ± 0.1317

Table 15: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **hybrid (simulated and real) time-series signals**, and the CNNs were trained on **CVAE generated time-series signals only** with **multivariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.7291 ± 0.0700	0.6627 ± 0.0673	0.8208 ± 0.1144	0.7310 ± 0.0780	0.7816 ± 0.0961	0.8087 ± 0.1084
0.1	0.7167 ± 0.0556	0.6609 ± 0.0637	0.7877 ± 0.1205	0.7131 ± 0.0647	0.7545 ± 0.0929	0.7774 ± 0.1117
0.2	0.7568 ± 0.0617	0.7337 ± 0.0772	0.7387 ± 0.1099	0.7314 ± 0.0723	0.7346 ± 0.0917	0.7373 ± 0.1042
0.3	0.7863 ± 0.0632	0.8030 ± 0.0763	0.7038 ± 0.1077	0.7463 ± 0.0809	0.7194 ± 0.0961	0.7082 ± 0.1043
0.4	0.7735 ± 0.0430	0.8635 ± 0.0756	0.6009 ± 0.0861	0.7035 ± 0.0653	0.6375 ± 0.0791	0.6112 ± 0.0842
0.5	0.7850 ± 0.0494	0.9159 ± 0.0545	0.5830 ± 0.1214	0.7032 ± 0.0962	0.6247 ± 0.1143	0.5946 ± 0.1196
0.6	0.7675 ± 0.0310	0.9394 ± 0.0489	0.5217 ± 0.0654	0.6680 ± 0.0589	0.5715 ± 0.0641	0.5354 ± 0.0652
0.7	0.7564 ± 0.0411	0.9605 ± 0.0516	0.4830 ± 0.0967	0.6366 ± 0.0886	0.5340 ± 0.0955	0.4969 ± 0.0965
0.8	0.7389 ± 0.0382	0.9756 ± 0.0276	0.4358 ± 0.0897	0.5963 ± 0.0861	0.4880 ± 0.0907	0.4499 ± 0.0902
0.9	0.7325 ± 0.0388	0.9774 ± 0.0345	0.4208 ± 0.0917	0.5814 ± 0.0919	0.4726 ± 0.0937	0.4347 ± 0.0924
1.0	0.7038 ± 0.0271	0.9878 ± 0.0217	0.3509 ± 0.0615	0.5146 ± 0.0670	0.4019 ± 0.0647	0.3645 ± 0.0625

Table 16: Performance of trained CNN models when varying the Random Walk Noise multiplier of the CVAE generated time-series signals in the training set. The CVAE was trained on **hybrid (simulated and real) time-series signals**, and the CNNs were trained on **hybrid (CVAE generated and real) time-series signals** with **multivariate features**.

Random Walk Noise Multiplier	**Accuracy**	**Precision**	**Recall**	F_1	F_2	F_4
0.0	0.8201 ± 0.0455	0.7294 ± 0.0591	0.9745 ± 0.0225	0.8324 ± 0.0340	0.9116 ± 0.0159	0.9550 ± 0.0177
0.1	0.8380 ± 0.0398	0.7518 ± 0.0516	0.9679 ± 0.0267	0.8451 ± 0.0338	0.9144 ± 0.0244	0.9515 ± 0.0249
0.2	0.8252 ± 0.0454	0.7388 ± 0.0594	0.9651 ± 0.0306	0.8350 ± 0.0350	0.9078 ± 0.0225	0.9474 ± 0.0263
0.3	0.8286 ± 0.0423	0.7550 ± 0.0561	0.9311 ± 0.0364	0.8322 ± 0.0360	0.8884 ± 0.0300	0.9181 ± 0.0335
0.4	0.8244 ± 0.0349	0.7687 ± 0.0589	0.8896 ± 0.0585	0.8215 ± 0.0312	0.8602 ± 0.0403	0.8806 ± 0.0525
0.5	0.8299 ± 0.0322	0.7920 ± 0.0521	0.8547 ± 0.0405	0.8204 ± 0.0293	0.8402 ± 0.0304	0.8503 ± 0.0370
0.6	0.8372 ± 0.0399	0.8207 ± 0.0579	0.8283 ± 0.0631	0.8217 ± 0.0409	0.8250 ± 0.0508	0.8272 ± 0.0592
0.7	0.8350 ± 0.0231	0.8330 ± 0.0584	0.8057 ± 0.0651	0.8151 ± 0.0268	0.8086 ± 0.0478	0.8064 ± 0.0600
0.8	0.8308 ± 0.0508	0.8388 ± 0.0632	0.7849 ± 0.1150	0.8039 ± 0.0725	0.7912 ± 0.0984	0.7866 ± 0.1102
0.9	0.8150 ± 0.0439	0.7992 ± 0.0843	0.8151 ± 0.0697	0.8006 ± 0.0362	0.8077 ± 0.0483	0.8127 ± 0.0629
1.0	0.8222 ± 0.0334	0.8244 ± 0.0676	0.7840 ± 0.0658	0.7995 ± 0.0378	0.7893 ± 0.0513	0.7854 ± 0.0613

G PREDICTIONS BY THE BEST MODEL ENSEMBLE ON THE XMM TEST SET

In this section we display a few time-series from the 4XMM-DR14 filtered catalogue which were predicted by the best model ensemble to contain bursts and not contain bursts.

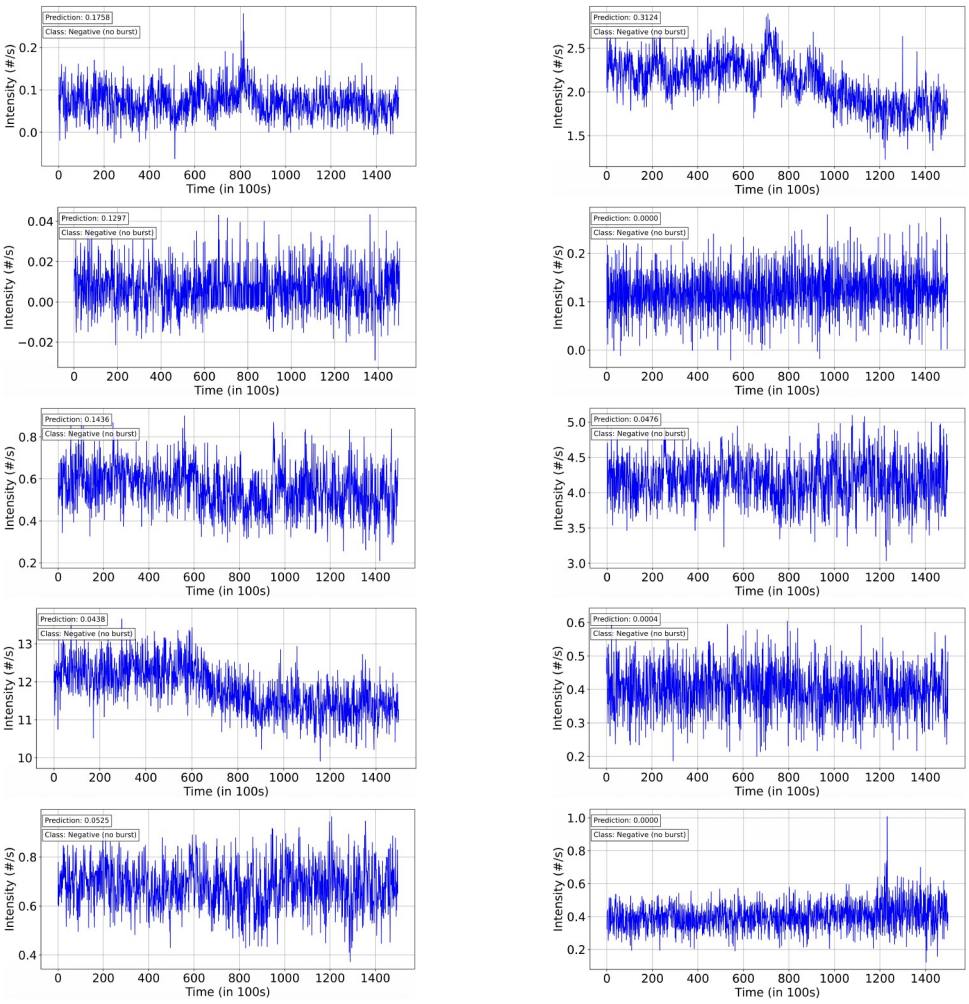


Figure 10: Ten time-series signals from the 4XMM-DR14 filtered catalogue predicted by best model ensemble to not contain stochastic bursts

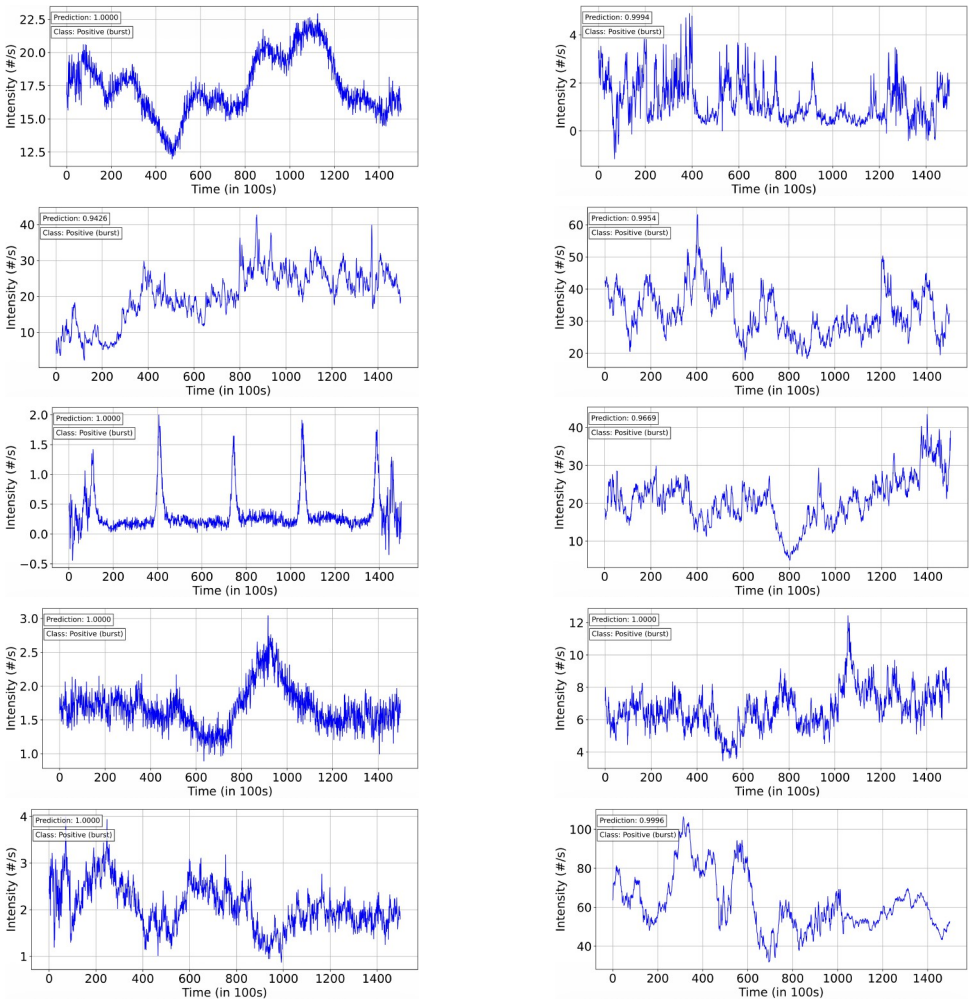


Figure 11: Ten time-series signals from the 4XMM-DR14 filtered catalogue predicted by best model ensemble to contain stochastic bursts