Available at https://jtsiskom.undip.ac.id (3 July 2020)
DOI:10.14710/jtsiskom.2020.13648

Jurnal Teknologi dan Sistem Komputer, 8(3), 2020, 246-254

Optimasi nilai k dan parameter lag algoritme k-nearest neighbor pada prediksi tingkat hunian hotel

Optimization of k value and lag parameter of k-nearest neighbor algorithm on the prediction of hotel occupancy rates

Agus Subhan Akbar[1*)], R. Hadapiningradja Kusumodestoni[2)]

[1)]*Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Islam Nahdlatul Ulama Jepara*
Jl. Taman Siswa, Pekeng, Tahunan, Jepara, Jawa Tengah, Indonesia 59451

[2)]*Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Nahdlatul Ulama Jepara*
Jl. Taman Siswa, Pekeng, Tahunan, Jepara, Jawa Tengah, Indonesia 59451

Cara sitasi: A. S. Akbar and R. H. Kusumodestoni, "Optimasi nilai k dan parameter lag algoritme k-nearest neighbor pada prediksi tingkat hunian hotel," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 246-254, 2020. doi: 10.14710/jtsiskom.2020.13648, [Online].

Abstract* – *Hotel occupancy rates are the most important factor in hotel business management. Prediction of the rates for the next few months determines the manager's decision to arrange and provide all the needed facilities. This study performs the optimization of lag parameters and k values of the k-Nearest Neighbor algorithm on hotel occupancy history data. Historical data were arranged in the form of supervised training data, with the number of columns per row according to the lag parameter and the number of prediction targets. The kNN algorithm was applied using 10-fold cross-validation and k-value variations from 1-30. The optimal lag was obtained at intervals of 14-17 and the optimal k at intervals of 5-13 to predict occupancy rates of 1, 3, 6, 9, and 12 months later. The obtained k-value does not follow the rule at the square root of the number of sample data.

Keywords *– hotel occupancy rate; kNN regression; k optimization; lag; kNN prediction*

Abstrak* – *Tingkat hunian hotel merupakan faktor terpenting dalam pengelolaan bisnis perhotelan. Prediksi tingkat hunian hotel untuk beberapa bulan ke depan menentukan keputusan pengelola untuk mengatur dan menyediakan semua fasilitas yang diperlukan di hotel tersebut. Penelitian ini melakukan optimalisasi parameter lag dan nilai optimal k dari algoritme k-Nearest Neighbor pada data histori tingkat hunian hotel. Data histori tingkat hunian hotel disusun dalam bentuk data pelatihan supervised dengan jumlah kolom setiap baris sesuai dengan parameter lag dan jumlah target prediksi. Algoritme kNN diterapkan dengan menggunakan validasi silang 10-fold dan variasi nilai k dari 1-30. Dari hasil uji coba didapatkan lag optimal diperoleh pada interval 14-17 dan nilai k optimal pada interval 5-13 untuk prediksi tingkat hunian 1, 3, 6, 9, dan 12 bulan berikutnya. Nilai k terbaik yang diperoleh tidak mengikuti kaidah pada akar kuadrat jumlah sampel data.

Kata kunci *– tingkat hunian hotel; regresi kNN; optimalisasi k; lag; prediksi kNN*

[*)] Penulis korespondensi (Agus Subhan Akbar)
Email: agussa@unisnu.ac.id

I. PENDAHULUAN

Hotel merupakan tempat singgah sementara bagi tamu selama melaksanakan kegiatannya. Kegiatan tersebut dapat berupa urusan kantor, bisnis, wisata, ataupun ketiganya. Selama singgah di hotel, para tamu mengharapkan kenyamanan dan pelayanan yang terbaik dari hotel tesebut.

Jumlah tamu yang menginap di hotel sangat dipengaruhi oleh waktu. Ada saat-saat tertentu dimana tamu yang menginap sangat banyak. Ada juga waktu-waktu tertentu dimana jumlah tamu yang menginap menurun. Waktu-waktu dengan jumlah tamu banyak adalah saat liburan sekolah, liburan natal, liburan hari raya idul fitri, dan liburan tahun baru atau saat awal tahun saat pekerja kantor atau bisnis sering melakkan aktivitas seminar atau pelatihan. Sebagian peserta seminar atau pelatihan tersebut juga memerlukan untuk menginap.

Tingkat hunian hotel merupakan rasio jumlah tamu yang menginap terhadap jumlah kamar yang tersedia. Semakin banyak kamar yang terpakai, maka tingkat hunian semakin tinggi. Demikian juga sebaliknya, semakin sedikit kamar yang terpakai, tingkat hunian semakin turun.

Tingkat hunian hotel ini merupakan faktor terpenting bagi bisnis perhotelan. Dengan tingkat hunian yang tinggi, maka penghasilan bisnis pun menjadi meningkat. Tingkat hunian yang tinggi memerlukan penyediaan fasilitas pendukung pun mengikuti tingkat hunian yang meningkat. Begitu juga sebaliknya, tingkat hunian yang menurun menyebabkan penghasilan bisnis menurun. Penyediaan fasilitas juga perlu diturunkan untuk keseimbangan. Dari hal tersebut, pengetahuan tentang tingkat hunian bulan-bulan berikutnya diperlukan.

Pengetahuan tingkat hunian bulan-bulan berikutnya bisa dilakukan dengan memprediksi berdasarkan data-data

Submitted: 30 January 2020; Revised: 26 April 2020; Accepted: 6 May 2020; Published: 31 July 2020

terdahulu. Teknologi dan algoritme prediksi yang digunakan telah berkembang pesat, mulai dari penggunaan *machine learning* sampai penggunaan *deep learning*. Algoritme yang sering digunakan untuk analisis, khususnya prediksi, adalah k-Nearest Neighbor (kNN) [1].

Algoritme kNN ini sederhana, namun cukup efektif untuk klasifikasi dan prediksi. Algoritme ini tidak memerlukan proses pelatihan seperti algoritme lain, namun langsung menggunakan data pelatihan tersebut sebagai basis modelnya sehingga tidak perlu menyimpan parameter bobot seperti di model jaringan syaraf tiruan. Dengan kata lain, algoritme ini dikelompokkan ke dalam *lazy algorithm* [2].

Data uji diklasifikasikan dengan cara menghitung jarak data tersebut dengan semua data latih yang ada. Perhitungan jarak menggunakan sejumlah alternatif, di antaranya *euclidean*, *manhattan*, dan *minkowski* [3]. Data jarak tersebut diurutkan dari yang terdekat dan diambil sejumlah *k* data yang memiliki jarak terdekat dengan data uji. Untuk proses klasifikasi, *k* data tersebut dicari label dengan frekuensi terbanyak. Data uji dikategorikan ke dalam label dengan frekuensi terbanyak. Untuk proses regresi, luaran dari data uji didapatkan dari rata-rata *k* data terdekat tersebut [4], [5].

Penggunaan kNN di bidang klasifikasi dan regresi telah banyak digunakan, di antaranya pengenalan tulisan tangan dengan huruf Yoruba [6]. Prediksi kepadatan lalu lintas di kota-kota besar juga telah memanfaatkan kNN dalam pengelolaan menjadi kota cerdas [7]. Klasifikasi uang kertas Rupiah juga memanfaatkan kNN dalam [8]. Deteksi kegagalan pada perangkat lunak dilakukan menggunakan kNN dalam [5]. Penerapan kNN untuk data *timeseries* juga telah dilakukan dalam [9]. Riset di bidang pertanian juga telah menggunakan kNN dengan membandingkan dengan sejumlah algoritme klasifikasi lainnya [10]. Presisi yang dicapai oleh beberapa variasi kNN mencapai 85 %.

Kajian penggunaan algoritme kNN tersebut menunjukkan bahwa kemampuan kNN sangat tergantung pada nilai *k* yang digunakan. Beragam kajian mengkaji bagaimana menentukan nilai *k* terbaik, seperti dalam [4], [11]-[13]. Namun, penentuan nilai *k* masih trivial dan sangat tergantung pada dataset yang digunakan. Secara khusus, Zhang dkk. [13] menggunakan *k-means clustering* untuk membagi dataset menjadi sejumlah klaster dan menerapkan kNN untuk masing-masing klaster secara terpisah. Di sisi lain, Song dkk. [14] menerapkan pembobotan pada algoritme kNN untuk klasifikasi dengan *direct cost sensitive* dan *distance cost sensitive* kNN. Penanganan prediksi data *timeseries* yang memiliki sifat musiman juga telah dikaji dalam [15]. Penanganannya dengan membagi data training sesuai periode musiman tersebut dan hasil prediksi merupakan agregat dari sejumlah nilai target.

Perhitungan nilai *k* masih menjadi tantangan yang menarik, terutama untuk prediksi tingkat hunian hotel yang belum dikaji. Artikel ini berfokus untuk mengkaji penerapan alogoritma kNN untuk data *timeseries* pada tingkat hunian hotel. Selain itu, penerapan kNN di data ini digunakan untuk melakukan prediksi n-bulan ke depan dan menentukan jumlah data tingkat hunian sebelumnya yang tepat untuk bisa digunakan melakukan prediksi tingkat hunian bulan-bulan berikutnya. Kajian ini menggunakan formulasi perhitungan tingkat kesalahan prediksi sebagai ukuran kinerja, yaitu dengan *symetric mean absolute precentage error* (SMAPE), *mean forecast error* (MFE), *mean absolute error* (MAE), dan *root mean square error* (RMSE).

II. METODE PENELITIAN

A. Dataset

Dataset yang digunakan dalam penelitian ini berasal dari data tingkat hunian hotel K di Kudus. Data yang tersedia adalah mulai dari tahun 2006 sampai dengan bulan juni 2019. Data tersebut berbentuk tingkat hunian hotel bulanan dalam persentase. Jumlah data yang tersedia 162. Data tingkat hunian tersebut dinyatakan seperti pada Gambar 1. Masing-masing titik menyatakan tingkat hunian dalam satu bulan dalam persentase terhadap total.

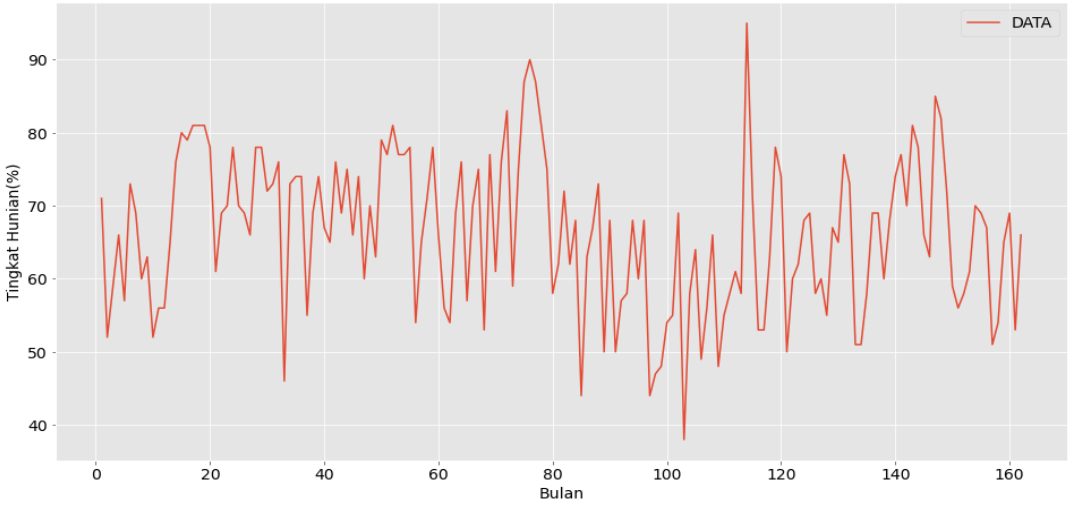
Data tingkat hunian ini berbentuk *timeseries*. Untuk penerapan algoritme kNN, data ini harus disusun dalam bentuk yang bisa digunakan untuk pelatihan *supervised* yang bisa diterima kNN.

B. Prosedur

Data tingkat hunian yang berbentuk data *timeseries* dan disusun dalam bentuk yang bisa diterima oleh algoritme kNN. Untuk memprediksi data pada waktu *t*, digunakan sejumlah data sebelumnya. Data sebelumnya bisa berjumlah 1, 2 atau *n*. Jumlah *n* data sebelumnya yang digunakan ini dinamakan dengan *lag*.

Jika terdapat data pada *t0* sampai dengan *t6* seperti yang terlihat di Tabel 1, maka data tersebut bisa disusun dalam bentuk model *supervised* dengan menentukan *lag* yang digunakan. Nilai *lag=2* berarti bahwa untuk memprediksi data ke *t* dibutuhkan 2 data sebelumnya. Untuk *lag=3*, maka 3 data sebelumnya dibutuhkan untuk memprediksi data ke *t*. Penyusunan data latih dari data *timeseries* ke data latih untuk *lag=2* dan *lag=3* dinyatakan dalam Tabel 1 [9]. Pemilihan *lag* yang tepat untuk memprediksi nilai luaran yang mendekati nilai riilnya menjadi tantangan. Kajian ini menggunakan beberapa variasi *lag*, yaitu mulai dari 1 sampai 36, untuk menghasilkan satu nilai luaran tersebut.

Eksperimen berikutnya dilakukan dengan menyusun data dengan *lag* tertentu untuk menghasilkan nilai luaran satu atau beberapa bulan berikutnya. Prediksi yang dilakukan terdiri atas 1 bulan, 3 bulan, 6 bulan, dan 12 bulan berikutnya. *Lag* yang digunakan dimulai dari 1 sampai dengan 36. Dengan lag tersebut, data sampel berada pada interval 115-161 data. Nilai *k* dalam kNN diuji untuk nilai 1 sampai dengan 30. Pemilihan nilai *k* ini untuk pembuktian bahwa nilai *k* berada pada nilai akar dari jumlah sampel seperti dinyatakan dalam [1].



Gambar 1. Data tingkat hunian hotel K (2006 - 2019)

Pemilihan nilai *k* dalam kNN dilakukan secara empiris. Setiap nilai *k* diuji dan dibandingkan untuk mendapatkan nilai prediksi dengan kinerja terbaik. Ukuran yang digunakan untuk menunjukkan kinerja prediksi dalam kajian ini adalah SMAPE, RMSE, MAE, dan MFE. Tahapan pemrosesan dalam kajian ini dinyatakan dengan diagram alir pada Gambar 2.

C. Pengukuran kinerja

Pengukuran kinerja hasil prediksi dengan data riil menjadi ukuran seberapa dekat hasil prediksi tersebut. Pengukuran kinerja dilakukan dengan menghitung tingkat kesalahan hasil prediksi dan data riil. Kajian ini menggunakan formulasi perhitungan tingkat kesalahan prediksi sebagai ukuran kinerja, yaitu SMAPE, RMSE, MFE, dan MAE. Ukuran kinerja model tersebut digunakan untuk menilai seberapa baik model tersebut.

Ukuran SMAPE digunakan untuk menyatakan persentase absolut kesalahan. Nilai kesalahan dihitung dengan mengurangkan nilai aktual dengan nilai hasil prediksi. Persamaan 1 digunakan untuk menghitung SMAPE, dengan *F* merupakan nilai prediksi dan *A* merupakan nilai aktual [16]. Persamaan tersebut merupakan modifikasi dari persamaan dalam [17] yang bisa menghasilkan nilai presentase 200 % sehingga modifikasi SMAPE ini akan menghasilkan nilai maksimal 100 %.

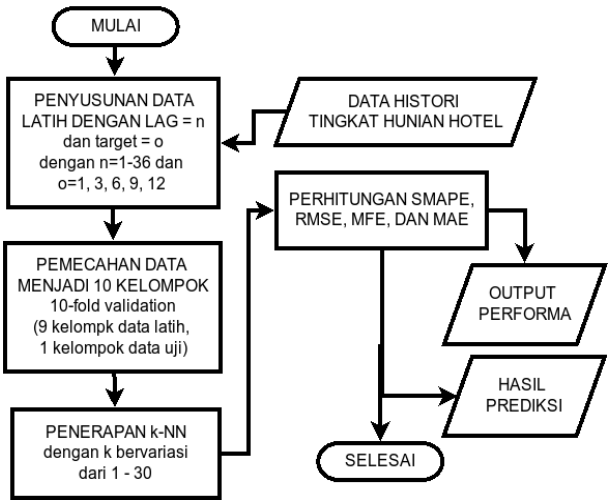
$$SMAPE = \frac{100\,\%}{n} * \sum_{t=1}^{n} \frac{|F_t - A_t|}{|F_t| + |A_t|} \quad (1)$$

RMSE menyajikan standar deviasi dari tingkat kesalahan prediksi. RMSE ini digunakan untuk mengukur sebaran tingkat kesalahan yang terjadi dari nilai prediksi dengan nilai riil. RMSE digunakan di sejumlah kajian, di antaranya dalam [3], [18], dan [19]. RMSE dihitung dengan Persamaan 2.

$$RMSE = \sqrt{\sum_{t=1}^{n} (F_t - A_t)^2 / n} \quad (2)$$

Tabel 1. Susunan data pelatihan

Data Asal	Lag=2		Lag=3	
	Masukan	Target	Masukan	Target
t0				
t1				
t2	t0,t1	t2		
t3	t1,t2	t3	t0,t1,t2	t3
t4	t2,t3	t4	t1,t2,t3	t4
t5	t3,t4	t5	t2,t3,t4	t5
t6	t4,t5	t6	t3,t4,t5	t6

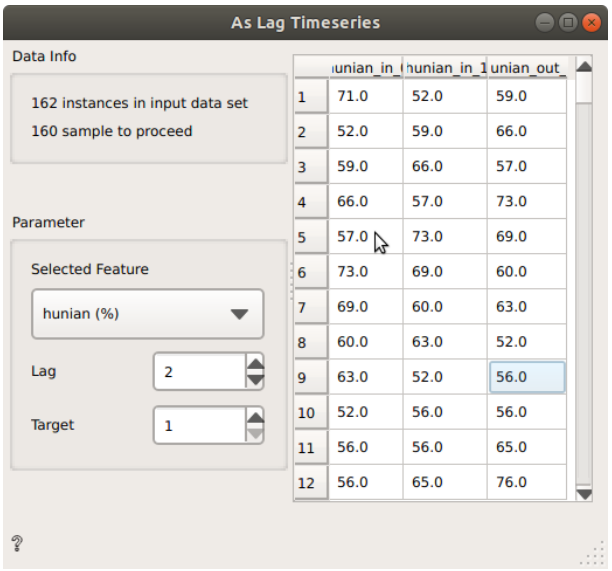


Gambar 2. Tahapan proses prediksi

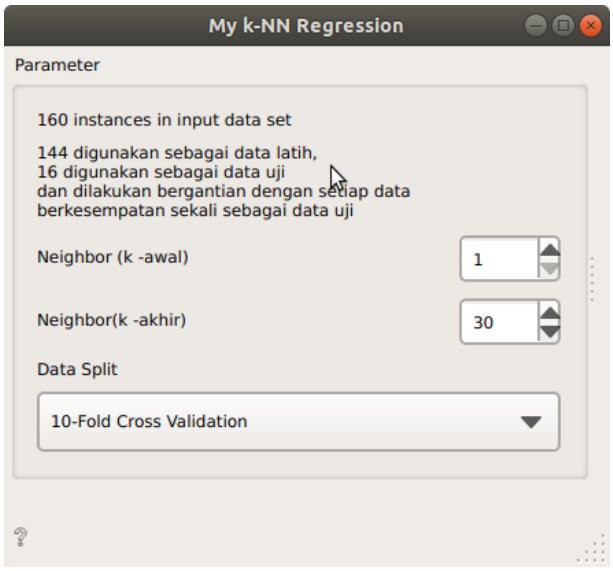
MAE digunakan untuk menghitung nilai rata-rata absolut kesalahan prediksi. MAE dihitung dengan menggunakan Persamaan 3.

$$MAE = \frac{\sum_{t=1}^{n} |F_t - A_t|}{n} \quad (3)$$

MFE menyatakan deviasi rata-rata dari hasil prediksi dibandingkan dengan nilai aktualnya. Nilai MFE bisa



Gambar 3. Widget As Lag Timeseries



Gambar 4. Widget My k-NN Regression

positif ataupun negatif. Nilai positif menyatakan prediksi yang lebih rendah dibandingkan dengan nilai aktual, sedangkan nilai negatif menyatakan nilai prediksi yang lebih tinggi dibandingkan dengan nilai aktual. MFE dihitung dengan menggunakan Persamaan 4.

$$MFE = \frac{\sum_{t=1}^{n} (F_t - A_t)}{n} \quad (4)$$

D. Perangkat lunak

Penelitian ini menggunakan perangkat Orange data mining versi 3.24. Orange data mining merupakan alat yang memiliki visualisasi berbasis grafis dengan memanfaatkan pustaka Scikit [20], [21]. Basis bahasa pemrograman yang digunakan adalah Python. Sejumlah *widget* digunakan untuk analisis, yaitu *As lag Timeseries* dan *My kNN Regression.*

Widget As Lag Timeseries dan *My kNN Regression* dibuat dengan menggunakan perangkat lunak PyCharm [22]. *As Lag Timeseries* digunakan untuk mengubah data timeseries menjadi data untuk pelatihan *supervised* berdasarkan nilai lag (berapa data sebelumnya) yang digunakan untuk memprediksi data berikutnya. Parameter yang dibutuhkan meliputi fitur data yang akan digunakan, nilai lag, dan jumlah nilai target yang akan diprediksi seperti yang disajikan di Gambar 3. Untuk data hunian, fitur yang dipilih adalah tingkat hunian. Pengujian untuk mendapatkan data prediksi 1 bulan berikutnya dari 2 bulan sebelumnya dengan cara mengisi nilai lag 2 dan target 1.

Widget My kNN Regression digunakan untuk memproses data masukan dari *widget* sebelumnya yang berupa data pelatihan *supervised*. Parameter yang diperlukan untuk *widget* ini meliputi range nilai *k* (awal dan akhir). Model pengujiannya dinyatakan di Gambar 4.

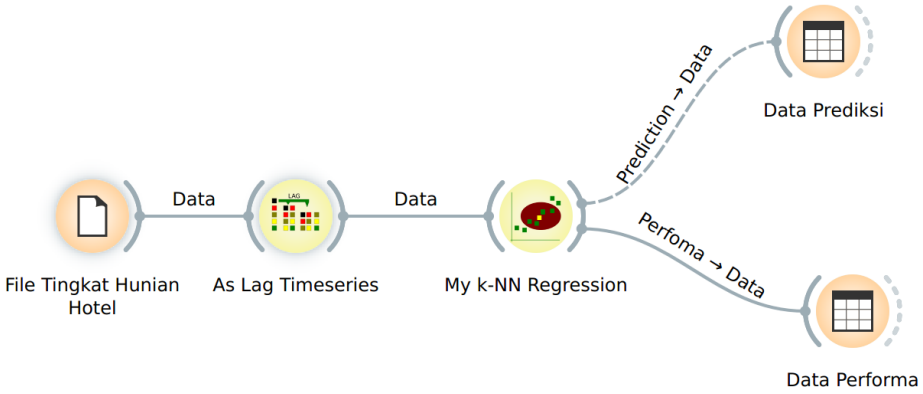
Model pengujian yang disediakan 25:75, 50:50, 75:25, 5-*fold cross validation*, dan 10-*fold cross validation.* Model pengujian 25:75 berarti dari data yang digunakan 25 % data digunakan sebagai data latih dan 75 % data sisanya sebagai data uji. Persentase ini juga berlaku untuk model 50:50 dan 75:25. Model pengujian *fold cross validation* yang terdiri atas 5 dan 10 berarti bahwa data dibagi menjadi 5 atau 10 kelompok data. Dari 5 kelompok tersebut, 4 kelompok data digunakan sebagai data latih dan 1 kelompok digunakan sebagai data uji. Untuk 10-*fold cross validation*, 9 kelompok digunakan sebagai data latih dan 1 kelompok digunakan sebagai data uji. Satu kelompok data uji dipilih dari setiap kelompok sehingga setiap kelompok data pasti pernah menjadi data uji. Pembagian kelompok ini tanpa memungkinan overlap untuk setiap data yang ada.

Widget My kNN Regression ini disusun dengan menggunakan pustaka Sckit. Fungsi pustaka yang digunakan adalah *sklearn.neigbors*. Sebagian grafik yang ditampilkan menggunakan perangkat Google Colaboratory yang tersedia secara bebas dengan menggunakan pustaka Matplotlib [23].

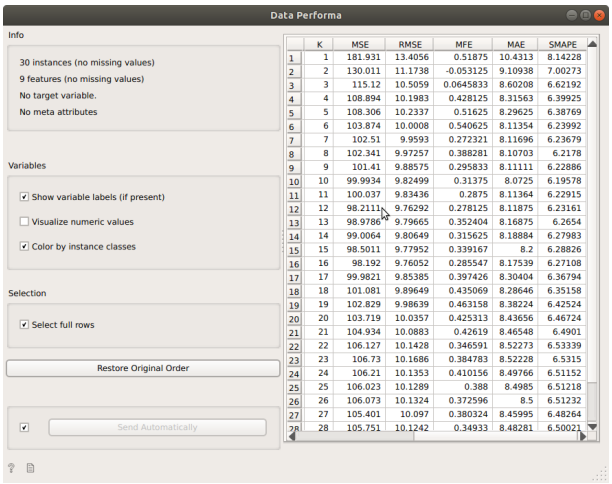
E. Arsitektur

Arsitektur penelitian yang digunakan dinyatakan pada Gambar 5. Arsitektur ini dirancang di perangkat Orang data mining versi 3.24. Widget yang digunakan meliputi *File, As Lag Timeseries, My kNN Regression*, dan *Data Table*. *Wdiget File* digunakan untuk membaca data dari file berformat csv yang berisi data tingkat hunian dengan 6 kolom tahun, bulan, jumlah kamar (deluxe, superior, dan standar) dan tingkat hunian per-bulan dalam persentase.

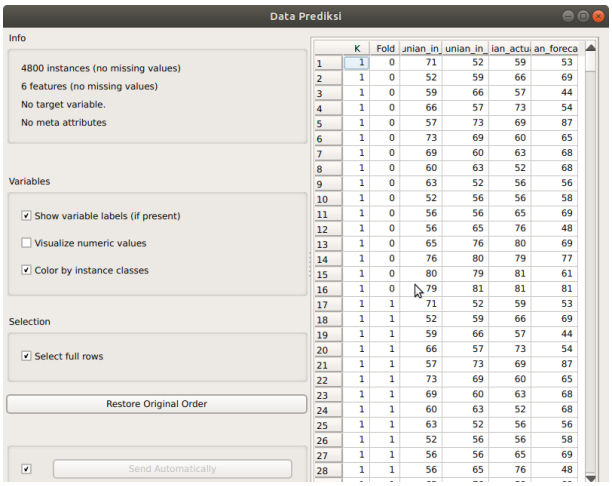
Widget As Lag Timeseries digunakan untuk memilih fitur yang akan digunakan. Fitur yang dipilih adalah tingkat hunian. Untuk pemilihan nilai *lag* dan target diuji untuk beberapa alternatif. Nilai *lag* diuji untuk nilai 1 sampai dengan 36 (1 bulan dan 36 bulan sebelumnya). Nilai target dipilih untuk 1 bulan berikutnya, 3 bulan, 6 bulan, 9 bulan, dan 12 bulan berikutnya.



Gambar 5. Arsitektur penelitian



Gambar 6. Widget data performa



Gambar 7. Widget data prediksi

Widget My kNN Regression digunakan untuk menerapkan dan menguji penggunaan algoritme kNN. Parameter k yang digunakan meliputi interval nilai 1 sampai dengan 30. Pengujian menggunakan 10-*fold cross validation*. Luaran dari *widget* ini ada dua, yaitu data performa (kinerja) dan data prediksi. Data performa berisi hasil perhitungan tingkat kesalahan yang meliputi MSE, RMSE, MFE, MAE, dan SMAPE. Data prediksi menyajikan hasil prediksi untuk setiap data uji sesuai metode pengujian dan masing-masing parameter yang digunakan. Tiap luaran ini ditampilkan dengan *widget data table* dalam Gambar 6 dan Gambar 7.

III. HASIL DAN PEMBAHASAN

A. Pengaruh variasi nilai k dan lag untuk prediksi 1 bulan berikutnya

Nilai *k* diuji mulai dari 1 sampai dengan 30. Nilai *lag* diuji untuk nilai 1 sampai 36 untuk prediksi nilai tingkat hunian 1 bulan ke depan. Hasil yang diperoleh disajikan dalam Tabel 2. JS pada judul kolom menyatakan jumlah sampel.

Nilai SMAPE terbaik diperoleh saat *lag* 16 dan parameter *k* bernilai 7. Jumlah *lag* yang meningkat tidak otomatis memberikan hasil SMAPE yang lebih rendah. Ada jumlah *lag* tertentu yang menghasilkan SMAPE yang lebih baik. Hal ini konsisten dengan nilai-nilai pengukuran kesalahan lainnya, yaitu MAP, MFE, dan RMSE. Perbedaan dengan nilai MFE yang terbaik dicapai pada saat *lag* bernilai 18 dan *k* bernilai 4. Nilai MFE merepresentasikan tingkat bias prediksi dengan nilai aktual. Jadi, semakin mendekati 0, maka biasnya semakin kecil. Namun, bias yang mendekati 0 akan menimbullkan dugaan terjadi *overfitting* seperti dinyatakan [24]. Nilai *k* yang terbaik juga bervariasi, tidak terlalu dekat dengan akar dari jumlah sampel seperti disebutkan dalam [1], namun lebih bersifat trivial atau uji coba seperti dalam [11].

B. Pengaruh variasi nilai k dan lag untuk prediksi 3 bulan berikutnya

Prediksi data 3 bulan berikutnya menggunakan variasi nilai *k* dari 1 sampai dengan 30 dan variasi *lag* dari 1 sampai dengan 36 bulan berikutnya. Hasil pengujian dinyatakan dalam Tabel 3.

Faktor *lag* data pada tingkat tertentu menghasilkan nilai SMAPE yang optimal. Namun, seiring dengan bertambahnya lag, nilai SMAPE tidak menurun. Nilai

Tabel 2. Kinerja prediksi 1 bulan

JS	LAG	K	RMSE	MFE	MAE	SMAPE
161	1	22	9.606	**0.100**	7.891	6.077
160	2	10	9.825	0.314	8.073	6.196
...	...	...	...	...	...	...
147	15	9	9.573	-0.936	7.788	5.976
146	16	7	9.332	-0.669	**7.471**	**5.747**
145	17	6	**9.125**	-0.285	7.544	5.812
...	...	...	...	...	...	...
127	35	15	9.874	-1.671	8.202	6.317
126	36	9	9.963	-1.027	8.258	6.359

Tabel 3. Kinerja prediksi 3 bulan

JS	LAG	K	RMSE	MFE	MAE	SMAPE
159	1	18	10.365	0.120	8.657	6.619
158	2	9	10.541	**0.038**	8.806	6.729
157	3	11	10.441	**-0.081**	8.734	6.670
...	...	...	...	...	...	...
143	17	6	**9.666**	-0.519	**7.919**	**6.089**
142	18	5	9.779	-0.602	8.004	6.155
...	...	...	...	...	...	...
124	36	8	10.314	-1.679	8.471	6.521

SMAPE untuk prediksi 3 bulan berikutnya ini konsisten dengan nilai MAE dan RMSE.

Untuk nilai MFE yang terdekat dengan 0 dengan menggunakan *lag* 2 atau 3, namun diperkirakan terjadi *overfitting*. Nilai RMSE, MAE, dan SMAPE berjarak jauh dari yang terendah. Nilai *k* terbaik berada agak jauh dari akar kuadrat jumlah sampel yang berbeda dengan [1], namun lebih bersifat trivial [11].

C. Pengaruh variasi nilai k dan lag untuk prediksi 6 bulan berikutnya

Prediksi 6 bulan berikutnya dilakukan dengan menggunakan variasi *k* dari 1 sampai 30 dan *lag* dari 1 sampai 36. Hasil yang diperoleh disajikan pada Tabel 4. Nilai SMAPE, MAE, dan RMSE tidak berbeda untuk nilai *lag* yang terbaik. Nilai MFE yang paling mendekati 0 memiliki nilai SMAPE, MAE, dan RMSE yang berbeda dari nilai optimal. Nilai *k* terbaik mendekati nilai akar kuadrat jumlah sampel sehingga tidak terlalu menyelisihi yang disebutkan dalam [1].

D. Pengaruh variasi nilai k dan lag untuk prediksi 9 bulan berikutnya

Prediksi 9 bulan berikutnya menggunakan variasi *lag* dari 1 sampai 36 dan *k* dipilih dari 1 sampai dengan 30. Hasil yang diperoleh disajikan dalam Tabel 5. Nilai RMSE, MAE, dan SMAPE optimal dicapai dengan menggunakan *lag* 14 dan *k* 13, sedangkan nilai MFE yang mendekati 0 dicapai dengan *lag* 1 dan *k* 20. Nilai *k* cukup dekat dengan akar kuadrat dari jumlah sampel sehingga lebih sesuai dengan yang disebutkan dalam [1].

Tabel 4. Kinerja prediksi 6 bulan

JS	LAG	K	RMSE	MFE	MAE	SMAPE
156	1	20	10.452	**0.014**	8.679	6.632
155	2	25	10.455	**0.066**	8.744	6.677
...	...	...	...	...	...	...
143	14	13	9.910	-1.288	8.130	6.235
142	15	9	**9.864**	-0.917	**8.116**	**6.234**
141	16	10	9.901	-1.240	8.123	6.244
...	...	...	...	...	...	...
122	35	5	10.676	-1.804	8.738	6.721
121	36	8	10.673	-2.198	8.804	6.768

Tabel 5. Kinerja prediksi 9 bulan

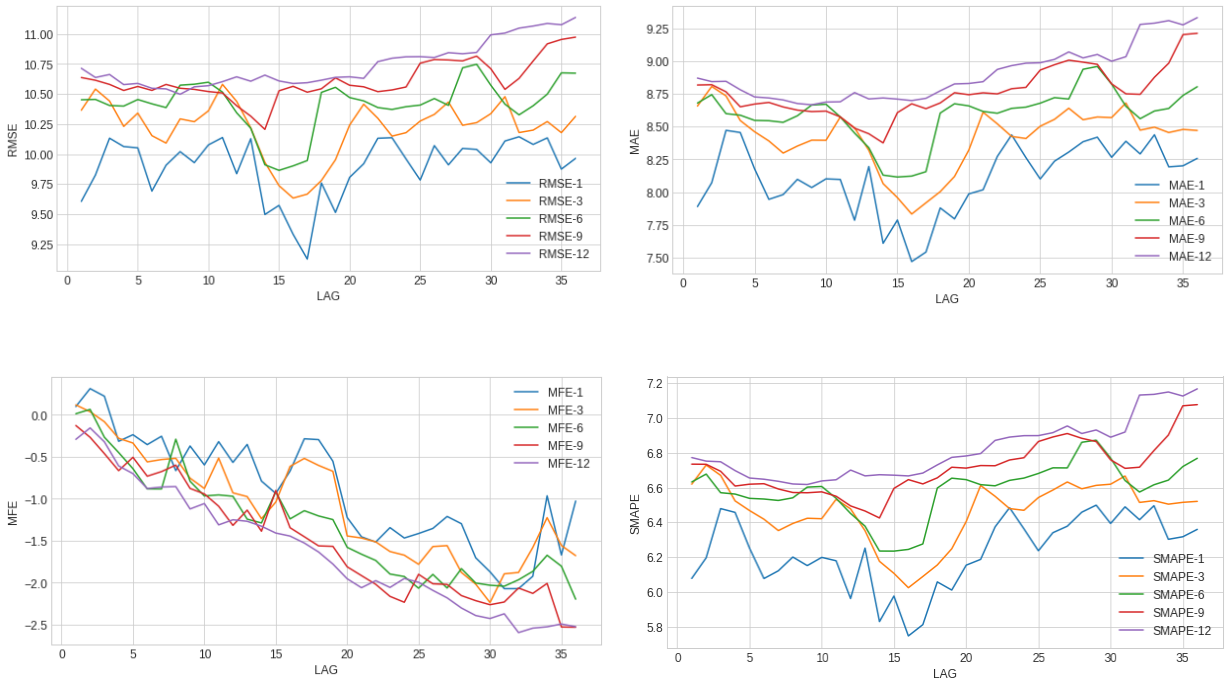
JS	LAG	K	RMSE	MFE	MAE	SMAPE
153	1	20	10.637	**-0.126**	8.817	6.734
152	2	30	10.614	-0.267	8.819	6.733
…	…	…	…	…	…	…
141	13	7	10.315	-1.135	8.446	6.465
140	14	13	**10.205**	-1.387	**8.376**	**6.425**
139	15	11	10.526	-0.901	8.606	6.595
…	…	…	…	…	…	…
119	35	30	10.954	-2.529	9.202	7.070
118	36	30	10.973	-2.532	9.212	7.076

Tabel 6. Kinerja prediksi 12 bulan

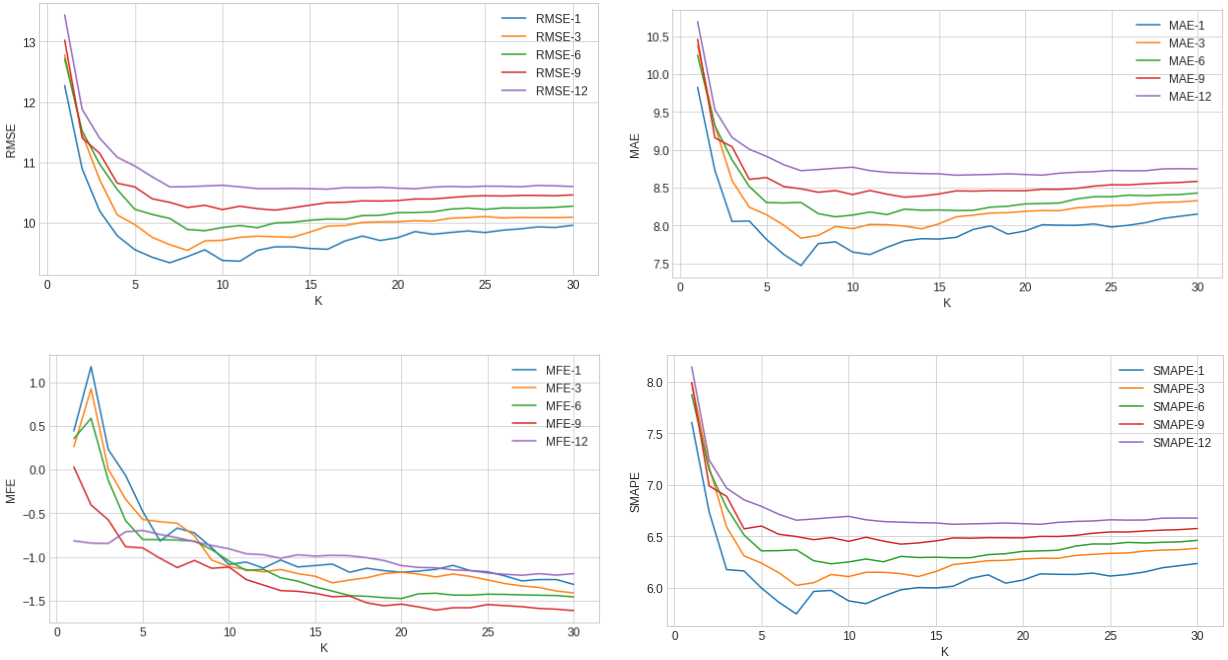
JS	LAG	K	RMSE	MFE	MAE	SMAPE
150	1	11	10.714	-0.291	8.870	6.772
149	2	29	10.637	**-0.155**	8.843	6.752
…	…	…	…	…	…	…
143	8	13	**10.497**	-0.855	8.676	6.620
142	9	21	10.559	-1.122	**8.666**	**6.617**
141	10	12	10.569	-1.057	8.687	6.638
…	…	…	…	…	…	…
116	35	30	11.076	-2.494	9.275	7.126
115	36	29	11.137	-2.526	9.331	7.167

E. Pengaruh variasi nilai k dan lag untuk prediksi 12 bulan berikutnya

Prediksi 12 bulan berikutnya menggunakan variasi *lag* dari 1 sampai 36 dan *k* dipilih dari 1 sampai dengan 30. Hasil yang diperoleh disajikan dalam Tabel 6 Hasil tersebut menunjukkan bahwa terdapat perbedaan nilai SMAPE, MAE dengan RMSE pada *lag* dan *k* yang menghasilkan nilai optimal. Nilai *lag*-nya berada di bawah 12 bulan sebelumnya. Hal ini menunjukkan bahwa data yang digunakan untuk memprediksi dengan tingkat SMAPE, MAE yang rendah membutuhkan data kurang dari 12 bulan sebelumnya. Hal ini berbeda dengan prediksi untuk 3, 6, dan 9 bulan berikutnya dengan nilai *lag* selalu lebih besar daripada jumlah target bulan prediksi. Nilai *k* bervariasi pada rentang yang cukup berjauhan dengan SMAPE dan MAE terendah pada nilai 21 dan RMSE terendah pada nilai 13. Ini cukup jauh dari nilai akar kuadrat dari jumlah sampel seperti dinyatakan dalam [1].



Gambar 8. Relasi lag dan kinerja (RMSE, MFE, MAE, SMAPE)



Gambar 9. Relasi k dan kinerja (RMSE, MFE, MAE, SMAPE)

F. Pengaruh variasi lag dan k terhadap kinerja

Pengaruh variasi *lag* terhadap kinerja prediksi dinyatakan dalam Gambar 8. Nilai *lag* yang optimal berada pada kisaran nilai 14–17 untuk prediksi 1 bulan, 3 bulan, 6 bulan, dan 9 bulan, sedangkan untuk prediksi 12 bulan berbeda, yaitu dengan *lag* 9. Hal ini terlihat dari nilai SMAPE, MAE, dan RMSE pada interval tersebut yang mencapai nilai minimum. Pada interval tersebut, bias (MFE) juga ada peningkatan untuk mendekati nilai 0. Nilai 0 di awal dengan *lag* kurang dari 5 tidak dipilih karena rawan terhadap kejadian *overfitting* [24].

Pengaruh nilai *k* terhadap kinerja prediksi dinyatakan dalam Gambar 9. Nilai parameter *k* terbaik berada pada interval 5–13 dengan jumlah sampel 140–150. Semakin besar nilai *k* di luar interval tersebut, maka semakin memperburuk kinerja prediksi dengan semakin tingginya

nilai kinerja, kecuali untuk prediksi 12 bulan yang berada pada nilai *k* 21. Nilai-nilai ini agak berbeda dengan kaidah yang menyatakan bahwa nilai *k* optimum berada pada nilai akar dari jumlah sampel seperti dinyatakan dalam [1], namun nilai akar dari jumlah sampel tetap bisa dijadikan sebagai acuan awal.

IV. KESIMPULAN

Nilai *lag* dan parameter *k* dalam penerapan algoritme kNN sangat menentukan untuk mendapatkan nilai prediksi dengan tingkat kesalahan yang rendah. Nilai lag yang digunakan untuk prediksi data tingkat hunian hotel berada pada interval 14-17 sehingga nilai *lag* untuk prediksi data sampai 12 bulan ke depan harus lebih dari 12 bulan sebelumnya. Nilai *k* terbaik tidak selalu berada pada sekitar nilai akar dari jumlah sampel seperti kaidah umum yang sering digunakan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Unisnu Jepara lewat Program Akselerasi Lektor dari LPPM yang memberikan bantuan dana penelitian ini dengan kontrak nomor 13/SP3R/LPPM/UNISNU/IV/2019.

DAFTAR PUSTAKA

[1] P. Nadkarni and P. Nadkarni, “Core technologies: data mining and big data,” in *Clinical Research Computing*, Academic Press, 2016, pp. 187–204. doi: 10.1016/B978-0-12-803130-8.00010-5

[2] D. A. Adeniyi, Z. Wei, and Y. Yongquan, “Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method,” *Applied Computing and Informatics*, vol. 12, no. 1, pp. 90–108, 2016. doi: 10.1016/j.aci.2014.10.001

[3] B. Santosa dan A. Umam, *Data Mining dan Big Data Analytics, edisi 1*. Yogyakarta: Penebar Media Pustaka, 2018, pp. 110–113.

[4] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, “A novel kNN algorithm with data-driven k parameter computation,” *Pattern Recognition Letters*, vol. 109, pp. 44–54, 2018. doi: 10.1016/j.patrec.2017.09.036

[5] R. Goyal, P. Chandra, and Y. Singh, "Suitability of kNN regression in the development of interaction based software fault prediction models," *IERI Procedia*, vol. 6, pp. 15-21, 2014. doi: 10.1016/j.ieri.2014.03.004

[6] J. F. Ajao, D. O. Olawuyi, and O. O. Odejobi, "Yoruba handwritten character recognition using Freeman chain code and k-nearest neighbor classifier," *Jurnal Teknologi dan Sistem Komputer*, vol. 6, no. 4, pp. 129-134, 2018. doi: 10.14710/jtsiskom.6.4.2018.129-134

[7] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148-163, 2018. doi: 10.1016/j.pmcj.2018.07.004

[8] A. Priadana and A. W. Murdiyanto, "Metode SURF dan FLANN untuk identifikasi nominal uang kertas Rupiah tahun emisi 2016 pada variasi rotasi," *Jurnal Teknologi dan Sistem Komputer*, vol. 7, no. 1, pp. 19-24, 2019. doi: 10.14710/jtsiskom.7.1.2019.19-24

[9] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying k-nearest neighbor to time series forecasting," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 2019-2037, 2019. doi: 10.1007/s10462-017-9593-z

[10] V. Nguyen Thanh Le, B. Apopei, and K. Alameh, "Effective plant discrimination based on the combination of local binary pattern operators and multiclass support vector machine methods," *Information Processing in Agriculture*, vol. 6, no. 1, pp. 116-131, 2019. doi: 10.1016/j.inpa.2018.08.002

[11] M. A. Mabayoje, A. O. Balogun, H. A. Jibril, J. O. Atoyebi, H. A. Mojeed, and V. E. Adeyemo, "Parameter tuning in kNN for software defect prediction: an empirical analysis," *Jurnal Teknologi dan Sistem Komputer*, vol. 7, no. 4, pp. 121-126, 2019. doi: 10.14710/jtsiskom.7.4.2019.121-126

[12] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3:43, pp. 1-19, 2017. doi: 10.1145/2990508

[13] Y. Song, J. Liang, J. Lu, and X. Zhao, "An efficient instance selection algorithm for k nearest neighbor regression," *Neurocomputing*, vol. 251, pp. 26-34, 2017. doi: 10.1016/j.neucom.2017.04.018

[14] S. Zhang, "Cost-sensitive KNN classification," *Neurocomputing*, vol. 391, pp. 234-242, 2019. doi: 10.1016/j.neucom.2018.11.101

[15] F. Martínez, M. P. Frías, M. D. Pérez-Godoy, and A. J. Rivera, "Dealing with seasonality by narrowing the training set in time series forecasting with kNN," *Expert Systems with Applications*, vol. 103, pp. 38-48, 2018. doi: 10.1016/j.eswa.2018.03.005

[16] B. E. Flores, "A pragmatic view of accuracy measurement in forecasting," *Omega*, vol. 14, no. 2, pp. 93-98, 1986. doi: 10.1016/0305-0483(86)90013-7

[17] J. S. Armstrong, *Long-range Forecasting: from crystal ball to computer, 2ed*. Wiley, 1985. doi: 10.1016/0169-2070(86)90059-2

[18] Y. Cai, H. Huang, H. Cai, and Y. Qi, "A K-nearest neighbor locally search regression algorithm for short-term traffic flow forecasting," in *9th International Conference on Modelling, Identification and Control*, Kunming, China, Jul. 2017, pp. 624-629. doi: 10.1109/ICMIC.2017.8321530

[19] S. P. Mahasagara, A. Alamsyah, and B. Rikumahu, "Indonesia infrastructure and consumer stock portfolio prediction using artificial neural network backpropagation," in *5th International Conference on Information and Communication Technology*

(ICoIC7), Malacca City, Malaysia, May 2017, pp. 1-4. doi: 10.1109/ICoICT.2017.8074710
[20] J. Demšar et al., "Orange: data mining toolbox in Python," *Journal of Machine Learning Research*, vol. 14, pp. 2349-2353, 2013.
[21] F. Pedregosa et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
[22] JetBrains, "PyCharm: the Python IDE for professional developers by JetBrains," 2017. [online]. Available: https://www.jetbrains.com/pycharm/
[23] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007. doi: 10.1109/MCSE.2007.55
[24] P. Mehta et al., "A high-bias, low-variance introduction to machine learning for physicists," *Physics Reports*, vol. 810, pp. 1-124, 2019. doi: 10.1016/j.physrep.2019.03.001