

***1e0a*: A COMPUTATIONAL APPROACH TO RHYTHM TRAINING**

Noel Alben

Georgia Institute Of Technology

noelalben@gatech.edu

Ranjani H.G.

IEEE Member

ABSTRACT

We present a computational assessment system that promotes the learning of basic rhythmic patterns. The system generates multiple rhythmic patterns with increasing complexity within various cycle lengths. For a generated rhythm pattern the performance assessment of the learner is characterized with summary statistics estimated from the onsets of the learner's performance and provided as feedback. When performance assessment feedback is within certain error bounds, the system proceeds to generate a new pattern of increased complexity. The system thus mimics a learner-teacher relationship as the learner progresses in their feedback-based learning. The choice of progression within a cycle for each pattern is determined by a predefined complexity metric. This metric is based on a coded element model for the perceptual processing of sequential stimuli. A model earlier proposed for a sequence of tones and non-tones, is now extended for onsets and silences. This system is developed into a web-based application and provides accessibility for learning purposes. Analysis of learner performance assessments from a pilot user study shows us that the complexity metric is moderately indicative of the perceptual processing of rhythm patterns and with further enhancements can be used for rhythm learning and future syllabus generation of rhythm training tools.

1. INTRODUCTION

Learning to play a musical instrument is a complex process that requires the acquisition of many interlinked skills such as tone, intonation, note accuracy, timing, rhythm precision, clear articulation, and dynamic variation [1]. An acknowledged important skill is the understanding of rhythm, timing, and rhythm precision. The word rhythm is used to refer broadly to all temporal aspects of musical works, and more specifically the duration and durational patterns of musical notes [2], [3]. Over the course of their musical education and understanding of rhythmic concepts one gradually starts to develop an intuition towards family of rhythmic patterns and similarity between different patterns, this intuition and understanding is usually the first step towards mastering the art and developing the skill set. Two main contexts within which formal musical learning takes place are the classroom, through playing and inter-

acting with a teacher, and at home, practicing alone [1]. From the perspective of a drum kit or percussive instrument, a typical classroom setting involves the teacher and a learner. The teacher decides what the learner must practice and the teacher's feedback helps them achieve accuracy in timing and quality in performance. The learner's practicing habits at home aid in reaching performance goals. A software application cannot substitute a music teacher but it can be a supplement in the learning process, especially in the performance assessment of daily practice at home [1].

1.1 Related Literature

Most assessment methods are either the assessment of learning, used to measure competence, or the assessment for learning used to enhance learning [4], [5], [6]. The rapid development of music technology over the last few decades has dramatically changed the way that people interact with music. In [7], the authors explore the use of Music Information Retrieval (MIR) techniques in music education and their integration in learning software. They outline Song2See as a representative example of a music learning system developed within the MIR community. Songs2See is a music game developed based on pitch detection, sound separation, music transcription, interface development, and audio analysis technologies [8]. Despite advancements in these sound analysis technologies, Eremenko V, Morsi A, et al. in [1], revealed that the performance assessment tools present in these applications are still not reliable and effective enough to support the strict requirements of a professional music education context. In [1], the authors review some of the work done in the practice of music assessment and present a proof of concept for a complete assessment system for supporting guitar exercises. Furthermore, they identify the challenges that should be addressed to further advance these assessment technologies and their useful integration into professional learning contexts, particularly those that will help learners to plan, monitor, and evaluate their learning progression. They also highlight the importance and requirement of presenting the assessment results to a learner in a way that promotes learning [1]. Similarly, Lerch, A, et al. in [9] surveyed the field of Music Performance Analysis (MPA) from the perspective of performance assessment and its ability to provide insights and interactive feedback by analyzing and assessing the audio of practice sessions. Furthermore, they highlight the necessity for individual assessment and curriculum in the context of a music education tool. The work of Chih-Wei Wu and Alexander Lerch in [10], presents an unsupervised feature

learning approach to derive features for assessing percussive instrument performances amongst a large data of audition performances, mimicking a judge in this scenario. Specifically, they proposed using a histogram-based input representation to sparse coding to allow the sparse coding to take advantage of temporal rhythmic information. Various (commercially available) solutions exist specifically for rhythm education and performance assessment. Examples for rhythm tutoring applications are Perfect Ear [11], Rhythm Trainer [12], and Complete Rhythm Trainer [13]. These form a class of mobile applications that help learners practice and assess their rhythm skills by following a set curriculum presented in each of these applications and touchscreen-based feedback for performance assessment. Other apps such as SmartMusic [14], Yousician [15], Music Prodigy [16], and SingStar [17] are available for music education purposes however, they do not have a dedicated rhythm training section. Also, none of the previous works address musical rhythm complexity measures as a feature to aid in a learner's progression. Evaluating various rhythm complexity measures based on how accurately they reflect human-based measures were studied by Eric Thul in his thesis [3]. Although his results suggest that none of the measures accurately reflect the difficulty humans have performing or listening to rhythm, the measures do accurately reflect how humans recognize a rhythm's structure.

1.2 Our Contribution

In this work, we consider the individual assessment of learning a particular rhythmic pattern at a fixed tempo. We present the proof of concept for a system that will assess a learner's audio recording and provide performance feedback to them. We will only focus on assessing the learner's temporal proficiency, keeping the performance parameter categories of tempo, dynamics, pitch, and timbre fixed. Our main contribution is employing a pre-defined complexity metric found in [3] to enable the learner to progress from one rhythm pattern to the next, similar to a teacher who decides what the learner must practice.

This manuscript is organized as follows, we first define the terms, notations, and conventions to be held throughout that will help assess the performance of an individual. Followed by the proposed system for the assessment where we go over the subsequent blocks that make up the assessment system and understand the coded element model used to define the complexity metric. Finally, we discuss the pilot user study results and evaluation of this system through our web application platform and a conclusion with the future enhancements and improvements to the system.

2. TERMINOLOGY

2.1 Definitions

We first provide a brief background of some of the terms used to define a rhythmic pattern.

In this work, we consider the fundamental and individual time unit for a rhythmic pattern to be a *pulse*. Pulse is defined to be one of a series of regularly recurring, precisely equivalent stimuli [18]. The duration of a rhythmic pattern

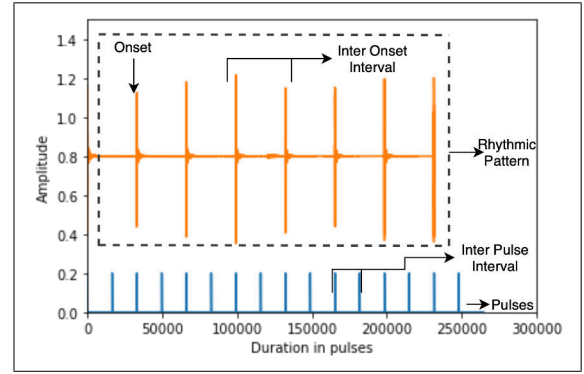


Figure 1. An illustrative diagram of the terms used in this work.

is composed of a number of pulses. Tempo refers to the rate at which the musical pulses are generated and it also controls the number of pulses for a given duration. For our proposed system the total duration of a rhythmic pattern is determined by the total number of samples present in the audio signal. The uniform number of samples or duration, between two subsequent pulses, is defined to be the inter-pulse interval (IPI). The IPI depends on the sampling frequency and the tempo used for the rhythmic pattern. Figure 1 shows a duration of 6.80 seconds with pulses generated at a fixed Inter Pulse Interval (IPI), determined by a tempo of 160 Pulses Per Minute (PPM).

To generate a rhythmic pattern, we consider instantaneous attacks on the marked time units (pulses) i.e., striking a drum, clapping, or tapping. The instantaneous attacks are defined as onsets. The duration that is marked between two onsets is defined as the inter-onset interval (IOI). The IPI, onsets, and the IOI together create a rhythmic pattern as shown in figure 1.

A cycle determines the number of pulses beyond which the rhythmic pattern begins to repeat itself. The cycle represents a structure for the pulses, and this structure is realised by the rhythmic pattern. For example, we use 7 to indicate a repeating rhythmic pattern of 7 pulses. Rhythm patterns can be composed in any predetermined cycle number, for this article we will only be following cycles of 4,3,5,7, and 16.

2.2 Representation

We will now introduce the rhythm pattern representation used in the proposed system with the help of a famous Afro-Cuban rhythm, the clave son, shown in figure 2 (a).

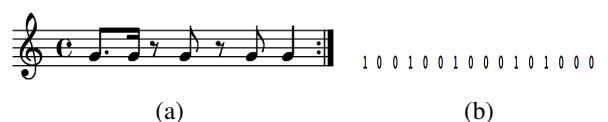


Figure 2. (a) The clave son rhythm in music (stave) notation. (b) The clave son rhythm in binary notation.

Figure 2 (b) shows the clave son rhythm as a binary string, a notation that describes the position of the onsets

and silences. This notation is used in the domain of computer science [19]. Each digit represents a single pulse, with the total number of digits indicating the cycle length. The value of the digit itself indicates whether, in a particular rhythmic pattern, an onset occurs on this pulse (1) or not (0). This representation enables faster computation and integration of rhythm patterns into our system. This representation of a rhythm pattern henceforth will be referred to as the binary representation.

2.3 Naming *1e0a*

The name *1e0a* for the proposed system is derived from a mixture of the binary representation with its 1's and 0's and a western rhythm counting technique used by a large majority of music educators, called the Harr system. The system that Harr used assigned a set of numbers and syllables. For a cycle of length 4, numbers are assigned to onsets that occur at the beginning of the rhythmic pattern's duration and its repeats. Syllables are assigned to those onsets that fall everywhere else. The syllables remain consistent for all the pulses with only the number for each repeat changing. For example, the rhythm [11111111] would be read, *1 e & a 2 e & a* [20]. In our work, we adopt the same system, we use the name *1e0a*.

3. PROPOSED SYSTEM

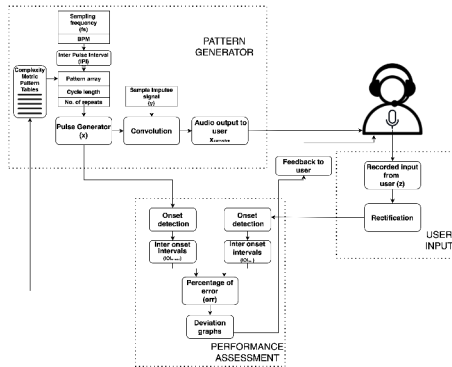


Figure 3. Block diagram of the proposed system for *1e0a*.

Figure 3 shows the block diagram of our proposed system for rhythm training and progression in learning. This system consists of two distinct blocks: (1) The rhythm pattern generator with the complexity metric, and, (2) the performance assessment of the learner with feedback.

3.1 Pattern Generator

3.1.1 Pattern Generation

The pattern generator as shown in figure 5 begins with a pulse generator function. This function takes input parameters that determine the cycle length, pattern array, tempo (PPM), and the total number of repeats, required to generate a collection of pulses whose inter-pulse interval (IPI) is determined by the equation 1.

$$IPI = (60/PPM)/(1/f_s) \quad (1)$$

PPM represents the pulses per minute of the generated collection of pulses, f_s represents the sampling frequency and *IPI* represents the fixed inter-pulse interval.

The pattern array in binary representation determines the location of onsets and the number of silences to be incorporated into the duration of pulses. We also propose the selection of pattern arrays for generating subsequent sequences using complexity metrics detailed in section 3.1.2. For the clave son: [1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0], at 160 *PPM*, with 4 repeats, at an f_s of 44100. The function generates an array of pulses as shown in figure 4(a).

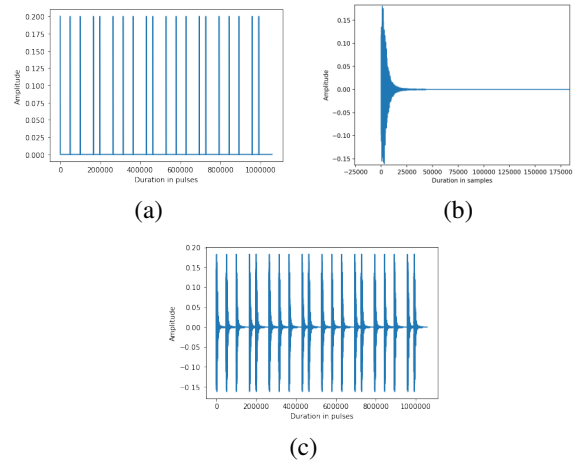


Figure 4. Pattern Generator plots. (a) Pulse Generator Output; (b) Hi-Hat Drum sample impulse signal; (c) Convolution result of (a) and (b).

The next stage of the pattern generator block adds tonal quality to the output of the pulse generator, by convolving the pulse generator output with an appropriate drum sample impulse signal. The drum sample impulse signal is a short recording of an onset played using a particular drum sound. The drum sound selected must have an instantaneous attack and a short decay as shown in figure 4(b). The convolution process requires the loaded sample impulse to have the same f_s as the pulse generator output. For a pulse generator output and drum impulse, figure 4(c) shows the output of the convolution function. We make use of the Discrete Fourier Transform convolution property [21]. This allows our result to have the same number of samples as the pulse generator output. We use the Fast Fourier Transform (FFT) [21] algorithm to achieve this result. The resultant clave son rhythm from the pattern generator block, at 160 *PPM* and 4 cycles, is an audio signal whose plot is shown in figure 4(c).

3.1.2 Complexity Metric

Various measures of musical rhythm complexity have been studied in [3]. Rhythm complexity measures have been studied as:

Rhythm syncopation measures that use metrical weights determined from the metrical hierarchy to find syncopated

onsets in a rhythm [[19], [22]].

Pattern matching measures which chop up a rhythm as determined by the levels of the metrical hierarchy and search each sub-rhythm for patterns that indicate the complexity [[23], [24]].

Distance measures, which measure how far away a rhythm is from a more simple rhythm composed of beats [19], [25], [26]].

Shannon's information entropy [27], [28], which are mathematical measures which are based upon **Shannon's information entropy** uncertainty equations given in equation 3, 4, 5.

We estimate the rhythmic complexity measure using information entropy approach by estimating the entropy for a particular rhythm pattern. Vitz and Todd in [28], present a method that can be used to calculate the complexity of cyclic binary patterns. Their work defined that the complexity measure of a certain periodic or repeating pattern is the sum of the total amount of uncertainty evaluated in processing the stimulus sequence. In [28], the authors achieved this by proposing a set of rules (axioms) describing a perceptual coding process. As the method implies an emphasis on coded elements, it is referred to as a coded element processing system (CEPS) [3].

We consider this model for our application due to its algorithmic approach to measure rhythm complexity, thus making it viable for coding, as against rhythm syncopation models in [19]. CEPS also has an improved ability to distinguish rhythm patterns within a cycle based on complexity as compared to pattern matching [23], [24] and distance measures [19], [25], [26].

The axioms in [28], proposed originally for tones and non-tones, break down sequences into code levels. In our work, we extend it to rhythmic patterns, composed of onsets and silences. Each code level successfully codes the pattern from single elements to larger elements which continues until the entire pattern is constructed. To calculate the complexity of the rhythmic pattern, we measure two uncertainty values at each Code Level in the algorithm as proposed in [28]. Uncertainty in this case means information entropy. The uncertainty values are the maximum uncertainty, H_{max} , and the joint uncertainty, H_{joint} represented in equations 2 and 3.1.2 respectively. Let X, Y, Z, W represent random discrete variables.

$$H_{max}(X, Y, Z, W) = H(X) + H(Y) + H(Z) + H(W) \quad (2)$$

$$H_{joint}(X, Y, Z, W) = H(X) + H(Y|X) + H(Z|X, Y) + H(W|X, Y, Z) \text{ where,}$$

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x) \quad (3)$$

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x)p(y) \log_2 p(x, y) \quad (4)$$

$$H(X|Y) = - \sum_{x \in X} \sum_{y \in Y} p(y, x) \log_2 p(y/x) \quad (5)$$

Let the value H^k be the sum of H_{max}^k and H_{joint}^k at Code Level k . The total sum of H at each Code Level represents the complexity of the pattern. This value is termed H_{code} .

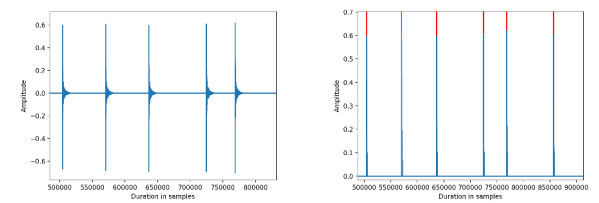
$$H_{code} = \sum_{k=1}^n H^k = \sum_{k=1}^n H_{max}^k + H_{joint}^k \quad (6)$$

The complexity measure for [1001001000101000] is calculated to be **20.5538**. The complexity measures are pre-calculated for all the combinations of patterns within the cycle lengths of 4,3,5, and 7. Then, the patterns are ordered based on increasing complexity as shown in *sheets*¹. We observed that for patterns that are a shifted version of another, where 1's (onsets) replace the 0's (silences), and vice versa, the complexity measures are identical. For example, [1100] and [0011] have a complexity measure of 5.0. To tackle this, we considered those patterns that begin with a 0 as off-beat patterns and assumed that they present a higher challenge to learners. Hence, we incremented a constant value to the complexity measures for all those patterns that begin with a 0 (silence). The constant value within a cycle was determined by the highest complexity measure for the patterns that begin with 1 (onsets), for that cycle length. This can be observed in *sheets*¹.

However, this complexity metric does not account for the complexity that arises with changes in tempo. Therefore in this work, we focus on studying the performance assessment at a constant *PPM*.

3.2 Performance Assessment

Once the learner listens to the generated pattern their performance is recorded and loaded onto the system. We rectify the recorded audio signal (z) with a threshold, $z_{threshold} = z_{max}/4$, to improve the results of the onset detection function and to account for low magnitude noise in the live recording environment.



(a) Recorded input. (b) Onsets from rectified inputs.

Figure 5. Plots of onset detection in one cycle.

Any onset detection function can be used to extract the onsets from the rectified recorded input. We used the onset detection function defined by the Librosa python library [29]. A sample of the recorded input and the extracted onsets are shown in figure 5.

Inter onset intervals are calculated from onsets estimated from both recorded (IOI_{input}) and generated signals ($IOI_{generated}$).

¹ https://docs.google.com/spreadsheets/d/e/2PACX-1vTSDSqi1264H6-M2kLdQUuD6VYnRqYPQePhKl8CSTD1j-54rFdhbLag-ep_15ggiHpo8-DOPJWYQ-O/pubhtml

| | | | | |
|--------|-------|------|-------|--------|
| [0.15 | -0.04 | 0.01 | -0.14 | 0.01] |
| [-0.04 | 0.15 | 0.01 | -0.14 | 0.01] |
| [-0.04 | -0.04 | 0.01 | 0.15 | 0.01] |
| [-0.04 | -0.04 | 0.01 | 0.15 | -0.14] |

Figure 6. 4X5 Matrix of error deviation result.

The deviation: $IOI_{generated} - IOI_{input}$, for each inter onset interval is calculated to be the error in temporal performance at each onset. If the user inputs different number of onsets as compared to the generated pattern, we pad the audio if it's lower or truncate the audio if it's longer.

The error is recorded in a matrix format where the number of columns is equivalent to the number of onsets in a cycle and the number of rows is determined by the total number of cycles. For the son clave rhythm, there are 5 *l*'s (onsets) and 11 *0*'s (silence) played for 4 cycles, resulting in a 4X5 matrix as shown in 6.

From the error matrix, two sets of averages are calculated. One set of average values represent the deviation of each inter onset interval within a cycle (average of all the values in one column of the error matrix). This results in a 5X1 average onset error array. This average onset error array is plotted on a graph with the horizontal axis representing each onset and the vertical axis representing the percentage of deviation (-100 to 100) This is shown in figure 7(left). This result indicates to the learner their timing error at each onset. The negative error implies slowly playing speed and the positive indicates fast playing speed.

The next set of average values represent the error deviation of a learner across the cycles (average all the values in one row of the error matrix) which results in a 4X1 average result array. This result is plotted on a graph with the horizontal axis representing the number of cycles and the vertical axis representing the deviation (-100 to 100) shown in figure 7(right). This result gives the learner information on their consistency throughout the performance and indicates if the timing error is consistent across cycles.

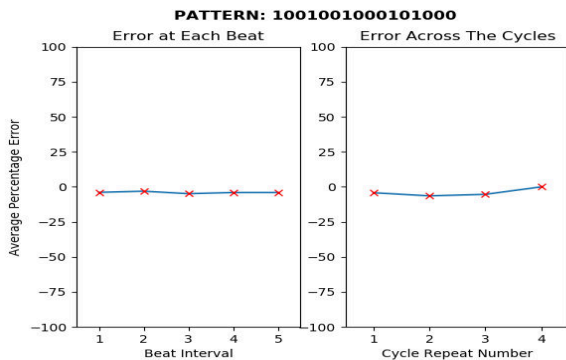


Figure 7. Deviation graph results of the performance assessment block.

The graphs in figure 7 give the learner comprehensive performance feedback concerning their timing and consistency. We achieve this through the graphical representation wherein, a positive deviation (above 0 line) in figure 7 (left) infers that the learner is rushing or playing the beat faster

than intended and a negative deviation (below 0 line) infers that the learner is dragging or playing the beat slower than intended. Each onset in figure 7 (left) is marked by an *x*.

4. 1E0A - THE WEB APPLICATION

For improved accessibility to this system to aid in learning, we developed a web application hosted at <https://1e0a.noelalben.com>

Our goal was to implement the proposed system with a simple user interface and a server to collect learners' data. We have been able to achieve both by using Flask a micro web framework written in Python [30]. The website was hosted on AWS [17] with certbot SSL certification [31].



(a) Learning platform (b) Performance Assessment results

Figure 8. 1e0a Web Application.

4.1 Data collection

Every new learner must register with the website and provide consent to use their registration information and performance assessment results for research purposes. Post-registration the learner is given a login id and password. After logging in to the system, they enter the learning platform as shown in figure 8 (a). They must listen to the generated pattern and record their respective performance. The *Analyze* button conducts performance assessment on the learner's recording and takes them to the results as shown in figure 8 (b). If they choose to progress in their learning by clicking *Learn Another*, the system will generate the next pattern based on the complexity metrics defined in section 3.1.2 and take the learner back to the learning platform. When they log out, the most recent generated pattern is stored into the database and they can continue where they left, upon logging in again.

5. PILOT STUDY

5.1 Setup

To assess and quantify the effectiveness of our rhythm learning system, each learner on the 1e0a web application was required to perform and record 23 distinct patterns over 4 different cycle lengths of (4,3,5 and 7), at a constant PPM of 160. The learners were asked to use a desktop/laptop to access the web application and to maintain a constant interface to perform the rhythmic patterns (clapping or a single percussive instrument). They were not allowed to use accents in their performance nor take breaks while performing a given pattern. Throughout the experiment, the application recorded the number of attempts

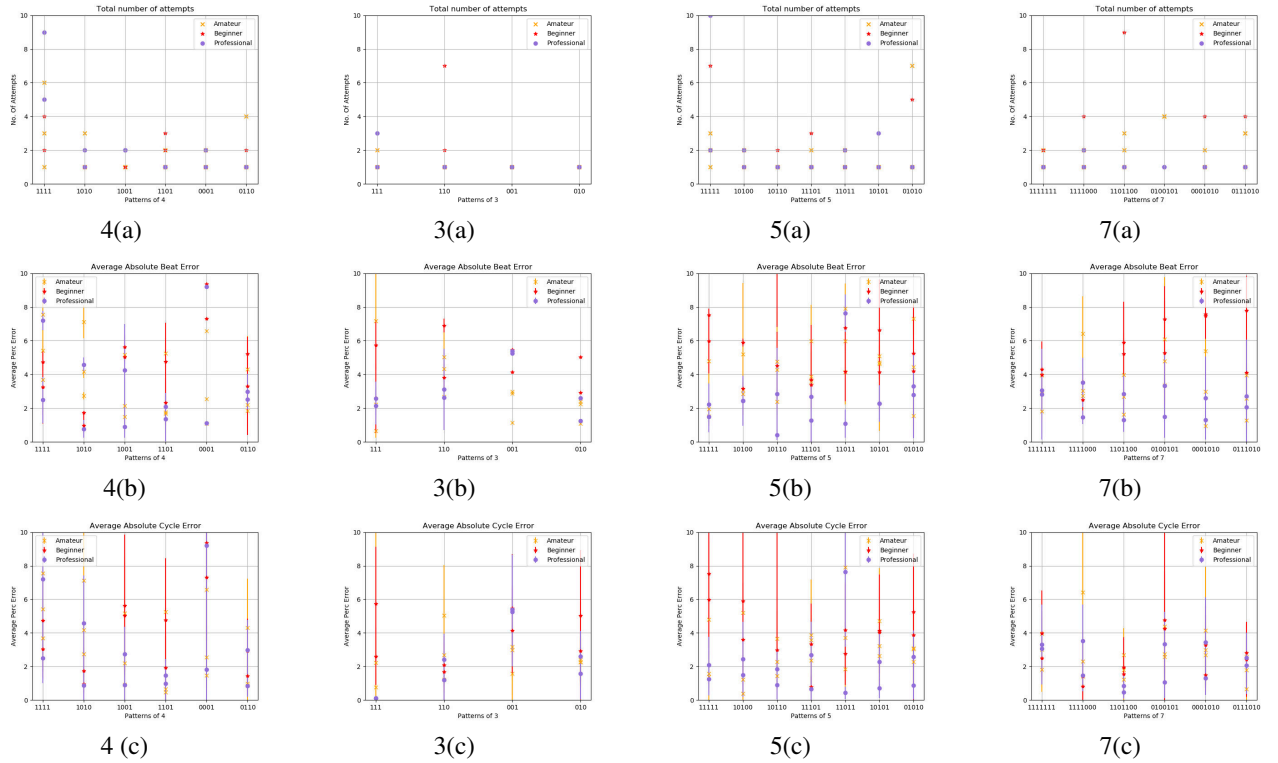


Figure 9. Graphical analysis of performance assessment for 7 learners categorised by professional(●), amateur(×), and beginner(★) on the 1e0a web application at each cycle. (a)Top Panel: Total number of attempts taken by learners. (b) Center Panel: Absolute average beat error in performance. (c) Bottom Panel: absolute average error over the cycle in performance.

each learner takes to achieve performance assessment results within the error bounds presented in section 3.2. Furthermore, we stored the performance assessment results of the learner for each rhythmic pattern where their results were within the error bounds. The results of 7 learners are shown in fig 9. Each learner is categorized based on their years of experience with music education: (2) Professionals (>10 years), (3)Amateur (>2 years), and (2) Beginners(0-2 years).

5.2 Results

From the resulting graphs in figure 9, it can be observed that compared to an amateur or a beginner, for any given pattern the professional requires less number of attempts to achieve performance accuracy within the error bounds. This is reflective of a learning system where beginners make more mistakes and progressively improve in their performance as they spend more time on the application. We also observe that the learners were able to correct their mistakes and achieve performance accuracy within the error bounds after a sufficient number of attempts. This indicates that the performance assessment feedback of the system aids the learner in understanding and correcting their performance to achieve accuracy. Also, it can be observed that across different patterns within the cycle there exists complexity because the number of attempts taken by the

learner increases with the progression of patterns within a cycle. However, this does not directly reflect the difficulty that learners have while performing or listening to the rhythm; but does accurately reflect how beginners and amateurs struggle to recognize a rhythm’s structure as the complexity increases. For this analysis we did not look into those attempts where the user made mistakes due to the constraints in storage within the application.

6. CONCLUSION

In this article, we introduce *1e0a*, the proof of concept for a rhythm learning and assessment system that helps learners progress in their learning with a new pattern generated based on a pre-defined rhythm complexity measure. We describe the rhythm complexity measure that was used and present the methodology for rhythm performance assessment. The pilot results of the proposed system show that we can successfully test a learner’s proficiency on rhythm patterns over a single PPM and monitor their progression within a cycle of patterns of increasing complexity values. This system does not test the learner’s proficiency across different tempos neither does it test for mixed cycle complexity.

The future work is aimed at perceptual evaluation on the pedagogical significance of presenting the rhythms progressively using this complexity metric. We are also fo-

cused on building a rhythm generative algorithm using relevant rhythm similarity metrics to obtain a robust syllabus that extends to accented notes and mixed cycle length. Furthermore we would like to improve the system as a learning tool and extend it's capabilities in onset detection, noise removal and user interaction.

7. REFERENCES

- [1] V. Eremenko, A. Morsi, J. Narang, and X. Serra, "Performance assessment technologies for the support of musical instrument learning," 2020.
- [2] J. London, "Rhythm in l. macy, editor, grove music online. oxford university press," 2008.
- [3] E. Thul, "Measuring the complexity of musical rhythm," 2008.
- [4] R. Mantie, "The philosophy of assessment in music education," in *The Oxford Handbook of Assessment Policy and Practice in Music Education, Volume 1*. Oxford University Press, 2019, p. 33.
- [5] M. C. Schneider, J. S. McDonel, and C. A. DePascale, "Performance assessment and rubric design," in *The Oxford Handbook of Assessment Policy and Practice in Music Education, Volume 1*, 2019.
- [6] B. C. Wesolowski, "Understanding and developing rubrics for music performance assessment," *Music Educators Journal*, vol. 98, no. 3, pp. 36–42, 2012.
- [7] C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch, "Music information retrieval meets music education," in *Dagstuhl Follow-Ups*, vol. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.
- [8] E. Cano, C. Dittmar, and S. Grollmisch, "Songs2see: learn to play by playing," in *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. Miami USA, 2011, pp. 2231–2240.
- [9] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "Music performance analysis: A survey," *arXiv preprint arXiv:1907.00178*, 2019.
- [10] C.-W. Wu and A. Lerch, "Learned features for the assessment of percussive music performances," in *2018 IEEE 12th International conference on semantic computing (ICSC)*. IEEE, 2018, pp. 93–99.
- [11] "Crazy ootka software ab, perfect ear; version: 3.8.71," last accessed 06/02/2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.evilduck.musiciankit>
- [12] Demax, "Rhythm trainer; version:0.2104261849," last accessed 06/02/2022. [Online]. Available: <https://play.google.com/store/apps/details?id=ru.demax.rhythmerr>
- [13] S. Dupont, "Complete rhythm trainer; version:1.3.10," last accessed 06/02/2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.binaryguilt.completerhythmtrainer>
- [14] I. MakeMusic, "Smartmusic, april 2019," last accessed 06/02/2022. [Online]. Available: <https://www.smartmusic.com>
- [15] Y. Oy, "Yousician, april 2019," last accessed 06/02/2022. [Online]. Available: <https://www.yousician.com>
- [16] I. d. M. P. The Way of H, "Music prodigy, april 2019," last accessed 06/02/2022. [Online]. Available: <http://www.musicprodigy.com>
- [17] S. I. E. Europe, "Singstar, april 2019," last accessed 06/02/2022. [Online]. Available: <http://www.singstar.com>
- [18] G. W. Cooper and L. B. Meyer, "The rhythmic structure of music. the university of chicago press, 1960."
- [19] G. Toussaint, "A mathematical analysis of african, brazilian, and cuban clave rhythms," in *Bridges: Mathematical Connections in Art, Music, and Science*, 2002, pp. 157–168.
- [20] B. A. Janssen, *A preliminary comparative study of rhythm systems employed within the first-year college aural skills class*. Kansas State University, 2017.
- [21] J. G. Proakis, *Digital signal processing: principles algorithms and applications*. Pearson Education India, 2001.
- [22] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception*, vol. 1, no. 4, pp. 424–441, 1984.
- [23] J. Pressing, "Cognitive complexity and the structure of musical patterns," in *Proceedings of the 4th Conference of the Australasian Cognitive Science Society*, 1999.
- [24] M. Keith, "From polychords to polya: adventures in musical combinatorics," 1991.
- [25] G. Toussaint, "Classification and phylogenetic analysis of african ternary rhythm timelines," in *Meeting Alhambra, ISAMA-BRIDGES Conference Proceedings*, 2003, pp. 25–36.
- [26] F. Gómez, A. Melvin, D. Rappaport, and G. T. Toussaint, "Mathematical measures of syncopation," in *Renaissance Banff: Mathematics, Music, Art, Culture*, 2005, pp. 73–84.
- [27] P. C. Vitz, "Information, run structure and binary pattern complexity," *Perception & Psychophysics*, vol. 3, no. 4, pp. 275–280, 1968.
- [28] P. C. Vitz and T. C. Todd, "A coded element model of the perceptual processing of sequential stimuli," *Psychological Review*, vol. 76, no. 5, p. 433, 1969.
- [29] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

- [30] M. Grinberg, *Flask web development: developing web applications with python.* ” O'Reilly Media, Inc.”, 2018.
- [31] C. Tiefenau, E. von Zezschwitz, M. Häring, K. Kromholz, and M. Smith, “A usability evaluation of let’s encrypt and certbot: usable security done right,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1971–1988.