

## Implementasi *Support Vector Machine* untuk Klasifikasi Ucapan Huruf Hijaiyah Bertanda Baca Berbasis Ekstraksi Ciri MFCC

Wisnu Adhi Pradana<sup>1</sup>, Adiwijaya<sup>2</sup>, Untari Novia Wisesty<sup>3</sup>

<sup>1,2,3</sup>School of Computing, Telkom University, Bandung 40257, Indonesia

[wa.pradana01@gmail.com](mailto:wa.pradana01@gmail.com)<sup>1</sup>, [adiwijaya@telkomuniversity.ac.id](mailto:adiwijaya@telkomuniversity.ac.id)<sup>2</sup>,  
[untarinw@telkomuniversity.ac.id](mailto:untarinw@telkomuniversity.ac.id)<sup>3</sup>

---

**Abstrak.** *Support vector machine* atau yang biasa disebut SVM adalah salah satu metode yang dapat digunakan untuk proses klasifikasi suatu data. Pada penelitian kali ini, sistem dibangun dengan menggunakan pendekatan tersebut dalam pengembangan *arabic speech recognition*. Dalam pembangunan sistem, ada 2 macam tipe *speaker* yang telah diuji yaitu *dependat speaker* dan *independent speaker*. Hasil yang diperoleh dari sistem ini adalah nilai akurasi sebesar 85,32% untuk *dependat speaker* dan 61,16% untuk *independent speaker*.

**Kata Kunci:** *Arabic Speech Recognition, Support Vector Machine, MFCC.*

---

**Abstract.** Support Vector Machine or commonly called SVM is one method that can be used to process the classification of a data. In this study, the system was built using the approach in the development of Arabic Speech Recognition. In the development of the system, there are 2 kinds of speakers that have been tested that is dependat speakers and independent speakers. The results from this system is an accuracy of 85.32% for speaker dependent and 61.16% for independent speakers.

**Keywords:** Arabic Speech Recognition, Support Vector Machine, MFCC.

---

### 1. Introduction

Bahasa arab merupakan salah satu bahasa yang luas dan tua di dunia, hanya ada sedikit Arabic Speech Recognition dari pada bahasa yang lain. Dalam pengembangannya, Arabic Speech Recognition memiliki banyak problem yang membuatnya sulit untuk dikembangkan. Faktor yang paling mempengaruhi dalam mengembangkan Arabic Speech Recognition adalah banyaknya variasi dialek dan kompleksitas morfologinya [10]. Variasi dialek dapat menyebabkan perbedaan makna pada kata yang diucapkan. Selain itu, dalam bahasa arab ada beberapa aturan tentang cara baca (tajwid) sehingga mempengaruhi makna dari sebuah kata.

Speech Recognition atau dapat disingkat SR adalah teknologi yang memungkinkan computer untuk mengenali kata yang diucapkan oleh manusia di mic atau telepon. Dalam penggunaannya, SR telah digunakan dalam berbagai macam permasalahan, semisal penggunaan command computer dengan suara atau belajar bahasa asing. Selain itu SR juga membantu berinteraksi dengan masyarakat asing.

Ada banyak metode yang telah digunakan dalam membangun sebuah SR. namun, metode yang sering digunakan untuk membangun sebuah SR adalah dengan menggunakan Hidden Markov Model atau biasa disebut HMM. HMM adalah model yang mengasumsikan proses markov dengan parameter yang tidak diketahui serta menentukan hidden parameters dari observable parameter. Oleh karena itu, kebanyakan Arabic Speech Recognition yang telah berkembang saat ini adalah dengan menggunakan pendekatan HMM [6].

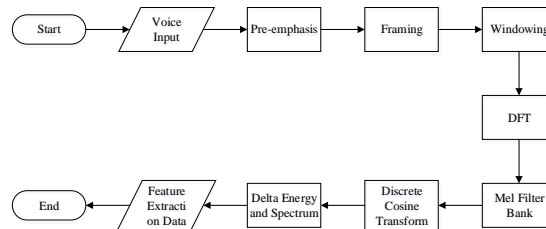
Support Vector Machine atau yang biasa disebut SVM adalah salah satu metode yang dapat digunakan untuk proses klasifikasi suatu data [4]. Penulis bermaksud untuk menggunakan pendekatan tersebut dalam pengembangan Arabic Speech Recognition. Dengan adanya sistem ini, maka diharapkan

menambah pilihan dalam menggunakan Arabic Speech Recognition sebagai classifier lafas huruf hijaiyah bertanda baca.

**2. Mel frequency cepstrum coefficient**

Mel frequency cepstrum coefficient atau MFCC merupakan pendekatan yang paling umum digunakan untuk pengenalan suara. MFCC adalah satu-satunya pendekatan akustik yang mengambil persepsi manusia (fisiologi dan aspek perilaku dari organ produksi suara) sebagai pertimbangan untuk membangun speech recognition yang berarti pendekatan ini tidak dapat memproses frekuensi diatas 1KHz. Pendekatan ini mempunyai dua tipe filter yang berjarak secara linear pada frekuensi rendah dibawah 1000Hz dan jarak logarithmic diatas 1000Hz.

Langkah-langkah yang dilakukan dalam ekstraksi adalah pre-emphasis, frame blocking, windowing, Fast Fourier Transform, Mel Filter Bank Processing, discrete cosine transformation dan Delta Energy and Delta Spectrum. Hasil dari MFCC adalah berupa vektor ciri yang nantinya akan digunakan sebagai database untuk menentukan klasifikasi data suara.



Gambar Error! No text of specified style in document..1 Langkah-langkah MFCC untuk menentukan vektor ciri[9]

**2.1. Pre-emphasis**

Proses ini akan meningkatkan energi sinyal pada frekuensi yang lebih tinggi. Misalkan a = 0,95, dalam artian 95% dari salah satu sampel yang diduga berasal dari sampel sebelumnya.

$$Y[n] = X[n] - 0.95X[n - 1] \tag{1}$$

Y[n] = sinyal output

X[n] = sinyal input

**2.2. Framing**

Proses ini adalah memotong sampel suara analog menjadi digital menjadi frame kecil dengan range 20 hingga 40msec. Sinyal suara dibagi menjadi frame sejumlah N sampel. Frame yang berdekatan dipisahkan oleh M (M<N)

**2.3. Windowing**

Hamming window digunakan sebagai bentuk jendela dengan mempertimbangkan blok berikutnya dalam rantai pengolahan fitur ekstraksi dan mengintegrasikan semua lini frekuensi terdekat. Jika window didefinisikan sebagai W(n) dan 0 ≤ n ≤ N-1, maka

$$Y(n) = X(n) \times W(n) \tag{2}$$

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N - 1}\right), \quad 0 \leq n \leq N - 1 \tag{3}$$

N = jumlah sampel tiap frame

Y(n) = sinyal output

X(n) = sinyal input

W(n) = hamming window

**2.4. DFT**

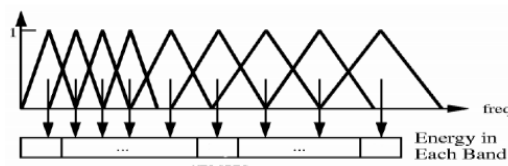
Proses ini dilakukan untuk mengkonversikan setiap N sampel frame dari domain waktu ke domain frekuensi. *Fast fourier transform* bertujuan untuk mengubah perubahan bentuk gelombang dari bunyi yang dihasilkan oleh pita suara  $U[n]$  dan respon impuls saluran vokal  $H[n]$  pada domain waktu. Lihat persamaan 4

$$Y(w) = FFT[H(t) * X(t)] = H(w) * X(w) \quad (4)$$

$X(w)$ ,  $H(w)$ , dan  $Y(w)$  adalah hasil *fourier transform* dari  $X(t)$ ,  $H(t)$ , dan  $Y(t)$ .

### 2.5. Mel Filter Bank

Range frekuensi yang berada dalam spektrum FFT sangat luas dan sinyal suara belum berupa skala linier. Bank dari filter sesuai dengan skala Mel yang ditunjukkan gambar 2.3.



Gambar **Error! No text of specified style in document.** 2 mel scale filter bank[9]

Gambar 2.3 menyajikan himpunan dari *filter* yang berbentuk segitiga yang digunakan untuk menghitung *weighted sum* dari komponen *filter spectral* sehingga keluaran dari proses ini mendekati skala mel.

$$f_{mel} = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (5)$$

Hasil dari proses ini adalah mel dengan satuan frekuensi dalam Hz.

### 2.6. Discrete Cosine Transform

Bagian ini adalah proses untuk mengkonversi *log mel* spectrum menjadi domain menggunakan *Discrete Cosine Transform* (DCT). Hasil dari konversi disebut *Mel frequency cepstrum coefficient*. Set koefisien tersebut disebut juga vektor akustik (vektor ciri).

### 2.7. Delta Energy and Spectrum

Sinyal suara dan perubahan *frame*, seperti kemiringan *formant* di transisinya. Oleh karena itu, ada kebutuhan untuk menambahkan fitur yang berkaitan dengan perubahan fitur cepstral dari waktu ke waktu. 13 *delta* atau *velocity features* (12 *cepstral features* ditambah energi), dan 39 *features double delta* atau percepatan fitur ditambahkan. Energi dalam *frame* untuk sinyal  $x$  pada *window* dari waktu  $t_1$  hingga  $t_2$  dapat direpresentasikan pada persamaan 6,

$$Energy = \sum X^2[t] \quad (6)$$

## 3. Support Vector Machine

*Support vector machine* atau bisa disebut SVM adalah salah metode *supervised learning* yang dapat digunakan untuk klasifikasi data[4]. SVM membagi data menjadi dua kelas dengan memaksimalkan margin antara *hyperplane* dengan kedua data terdekat pada kedua kelas[4]. *Hyperplane* yang paling optimal disebut sebagai *support vector*. Untuk memperoleh *hyperplane* yang terbaik diperlukan perhitungan untuk menentukan jarak antara data dengan *hyperplane* yang ada. Pada proses *learning* yang pertama, akan dicari jarak minimum dari setiap data *training* terhadap *hyperplane* dengan menggunakan rumus 7 dan 8.

$$y_i(w^T \cdot x_i + b) \geq 1 \quad ; i = 1, 2 \dots n \quad (7)$$

$x_i$  = data input

$y_i$  = target

$b$  = posisi bidang terhadap pusat koordinat

$w$  = bidang normal

Persamaan 8 dapat dipecahkan menggunakan titik optimal dari fungsi lagrange.

$$L_p = L_{(w,b,\alpha)} = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1) \tag{8}$$

Alpha ke-I adalah pengali lagrange. Untuk mencari titik optimal, maka w dan b harus diminimumkan sehingga L = 0. Dengan kondisi tersebut, maka  $\alpha$  dapat diketahui.

Untuk membuat keputusan dari data *testing*, dibutuhkan sebuah fungsi keputusan yang dibuat dari fungsi *kernel*. Ada banyak fungsi *kernel* yang bisa dipakai untuk pengambilan keputusan, berikut adalah contoh fungsi yang sering dipakai:

Linear kernel :  $K(x_i, x_j) = x_i^T x_j$  (9)

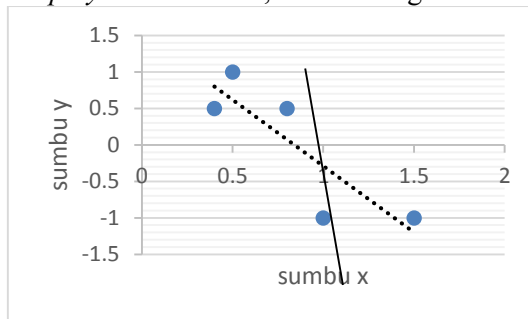
Polynomial kernel :  $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$  (10)

d = orde

$x_i$  dan  $x_j$  = data berupa *vector*

$\gamma$  dan r = parameter

Quadratic kernel juga termasuk *polynomial kernel*, namun dengan orde dua [16].



Gambar Error! No text of specified style in document..3 pencarian hyperplane terbaik

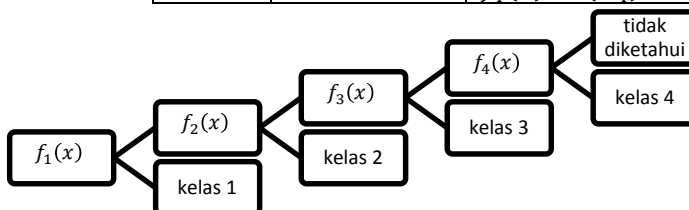
### 3.1. Multiclas SVM

Pada awalnya SVM hanya bisa digunakan untuk mengelompokkan 2 kelas saja. Namun, dalam perkembangannya para peneliti telah mengembangkan metode yang membuat SVM dapat digunakan untuk klasifikasi lebih dari 2 kelas data. Ada cara untuk mengimplementasikan *multi class SVM* yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data ke dalam sebuah sistem optimasi.

Metode “*one against all*” adalah salah satu metode yang dapat digunakan untuk permasalahan *multi class SVM*. Metode ini membangun n buah model SVM biner (n adalah jumlah kelas). Setiap model di-*training* dengan semua data untuk dicari solusinya. Lihat tabel 2.1 dan gambar 2.4

Tabel Error! No text of specified style in document..1 contoh SVM biner one-against-all

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f_1(x) = (w_1)x + b_1$
Kelas 2	Bukan kelas 2	$f_2(x) = (w_2)x + b_2$
Kelas 3	Bukan kelas 3	$f_3(x) = (w_3)x + b_3$
Kelas 4	Bukan kelas 4	$f_4(x) = (w_4)x + b_4$



Gambar Error! No text of specified style in document..4 contoh klasifikasi one-against-all

## 4. Hasil

### 4.1. Dependent speaker

Akurasi adalah ukuran seberapa dekat suatu hasil pengukuran dengan nilai yang benar atau diterima dari kuantitas besaran yang diukur. Akurasi didapat dari jumlah hasil uji yang benar dibagi dengan jumlah data. Tabel 4.1 diberikan untuk merepresentasikan hasil dari tipe *dependent speaker* yang telah diperoleh selama penelitian.

Tabel 4.1 menjelaskan tentang akurasi yang didapat pada setiap huruf yang telah diuji terhadap model dengan 2 fungsi *kernel* yang berbeda. Akurasi terbesar yang diperoleh dengan menggunakan fungsi *kernel linear* dan *quadratic* adalah huruf *ro* (ر) dengan nilai 97,22% untuk fungsi *kernel linear* dan 61,11% untuk fungsi *kernel quadratic*. Akurasi terkecil yang diperoleh dengan menggunakan fungsi *kernel linear* dan fungsi *kernel quadratic* adalah huruf *tsa* (ث) dengan nilai 72,22% untuk fungsi *kernel linear* dan 22,22% untuk fungsi *kernel quadratic*. Hal ini mengindikasikan model yang diperoleh lebih optimal digunakan untuk mengklasifikasikan huruf *ro* daripada huruf *tsa*. Selain itu, rata-rata akurasi yang diperoleh dari sistem dengan fungsi *kernel linear* adalah 85,32% dan fungsi *kernel quadratic* adalah 43,45%. Pembagian huruf yang dijalankan pada sistem juga mempengaruhi hasil akurasi. Dengan pembagian ini, sistem tidak perlu membuat 168 model kelas secara langsung. Selain itu proses pembuatan model dapat diperoleh lebih cepat.

Tabel Error! No text of specified style in document..2 hasil pengukuran dengan dependent speaker

No	Huruf	Hijaiyah	Akurasi		No	Huruf	Hijaiyah	Akurasi	
			Linear	Quadratic				Linear	quadratic
1	Alif	ا	88,89%	44,44%	15	Dhod	ض	88,89%	47,22%
2	Ba	ب	75%	33,33%	16	Tho	ط	91,67%	52,78%
3	Ta	ت	77,78%	38,89%	17	Zho	ظ	80,56%	44,44%
4	Tsa	ث	72,22%	22,22%	18	Ain	ع	86,11%	58,33%
5	Jim	ج	86,11%	27,78%	19	Ghoin	غ	77,78%	52,78%
6	Ha	ح	88,89%	52,78%	20	Fa	ف	83,33%	33,33%
7	Kho	خ	94,44%	44,44%	21	Qof	ق	86,11%	44,44%
8	Dal	د	88,89%	38,89%	22	Kaf	ك	86,11%	38,89%
9	Dzal	ذ	86,11%	41,67%	23	Lam	ل	88,89%	47,22%
10	Ro	ر	97,22%	61,11%	24	Mim	م	86,11%	30,56%
11	Zai	ز	88,89%	50%	25	Nun	ن	80,56%	47,22%
12	Sin	س	94,44%	47,22%	26	Wa	و	80,56%	41,67%
13	Syin	ش	80,56%	36,11%	27	Ha	ه	88,89%	44,44%
14	Shat	ص	83,33%	38,89%	28	Ya	ي	80,56%	55,56%

### 4.2. Independent speaker

Pengujian pada tipe *independent speaker* dilakukan dengan menggunakan fungsi *kernel linear*. Tabel 4.2 merepresentasikan hasil dari pengujian menggunakan fungsi *kernel linear*. Pada pengujian ini, data *testing* yang digunakan sejumlah 672 data. Data ini berasal dari 1 orang yang mengucapkan semua huruf dengan *harakat* yang berbeda.

Pada tabel 4.2 dapat dilihat bahwa akurasi terbesar diperoleh huruf *sin* (س) dan *tho* (ط) dengan akurasi sebesar 87,5%. Sedangkan akurasi terkecil yang diperoleh dari hasil uji model ini adalah huruf *fa* (ف) dengan akurasi sebesar 25%. Hal ini mengindikasikan model yang diperoleh lebih optimal digunakan untuk mengklasifikasikan huruf *sin* dan *tho* daripada huruf *fa*. Selain itu, rata-rata akurasi yang diperoleh dari sistem dengan ini adalah sebesar 61,16%.

Tabel Error! No text of specified style in document..3hasil pengukuran dengan independent speaker

No	Huruf	Hijaiyah	Akurasi	No	Huruf	Hijaiyah	Akurasi
1	Alif	ا	75%	15	Dhod	ض	75%
2	Ba	ب	54,17%	16	Tho	ط	87,50%
3	Ta	ت	75%	17	Zho	ظ	83,33%
4	Tsa	ث	54,17%	18	Ain	ع	75%
5	Jim	ج	66,67%	19	Ghoin	غ	41,67%
6	Ha	ح	41,67%	20	Fa	ف	25%
7	Kho	خ	70,83%	21	Qof	ق	41,67%
8	Dal	د	83,33%	22	Kaf	ك	66,67%
9	Dzal	ذ	54,17%	23	Lam	ل	62,50%
10	Ro	ر	66,67%	24	Mim	م	29,17%
11	Zai	ز	58,33%	25	Nun	ن	37,50%
12	Sin	س	87,50%	26	Wa	و	50%
13	Syin	ش	58,33%	27	Ha	ه	62,50%
14	Shat	ص	50%	28	Ya	ي	79,17%
Rata-rata				61.16%			

### 4.3. Perbandingan

Hasil akurasi terbaik yang diperoleh dari model *support vector machine* dengan fungsi *kernel linear* adalah huruf *ro* (ر) dengan nilai akurasi 97,22% untuk data *testing dependent speaker*. Sedangkan untuk data *testing independent speaker*, Hasil akurasi terbaik yang diperoleh dari model *support vector machine* dengan fungsi *kernel linear* adalah huruf *sin* (س) dan *tho* (ط) dengan akurasi sebesar 87,5%. Selain itu rata-rata akurasi yang didapat dari kedua tipe tersebut adalah 85,32% untuk *dependent speaker* dan 61,16% untuk *independent speaker*.

Sementara itu, model *support vector machine* dengan fungsi *kernel quadratic* hanya diperoleh dari tipe *dependent speaker*. Hal ini terjadi dikarenakan model *support vector machine* dengan fungsi *kernel quadratic* gagal mengklasifikasikan data *testing* dari *independent speaker*. Hasil akurasi terbaik diperoleh dari model *support vector machine* dengan fungsi *kernel quadratic* adalah huruf *ro* (ر) dengan nilai akurasi 61,11%. Selain itu rata-rata akurasi yang didapat dari model ini adalah 43,45%

Dari hasil pengujian yang telah dilakukan didapat bahwa model *support vector machine* dengan tipe *dependent speaker* lebih baik daripada tipe *independent speaker*. Hal ini membuktikan *dependency* sebuah data mempengaruhi hasil dari klasifikasi. Selain itu, fungsi *kernel* juga mempengaruhi hasil dari klasifikasi, terbukti bahwa fungsi *linear* lebih baik dari fungsi *quadratic*.

### 5. Kesimpulan

Berdasarkan dari hasil pengujian yang telah dilaksanakan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Perlakuan training dan testing menjadi setiap huruf membuat sistem bekerja lebih cepat daripada semua data diproses bersama-sama.
2. Fungsi kernel mempengaruhi akurasi dari sistem dengan penggunaan fungsi linear menghasilkan rata-rata akurasi sebesar 85,32% untuk *dependent speaker* dan 61,16% untuk *independent speaker*.
3. *Dependency* sebuah data mempengaruhi hasil klasifikasi. Model yang menggunakan *dependent speaker* memiliki akurasi yang lebih bagus dari pada *independent speaker*

### References

- [1] Abushariah, M.A.A.M., Ainon, R., Zainuddin, R., Elshafei, M. and Khalifa, O.O., 2012. Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and

- balanced speech corpus. *International Arab Journal of Information Technology (IAJIT)*, 9(1), pp.84-93.
- [2] AbuZeina, D. and Elshafei, M., 2012. Arabic Speech Recognition Systems. In *Cross-Word Modeling for Arabic Speech Recognition* (pp. 17-23). Springer US.
- [3] Alotaibi, Y.A. and Hussain, A., 2010. Comparative analysis of arabic vowels using formants and an automatic speech recognition system.
- [4] Durgesh, K.S. and Lekha, B., 2010. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 12(1), pp.1-7.
- [5] El Choubassi, M.M., El Khoury, H.E., Alagha, C.J., Skaf, J.A. and Al, M.A., 2003. Arabic speech recognition using recurrent neural networks. In *IEEE Intl. Symp. Signal Processing and Information Technology ISSPIT*.
- [6] Juang, B.H. and Rabiner, L.R., 1991. Hidden Markov models for speech recognition. *Technometrics*, 33(3), pp.251-272.
- [7] Kalinli, A. and Sagiroglu, S., 2006. Short Paper\_. *Journal of Information Science and Engineering*, 22, pp.1555-1568.
- [8] Meng, Y.U.A.N., 2004. Speech recognition on DSP: Algorithm optimization and performance analysis (Doctoral dissertation, The Chinese University of Hong Kong).
- [9] Muda, L., Begam, M. and Elamvazuthi, I., 2010. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:1003.4083.
- [10] Satori, H., Hiyassat, H., Harti, M. and Chenfour, N., 2009. Investigation arabic speech recognition using CMU sphinx system. *Int. Arab J. Inf. Technol.*, 6(2), pp.186-190.
- [11] Singh, S. and Rajan, E.G., 2011. Vector quantization approach for speaker recognition using MFCC and inverted MFCC. *International Journal of Computer Applications*, 17(1), pp.1-7.
- [12] Shady, Y. and Zayed, S.H.H., 2009. Speaker independent Arabic speech recognition using support vector machine.
- [13] Susanto, C.P. and Setiawan, E.I., 2015. Algoritma Support Vector Machine Untuk Mendeteksi Sms Spam Berbahasa Indonesia.
- [14] U.N. Wisesty, M.S. Mubarak, Adiwijaya, A classification of marked hijaiyah letters' pronunciation using hidden Markov model, *AIP Conference Proceedings* 1867(1), 020036
- [15] Zhang, S.X., 2014. Structured support vector machines for speech recognition.
- [16] Chang, Y.W., Hsieh, C.J., Chang, K.W., Ringgaard, M. and Lin, C.J., 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, 11(Apr), pp.1471-1490.