StepMania annotated music as TSAD benchmark dataset

Anosh Billimoria

Supervisor: Billy Joe Franks

RPTU, Kaiserslautern, Germany

billimor@rptu.de

Abstract. The search for anomalies in music is not a common practice. In this paper, we propose a scenario where abrupt changes in music beats can be considered as anomalies. Our efforts are directed towards solving a data problem where the evaluation of methods in Time Series Anomaly Detection is hindered due to the absence of a reliable benchmark dataset. This unavailability complicates the search for effective algorithms. Stepmania, a community-driven game, involves people from around the world annotating beat changes in songs as part of a gameplay mechanic configuration, making it a vast data source. Our work evaluates this dataset in several key ways, such as measuring the length of anomalies, their positional occurrence, and the variation that beat changes introduced to the frequencies that describe the music. Subsequently, we apply various well-known learning methods to observe how the dataset responds to them.

1 Introduction

Anomaly Detection in Time Series data is a heavily researched area that garners significant interest from a large part of the academic community. It has been a significant field of study for many years, with wide-ranging applications across finance, security, and manufacturing sectors. However, the effectiveness and accuracy of anomaly detection methods are increasingly being tested by the vast and complex datasets characteristic of today's big data landscape. The challenges faced in finding the right datasets and methods, and how to evaluate them, are highlighted in research by Wagner et al., 2023, where the datasets, methods, and evaluation metrics are critically assessed across different aspects thereafter logical improvements are suggested. As a follow-up, Si et al., 2024 proposes a Benchmarking paradigm and acknowledges criticisms of the prior. One of the main concerns in Time Series Anomaly Detection (TSAD) is the quality of datasets that are currently used in benchmarks. Some main issues highlighted by Wagner et al., 2023 include the distributional shift of anomalies, anomaly density, long anomalies, positional bias, and constant features. All of these contribute to skewed learning and thus present a foundational dilemma over the methods that have been tested using these datasets.

This work proposes a new dataset comprised of musical tracks, that are annotated with beats per minute (BPM) changes. Music is simply a well structured pattern of frequencies that our ears perceive as pleasing as opposed to noise. It can also be regarded as organized noise Godt, 2005. But for music to be engaging, there are variations and shifts in tone, tempo, pitch, and sometimes sudden halts. This makes them a seemingly good candidate for studying Anomaly Detection.

The procurement of music tracks is not the challenge but instead being aware of the sudden shifts in a sequence is the primary goal. We take the help of a community-driven collection of StepMania “Stepmania

Data”, 2012 configuration files formatted as (.sm) or (.ssc) files. These hold information on when the tempo changes away from the initial reference beat sequence of the track. We aim to evaluate this Stepmania Dataset as a potential benchmark dataset for TSAD. To achieve this we will convert the time-based tracks to frequency-based mel-spectrograms spread across over frames, which are chopped up timestamps. This dataset will be multivariate since at every frame the intensities of the corresponding frequency levels act as attributes, the columns of the dataset . Then we shall apply well-known unsupervised Machine Learning and Deep Learning algorithms to present a cohesive evaluation.

2 Background

We want to apply AD to Music files, to do that we need to prepare the data, for which we must first identify ground-truth anomalies, and finally find an accurate and effective method to evaluate the results. Music files must be converted to the frequency domain to make a conducive representation for learning algorithms. Initially, these files are in standard audio formats such as .ogg, .wav, or .mp4. The first step involves applying the short-term Fourier transform(STFT) and then plotting a spectrogram that reports the intensity across various frequencies at sub-second intervals. This representation significantly emphasizes changes in Beats Per Minute (BPM). Which are otherwise hidden in time-domain representations.

The subsequent step entails extracting and annotating BPM variations as anomalies, which are identified from the simfiles across a series of time steps whenever the beat strays away from the reference beat.

Lastly, an essential component is the evaluation pipeline utilized for this dataset. It consists of a library that is easily expandable, designed for training, and modified metrics for evaluating the performance of different model interactions with the data.

2.1 Mel-spectrograms

The data is essentially music tracks, which are converted into Mel-spectrograms, These are human a perceptible version of the spectrogram.

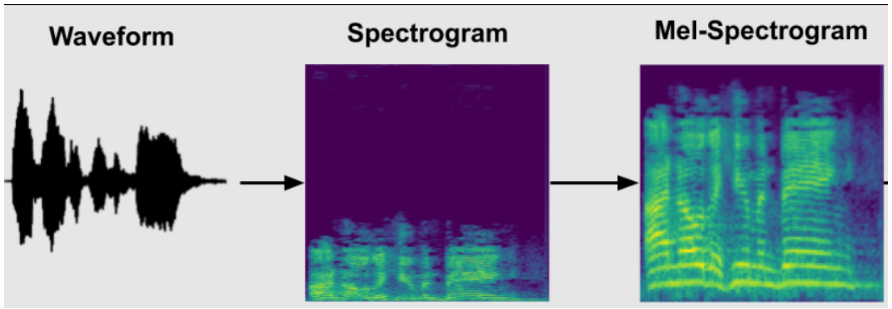


Fig. 1: From the left, a sound waveform is the most basic representation. It depicts the amplitude of sound measured in decibels over time. The spectrogram is a representation of the same audio in the frequency domain. As such it shows the intensities of specific frequencies occurring over frames. Frames are simply sub-second samples of the song providing more granularity and information in the frequency domain. The Mel-spectrogram is just a differently scaled version of the spectrogram. It reveals how humans will perceive music and scales the frequencies between the audible range of the human ear making it more informative.

Visually from Figure 1, a Mel-spectrogram looks similar to the spectrogram. The only difference is that of the scale. For humans, our limited range of audible frequencies prevents us from perceiving sound from different frequencies as if there are mild differences or large ones. For example, we can easily tell the difference between 500 and 1000 Hz, but we will hardly be able to tell a difference between 10,000 and 10,500 Hz, even though the distance between the two pairs is the same. This is better represented on Mel-scale, which was derived empirically in O'Shaughnessy, 1987 shown in 1

$$m = 2959 * \log_{10}(1 + (\frac{freq}{1000})) \tag{1}$$

Where, m is the value in the scale of the corresponding frequency $freq$ in Hertz(Hz). To acheive this conversion we use the librosa library with discrete short fourier transform parameters set to the following: $sampling\_rate = 22050$, $n\_fft = 2048$ segment length in Short time fourier transform(STFT) and $hop\_length = 512$ number of segments to skip during windowing. We also establish $n\_mels = 128$ number of mel filter banks that will be used in in the final representation. These parameters result in every track being a matrix of $(\frac{sampling\_rate}{hop\_length} * D, n\_mels)$, where D is the duration of the track. For example a one minute track would have a resulting dimension of $(2583, 128)$. The values contained in this matrix are the amplitude of corresponding frequency represented on the mel scale at a particular frame.

2.2 Anomaly-extractions

The stepmania configuration files are called simfiles and can have two formats (.sm) and (.ssc), we only use the (.sm) format when (.ssc) is not available since the latter is newer. These files contain several attributes about how the track will be presented on screen and in-game. The files define when and how will the arrows be displayed and behave while the track is being played.

The question arises as to what in these files could constitute as an anomaly. We can consider as anomalies the fields such as BPMS(beat per minute changes), Stops (seconds to stop), and Warps (skips beats). Out of these, our primary focus is on the BPMs attribute since initial analysis shows that the annotated parts correspond well with the actual mel-spectrogram of the track as shown in Figure 2. A long tempo shift between frames 3500-4000, and another shorter change right after 2000. The underlying spectrogram, wherever annotated, appears visually distorted compared to it's otherwise regular surrounding patterns.

2.3 TimeSeAD library

We used the benchmarking library by Wagner et al., 2023 to run all experiments. It contains datasets, methods, and metrics as defined in their research. The datasets are well-known and used in literature such as SMD, Exathlon, WADI, and SWAT. The methods incorporated are also from recent research in time series anomaly detection. They propose a modified form of F1-score and Area under the Precision-Recall Curve (AUPRC) from point-wise to windowed evaluation of time series. This is done to evaluate neighboring time steps and not end up isolating anomalies. The changes are meant to incorporate consistencies between neighboring anomalies and add context to evaluation resulting in a more accurate representation of how models perform on different datasets. These evaluation methods reveal the onset and drop of anomalies which works well for our use case since beat music changes are not sudden but rather have an onset and a build-up before the tempo fluctuates.

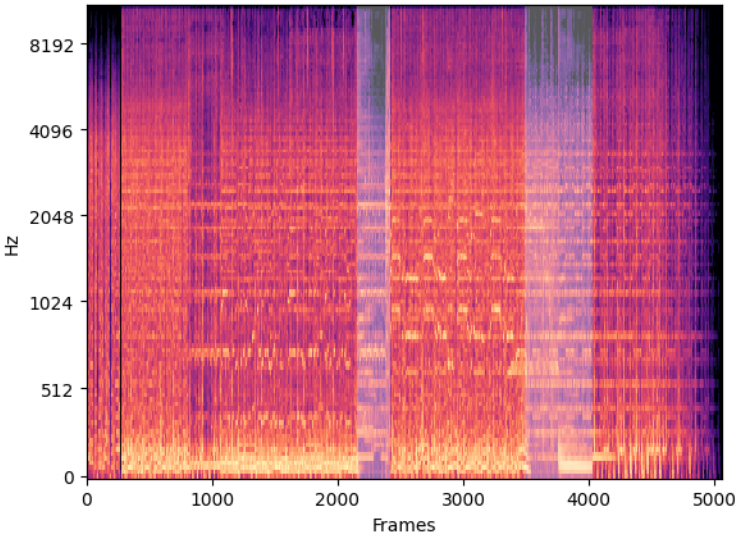


Fig. 2: A visual representation of how music files are fed to the model. The y-axis shows frequencies within the human perceivable limit and x-axis shows frames which are akin to time stamps but sampled in a windowing fashion for increased granularity. The colours within represent intensities of corresponding frequencies at particular frame events. The whitened portions are beat shifts derived from simfile. These correspond to actual disturbances in the Mel-spectrogram as seen

3 Methods

For fairness and proper varied evaluation, we have made use of methods across the spectrum. Majorly we divide them into shallow and deep methods. The former includes the likes of Kmeans clusteringMacQueen et al., 1967 and Isolation ForestLiu et al., 2008. We will consider the following sub-categories within the latter. Reconstruction-based methods where an encoder and decoder are used to map the input onto a latent space and then back, anomalies are identified based on the reconstruction loss from the original. We will be using the technique LSTM_AEMalhotra et al., 2016. Generative-adversarial networks (GANs) and Variational Autoencoder (VAEs) are both generative techniques that directly model the distribution that generates the data by training a generative model in a certain latent space. GANs use two networks a discriminator that looks for difference between the actual and generated image, it's feedback is used to modify the generator network parameters to generate samples that closely resemble the actual data. While VAEs map actual data to high dimensional latent space after which the original is reproduced by the decoder. We have used TADGANGeiger et al., 2020 and SIS_VAE Li et al., 2020. Lastly, we have prediction methods that predict the time series a few timestamps ahead. This method uses a prediction error between the original sample and predicted one to find anomalies. We used TCN He and Zhao, 2019 for our research.

3.1 Shallow baselines

We utilize some easy to apply and reliable shallow methods as a sanity check on how well our data really is. The results of which can be considered a baseline for other Deep Anomaly Detection methods.

Kmeans Clustering: First appeared in MacQueen et al., 1967 is a powerful tool for grouping data points based on their similarities, making it widely used in various applications. Anomalies can be considered as data points that deviate markedly from the main body of the dataset. By measuring the distance from each data point to the closest cluster center, we can pinpoint those that are situated at a considerable distance from the clusters. In recent times the method is still very popular as an initial check upon the data for various kinds of Anomaly detection-related applications. The writers of Aziz and Bestak, 2024 utilize and highlight the effectiveness of the algorithm in the area of Mobile network security and analyzed Call Detail Records(CDR) to recognize anomalous voice calls. Researchers also tried to analyze electricity consumption for public street lighting, and find anomalous patterns using the algorithm Anaedevha, 2024. In this sense, we also decide to use the algorithm. However, we use a modified version thanks to Wagner et al., 2023 algorithm does not just select points in isolation but instead cluster a window of timesteps for fair and accurate identification of BPM changes in our dataset. In other words, the algorithm is input with sliding windows instead of points to alleviate shortcomings of point-wise evaluation. The best parameters that emerged after running an extensive grid search are a window size of 50, a step size of 1, and a value of k is 5. The results section will further illustrate comparisons.

Isolation Forest: Developed by Liu et al., 2008, the Isolation Forest algorithm works on the idea that it's easier to separate unusual data points from the rest. It does this by repeatedly dividing the data based on random attributes and values. This process can be visualized as a tree (called an Isolation Tree), where the number of divisions needed to isolate a point is the length of the path from the root to the end node in the tree. The fewer the divisions needed, the more anomalous the data point is considered to be. It is highly regarded and widely used for several AD applications to name a few, detecting desertification based on satellite imagery Harrou et al., 2024, detection of anomalous protein activation in cells particles Nascimben et al., 2024 and monitoring quality of food using data from an IoT device Prasad and Raman, 2024. An academically relied upon method is bound to prove how well our tempo changes in music can be detected. The best parameters we found on our dataset after a grid search were 100 trees and a window size of 10. Similar to the previous method, isolation forest was also applied to a collection of points within a windowing function.

3.2 Deep AD

LSTM-AE: Malhotra et al., 2016 proposes an LSTM Autoencoder, a neural network model that uses LSTM units to encode and decode sequence data. It learns to compress the input into a fixed-length vector and then reconstructs the input from this vector. The trained encoder can be used for tasks like data visualization, dimension reduction, and as input to other models. Our grid search produced the following best parameters; hidden dimensions equal to 30, learning rate of $1 - e4$ and window size of 100.

TADGAN: Our choice of generative method was proposed by Geiger et al., 2020 and is called Time Series Anomalies detection using GANs or TADGAN. Their technique uses bidirectional LSTM as an Autoencoder (AE) and two distinct Wasserstein GANs. The AE's decoder serves as the generator for the first GAN, while the AE's encoder serves as the generator for the second GAN. Both GANs employ TCN-based discriminators. The loss function incorporates the AE's Mean Squared Error (MSE) as a measure of reconstruction error. During the detection phase, the MSE and discriminator score are calculated, normalized, and combined to produce a final anomaly score. The method is considered an online method as the statistics of both scores are computed during training. The best parameter we found for our dataset is detector parameter alpha as 0.9, which controls the importance of the two scores. Latent size 20, a window size of 100, reconstruction co-efficient 1, and learning rate $1 - e4$.

SIS-VAE: Smoothness-Inducing Sequential VAE proposed by Li et al., 2020 is a Gated Recurrent Unit-base(GRU) VAE. The researchers enhance the Variational Autoencoder (VAE) to reconstruct smooth time series by introducing a KL divergence term between successive time steps to the Evidence Lower Bound (ELBO). This term intuitively promotes similarity in distributions for adjacent points in the predicted time series. Following common practice in VAE-based methods, the SIS-VAE employs the reconstruction probability as the anomaly score. Best found parameters for this method turned out to be window size of 100 and learning rate of $1 - e3$.

TCN: Temporal convolution networks (TCN) proposed by He and Zhao, 2019 R use a dilated causal TCN to process the input window. The final three layers' outputs are concatenated and passed through a convolution layer with D filters. The result is a shifted window of size w × D. The method uses MSE loss for training and fits a Gaussian distribution to prediction errors. Only the last point in the predicted window is used for detection due to the online setting requirement. the best found parameters are 64 resolution filters, kernel sizes of 3, exponential dilations from 2 to 16, concatenate 3 layers, window size 50 and learning rate $1 - e3$.

4 Results

We chose to use the best adjusted F1 score calculation as the evaluation metric for all the mentioned methods. Table 1 reveals how inconsistent results can be with the classical calculation, mainly when calculating precision. On the other hand, we are more interested in how all methods, shallow or deep, produce similar results on the Stepmania dataset. This is a good sign when it comes to considering it as a benchmarking dataset since the deep methods perform nearly as well as the shallow baselines. We also note that the recall is generally much higher than precision, meaning that while there are fewer false negatives, there is a high number of false positives and a low number of true positives. The prime suspect for this behavior is the fact that beat change annotations are inconsistent with the actual detection of anomalies by all methods. We dive deeper into this in the following Section 5.

	Precision	Recall	F1	Precision*	Recall*	F1*
K-Means	0.37564	0.87344	0.52534	0.32465	0.99765	0.48988
IForest	0.32048	0.73161	0.44571	0.32901	0.98549	0.49332
LSTM-AE	0.58676	0.99145	0.73721	0.33686	0.99965	0.50391
TADGAN	0.34235	0.91852	0.49879	0.33485	0.99622	0.50122
SIS-VAE	0.71676	0.99254	0.83240	0.33384	0.99943	0.50049
TCN	0.28823	0.74322	0.41537	0.33023	0.99912	0.49639

Table 1: F1 score comparison between methods. (*) indicates adjusted performance metric calculations that overcome point-wise bias of the classical evaluation methods. Wagner et al., 2023

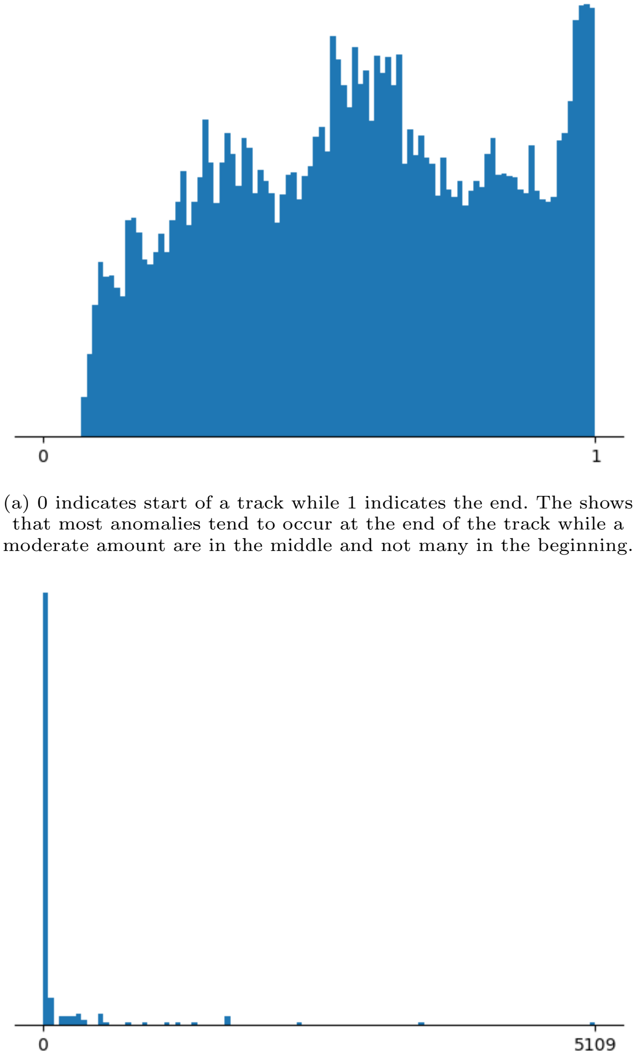
5 Analysis

Our Stepmania dataset does not resolve all the issues of a multivariate time series anomaly detection benchmark dataset, as pointed out by Wagner et al., 2023. However, it possesses a relatively promising structure on various fronts and can be regarded as having potential for future research. We plan to address some of these issues and evaluate the Stepmania dataset using the analysis tools provided in the TimeSeAD library.

The primary issue to address is positional bias. As illustrated in Figure 3a, anomalies occur with a good distribution across the track, without a significant skew towards the beginning or end. This distribution is advantageous, as it indicates that anomalies are not overly concentrated in any specific part of the song but can occur throughout. Given that the songs span various genres and styles, this diversity is a valuable attribute for a benchmark dataset in Time Series Anomaly Detection (TSAD).

The next issue is that of anomaly lengths, such as single events (points) or longer lasting for a sequence of frames. Initially, we considered BPM changes as point anomalies. In practice, we only marked points of change, while in many cases the beats can change for more frames. Thus, we shifted to marking them as a sequential anomaly across multiple frames of the spectrogram. This results in Figure 3b where we notice that the majority of tracks have different beat values only for a few seconds. A handful of outliers do exist where we notice that anomalies last for nearly the entire track. The reason for this is that some songs have a lot of small beat changes and sometimes don't deviate heavily from the reference beat but also never really come back to it. Next, we tackle how the frequencies change when an anomaly occurs.

On average the mean feature distributional shift for the dataset appears to be acceptable as a good dataset. This can be seen in Figure 4 Although looking closer we observe that for some tracks, their anomalies occur at major shifts in the frequency intensities as opposed to the rest of the normal song. From Figure 5 we can notice that some songs have minor shifts in frequencies that contribute to the beat change anomalies. At the same time, some songs can have a completely deviated frequency distribution when the anomalies occur. We can say that a manual choosing of the songs in the testing set can potentially make a very effective ground for consistent evaluations.



(a) 0 indicates start of a track while 1 indicates the end. The shows that most anomalies tend to occur at the end of the track while a moderate amount are in the middle and not many in the beginning.

(b) Number of continuous frames for which an anomaly or in other words BPM change occurs within every song.

Fig. 3: Analysis of how anomalies are placed within the dataset to reveal any bias or troubles that may come up for methods that are applied to it.

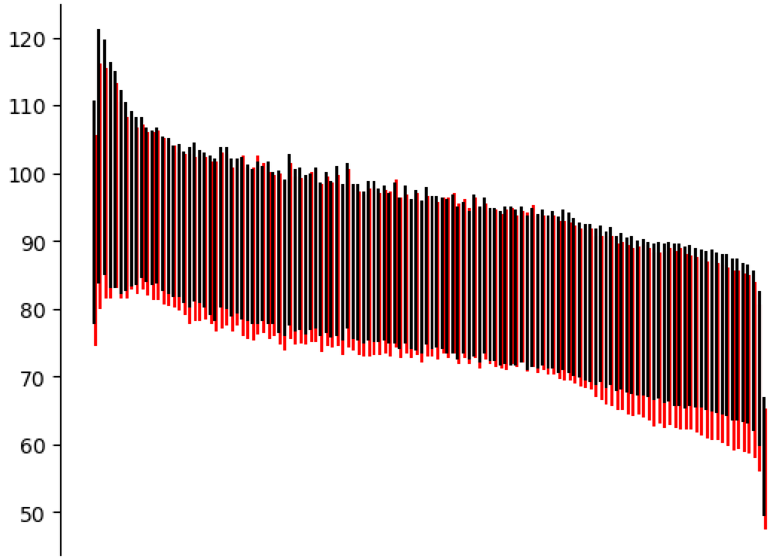
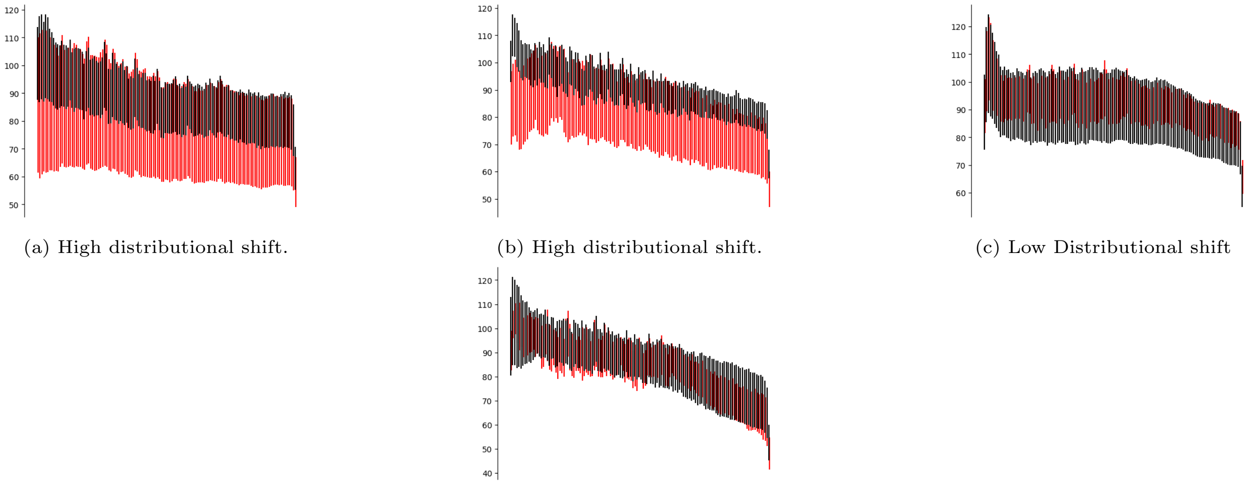


Fig. 4: The black bars show the non-anomalous change in intensities of each frequency on the Mel scale. Whereas, the red bars depict changes in frequency intensities when an anomaly occurs. This graph is a mean aggregation of standard deviation of the changes across all the songs in the dataset.



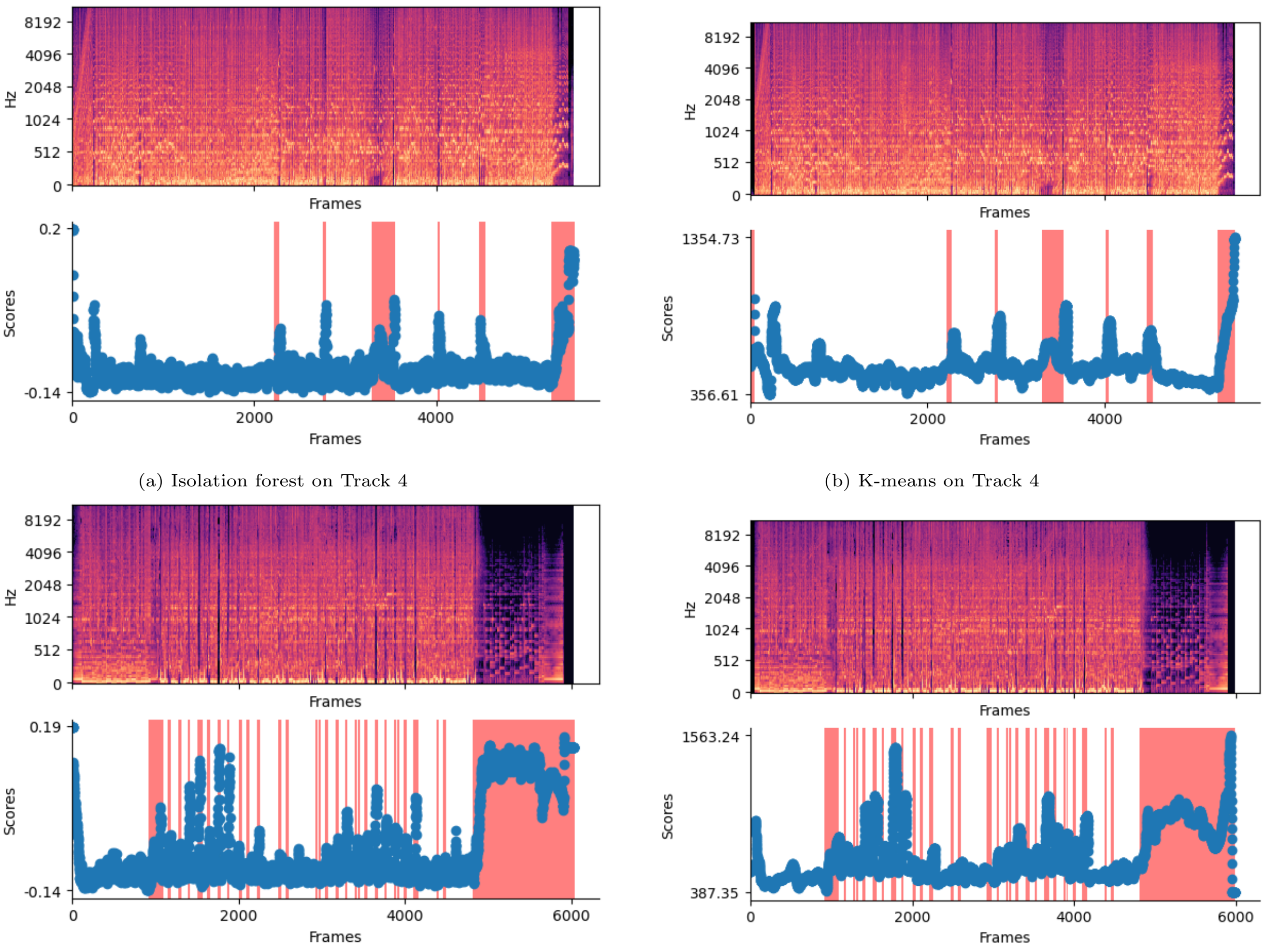
(a) High distributional shift.

(b) High distributional shift.

(c) Low Distributional shift

(d) Low Distributional shift

Fig. 5: Two songs each for high and low feature distribution shift. A high shift can essentially be a song on it's own.



(a) Isolation forest on Track 4

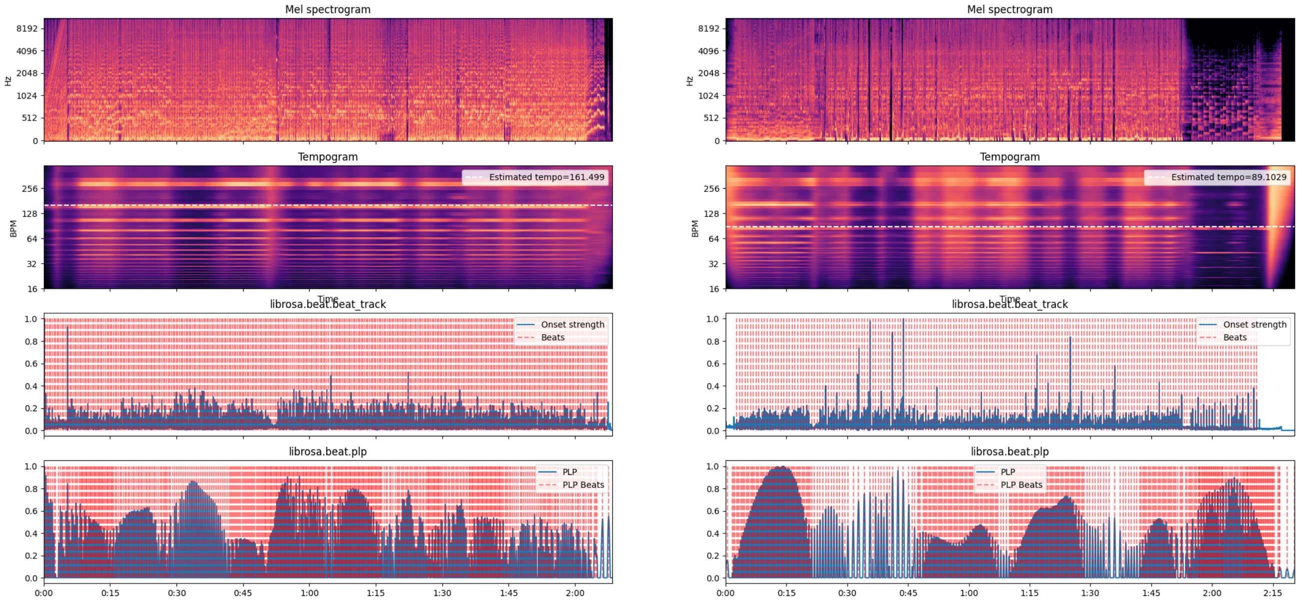
(b) K-means on Track 4

(c) Isolation forest on Track 16

(d) K-means on Track 16

Fig. 6: Detection by shallow methods on two sample song. Red areas indicate presence of annotated anomaly, blue scatter points are anomaly score assigned by the method, and above them is the Mel-spectrogram of the particular Track.

An In-depth manual inspection of the detection done by individual methods gives us some interesting behavioral insights. We notice that both methods can detect an unlabelled part of the track as an anomaly, visibly the Mel-spectrograms do appear to have discrepancies at that part of the track. In Figures 6c and 6d several labeled anomalies are being detected by both methods but not all of them. In the middle portion, two small sections are detected as normal by both methods. We verify the existence of beats by changing our perspective and looking at other representations of beat change such as Tempogram, Dynamic programming-based beat tracking Ellis, 2007 and Predominant Local Pulse (PLP) Grosche and Muller, 2010 in Figures 7a and 7b. All techniques are available as functions in the librosa python library. We notice that learning methods are detecting spikes or actual beat changes but leave out the onset of the beats. The Stepmania annotations do not always track every single beat change that occurs in the songs.

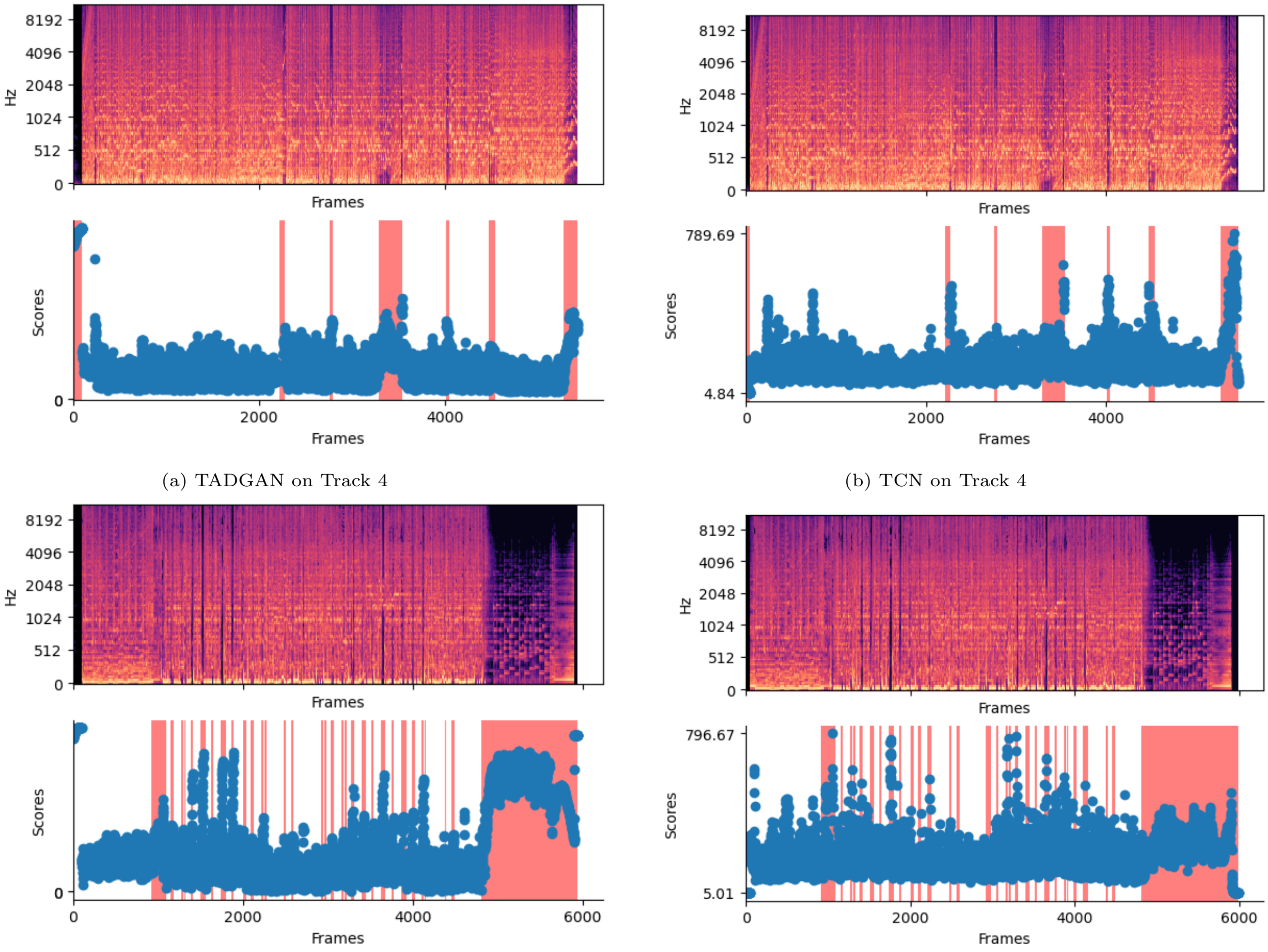


(a) Perspectives of Track 4

(b) Perspectives of Track 16

Fig. 7: This figure shows the same sample tracks as above. After applying some non-ML based methods such as Tempogram, DP based beat tracker, PLP respectively from after the Mel-spectrogram at the top. These three methods are capable of revealing actual beat changes as well the onset of a beat change which can act as false positives to learning methods. By comparing our results with these representations we can gather the correctness of annotations as well as our predictions.

We can now shift our focus to the deep learning methods, where we once again see similar patterns in Figure 8. Interestingly, we can observe the differing decision-making processes of both methods. Their conclusions in aggregation are similar, as they both seem to agree about the existence of beat changes at similar points. However, it seems the prediction method TCN falls short of longer anomaly windows, as we notice at the ends of both tracks. Meanwhile, the GAN method is certain of the presence of anomalies, regardless of annotation. We can argue, however, that neither method is perfect in detecting beat changes. Once there are multiple similar shifts in the music's tempo, the methods start treating them as normal.



(a) TADGAN on Track 4

(b) TCN on Track 4

(c) TADGAN on Track 16

(d) TCN on Track 16

Fig. 8: Detection by deep methods on two sample song.

6 Conclusion

In this paper, we have evaluated the potential of Stepmania annotated songs with beat changes as anomalies to act as a benchmarking dataset for Time Series Anomaly Detection. We analyzed the dataset in terms of how well the anomalies are distributed across different songs and their duration. We also plotted the variation in individual features or frequencies in the presence or absence of anomalies. These plots show us that the dataset has a reasonable distribution in terms of when the anomalies occur. The duration of anomalies can be much longer since the beat may change slightly and then remain so with minor changes, although some smart annotations or choice of music can fix these issues. Most frequency diversions in the presence of beat changes are not as extreme, meaning that their variations aren't too drastic or irregular, making them non-trivial to detect. We also looked at some learning methods across the board, from shallow to deep algorithms. Unfortunately, neither is better than the other, and all methods have similar evaluation metrics. On the other hand, some visual inspection reveals the learning patterns of these methods and thus, tells us that the models can detect beat changes as appearing in Mel-spectrograms, but those which are not labeled. This makes it solely an annotation problem that can be worked upon with some manual inspection and curated choice of music.

References

Anaedevha, R.-N. (2024). Application of machine learning models for device identification in wireless network traffic. *2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon)*, 104–110.

Aziz, Z., & Bestak, R. (2024). Insight into anomaly detection and prediction and mobile network security enhancement leveraging k-means clustering on call detail records. *Sensors*, *24*(6), 1716.

Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, *36*(1), 51–60.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., & Veeramachaneni, K. (2020). Tadgan: Time series anomaly detection using generative adversarial networks. *2020 ieee international conference on big data (big data)*, 33–43.

Godt, I. (2005). Music: A practical definition. *The Musical Times*, *146*(1890), 83–88. Retrieved April 10, 2024, from http://www.jstor.org/stable/30044071

Grosche, P., & Muller, M. (2010). Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(6), 1688–1701.

Harrou, F., Bouyeddou, B., Zerrouki, N., Dairi, A., Sun, Y., & Zerrouki, Y. (2024). Detecting the signs of desertification with landsat imagery: A semi-supervised anomaly detection approach. *Results in Engineering*, 102037.

He, Y., & Zhao, J. (2019). Temporal convolutional networks for anomaly detection in time series. *Journal of Physics: Conference Series*, *1213*(4), 042050.

Li, L., Yan, J., Wang, H., & Jin, Y. (2020). Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE transactions on neural networks and learning systems*, *32*(3), 1177–1191.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *2008 eighth ieee international conference on data mining*, 413–422.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14), 281–297.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*.

Nascimben, M., Abreu, H., Manfredi, M., Cappellano, G., Chiocchetti, A., & Rimondini, L. (2024). Extracellular vesicle protein expression in doped bioactive glasses: Further insights applying anomaly detection. *International Journal of Molecular Sciences*, *25*(6), 3560.

O'Shaughnessy, D. (1987). Speech communication, human and machine addison wesley. *Reading MA*, *40*, 150.

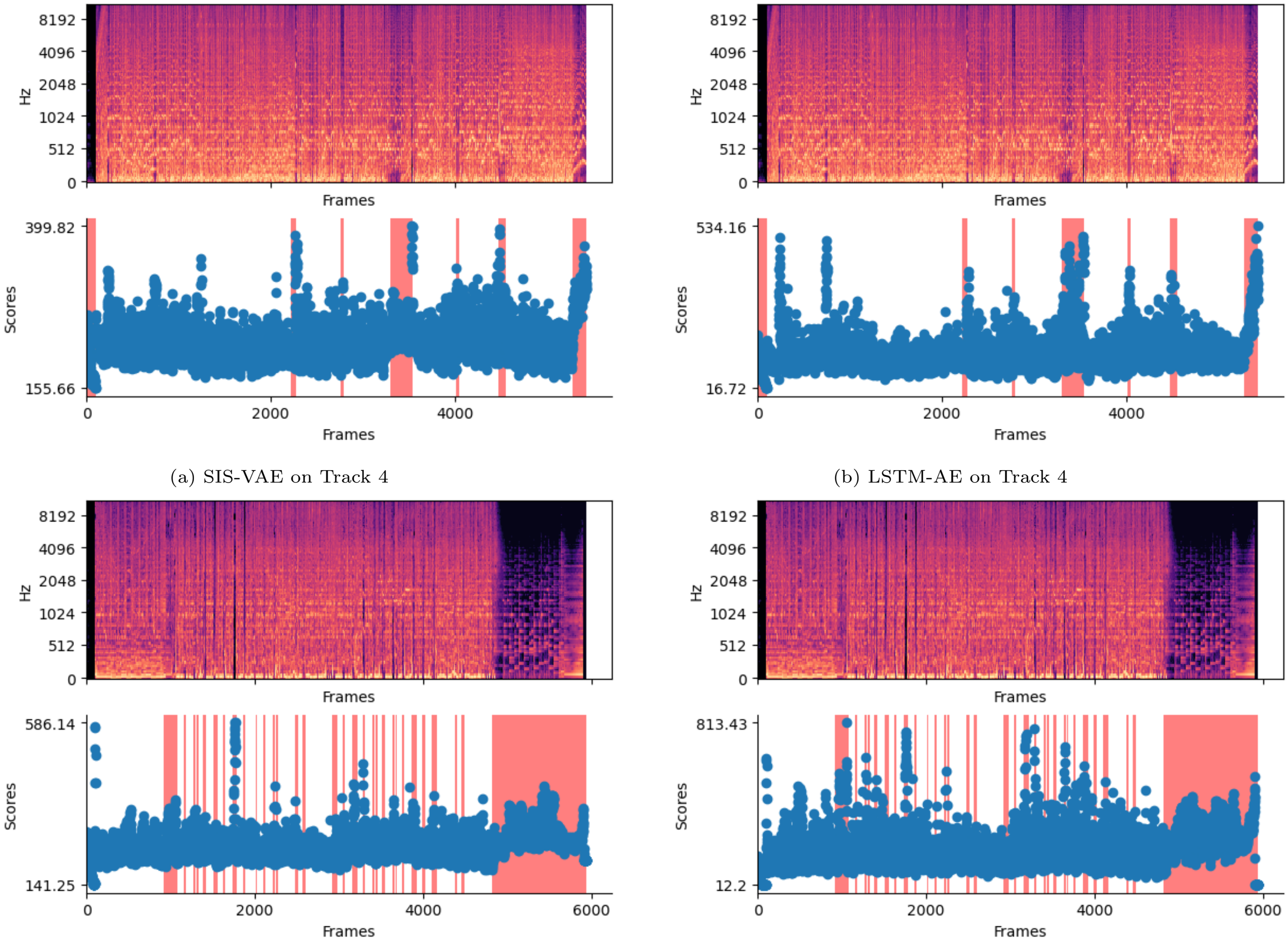
Prasad, S., & Raman, R. (2024). Isolation forests for anomaly detection in iot-enabled food quality monitoring system. *2024 International Conference on Automation and Computation (AUTOCOM)*, 364–368.

Si, H., Pei, C., Cui, H., Yang, J., Sun, Y., Zhang, S., Li, J., Zhang, H., Han, J., Pei, D., et al. (2024). Timeseriesbench: An industrial-grade benchmark for time series anomaly detection models. *arXiv preprint arXiv:2402.10802*.

Stepmania Data. (2012). https://search.stepmaniaonline.net/

Wagner, D., Michels, T., Schulz, F. C., Nair, A., Rudolph, M., & Kloft, M. (2023). Timesead: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research*.

A Extra plots of Deep learning methods



(a) SIS-VAE on Track 4

(b) LSTM-AE on Track 4

(c) SIS-VAE on Track 16

(d) LSTM-AE on Track 16

Fig. 9: Detection by deep methods on two sample song.