Symbolic Representation-based Melody Extraction using Multiclass Classification for Traditional Javanese Compositions

Arry Maulana Syarif[1], Khafiizh Hastuti[4]
Faculty of Computer Science
Universitas Dian Nuswantoro
Semarang, Indonesia

Azhari Azhari[2], Suprapto Suprapto[3]
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia

***Abstract*—Traditional Javanese compositions contain melodies and skeletal melodies. Skeletal melodies are an extraction form of melodies. The melody extraction problem is similar to the chord detection in Western music, where chords are extracted from a melody. This research aims to develop a melody extraction system for traditional Javanese compositions. Melodies which have a time series data structure were designed as a part of the supervised learning problem to be solved using the pattern recognition technique and the Feed-Forward Neural Networks method. The melody data source uses a symbolic format in the form of sheet music. The beats in melodies data are used as the input and notes in skeletal melodies are used as the target. An FFNN multi-class classifier was built with six classes as the targets, where the class represents notes of the musical scale system. The network evaluation was conducted using accuracy, precision, recall, specificity and F-1 score measurements.**

***Keywords*—Melody extraction; symbolic representation-based; multiclass classification; feed-forward neural network; Gamelan**

I. Introduction

This research is part of a program to preserve traditional Javanese music using artificial intelligence methods with the expectation of preserving the authenticity of the compositions throughout the ages. Traditional Javanese compositions known as Gamelan music consist of melodies and balungan (Javanese: skeletal melodies). The Gamelan composers create compositions by first composing a melody and then extracting it into a skeletal melody, or constructing a skeletal melody first which is then filled with harmonization into a melody. The melody extraction to form skeletal melodies is chosen as the background problems in this research. Skeletal melodies can be analogous to chords in Western music, and the challenge in this research is similar to the problem of determining chords to accompany the melody.

Chord detection uses time series data, and such data structures can be designed as part of a supervised learning problem to be solved using pattern recognition techniques. Chords are detected by extracting features from audio sources, filtering and matching the patterns [1], and this method has been used in various works [2-7]. Instead of using audio sources and performing feature extraction, a symbolic representation approach is proposed using sheet music as the dataset source. Learning directly from sheet music is to get original and complete information of the musical elements that is difficult to obtain through feature extraction from audio sources. Hence, a new method for a symbolic representation-based melody extraction was proposed by recognizing the note sequence pattern based on musical theory, which is carried out by calculating the duration of the notes. The proposed method in this research is in line with what [8] stated, the music theory approach without audio can be used as a complementary technique in the field of music information retrieval. In a different context, musical theory is disproved by using chord sequences found in the dataset so that theoretically unusual chord sequences are possible to learn [9]. The proposed method is also similar to that stated by [9] in the context of using all the sequences of notes or chords found in the dataset but the metrical structure in musical theory is still used as a reference to avoid metrical structure errors in composition. Further, a multi-class classifier using the Feed-Forward Neural Networks (FFNN) method was used to build a melody extraction system for traditional Javanese compositions. The FFNN network was trained using melodies as input and skeletal melodies as the output.

The availability of datasets is a challenge in building a melody extraction system for traditional Javanese compositions. Unlike western music, which has a well-organized composition documentation system that supports easy data access, traditional Javanese composition data in sheet music format is not well documented and difficult to access online. This causes a limited number of datasets. Data augmentation is a challenge in itself, and proper data mapping techniques are needed to increase the cardinality of the data.

This paper is structured as follows. Section II introduces traditional Javanese compositions. Section III describes the related work of chord detection which has in principle a similar task to melody extraction, as well as research on traditional Javanese compositions utilizing an AI approach. Section IV describes the methodology used in developing a melody extraction system for traditional Javanese compositions which consists of data preparation, beat detection, vector length adjustment, data mapping and feature selection, and binary representation. Section V discusses training and evaluation. Finally, Section VI discusses conclusions and future works.

II. The Traditional Javanese Composition

Traditional Javanese music called karawitan consists of a set of music instruments called Gamelan and compositions with or without vocal called gendhing. Gamelan consists of two musical scale systems, which are pelog and slendro. The pelog scale system consists of seven notes: 1, 2, 3, 4, 5, 6, 7. The slendro scale system consists of five notes: 1, 2, 3, 5 and 6. The pelog and slendro scale systems are different in their tuning. Moreover, there are dotted notes as addition that represent moments of silence. Based on their function in performing the composition, a set of Gamelan instruments is divided into three groups, which are ricikan garap, ricikan balungan and structural ricikan. The group of ricikan garap contains instruments to play the melody parts, such as gender, rebab, suling and gambang. The group of ricikan balungan contains instruments to play the skeletal melody parts, such as saron, demung, peking and slenthem. The group of structural ricikan contains instruments to play notes that form the type of compositions, such as kethuk, kempyang, kenong, kempul and gong.

There is a musical mode system called pathet on both the pelog and slendro scale systems. This system controls the dominant notes at certain positions in the sequence. The slendro mode system consists of manyura, nem and sanga, while the pelog mode system consists of barang, lima and nem. There are types of compositions, such as ladrang, lancaran and ketawang. The type of compositions is determined based on the number of beats in the skeletal melody, and it can be identified based on the play of the instruments of the group of structural ricikan. Fig. 1 shows illustration of the traditional Javanese compositions and Gamelan known as Gamelan music.

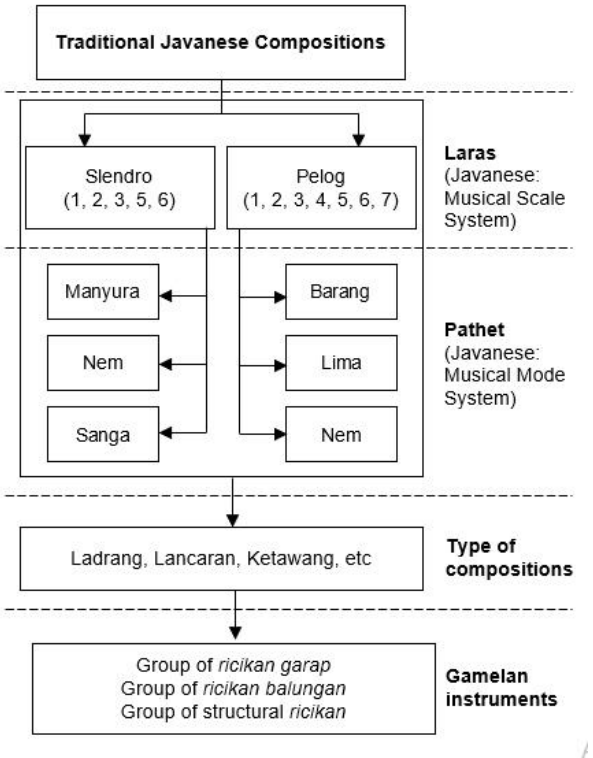

Fig. 1. Ilustration Traditional Javanese Compositions and Gamelan Instruments.

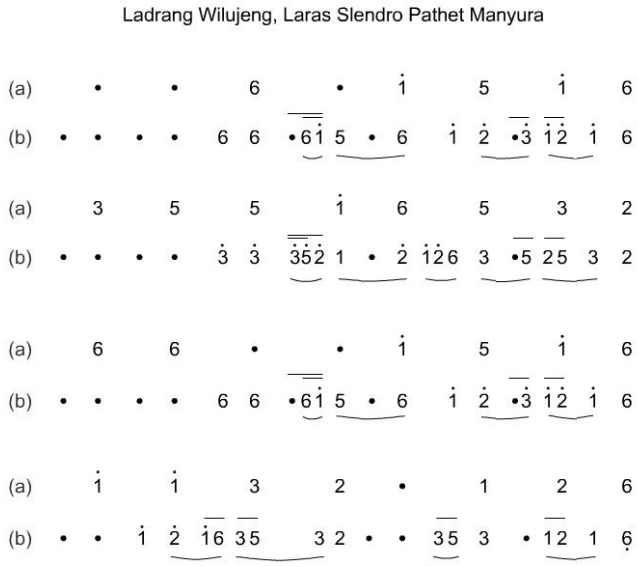

Fig. 2. A Traditional Javanese Sheet Music Example of a Composition Entitled Ladrang Wilujeng.

Time signature in Gamelan music is known as tempo, and it is divided into 1/1, 1/2, 1/4, 1/8 and so on. Tempo represents the note duration of beats of melodies and skeletal melodies. Tempo of 1/1 means that each beat in the melody and skeletal melody has note duration of 1, tempo of 1/2 means that each beat in the skeletal melody has note duration of 1 and each beat in the melody has note duration of 2. The note duration consists of 1, 0.5 or 0.25, and in the sheet music it is indicated by a single or double horizontal lines above the notes. The notes without any horizontal line have a note duration of 1, a single horizontal line represents a note duration of 0.5 and a double horizontal line represents a note duration of 0.25.

Musical element symbols, such as circular symbols and curves symbols above the notes of the skeletal melody, represent the type of the composition. Curve symbols below the notes in the melody represent the legato sign which is similar to those in Western music, horizontal line symbols above the notes in the melody represent the note duration, dotted notes above and below the notes in the melody represent the notes register (low, middle and high notes). Fig. 2 shows an example of a traditional Javanese music sheet of a composition played with a tempo of 1/2. The composition consists of four lines of the skeletal melody marked with (a) and the melody marked with (b).

III. Related Work

Pattern recognition are common for chord detection problems in which chords are detected by extracting features from audio source, giving filter and matching the pattern [1]. This approach was implemented in various works by [2-7], and the method usually uses audio signal as the input source, and a features extraction is conducted to set an input by selecting relevant features from the audio source. Features for chord detection are called pitch class profile (PCP), or known with chroma features, which consists of 12 semi-tones values attributes. Chroma features are still proven as the main features

in chord detection [10]. These 12 semi-tones were used to set 12 bin vectors for faster processing [4], while 178 bin vectors [3] and 180 bin vectors [7] were set from variations of 12 semi-tones to set a frequency range. Feature extraction for random variables data can be conducted using the $\chi 2$ statistics method in which the target and features are discrete finite values [11].

The decomposition of each chord label into a meaningful set of musical components was used to overcome the problem of insufficient sample size for model training in chord data quality [12]. Meanwhile, target and features can be determined based on data segmentation technique then followed by supervised learning implementation [13], and this technique can also increase the number of corpuses. Sequence mapping technique is used to calculate weighted moving average based on previous data of a time ordered sequence, such as works to predict stock index that implemented sliding window technique to map sequence data [14]. So, features for chord detection or melody extraction that uses a dataset collected from symbolic data can be determined by data segmentation and followed by data mapping using sliding window technique as proposed in this research. Sequence padding and sequence truncation are common techniques to solve a vector length problem in a time series prediction. Sequence padding adds a number of zeroes in the beginning (pre-sequence padding) or in the ending (post-sequence padding) of vectors as much as the maximum number of the vector's length. While sequence truncation chops a number of elements of vector in the beginning (pre-sequence truncation) or in the ending (post-sequence truncation) of vector to obtain a defined number of vector length. Sequence padding is better to find pattern in given data than to predict based on previous data [15]. This is also proven in a pre-experiment conducted in this research in which the use of sequence truncation achieves higher prediction on accuracy then sequence padding.

Several computer and music researches have been conducted to generate a note sequence of skeletal melodies. The grammar approach was used to formulize note sequence patterns of bars of the skeletal melody [16]. Meanwhile, the grammar based on a bar structure was analyzed to define note sequence patterns of bars of the skeletal melodies [17]. The rule-based method used for the solutions of the same problem, but the formulation includes note sequence patterns between bars, and then the note sequence rules were determined by segmenting data using the sliding window technique [18]. Further, the rules were implemented as constraints to generate note sequences of skeletal melodies using Genetic algorithm [19]. Different from existing researches, this research aims to extract note sequences of skeletal melodies from note sequences of melodies.

IV. Methodology

A feed-forward neural networks classifier with supervised learning approach and pattern recognition technique was proposed to build a melody extraction system for traditional Javanese compositions. The task of the classifier is to extract melodies into skeletal melodies, where the class is determined based on the notes of the musical scale system. Data of compositions are segmented into beats to reveal the patterns of the notes correlation between melodies and skeletal melodies as illustrated in Fig. 3. This technique can increase the number of corpuses, as more corpus results in better accuracy in the FFNN method. For example, a composition containing six lines contributes 12 bars and 48 beats if the beats are used as the corpus.

A collection of music sheet used as the data source was manually converted into a text-based format for computation process. Each composition data consists of melodies used as input, and skeletal melodies used as target. The difference in the number of notes in a melody beat determined by the note duration has an impact on the difference in vector length. The vector length adjustment was performed so that all vectors have the same element length as the FFNN input requirements. Further, the beats are mapped to restructure the data into time series format data and to determine features and to increase the cardinality of corpuses. Finally, the results of data mapping were converted into the binary format before being sent to train the network.

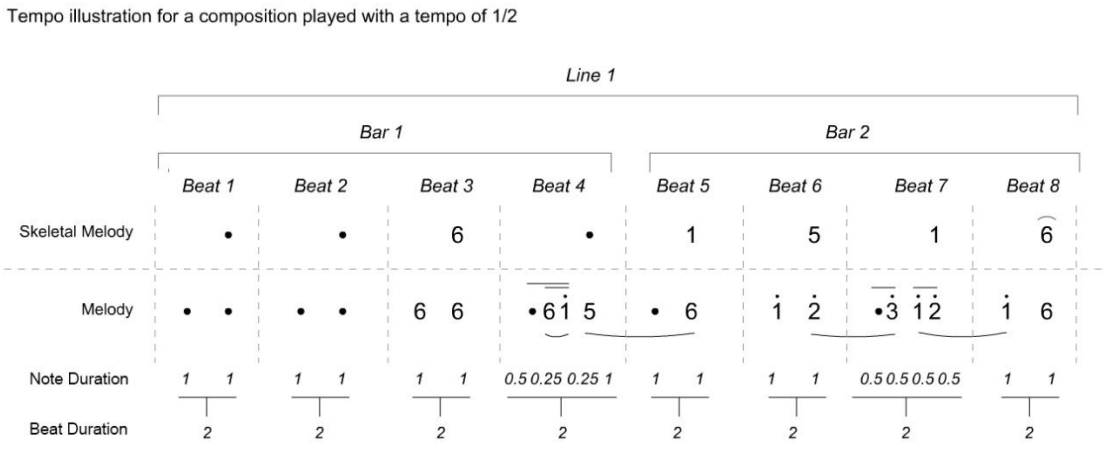

Fig. 3. Data Segmentation based on the Beats.

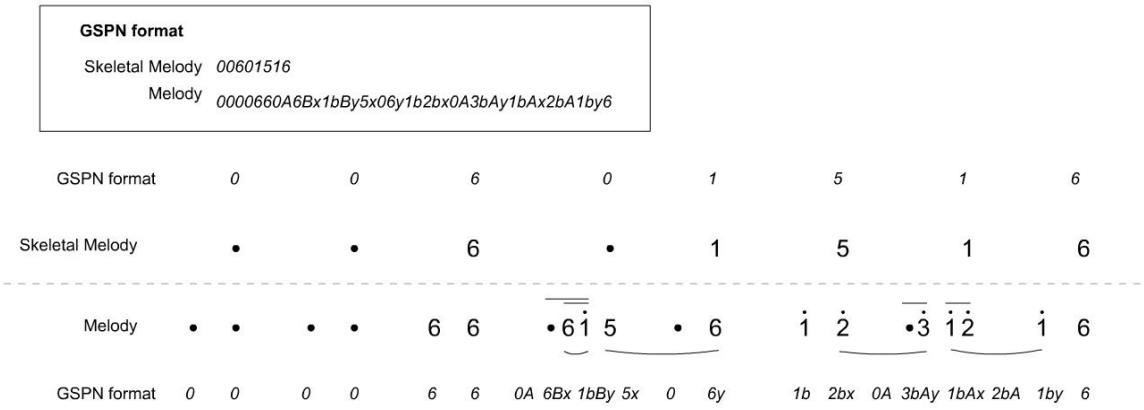

Fig. 4. Illustration of Text-based Format in the GSPN Model.

A. Data Preparation

The experiment was limited to the compositions of the *slendro* scale system played with a tempo of 1/2. The *slendro* scale system of slendro contains five notes, which are 1, 2, 3, 5 and 6, and also the dotted notes as an addition. The source of data which was 55 traditional Javanese music sheets was collected from www.gamelanbvg.com. Data were converted to a text-based format using a model of a text-based note writing for traditional Javanese compositions called *Ghending Scientific Pitch Notation* (GSPN). The model was developed to represent sheet music containing data of notes, note duration, note register and legato signs in a text-based format that can be read by human and computer [20]. The text-based conversion was manually conducted using a text editor program. Data of melodies and skeletal melodies were separately typed in two text files. To store information of the melody and its skeletal melody, each line in two text files was filled with one melody and one skeletal melody from each composition. Human errors are possible during conversion but the GSPN model supports a computational process that can detect typing errors. Note duration information can be calculated to detect the duration value of each beat so that typing errors can be detected if there are beats with different duration values in a composition. Once sheet music is converted to GSPN format, the data can be explored and calculated for information.

The code to represent the musical elements in GSPN format is case sensitive. Notes in *slendro* scale system, including the dotted note, are written in the numbers 0, 1, 2, 3, 5 and 6. Note duration is coded with no code, A and B where code A represents the note duration of 0.5, code B represents the note duration of 0.25, and no code represents the note duration of 1. Note register is coded with no code, a and b where code a represents the low notes, code b represents the high notes, and no code represents the middle notes. The legato sign is coded with no code, x and y where code x represents the beginning of the legato sign, code y represents the end of the legato sign, and no code represents notes that are between the beginning and end of a legato sign or notes that are not part of any legato sign. The code is written in order of note, note duration, note value and legato sign. Fig. 4 shows an illustration of a text-based format in the GSPN model converted from a music sheet.

TABLE I. GSPN IN TABULAR DATA FORMAT

	Note Sequence of Melody											
NT	0	0	0	0	6	6	0	6	1	5	…	6
NR									b		…	a
ND							A	B	B		…	
LS								x	y	x	…	

The following is an GSPN format example of a melody and its skeletal melody of a composition entitled Ladrang Wilujeng shown in Fig. 2.

Skeletal Melody:

00601516356165326600151611320126

Melody:

0000660A6Bx1bBy5x06y1b2bx0A3bAy1bAx2bA1by6000
03b3b3bBx5bB2bAy1x02by1bBx2bB6Ay3x0A5Ay2Ax5A3y
20000660A6Bx1bBy5x06y1b2bx0A3bAy1bAx2bA1by6001b
2bx1bA6Ay3Ax5A3y2003Ax5Ay301Ax2A1y6a

Table I shows illustration of GSPN in a tabular data format using melody data example above with NT stands for notes, ND stands for the notes duration, NR stands for the notes register and LS stands for the legato signs.

B. Beat Detection

In traditional Javanese sheet music, the melody part contains the musical elements of notes, note durations, note registers and legato signs, while the skeletal melody part usually contains only notes. The legato signs also do not used in the skeletal melody. The beats in the skeletal melody contains one note so every note in the skeletal melody has a duration note value of 1.

Beat detection was performed by converting letter codes in GSPN format into numbers. The note register encoded with a for low notes, and b for high notes, and no code for middle notes, was converted to 1, 0 and 2, respectively. The note duration encoded with A for the note of duration 0.5, and B for the note of duration 0.25, and no code for the note of duration 1 was converted to 0.5, 0.25 and 1, respectively. The legato sign

encoded with x for the beginning of legato, and y for the end of legato, and no code for notes that fall between the legato signs and notes that are not part of the legato signs, was converted to 1, 2 and 0, respectively.

Next, beat detection was done by calculating the notes duration value. The dataset uses compositions with a tempo of 1/2 which means each beat has a duration value of 2. The following is the pseudocode for detecting beats using data of a composition entitled Ladrang Wilujeng. The first pseudocode is to detect the number of notes duration (ND) in the sequence by adding up the note duration values, then dividing by the beat duration values based on the tempo (TP). Given NB as the number of beats then the pseudocode is:

```
TP = 2
NB = 0
ND = (1, 1, 1, 1, 1, 1, 0.5, 0.25, 0.25, 1, …, 1)
for (i = 0; i < ND.length; i++) {
    NB += ND [i]
}
NB = NB/TP
```

Next is to group the sequence of notes (NT), notes register (NR), note duration (ND), legato signs (LS) into beats. This grouping is performed using the following pseudocode:

```
NT = (0, 0, 0, 0, 6, 6, 0, 6, 1, 5, …, 6)
NR = (0, 0, 0, 0, 0, 0, 0, 0, 2, 0, …, 1)
LS = (0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, …, 0)
for (i = 0; i < NB; i++) {
    BT [i] = []
    BR [i] = []
    BD [i] = []
    BL [i] = []
}
k = 0
m = 0
for (i = 0; i <= ND.length; i++) {
    if (m < 2) {
      BT[k].push(NT[i])
      BR[k].push(NR[i])
      BD[k].push(ND[i])
      BL[k].push(LS[i])
    }
    m += ND [i]
    if (m > 2) {
      m = 0
      k += 1
    }
}
```

Based on the pseudocode above, the sequence breakdown of NT, NR, ND, and LS into beats is stored in notes per beat (BT), notes register per beat (BR), notes duration per beat (BD) and legato signs per beat (BL) respectively. Furthermore, each beat that has been defined is correlated with each note in the skeletal melody (SM) as follows:

BT = ((0, 0), (0, 0), (6, 6), (0, 6, 1, 5), …, (…, 6))

BR = ((0, 0), (0, 0), (0, 0), (0, 0, 2, 0), …, (…, 1))

BD = ((1, 1), (1, 1), (1, 1), (0.5, 0.25, 0.25, 1), …, (…, 1))

BL = ((0, 0), (0, 0), (0, 0), (0, 1, 2, 1), …, (…, 0))

SM = (0, 0, 6, 0, …, 6)

Table II shows illustration of the beat detection using the pseudocodes above.

TABLE II. BEAT DETECTION RESULTS

	Melody											
NT	0	0	0	0	6	6	0	6	1	5	…	6
NR	0	0	0	0	0	0	0	0	2	0	…	1
ND	1	1	1	1	1	1	0.5	0.25	0.25	1	…	1
LS	0	0	0	0	0	0	0	1	2	1	…	0
	Skeletal Melody											
SM	0		0		6		0				…	6

C. Vector Length Adjustment

The length of the beat element of melody data varies whereas the FFNN requires the same element length for the vector. Sequence padding and sequence truncation techniques are commonly used to solve element length problems. Truncation techniques that cut data elements can lose important information, while padding techniques that add elements in the data are computationally expensive.

Pre-experiments were conducted to compare the use of sequence padding and sequence truncation techniques based on prediction accuracy. By using the same dataset, the results show that the data managed by the sequence truncation technique achieves higher prediction accuracy results. The risk of losing important information due to truncation seems to be reduced by mapping the data using a sliding window technique after the data is truncated. So, sequence truncation technique was chosen to solve the beat element length problem.

The post-sequence truncation technique was implemented to notes per beat (BT), notes register per beat (BR), notes duration per beat (BD) and legato signs per beat (BL). Table III shows an example of the implementation of the post-sequence truncation technique to set the vector length of the notes per beat (BT) data, and the elements that are retained are two bits.

TABLE III. BEAT SEQUENCE TRUNCATION

ID	Beat Sequence	Truncation
1	(0, 0)	(0, 0)
2	(0, 0)	(0, 0)
3	(6, 6)	(6, 6)
4	(0, 6, 1, 5)	(0, 6)
…	…	…
32	(1, 6)	(1, 6)

D. Data Mapping & Feature Selection

Data mapping was performed based on the beats using the sliding window technique, a technique for data mapping by restructuring time series data to be used in a classification problem. This technique produces data segmentation based on the previous or the following sequences. The previous beat, selected beat and next beat are selected as the features. So, the sliding window implementation defines a pattern of (B_{n-1}, B_n, B_{n+1}) for the data mapping, where B stands for beat and n stands for sequence index. Melody has repetitive pattern; after reaching the last pitch, melody continues to restart from the first pitch. This solves the problem in indexing a sequence. The data mapping for the first beat is set to (B_{last}, B_1, B_2), and for the last beat is set to (B_{last-1}, B_{last}, B_1). The sliding window technique was implemented to BT, BR, BD and BL. Table IV shows an example of the implementation of the sliding window technique to set the data mapping of the notes per beat (BT).

Features are selected based on the data mapping implemented to BT, BR, and BD. Meanwhile, BL was not used as a feature to reduce the computation cost. Beat per bar index was also used as a feature because the position of the beat order per bar affects the musical mode system so it is important to use it as a feature. Each bar consists of four beats so that the sequence of beats per bar is a repeating pattern of 1, 2, 3, 4 and back to 1. Table V shows illustration of the feature selection based on BT, BR, BD, LS and beats ID per bar.

TABLE IV. DATA MAPPING_BT

BT (Input)			SM (Output)
ID	Beats	Data Mapping	
1	(0, 0)	(1, 6, **0, 0**, 0, 0)	0
2	(0, 0)	(0, 0, **0, 0**, 6, 6)	0
3	(6, 6)	(0, 0, **6, 6**, 0, 6)	6
4	(0, 6)	(6, 6, **0, 6**, 1, 2)	0
…	…	…	…
32	(1, 6)	(0, 1, **1, 6**, 0,0)	6

TABLE V. DATA MAPPING_BEATS

Beats (Input)				SM (Output)
ID	BT	BR	BD	
1	(1, 6, 0, 0, 0, 0)	(0, 1, 0, 0, 0, 0)	(1, 1, 1, 1, 1, 1)	0
2	(0, 0, 0, 0, 6, 6)	(0, 0, 0, 0, 0, 0)	(1, 1, 1, 1, 1, 1)	0
3	(0, 0, 6, 6, 0, 6)	(0, 0, 0, 0, 0, 0)	(1, 1, 1, 1, 0.5, 0.25)	6
4	(6, 6, 0, 6, 1, 2)	(0, 0, 0, 0, 0, 0)	(1, 1, 0.5, 0.25, 1, 1)	0
…	…	…	…	…
4	(0, 1, 1, 6, 0,0)	(0, 0, 0, 1, 0, 0)	(1, 0.5, 1, 1, 1, 1)	6

E. Binary Representation

Binary representation was implemented using the localist representation technique. The localist representation uses values of 0 and 1 to control an activation of variables. The *slendro* scale system consists of five notes: 1, 2, 3, 5 and 6, and the dot notation is converted into the number 0. Thus, the localist representation for each note consists of six bits, which is: 0 = 100000, 1 = 010000, 2 = 001000, 3 = 000100, 5 = 000010, and 6 = 000001. The notes register consists of three values: 1 represents the low notes, 2 represents the high notes and 0 represents the middle notes. Thus, the localist representation for each note register consists of three bits, which is: 1 = 100, 2 = 010, and 0 = 001. The notes duration consists of three values: 1 represents the value of 0.25, 2 represents value of 0.5 and 0 represents value of 1. Thus, the localist representation for each note duration consists of three bits, which is: 2 = 100, 1 = 010, and 0 = 001. The beat ID consists of four values: 1 represents the first beat in a bar, 2 represents the second beat in a bar, 3 represents the third beat in a bar, and 4 represents the fourth beat in a bar. Thus, the localist representation for each beat ID consists of four bits, which is: 1 = 1000, 2 = 0100, 3 = 0010, and 4 = 0001. Thus, the localist representation of the input data yields the length of each input: (6 × 6) + (6 × 3) + (6 × 3) + (1 × 4) = 36 + 18 + 18 + 4 = 76 bits, and the length output is 6 × 1 = 6 bits which is the notes elements of the *slendro* scale system.

V. TRAINING AND EVALUATION

An FFNN classifier was developed using the supervised learning approach and scaled conjugate gradient backpropagation algorithm to extract melodies into skeletal melodies. There are six classes for the melody extraction classification, where the output of each class is notes of the skeletal melodies. The length of the input vector is 76 bits and the length of the output vector is 6 bits.

The networks architecture consists of input, hidden and output layers. The best network performance was determined using an epoch with a parameter of six consecutive incorrect predictions. The number of hidden layer units was determined experimentally in multiples of 10 units and starts from 10 to 100 units. Each training was limited by 100 retrains. The training can be stopped before reaching the 100th training repetition if the results of the training meet the best prediction parameters. Later, experiments showed that the configuration of the number of hidden layer units of 40 units gives the best prediction results. Less than 40 units, the prediction error by the FFNN network is more than 40%, and more than that number the predictions are trapped in the local minima.

The best prediction parameters are determined based on the balance of the number of prediction errors between training, validation and test data in the FFNN network training with the maximum value of prediction error is 40%. The experiments used 55 music sheets, each of which contains a melody and its skeletal melody. The dataset mapped based on beats produced 2,808 beats for the corpus. Fig. 5 shows the FFNN architecture for the melody extraction.

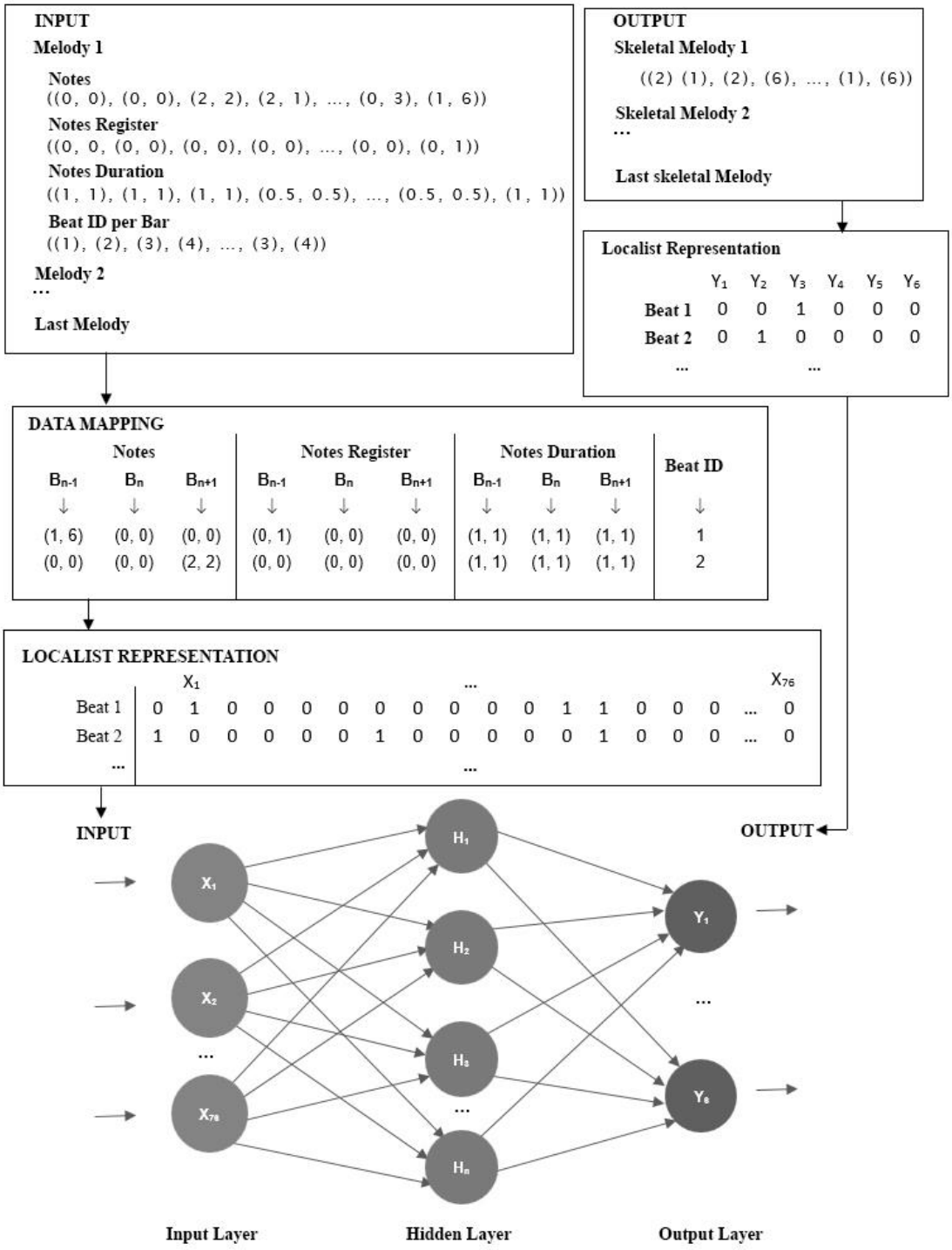

Fig. 5. The FFNN Architecture.

Of the 55 compositions used as datasets, 50 compositions were used for training data, while the remaining 5 compositions were used for test data. Of the 50 compositions used as training data, the input matrix is 76 × 2,568 bits and the output matrix is 6 × 2,568 bits. The distributed corpus in class 1, 2, 3, 4, 5, and 6 is 424, 387, 486, 525, 324, and 422 samples, respectively. Furthermore, the data is divided randomly into training, validation and test data with a proportion of 80:10:10 and resulted 2,054, 257 and 257 samples, respectively. Table VI shows the dimensions of the matrix, after applying the data transpose, the results of dividing the dataset into training data, and test data consisting of five compositions for testing the composition separately.

TABLE VI. VECTOR MATRIX DIMENSIONS

Data		Input	Output
Training	Training	76 × 2054	6 × 2054
	Validation	76 × 257	6 × 257
	Test	76 × 257	6 × 257
Evaluation	Composition 1	76 × 48	6 × 48
	Composition 2	76 × 48	6 × 48
	Composition 3	76 × 48	6 × 48
	Composition 4	76 × 48	6 × 48
	Composition 5	76 × 48	6 × 48

The FFNN network with the number of hidden layer units of 40 units meets the best prediction criteria with the number of prediction errors in each training, validation and test data of 34,81012e-0%, 35,79766e-0% and 35,79766e-0%. The best validation performance was obtained with a cross-entropy value of 0.17844 at epoch 34 as shown in Fig. 6, while Fig. 7 shows the graph of the receiver operating characteristic (ROC) performance and Fig. 8 shows the results of calculating the confusion matrix.

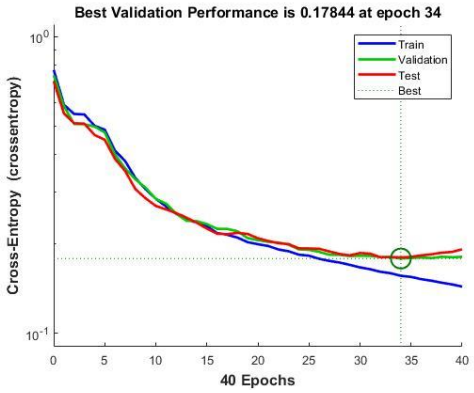

Fig. 6. The Best Validation Performance Graph.

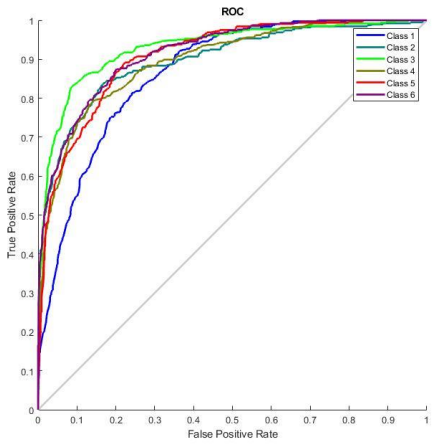

Fig. 7. The ROC Graph.

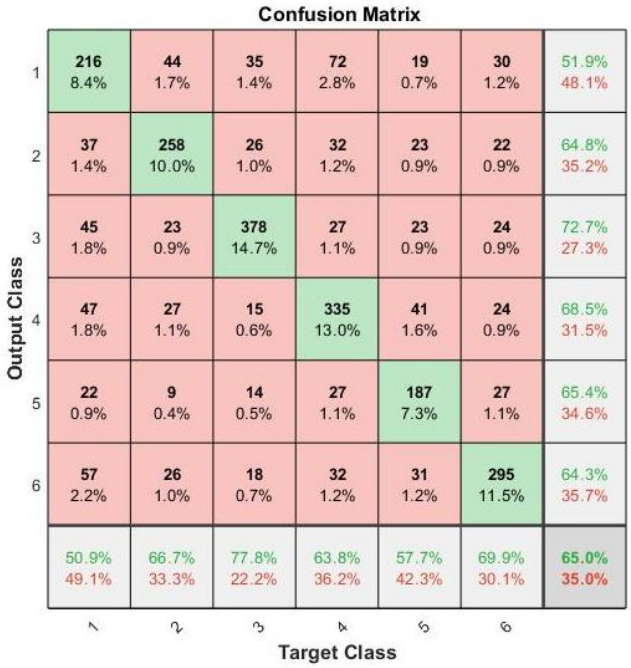

Fig. 8. The Confussion Matrix Results.

The ROC graph shows that the curve for class 1 initially moves along the curves of other classes before finally moving away. This condition is also shown by the results of the confusion matrix calculation, the prediction accuracy reaches 50.9%, or 216 of 424 class 1 data can be predicted correctly. In more detail, the calculation of the values of accuracy, precision, recall, specificity and F1-score of the FFNN network is shown in Table VII.

The experiment was continued by testing the networks using a hold-out test data which consisted of five compositions (abbreviated as C1, C2, C3, C4, C5). All of compositions, except C4, consist of 48 corpus, while C4 which consists of 64 corpus. The evaluation results show that, all measurements have improved performance in all compositions except C4, in which there was a decrease of 3.1% in the recall measure and 0.3% in the F1 measurement. Table VIII shows the comparison of evaluation results on training data and evaluation data which are divided into data of five compositions which are measured separately and combined, with C1 to C5 representing the first composition to the fifth composition.

TABLE VII. VECTOR MATRIX DIMENSIONS

Measurement	Training (%)	Evaluation (%)				
		C1	C2	C3	C4	C5
Accuracy	65.0	70.8	85.4	68.8	67.2	68.8
Precision	64.5	71.3	85.1	71.7	64.8	68.3
Recall	64.6	72.3	84.3	71.3	63.7	72.2
Specificity	65.0	70.8	85.5	68.8	67.1	68.7
F1 score	64.5	71.8	84.7	71.5	64.2	70.2

VI. Conclusion and Future Work

Overall, the proposed method successfully combines musical theory in notes sequence pattern recognition by extracting melodies using a multi-class classification based on notation duration and beat rules. Based on the evaluation of the measurement of accuracy, precision, recall, specificity and F1 score on the melody extraction per composition, the performance of the networks in correctly classifying the notes of skeletal melodies from the beats of melodies, including distinguishing extracted notes from targets that are not in their class, has increased compared to the results obtained in training.

Table VIII shows the results of the accuracy test per class per composition. On the melody extraction with target class 3 (note 2), the performance of the FFNN network looks good with the lowest accuracy of the five compositions being 83.3%, which are at C1 and C3, even at C2 and C5, the accuracy reaches 100%. Good performance is also shown in the class 6 (note 6) where the lowest accuracy is 70% at C4, and overall C4 does contribute to a decrease in accuracy. Similar conditions also occur in class 2 (note 1) with 100% accuracy results in C2 and C5, but there is a decrease in accuracy in C3 even though it can still be categorized as a good achievement, which is 87.5%. Meanwhile in C1 and C4, there was a decrease to 66.7% and 60%, respectively. Class 1 (note 0), 4 (note 3) and 5 (note 5) gave a poor contribution to the accuracy of the melody extraction.

TABLE VIII. Accuracy Test Results Per Note Class

Target	Output (%)				
	C1	C2	C3	C4	C5
1 (note 0)	71.4	50	83.3	62.5	40.0
2 (note 1)	66.7	100	87.5	60.0	100
3 (note 2)	83.3	100	83.3	84.6	100
4 (note 3)	45.5	100	33.3	80.0	50.0
5 (note 5)	77.8	66.7	50.0	25.0	55.6
6 (note 6)	88.9	88.9	90	70.0	87.5

The accuracy of the melody extraction in notes 2 and 6 is directly proportional to the fact that the dominant notes or note strength in the *manyura* musical mode system in the *slendro* scale system used as a dataset is in both notes. Thus, it can be concluded that the networks can recognize the musical mode system and the musical scale system using the melody extraction approach. On the other hand, the conditions of the other four notes, which area 0, 1, 3 and 5, and some of which are still not well predicted. It still cannot be used to conclude that the networks failed in extracting melodies. The unique characteristics of *Gamelan* music allow differences in notes in the same condition not to be a mistake as long as all the different tones do not change the meaning of the composition and can still be accepted by the *Gamelan* music community (Hastuti et al., 2017).

The performance of the networks in extracting melodies into skeletal melodies can be said to be quite successful and promising. However, there are still several factors that need to be explored further to improve networks performance, such as the need to apply artificial intelligence methods for feature selection and data mapping to increase the cardinality of the corpus to overcome the problems of limited number of datasets and confusion faced by the networks.

References

[1] T. Cho, R.J. Weiss, and J.P. Bello,"Exploring Common Variations in State of the art Chord Recognition Systems", in Proceedings of the 7th Sound and Music Computing Conference (SMC), Barcelona, Spain, 2010, pp. 1–8.

[2] T. Carsault, J. Nika, and P. Esling, "Using Musical Relationships between Chords Label in Automatic Chord Extraction Task", in 19th International Society for Music Information Retrieval Conference, Paris, France, 2018, pp. 18-25.

[3] F. Korzeniowski, and G. Widmer, "Improved Chord Recognition by Combining Duration and Harmonic Language Model", in 19th International Society for Music Information Retrieval Conference, Malaga, Spain, 2016, pp. 10-17.

[4] J. Osmalskyj, J.J. Embrechts, S. Piérard, and M. Van Droogenbroeck, "Neural Networks for Musical Chord Recognition". In Actes des Journées d'Informatique Musicale (JIM 2012), Belgique, 2012, pp. 39-46.

[5] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A Bi-directional Transformer for Musical Chord Recognition", in 20th International Society for Music Information Retrieval Conference, Delft, Netherlands, 2019, pp. 620-627.

[6] Z. Rao, X. Guan, and J. Teng, "Chord Recognition Based on Temporal Correlation Support Vector Machine". Applied Science, Vol. 6, No. 5, 2016, pp. 1-14.

[7] X. Zhou, and A. Lerch, "Chord Detection using Deep Learning", in 16th International Society for Music Information Retrieval Conference, Malaga, Spain, 2015, pp. 52-58.

[8] L. Marques, "A chord distance metric based on the Tonal Pitch Space and a key-finding method for chord annotation sequences", in 17th Brazilian Symposium on Computer Music - SBCM 2019, São João del-Rei, 2019, pp. 136-140.

[9] H.V. Koops, W. Bas de Haas, J. Bransen, and A. Volk, "Automatic chord label personalization through deep learning of shared harmonic interval profiles", Neural Computing and Applications (2020) 32:929–939, DOI: https://doi.org/10.1007/s00521-018-3703-y.

[10] J. Pauwels, K. O'Hanlon, E. Gómez, and M.B. Sandler, "20 Years of Automatic Chord Recognition from Audio", in 20th International Society for Music Information Retrieval Conference, Delft, Netherlands, 2019, pp. 54-63.

[11] B. Ghojogh, M.N. Samad, S.A. Mashhadi, T. Kapoor, W. Ali, F. Karray and M. Crowley,"Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review", arXiv:1905.02845 [cs.LG], 2019.

[12] J. Jiang, K. Chen, W. Li, G. Xia, "Large-Vocabulary Chord Transcription via Chord Structure Decomposition", in 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019, pp. 644-651.

[13] D. Efrosinin, and V. Sturm, "Time Series Segmentation of Linear Stochastic Processes for Anomaly Detection Problem using Supervised Methods", in Proceedings of the 56th ESReDA Seminar, Linz, Austria, 2019, pp. 1-11.

[14] H.S. Hota, R. Handa, and A.K. Shrivas, "Time Series Data Prediction using Sliding Window based RBF Neural Network", International Journal of Computational Intelligence Research, Vol. 13, No. 5, 2017, pp. 1145-1156.

[15] M. Dwarampudi, and N.V.S. Reddy, "Effects of Padding on LSTMS and CNNS", arXiv:1903.07288 [cs.LG], 2019.

[16] J. Becker, and A. Becker, "A Grammar of the Musical Genre Srepegan", Asian Music, Vol. 14, No. 1, 1982, pp. 30-73.

[17] D.W. Hughes,"Deep Structure and Surface Structure in Javanese Music: A Grammar of Gendhing Lampah", Ethnomusicology, Vol. 32, No. 1, 1988, pp. 23-74.

[18] K. Hastuti, A. Azhari, A. Musdholifah, and R. Supanggah, "Building Melodic Feature Knowledge of Gamelan Music using Apriori based on

Functions in Sequence (AFiS) Algorithm", International Review on Computers and Software, Vol. 11, No. 12, 2016, pp. 1127-1137.

[19] K. Hastuti, A. Azhari, A. Musdholifah, and R. Supanggah, "Rule-based and Genetic Algorithm for Automatic Gamelan Music Composition", International Review on Modelling and Simulations, Vol 10, No. 3, 2017, pp. 202-212.

[20] A.M. Syarif, A. Azhari, S. Suprapto, and K. Hastuti, "A Model of Computation-based Naming System for Musical Elements of Java Traditional Song", IOP Conference Series: Materials Science and Engineering, Vol. 803 (012031), 2020.