

ANALISIS PERFORMANSI ALGORITMA SVM, CNN, DAN LSTM UNTUK PENGENALAN KEGIATAN MANUSIA DENGAN URAD FMCW RADAR

Azhar Y. R.¹, Figo A.D.F.², M. Adi N.³, Fiky Y. Suratman⁴, Istiqomah⁵

^{1, 2, 3, 4, 5} Fakultas Teknik Elektro, Universitas Telkom, Bandung

¹ramadhanazar@student.telkomuniversity.ac.id, ²figazdefitrah@student.telkomuniversity.ac.id,

³nurhidayatadi@student.telkomuniversity.ac.id,

⁴fysuratman@telkomuniversity.ac.id, ⁵istiqomah@telkomuniversity.ac.id

Diterima pada 13 Juni 2023; disetujui pada 7 Agustus 2023; dan diterbitkan pada 17 Oktober 2023.

Abstrak

Penelitian ini membahas tentang studi perbandingan antara Support Vector Machine (SVM), Convolutional Neural Network (CNN), dan Long Short-Term Memory (LSTM) sebagai algoritma pembelajaran mesin berbasis data Radar tipe FMCW yang umum digunakan pada permasalahan Pengenalan Kegiatan Manusia (PKM). Perbandingan dilakukan dengan membandingkan hasil evaluasi model terhadap data uji dan melihat waktu pelatihan model serta jumlah parameter yang dibutuhkan model untuk mencapai hasil tersebut guna mencari model yang memberikan hasil terbaik seefisien mungkin. Didapatkan bahwa model tipe LSTM_01 dengan satu lapis LSTM 16 unit memberikan hasil terbaik berdasarkan penilaian yang dilakukan terhadap beberapa model yang diujikan. Model tersebut mencapai akurasi hingga 86% pada data uji dengan jumlah parameter relatif sedikit, yaitu 294.725.

Kata Kunci: Radar, komputasi, FMCW, SVM, CNN, LSTM.

Abstract

In this study, we compare Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) algorithms as commonly used machine learning algorithms based on FMCW Radar data for Human Activity Recognition (HAR). The comparison is conducted by evaluating the models on test data and considering the fitting time and the number of parameters required by each model to achieve the desired results, to find the most efficient model that provides the best results. We discovered that the LSTM_01 model with one layer of 16 unit-LSTM produces the best result based on the scoring of several tested models. The model demonstrated an ability to achieve accuracy up to 86% on the test data with a relatively small number of parameters, i.e., 294,725.

Key Words: Radar, computation, FMCW, SVM, CNN, LSTM.

1. Pendahuluan

Sistem Pengenalan Kegiatan Manusia (PKM) saat ini menjadi riset yang cukup ditekuni[1]. Saat ini, sudah banyak metode yang dapat digunakan dalam perancangan sistem PKM dengan penggunaan kamera Closed-Circuit Television (CCTV) maupun wearable devices, seperti smart watch[2]. Hal ini didukung juga dengan pesatnya perkembangan Artificial Intelligence (AI) yang mampu menunjang perancangan sistem serupa[2].

Pada awalnya, pendekatan yang digunakan untuk sistem PKM hanya terbatas pada pemanfaatan citra kamera[3]. Seiring dengan perkembangan teknologi,

berbagai macam pendekatan mulai bermunculan yang salah satunya adalah dengan menggunakan Radio Detection and Ranging (Radar)[4]. Pendekatan Radar menawarkan solusi penerapan sistem PKM tanpa kontak langsung yang tidak akan mengganggu penggunanya saat melakukan aktivitas. Selain itu, penggunaan Radar juga akan menghasilkan sistem PKM yang lebih tahan terhadap gangguan, seperti kondisi pencahayaan yang kurang optimal dan hambatan objek fisik, seperti dinding, yakni sebuah kekurangan yang dimiliki oleh sistem PKM berbasis kamera [4]. Terlebih lagi, penggunaan kamera juga menimbulkan adanya kecamasan terkait privasi dari pengguna yang dalam

beberapa kondisi penggunaan kamera akan terasa mengganggu bagi mereka [4]

Meskipun begitu, penggunaan Radar dalam sistem PKM belum banyak diterapkan saat ini. Hal ini mengingat pengolahan data Radar yang cenderung memiliki biaya komputasi tinggi. Selain itu, data hasil olahan Radar juga sulit untuk diinterpretasikan sehingga banyak pendekatan sistem PKM berbasis Radar biasanya akan disandingkan dengan implementasi AI dalam bentuk algoritma pembelajaran mesin [2] [5].

Penerapan sistem PKM dengan melibatkan AI salah satunya dilakukan oleh Shahzad dkk., pada Radar *Frequency Modulated Continuous Wave* (FMCW) dengan memanfaatkan algoritma **Convolutional Neural Network** (CNN) dan *Long Short-Term Memory* (LSTM) untuk menghasilkan sistem PKM dengan akurasi mencapai 91% [6].

Dengan semakin banyaknya penelitian mengenai sistem PKM berbasis Radar dengan melibatkan AI, penulis merasa perlu adanya pemilihan algoritma AI yang tepat digunakan untuk merancang sistem tersebut. Penelitian ini berfokus pada perbandingan performansi dari berbagai metrik untuk algoritma *Support Vector Machine* (SVM), CNN, serta LSTM yang akan digunakan untuk melakukan klasifikasi aktivitas manusia dalam sistem PKM. Adapun penelitian ini difokuskan pada penggunaan Radar FMCW dengan teknik pengolahan *Fast Fourier Transform* (FFT) dan *Short-time Fourier Transform* (STFT) dengan dataset yang diambil secara mandiri.

2. Tinjauan Pustaka

2.1 Algoritma Acuan

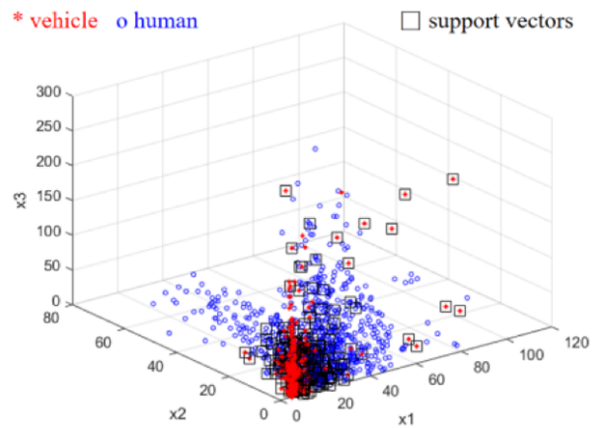
2.1.1 SVM (*Support Vector Machine*)

SVM pada dasarnya merupakan metode klasifikasi biner. Akan tetapi, dengan pendekatan *One vs One* (OvO) atau *One vs All* (OvA), algoritma ini dapat digeneralisasi untuk menyelesaikan permasalahan multi-class [7]. Konsep utama SVM terletak pada penyelesaian permasalahan optimasi untuk mencari vektor dalam ruang dimensi yang dapat mencapai pemisahan linear maksimum. Untuk sebuah sampel dataset training $\{(x_i, y_i), i = 1, \dots, n, x_i \in \{+1, -1\}\}$, di mana x_i merupakan vektor fitur, y_i adalah label terkait, untuk kondisi saat nilai pemisahan linear maksimum didapatkan dengan menyelesaikan permasalahan optimasi pada Pers. (1) dan Pers. (2),

$$\min \phi(w) = \frac{1}{2}(w \cdot w) \quad (1)$$

$$s.t. y_i(w \cdot x_i) + b \geq 1, i = 1, \dots, n$$

$$\max W(\alpha) = \sum_n \alpha_i - \frac{1}{2} \sum_n \sum_{ij=1} \alpha_i \alpha_j y_i y_j (y_i y_j) \quad (2)$$



Gambar 1. SVM diterapkan pada Doppler-Frequency divisualisasikan menggunakan Matlab [8]

$$s.t. \sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, \dots, n$$

Kesederhanaan algoritma SVM menjadi daya tarik untuk penerapan sistem prediksi terhadap data Radar. Ditunjukkan bahwa SVM dapat digunakan untuk membedakan kelas antar dua buah objek yang berbeda berdasarkan spektrum *Doppler-Frequency* yang diambil menggunakan Radar [8], SVM dapat melakukan pemisahan dua kelas tersebut menggunakan vektor ‘support’ yang dimaksimalkan nilai pemisahan linearnya.

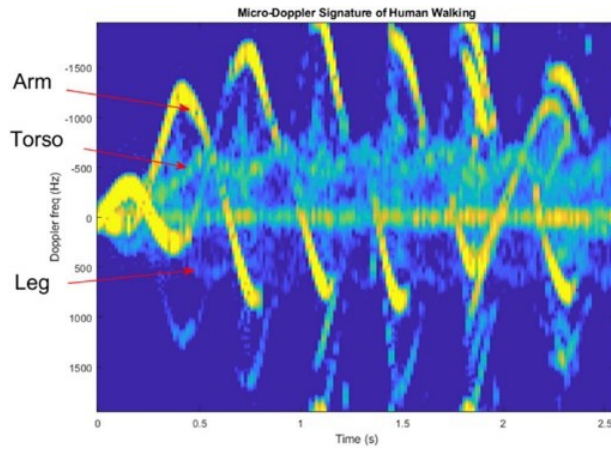
Gambar 1 menunjukkan bagaimana vektor ‘support’ ini divisualisasikan dalam ruang 3 dimensi menggunakan bantuan program Matlab. Bagian yang diberi persegi abu-abu merupakan vektor ‘support’ pada SVM, sementara titik biru dan merah menunjukkan fitur dari kelas yang diujikan [8].

2.1.2 Convolutional Neural Network

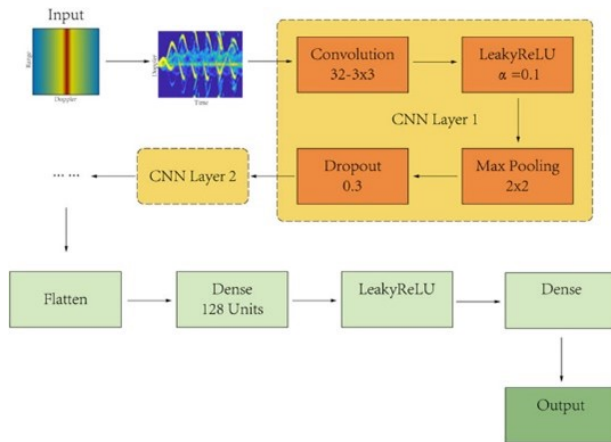
Convolutional Neural Network (CNN) adalah metode Deep Learning (DL) yang memanfaatkan proses konvolusi terhadap lapisan dalam Multi-layer Perceptron (MLP) [9]. CNN umumnya digunakan sebagai algoritma untuk mengklasifikasikan gambar atau objek visual karena CNN menerapkan respons dari neuron terhadap individu ketika melihat.

Proses konvolusi pada lapisan konvolusi kemudian akan dilanjutkan ke fungsi aktivasi. Dalam hal ini umumnya digunakan ReLU, yakni sebuah fungsi non-linear bernilai $ReLU = \max(0, x)$, untuk sebuah k kernel konvolusi berdimensi $n \times n \times q$ (n harus bernilai lebih kecil dari h atau w , q dapat bernilai sama dengan atau kurang dari r), keluaran dari CNN dengan ReLU activation berupa k feature maps dengan jumlah $(m-n-1)$.

Pers. (3) menunjukkan bagaimana keluaran dari satu buah lapisan konvolusi jika diberikan fungsi aktivasi ReLU.



Gambar 2. Penampakan Micro-Doppler ketika seseorang berjalan menggunakan CNN dan mmWave Radar [10]



Gambar 3. Arsitektur model CNN yang digunakan dengan masukan data spektrogram micro-doppler [10]

$$h^k = \max(0, f(W^k * x + b^k)) \quad (3)$$

CNN telah menjadi salah satu pilihan utama dalam sistem PKM, tercantum dalam sebuah penelitian yang dilakukan oleh Renyuan dkk., penelitian tersebut berfokus pada penggunaan algoritma CNN untuk membuat sebuah sistem PKM yang berjalan secara realtime menggunakan mmWave Radar, dengan memanfaatkan spektrum micro-Doppler yang dapat ditangkap oleh Radar bertipe tersebut [10].

Spektrum pada Gambar 2 merupakan spektrum micro-Doppler yang ditangkap oleh mmWave Radar. Spektrum ini akan digunakan untuk membangun sebuah model CNN dengan arsitektur yang ditunjukkan pada Gambar 3. Arsitektur terdiri tiga lapisan CNN dengan fungsi aktivasi LeakyReLU, Max Pooling 2D, serta Dropout [10].

2.1.3 Long Short-Term Memory (LSTM)

LSTM merupakan salah satu turunan dari keluarga Neural Network (NN) tipe Recurrent Neural Network (RNN) dengan kemampuan untuk menanggulangi permasalahan gradien meledak atau menghilang yang kerap terjadi pada RNN [11]. LSTM memiliki Cell State yang berfungsi sebagai ‘Long-term memory’ yang akan diteruskan pada setiap lapisan sekuen LSTM, serta umpan balik yang diteruskan dari satu sekuen sebelumnya yang memiliki peran sebagai ‘Short-term memory’.

Untuk sebuah vektor input $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ dengan vektor output $y(t) = [y_1(t), y_2(t), \dots, y_m(t)]$, maka bobot untuk satu unit LSTM untuk timestep t tertentu dapat dinyatakan dalam Pers. (4) hingga Pers. (7),

$$f(t) = \sigma_g(W_f x(t) + R_f h(t-1) + b_f) \quad (4)$$

$$i(t) = \sigma_g(W_i x(t) + R_i h(t-1) + b_i) \quad (5)$$

$$g(t) = \sigma_c(W_g x(t) + R_g h(t-1) + b_g) \quad (6)$$

$$o(t) = \sigma_g(W_o x(t) + R_o h(t-1) + b_o) \quad (7)$$

LSTM saat ini belum digunakan secara luas layaknya model SVM ataupun CNN dalam perancangan sistem PKM [12], namun terdapat beberapa publikasi telah menunjukkan bahwa LSTM tipe bidirectional dapat digunakan untuk menghasilkan sistem PKM berbasis Radar FMCW dengan akurasi yang sangat baik dan sistem tersebut dapat berjalan secara terus-menerus tanpa ada kendala [12, 13].

Fleksibilitas terhadap bentuk data masukan merupakan salah satu keunggulan LSTM (dan RNN secara umum) jika dibandingkan dengan model lain. LSTM dapat mengekstraksi fitur temporal dari frekuensi Doppler yang dihasilkan oleh Radar FMCW secara efisien dengan proses pemetaan menjadi representasi vektor dengan dimensi yang tetap [12].

2.2 Penelitian Sebelumnya

Permasalahan PKM dengan mengandalkan sebuah sistem berbasis AI dengan menggunakan data yang diambil dari sebuah Radar FMCW mempertimbangkan berbagai aspek, di antaranya adalah performansi model dalam segi akurasi, serta bobot komputasi yang digunakan. PKM telah menjadi perbincangan yang umum dikaitkan dengan kesulitan untuk mencapai akurasi yang baik.

Adapun, terdapat beberapa publikasi yang telah mengajukan solusi dengan menggunakan model yang memberikan hasil yang menjanjikan (cukup baik). Beberapa di antara algoritma yang paling umum ditemukan merupakan salah satu dari ketiga algoritma dalam fokus penelitian ini.

Tabel 1. Penelitian sebelumnya untuk PKM menggunakan Radar, dikategorikan berdasarkan tipe algoritma

Kategori	Judul Penelitian	Pencapaian	Keterangan
SVM	Y. Kim, Et al., "Human Activity Classification Based on MicroDoppler Signatures Using a Support Vector Machine," [14]	Hingga 92,8% akurasi tercapai untuk classifier SVM terhadap 12 subjek untuk 7 buah kelas kegiatan.	Bentuk penerapan secara ensemble learning.
SVM	P. R. Mohapatra, Et Al., "SVM-based Human Activity Recognition with Hyperparameter Tuning," [15]	±96,3% akurasi untuk classifier Linear_SVC terhadap 6 kelas kegiatan.	Model dilatih menggunakan metode K-Fold Cross Validation dengan K = 10.
CNN	R. Zhang, Et al., "Real-Time Human Motion Behaviour Detection via CNN Using mmWave Radar," [10]	64% akurasi untuk deteksi transisi duduk – berdiri dan berjalan, 91% akurasi untuk deteksi berjalan, 99% akurasi untuk gerakan melambai.	Tipe Radar yang digunakan merupakan model AWR1642 dari Texas Instruments dengan menggunakan ROS framework.
CNN	A. Helen Victoria, Et l., "Activity recognition of FMCW Radar human signatures using tower convolutional neural networks," [16]	Hingga 97,58% akurasi tercapai dengan menggunakan metode yang diajukan.	Tower CNN dengan menggabungkan CNN basis RGB, HSV, dan LAB color spaces.
LSTM	A. Shrestha, Et al., "Continuous Human Activity Classification From FMCW Radar With Bi-LSTM Networks," [12]	54% – 92% untuk model Doppler-LSTM, 69% – 98% untuk model Doppler Bi-LSTM.	Sistem deteksi dari model berjalan secara kontinu untuk 45 sekuens berbeda (1 sekuens berdurasi 35 detik).
LSTM	H. Li, Et al., "Bi-LSTM Network for Multimodal Continuous Human Activity Recognition and Fall Detection," [13]	96% akurasi untuk model diajukan menggunakan Bi-LSTM.	Model dengan metode penggabungan data Radar dan wrist IMU.

3. Metodologi Penelitian

3.1 Dataset

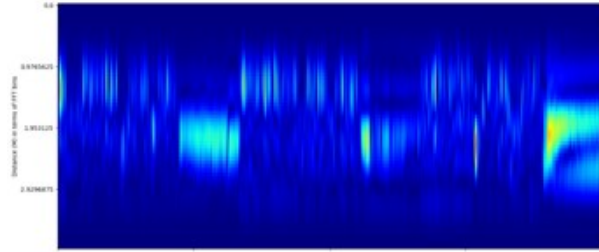
Akuisisi dilakukan dengan mengambil 50 sampel untuk setiap frame pengambilan yang untuk setiap detiknya terdapat 30 frame independen yang dapat diambil oleh Radar uRAD. Adapun data yang dihasilkan merupakan sinyal kompleks IQ dalam bentuk $Z = I + jQ$, untuk setiap sampel yang terdapat dalam setiap frame. Sinyal jenis IQ ini pada dasarnya merupakan sinyal kompleks yang dikomposisikan menjadi dua bagian, yaitu In-phase dan Quadrature.

Dengan meninjau cara kerja Radar FMCW yang untuk setiap interval frekuensinya akan dimodulasi dalam rentang bandwidth yang dimiliki oleh Radar, maka dengan melakukan FFT pada sinyal kompleks tersebut, kita dapat mengekstrak informasi mengenai jarak 1-D dari objek yang ditangkap pantulannya oleh Radar sehingga dapat dibuatkan spektrogram range-time distribution yang dapat digunakan sebagai acuan untuk merepresentasikan kegiatan manusia [8][9].

Adapun untuk 50 sampel yang diambil merupakan sampel dari gelombang pantul yang sumber gelombang tembaknya memiliki frekuensi yang dimodulasi berkisar pada rentang 24.005 GHz hingga 24.245 GHz. Gambar 4 menunjukkan bagaimana data diambil dengan subjek



Gambar 4. Posisi Radar beserta subjek saat pengambilan data



Gambar 5. Spektogram Range-Time Hasil FFT

yang diposisikan 2 hingga 3 meter dari Radar.

Tabel 2. Rangkaian Kegiatan pada Pengambilan Data

SEQ_1	SEQ_2	SEQ_3	SEQ_4
Berjalan	Merangkak	Berdiri	Berjalan
Merangkak	Berjalan	Berjalan	Berdiri
Berdiri	Berdiri	Duduk	Merangkak
Duduk	Duduk	Merangkak	Duduk

Tabel 2 menunjukkan 4 jenis rangkaian (diberi nama SEQ_1 hingga SEQ_4) kegiatan kontinyu yang dilakukan oleh 3 subjek yang berbeda, selama 30 detik untuk satu kali pengambilan, di mana keempat kelas tersebut dijalankan dalam rentang waktu 30 detik tersebut. Setiap rangkaian kegiatan tersebut terdiri dari 4 kelas yang nantinya akan dilakukan sistem PKM untuk mengenali kegiatan yang sedang dilakukan subjek menggunakan Radar FMCW. Sebuah kelas “null” ikut disertakan pada dataset untuk menunjukkan kondisi ketika subjek tidak berada dalam jangkauan deteksi Radar.

3.2 Pengolahan Data Radar

Data diberikan normalisasi terhadap nilai ADC-nya (data yang diberikan oleh uRAD memiliki lebar 12 bits, dengan rentang dari 0 – 4095) menjadi nilai voltase sinyal aslinya (nilai ADC untuk Raspberry Pi 4B+ adalah 3,3V untuk representasi bit ke-4095) [17]. Lalu data diteruskan untuk diberikan aplikasi FFT menggunakan modul Scipy.

$$V_{ADC} = \frac{X_{N_s}}{ADC_{intervals}} \times V_{ref} \quad (8)$$

$$X[k] = \sum_{n=0}^{N_s-1} x_{N_s}[n]e^{-j2\pi \frac{n}{N_s}[k]} \quad (9)$$

Proses ini dilakukan untuk mendekomposisi sinyal Radar dalam bentuk IQ yang diberikan menjadi sinyal kompleks penyusunnya dengan menggunakan FFT [18] dan menghasilkan spektogram seperti yang ditunjukkan oleh Gambar 5. Sinyal kompleks ini adalah sinyal yang merepresentasikan jarak 1-D dari objek yang ditangkap oleh Radar pada waktu tertentu. Pers. (8) dan (9) menunjukkan bagaimana normalisasi ADC dan FFT dilakukan.

X_{N_s} merupakan 50 sampel data yang diambil, di mana nilai tersebut akan diskalakan berdasarkan nilai tegangan V_{ref} Raspberry Pi 4B+ (3,3V) lalu diteruskan ke dalam proses FFT. K merupakan indeks frekuensi yang dapat direpresentasikan menjadi jarak dengan mengacu pada jarak maksimum yang didapatkan menggunakan Pers. [10],

$$Distance_{Max} = 75 \times \frac{N_s}{BW} \quad (10)$$

Setelah proses tersebut, dilakukan proses penghilangan komponen DC yang terdapat setelah proses FFT dan penghilangan objek statis yang akan menghambat model untuk mempelajari pola dari kegiatan yang dilakukan subjek.

Terakhir, data melalui satu kali tahapan FFT kembali dengan teknik STFT untuk mengekstrak informasi frekuensi bersama dengan waktu terjadinya frekuensi tersebut [19]. Pers. (11) menunjukkan bagaimana ini dilakukan dengan m adalah jarak dari jendela yang diaplikasikan bersama dengan fungsi windowing w . Hasil akhir data setelah dilakukan proses pengolahan data Radar adalah sebuah spektogram 3D yang terdiri dari dimensi $Waktu \times Jarak \times Frekuensi$, di mana besar ketiga dimensi tersebut setelah seluruh data diproses adalah (14400, 64, 32). Hal ini menunjukkan bahwa ada 14400 sampel berbeda yang diambil oleh Radar selama 480 detik, di mana untuk setiap sampel tersebut terdiri dari 64 indeks data jarak dan 32 indeks data frekuensi.

$$X[k, m] = \sum_{n=0}^{N-1} x[n]w[n + mH]e^{-j2\pi \frac{n}{N_s}[k]} \quad (11)$$

Data kemudian diubah ke dalam bentuk Windowed Dataset di mana setiap satu buah data terdiri dari 60 sampel beserta satu buah label yang mewakili sampel tersebut. Label ini ditentukan dari modulus label dari 60 sampel tersebut. Dataset kemudian dibagi menjadi dua buah subset data untuk pelatihan dan pengujian.

Tabel 3 menunjukkan bagaimana distribusi subset data setelah dibagi, dikategorikan berdasarkan kelas labelnya.

Tabel 3. Distribusi Kelas Windowed Datasets

Label	Subset dataset	
	Pelatihan	Pengujian
Berdiri	104	32
Berjalan	161	30
Duduk	123	22
Merangkak	151	38
'Kosong'	177	58

3.3 Algoritma yang Digunakan

Jumlah parameter pada model SVM dapat dihitung dari besarnya dimensi vektor ‘support’ dikalikan dengan sesamanya.

Tabel 4 menunjukkan berbagai arsitektur SVM yang akan digunakan untuk pengujian pada penelitian ini beserta jumlah parameter yang dimiliki oleh model terkait.

Tabel 4. Arsitektur SVM yang akan diujikan beserta jumlah parameternya

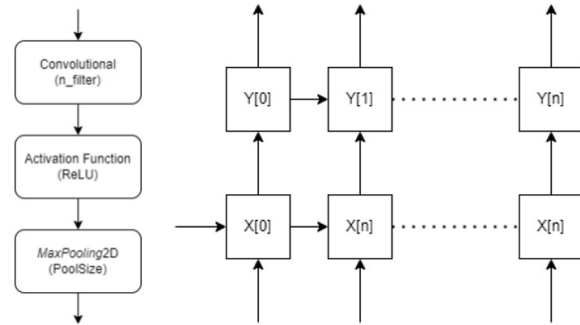
Arsitektur	Argumen Model	Total Param.
SVM_01	Kernel=Linear C=1	(485, 122880) 59.596.800,00
SVM_02	Kernel=Rbf; C=1	(588, 122880) 72.253.440,00
SVM_03	Kernel=Sigmoid; C=1	(494, 122880) 60.702.720,00
SVM_04	Kernel=Poly; C=1; Degree=2	(643, 122880) 79.011.840,00
SVM_05	Kernel=Poly; C=1; Degree=3	(666, 122880) 81.838.080,00

Untuk CNN dan LSTM, model dibangun dengan menggunakan Sequential API dari TensorFlow, di mana setiap lapisan dalam NN akan dibangun secara berurutan dari lapisan awal hingga lapisan akhir. CNN dan LSTM dalam hal ini akan berperan sebagai lapisan tengah yang akan menjembatani antara masukan dan keluaran dari model. Adapun parameter pelatihan model yang digunakan adalah sebagai berikut:

1. Epochs = 100
2. Batchsize = 32
3. Validationsplit = 0,1
4. Shuffle = False

Untuk model tipe CNN, akan terdiri dari beberapa lapis dari sebuah struktur lapisan yang selanjutnya disebut sebagai Convolutional Block. Untuk setiap Convolutional Block yang dibangun akan memiliki fungsi aktivasi ReLU serta Pooling Size bernilai 2. Untuk memudahkan notasi, Conv2D(16) menunjukkan sebuah Convolutional Block dengan $n_{filter}=16$.

Model LSTM dibangun serupa dengan CNN, dengan perbedaan hanya pada struktur lapisan LSTM, setiap lapis dihubungkan untuk dengan arsitektur sequence-to-sequence seperti yang ditunjukkan oleh Gambar 6. LSTM juga memiliki variasi dengan tipe bidirectional yang berfungsi untuk memberikan kemampuan LSTM memiliki unit umpan balik dua arah. Untuk memudahkan notasi, LSTM(16) memiliki arti



Gambar 6. Arsitektur Model Neural Network (a) CNN (b) LSTM Sequence-to-sequence

sebuah LSTM dengan 16 unit dan Bi(LSTM(16)) adalah sebuah LSTM dengan tipe bidirectional dengan 16 unit pada masing-masing arah.

Tabel 5 menunjukkan berbagai arsitektur dari model NN tipe CNN dan LSTM yang diujikan beserta struktur model yang digunakan.

Tabel 5. Arsitektur SVM yang akan diujikan beserta jumlah parameternya

Arsitektur	wStruktur	Total Param.
CNN_01	wConv2D(16)	62.917.285
CNN_02	wConv2D(16), Conv2D(32)	29.387.973
CNN_03	wConv2D(16), Conv2D(32), Conv2D(64)	12.662.021
CNN_04	wConv2D(16), Conv2D(32), Conv2D(64), Conv2D(64)	2.213.189
LSTM_01	wLSTM(16)	294.725
LSTM_02	wLSTM(16), LSTM(32)	331.717
LSTM_03	wBi(LSTM(16))	588.677
LSTM_04	wBi(LSTM(16)), Bi(LSTM(32))	666.757

4. Pembahasan dan Hasil

Pada model uji pada Tabel 4 dan Tabel 5 dilakukan proses pelatihan pada Jupyter Notebook yang dieksekusi dengan memanfaatkan Akselerasi Hardware apabila memungkinkan. Adapun spesifikasi sistem yang digunakan adalah sebagai berikut: AMD Ryzen™ 5 5600U, 16GB RAM, 1.5TB NVME SSD, GeForce RTX 3050 (Laptop) 35W.

Dalam pengujian performansi, model akan dinilai dari berbagai metrik evaluasi yang umumnya digunakan

pada penilaian model pembelajaran mesin. Adapun metrik tersebut meliputi Total Param., waktu pelatihan, Precision, Recall, F1-Score, dan akurasi. Metrik tersebut akan dikategorikan menjadi dua kategori, yakni metrik ‘positif’ dan ‘negatif’. Pembagian dari kategori ini dapat dilihat pada Tabel 6. Metrik yang dikategorikan ini akan dinormalisasi nilainya terhadap model SVM_01 , lalu digunakan untuk memberikan Scoring akhir untuk performansi secara keseluruhan sebuah model. Nilai dari Scoring akhir ini didapatkan berdasarkan Pers. (12).

Tabel 6. Metrik Evaluasi Model

Metrik ‘Positif’	Metrik ‘Negatif’
Precision	Total Param.
Recall	Waktu Pelatihan
F1-Score	
Akurasi	

$$Score = \frac{\prod Normalized_P_Param}{\prod Normalized_N_Param} \quad (12)$$

Total Param. beserta waktu pelatihan menunjukkan parameter negatif dari sebuah model yang nilainya proporsional dengan tingkat kompleksitas dari sebuah model. Pada dasarnya, waktu operasi yang harus dilakukan oleh model untuk memperoleh hasil prediksi bergantung pada jumlah parameter yang digunakan untuk melakukan operasi tersebut [20].

Sedangkan keempat metrik sisanya merupakan metrik umum yang digunakan untuk mengevaluasi permasalahan Supervised Learning tipe klasifikasi kelas, di mana semakin besar nilainya menunjukkan model dengan memiliki performansi yang lebih baik. Teknik umum yang digunakan untuk mengevaluasi berdasarkan metrik tersebut adalah dengan penggunaan Confusion Matrix, ditunjukkan pada Tabel 7.

Tabel 8 menunjukkan bagaimana hasil dari perancangan model serta hasil pengujian dengan menggunakan metrik evaluasi terhadap subset data uji. Model kemudian diberikan penilaian secara keseluruhan

Tabel 7. Confusion Matrix dengan metrik evaluasinya

		Actual		
		Positive	Negative	
Predicted	Positif	True Positive (TP)	False Positive (FP)	Precision: $\frac{TP}{TP+FP}$
	Negatif	False Negative (FN)	True Negative (TN)	
Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$		Recall: $\frac{TP}{TP+FN}$		F1-score: $\frac{TP}{TP+\frac{1}{2}(FP+FN)}$

menggunakan nilai yang dinormalisasi terhadap model SVM_01 .

Tabel 8. Hasil fitting untuk setiap model pengujian

Arsitektur	B1	G1	G2	G3	G4	S
SVM_01	25,50	0,85	0,8	0,8	0,84	1
SVM_02	36,75	0,78	0,75	0,74	0,8	0,53
SVM_03	19,69	0,73	0,72	0,7	0,77	1,12
SVM_04	46,45	0,83	0,57	0,52	0,64	0,32
SVM_05	49,44	0,61	0,4	0,35	0,46	0,20
CNN_01	254,24	0,86	0,82	0,81	0,85	0,09
CNN_02	455,87	0,83	0,81	0,81	0,84	0,11
CNN_03	526,89	0,85	0,83	0,83	0,86	0,23
CNN_04	548,17	0,82	0,79	0,79	0,82	1,22
LSTM_01	54,02	0,86	0,83	0,83	0,86	98,0

* nilai diatas sudah dinormalisasi terhadap SVM_01

** B1= Waktu Pelatihan, G1=Precision, G2=Recall, G3=F1-Score, G4=Akurasi, S=Scoring

Terlihat bahwa model SVM_01 adalah model yang paling cepat untuk dilakukan proses pelatihan. Hal ini masuk akal karena model tersebut memiliki jumlah parameter yang paling sedikit dibandingkan model tipe SVM lainnya dan karena SVM hanya melakukan iterasi sebanyak satu kali terhadap dataset, maka proses pelatihan yang dilakukan untuk model ini juga akan jauh lebih cepat jika dibandingkan NN.

Terlihat juga pada hasil pengujian ini, model CNN_03 dapat memberikan hasil evaluasi model yang paling baik, dan sebanding dengan model selanjutnya pada tipe model LSTM, yakni model $LSTM_01$ dengan satu buah layer LSTM 16 unit. LSTM dikenal sebagai salah satu model yang sangat mudah terjadinya overfitting sehingga satu buah layer dirasa sudah cukup untuk memberikan model tanpa menyebabkan overfitting yang berlebihan

5. Kesimpulan

Berdasarkan penelitian dan pengujian yang telah dilakukan oleh penulis, sistem PKM menggunakan Model SVM memiliki waktu train yang relatif cepat dibandingkan dengan model tipe lainnya, serta menawarkan performansi yang cukup bisa dipertimbangkan dibandingkan model lainnya. Model CNN menawarkan salah satu performansi yang terbaik, namun dengan pengorbanan pada waktu pelatihan yang lama serta jumlah parameter yang tinggi. Model tipe LSTM merupakan model yang paling mudah mengalami overfitting namun terbukti bahwa untuk sebuah model LSTM sederhana seperti model $LSTM_01$ dapat dengan mudah menyelesaikan permasalahan PKM dengan performansi yang baik dan efisien dengan waktu pelatihan cepat (54,02 detik saja) serta jumlah parameter yang minimal, yakni 294.725 Total Param.

Daftar Pustaka

- [1] Z. Hussain, M. Sheng, and W. E. Zhang, "Different approaches for human activity recognition: A survey," *arXiv preprint arXiv:1906.05074*, 2019.
- [2] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, and J. S. Suri, "Human activity recognition in artificial intelligence framework: A narrative review," *Artificial intelligence review*, vol. 55, no. 6, pp. 4755–4808, 2022.
- [3] J. K. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.
- [4] Z. Li, J. Le Kernec, Q. Abbasi, F. Fioranelli, S. Yang, and O. Romain, "Radar-based human activity recognition with adaptive thresholding towards resource constrained platforms," *Scientific Reports*, vol. 13, no. 1, p. 3473, 2023.
- [5] W. Z. Tee, R. Dave, J. Seliya, and M. Vanamala, "A close look into human activity recognition models using deep learning," in *2022 3rd International Conference on Computing, Networks and Internet of Things (CNIOT)*. IEEE, 2022, pp. 201–206.
- [6] S. Ahmed, J. Park, and S. H. Cho, "Fmcw radar sensor based human activity recognition using deep learning," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2022, pp. 1–5.
- [7] Y. F. D. Alita and H. Sulistiani, *Image Classification Using SVMs: One-against-One vs One-against-All*, vol. xiv, no. 2, pp. 86-91 ed., Jurnal TEKNOKOMPAK, June 2020.
- [8] E. Hyun and Y. Jin, "Doppler-spectrum feature-based human-vehicle classification scheme using machine learning for an fmcw radar sensor," *Sensors*, vol. 20, no. 7, p. 2001, 2020.
- [9] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16–28, 2019.
- [10] R. Zhang and S. Cao, "Real-time human motion behavior detection via cnn using mmwave radar," *IEEE Sensors Letters*, vol. 3, no. 2, pp. 1–4, 2018.
- [11] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International conference on machine learning*. PMLR, 2015, pp. 2342–2350.
- [12] A. Shrestha, H. Li, J. Le Kernec, and F. Fioranelli, "Continuous human activity classification from fmcw radar with bi-lstm networks," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13 607–13 619, 2020.
- [13] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-lstm network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1191–1201, 2019.
- [14] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE transactions on geoscience and remote sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.
- [15] P. R. Mohapatra, Y. S. Babu, and R. K. Sahu, "Svm-based human activity recognition with hyperparameter tuning."
- [16] A. Helen Victoria and G. Maragatham, "Activity recognition of fmcw radar human signatures using tower convolutional neural networks," *Wireless Networks*, pp. 1–17, 2021.
- [17] *Raspberry Pi*, "Raspberry Pi 4 Product Brief," June 2019, available at <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>.
- [18] S. Z. Gurbuz, *Deep Neural Network Design for Radar Applications*. SciTech Publishing, 2020.
- [19] L. Cohen, *Time-Frequency Analysis*, Pearson College Div, 1994.
- [20] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Knowledge and Information Systems*, vol. 63, pp. 2585–2619, 2021.