

# Peramalan Harga Bahan Pangan Harian dengan Model Berbasis Deep Learning

Rifqi Hafizuddin  
Departemen Matematika, FMIPA  
Universitas Indonesia  
Depok, Indonesia  
rifqi.hafizuddin@sci.ui.ac.id

Favian Sulthan Wafi  
Departemen Matematika, FMIPA  
Universitas Indonesia  
Depok, Indonesia  
favian.sulthan@sci.ui.ac.id

Aurelio Naufal Effendy  
Departemen Matematika, FMIPA  
Universitas Indonesia  
Depok, Indonesia  
aurelio.naufal@sci.ui.ac.id

**Abstract**—Ketidakstabilan harga bahan pangan merupakan permasalahan krusial yang dapat berdampak luas terhadap perekonomian dan kesejahteraan masyarakat. Fluktuasi harga bahan pokok dipengaruhi oleh berbagai faktor, seperti kondisi cuaca, kebijakan perdagangan, hari libur nasional, harga komoditas global, fluktuasi nilai tukar, dan perubahan pola konsumsi, sehingga diperlukan metode peramalan yang akurat untuk mengantisipasi perubahan harga dan mendukung pengambilan keputusan yang lebih baik. Dalam beberapa tahun terakhir, pendekatan model berbasis deep learning telah mencapai state-of-the-art (SOTA) pada peramalan deret waktu. Oleh karena itu, penelitian ini mengaplikasikan model peramalan berbasis deep learning dengan menggunakan PatchTST dan N-HiTS sebagai model utama, serta TimesNet, iTransformer, dan RMoK sebagai model *baseline*, dengan normalisasi RevIN. Model-model ini digunakan untuk melakukan prediksi 13 harga bahan pangan harian di 34 provinsi Indonesia dari tanggal 1 Oktober 2024 sampai dengan 31 Desember 2024. Berdasarkan hasil evaluasi metrik MAPE dengan metode evaluasi *rolling forecasting*, model PatchTST (0.0755) dan N-HiTS (0.0807) mengungguli seluruh model *baseline*. Prediksi harga daging sapi murni menjadi bahan pangan dengan metrik evaluasi terbaik, sedangkan prediksi harga bawang merah menjadi bahan pangan dengan metrik evaluasi terburuk. Menggunakan teknik *best fit*, prediksi akhir diambil dari hasil evaluasi terbaik antara model PatchTST dan N-HiTS dengan nilai MAPE sebesar 0.0694 pada data validasi.

**Keywords**—Forecasting, Bahan Pangan, RevIN, PatchTST, N-HiTS

## I. PENDAHULUAN

Pangan merupakan kebutuhan paling mendasar bagi manusia di setiap negara, termasuk Indonesia [1]. Setiap tahun, harga bahan pangan mengalami fluktuasi yang dipengaruhi oleh berbagai faktor, seperti ketersediaan stok, permintaan, alur dan biaya distribusi, serta tingkat keberhasilan produksi bahan pangan [2]. Menurut Peraturan Presiden (Perpres) No. 59 Tahun 2020 tentang Penetapan dan Penyimpanan Barang Kebutuhan Pokok dan Barang Penting, bahan pangan yang termasuk dalam kategori bahan pokok (Bapak) meliputi beras, kedelai, cabai, bawang merah, gula, minyak goreng, tepung terigu, daging sapi, daging ayam ras, telur ayam ras, dan ikan segar. Ketidakstabilan harga bahan pokok ini dapat berdampak luas terhadap berbagai aspek, mulai dari kebijakan ekonomi, rantai pasokan, hingga daya beli masyarakat. Jika tidak terkontrol, lonjakan harga dapat memicu inflasi dan mengganggu perekonomian nasional. Oleh karena itu, peramalan harga bahan pangan yang akurat menjadi hal yang krusial dalam mendukung stabilitas ekonomi dan ketahanan pangan. Pemerintah perlu menerapkan berbagai kebijakan yang didukung oleh peramalan harga berbasis data guna mengantisipasi fluktuasi harga dan memastikan keterjangkauan bahan pokok bagi masyarakat.

Namun, peramalan harga bahan pangan mempunyai berbagai tantangan, baik dari faktor internal maupun eksternal. Faktor-faktor seperti kondisi cuaca, kebijakan perdagangan, hari libur nasional, harga komoditas global, fluktuasi nilai tukar, dan perubahan pola konsumsi dapat menyebabkan pergerakan harga yang tidak selalu linear dan sulit diprediksi. Dalam hal ini, data historis yang berbentuk deret waktu berperan penting dalam memahami dinamika harga pangan serta membangun model prediksi yang dapat mengantisipasi perubahan harga di masa mendatang.

Deret waktu merupakan rangkaian titik data yang tersusun berdasarkan urutan waktu diskrit [3]. Berbeda dengan data gambar atau teks yang memiliki pola sintaksis atau intuitif yang jelas, informasi semantik dari data deret waktu berasal dari ketergantungan temporal dan korelasi antar variabel yang kompleks. Dengan menangkap ketergantungan ini, model deret waktu dapat secara efektif mengungkap dinamika yang mendasari objektif dari model, yaitu tugas peramalan [3].

Metode deret waktu tradisional, seperti *Autoregressive Integrated Moving Average* (ARIMA) telah lama digunakan dalam melakukan peramalan deret waktu [3]. Sebagai contoh, Muhardi *et al.* [4] telah menerapkan ARIMA untuk memprediksi produksi dan harga beras serta jagung di Sulawesi Tengah. Namun, ARIMA memiliki keterbatasan, seperti hanya dapat digunakan untuk satu variabel (univariat) saja dan mengasumsikan bahwa data bersifat linier dan stasioner, sehingga kurang mampu menangkap dinamika laten dalam data [5].

Dalam beberapa tahun terakhir, pendekatan peramalan berbasis *deep learning* telah mencapai *state-of-the-art* (SOTA) pada berbagai *dataset* [5][6]. Model-model seperti TimesNet [7] dan *Temporal Convolutional Network* (TCN) [8] yang berbasis *Convolutional Neural Network* (CNN), *Reversible Mixture of KAN Experts* (RMoK) [9] yang berbasis *Kolmogorov-Arnold Network* (KAN), serta xLSTMTIME [10], yang merupakan varian *Recurrent Neural Network* (RNN) yang dioptimalkan untuk peramalan deret waktu, telah menunjukkan performa unggul dalam menangkap pola kompleks dan nonlinear dalam data.

Meskipun peramalan berbasis *deep learning* telah mengalami kemajuan pesat, peramalan jangka panjang tetap menjadi tantangan besar. Dua tantangan utama yang dihadapi adalah volatilitas prediksi dan kompleksitas komputasi. Untuk mengatasi masalah ini, Challu *et al.* [11] memperkenalkan N-HiTS, sebuah model berbasis *Multi-Layer Perceptrons* (MLP) yang menggabungkan interpolasi hierarkis dan teknik pengambilan sampel data *multi-rate*. Pendekatan ini memungkinkan peramalan deret waktu jangka panjang yang lebih akurat, dapat diinterpretasi, serta efisien.

Selain model berbasis CNN, RNN, dan MLP, pendekatan *Transformer* juga telah digunakan secara luas dalam peramalan deret waktu. Model-model seperti Informer [12],

Autoformer [13], FEDformer [14], telah menunjukkan kemampuan dalam menangkap pola kompleks dan nonlinear, menghasilkan prediksi yang lebih akurat dibandingkan metode tradisional [5][6]. Namun, penelitian yang dilakukan oleh Zeng *et al.*, [15] menunjukkan bahwa model linier sederhana dapat mengungguli semua model sebelumnya pada berbagai *benchmark*, mempertanyakan efektivitasnya dalam peramalan. Model-model tersebut berfokus pada optimalisasi mekanisme *attention*, sehingga mencapai kinerja yang lebih baik dalam peramalan jangka waktu panjang. Namun, sebagian besar model menggunakan *attention* pada setiap titik, yang mengabaikan pentingnya informasi semantik lokal.

Pendekatan *patching* telah terbukti efektif dalam menangkap informasi semantik lokal, terutama dalam bidang *computer vision*. Model seperti *Vision Transformer* (ViT) [16] membagi gambar menjadi *patch* berukuran  $16 \times 16$  sebelum memasukkannya ke dalam *Transformer*. Metode serupa juga diterapkan dalam model BeiT [17] dan Masked Autoencoder [18], yang menggunakan *patch* sebagai token input.

PatchTST [19] mengadopsi konsep tersebut dalam peramalan deret waktu dengan membagi data menjadi *patch* tingkat subseri yang berfungsi sebagai token input untuk *Transformer*. Pendekatan ini tidak hanya memungkinkan model menangkap informasi semantik lokal dengan lebih baik, tetapi juga meningkatkan efisiensi komputasi dan penggunaan memori. Selain itu, PatchTST menerapkan *channel-independence*, di mana setiap variabel diperlakukan sebagai deret waktu univariat yang diproses secara independen. Pendekatan ini terbukti mampu meningkatkan akurasi peramalan jangka panjang dibandingkan model *Transformer* konvensional. Hal ini juga didukung oleh penelitian Wang *et al.* [5], yang menunjukkan bahwa PatchTST unggul dalam peramalan baik untuk jangka pendek maupun jangka panjang dibandingkan dengan model-model *Transformer* terbaik. Dengan strategi pembagian data menjadi *patch* serta pemrosesan independen tiap variabel, PatchTST mampu secara efektif menangkap pola kompleks dalam deret waktu, menjadikannya salah satu model yang kompetitif dalam tugas peramalan.

Oleh karena itu, untuk mengatasi tantangan dalam peramalan harga komoditas pangan, penelitian ini mengusulkan pendekatan berbasis deep learning dengan mengimplementasikan PatchTST dan N-HiTS menggunakan NeuralForecast. Kedua model ini dipilih karena kemampuannya dalam menangkap pola kompleks dalam data deret waktu serta keunggulannya dalam peramalan jangka panjang. Sebelum pelatihan model, data deret waktu akan dinormalisasi menggunakan *Reversible Instance Normalization* (RevIN) untuk mengatasi perbedaan skala antar waktu dan meningkatkan stabilitas prediksi. Selanjutnya, hasil prediksi dari PatchTST dan N-HiTS akan dikombinasikan menggunakan teknik *best fit*. Hasil peramalan akhir untuk setiap deret waktu akan diperoleh dari model dengan performa terbaik berdasarkan hasil validasi.

Pendekatan ini diharapkan dapat meningkatkan akurasi dan stabilitas prediksi harga pangan, sehingga dapat memberikan wawasan yang lebih baik bagi pembuat kebijakan, pelaku industri, dan masyarakat dalam mengantisipasi fluktuasi harga serta mendukung ketahanan pangan nasional.

## II. METODE ANALISIS

### A. Data Preparation

Data yang digunakan dalam penelitian ini terdiri dari 13 *dataset* harga bahan pangan harian di 34 provinsi Indonesia dari tanggal 1 Januari 2022 sampai dengan 31 Desember 2024 yang diperoleh dari Badan Pangan Nasional (Bapanas). Setiap *dataset* terbagi menjadi *training set*, dari tanggal 1 Januari 2022 sampai dengan 30 September 2024, dan *testing set* dari tanggal 1 Oktober 2024 sampai dengan 31 Desember 2024 yang akan diprediksi.

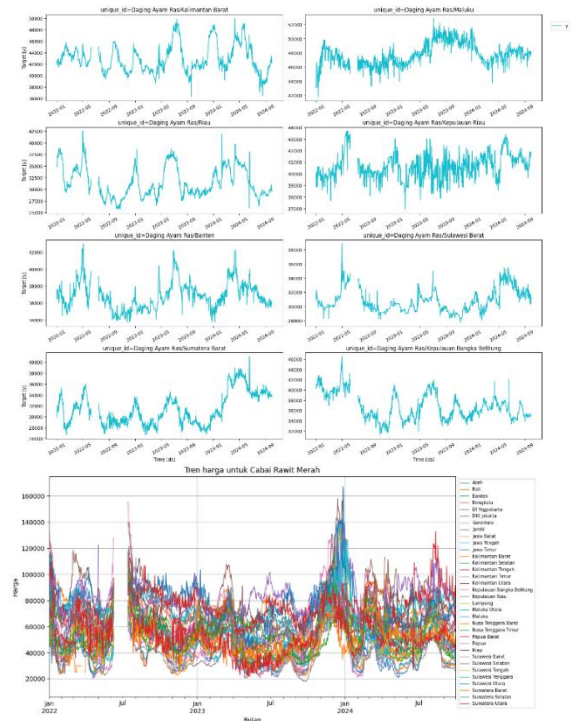
Untuk mempermudah proses analisis data *time-series* ini, seluruh *dataset* digabungkan menjadi satu ke dalam bentuk *long format*, seperti data panel, sehingga setiap baris merepresentasikan harga harian suatu bahan pangan dari suatu provinsi. Cuplikan dari *dataset* dapat dilihat pada Tabel 1.

TABEL 1. CUPLIKAN *DATASET* SETELAH PENGGABUNGAN

ds	unique_id	y
2022-01-01	Aceh/Bawang Merah	28,970
2022-01-02	Aceh/Bawang Merah	29,900
2022-01-03	Aceh/Bawang Merah	28,970

### B. Exploratory Data Analysis (EDA)

Sebelum melakukan pemodelan, akan dilakukan EDA untuk memahami karakteristik dan komponen penting dari *dataset* terlebih dahulu. EDA dilakukan dengan membuat visualisasi harga harian pangan per provinsi dan harga harian pangan seluruh provinsi, yang terlihat pada Gambar 1, untuk mengamati pola dan tren bulanan maupun tahunan yang ada pada setiap bahan pangan dan provinsi.



Gambar 1. Beberapa *Line Plot* Harga Harian Pangan per Provinsi.

Dari hasil *line plot*, ditemukan bahwa terdapat harga bahan pangan dengan missing values di beberapa tanggal. Beberapa *missing values* tersebut terlihat sangat jelas pada tanggal 10 Juni 2022 sampai dengan 13 Juli 2022 untuk setiap bahan

pangan. Kemudian, ditemukan juga pola dari beberapa bahan pangan, seperti daging sapi murni, bawang merah, cabai merah keriting, dan cabai rawit merah yang memiliki tren naik di bulan-bulan tertentu. Hal ini kemungkinan dipengaruhi oleh musim panen, perayaan hari besar, dan faktor musiman lainnya.

### C. Preprocessing

Mengetahui hasil dari EDA, perlu dilakukan *preprocessing* terlebih dahulu pada *dataset* untuk memastikan data yang digunakan “bersih” sehingga dapat memberikan hasil yang optimal. Hal pertama yang perlu diperhatikan adalah *missing values* di bulan Juni-Juli 2022. Data yang hilang terlalu banyak sehingga melakukan imputasi dikhawatirkan dapat mengubah distribusi aktual dari data *time-series*. Oleh karena itu, data dari bulan Juli 2022 dan sebelumnya tidak diikutsertakan dalam analisis. Untuk *missing values* lainnya, untuk setiap pangan, dilakukan imputasi menggunakan rata-rata harga harian (yang tidak *missing*) di tanggal yang sama.

Selain itu, melihat adanya pola tertentu dari setiap bahan pangan, kovariat-kovariat masa depan, “is\_holiday” (tanggal merupakan hari libur nasional), “prev\_holiday” (tanggal sebelumnya adalah hari libur nasional), dan “next\_holiday” (tanggal setelahnya adalah hari libur nasional), juga ditambahkan untuk membedakan hari-hari perayaan khusus di Indonesia, seperti Idul Adha, dan Idul Fitri yang berpotensi menaikkan harga bahan pangan nasional [20]. Data hari libur nasional diambil dari laman resmi Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan [21]. Cuplikan dari dataset akhir dapat dilihat pada Tabel 2.

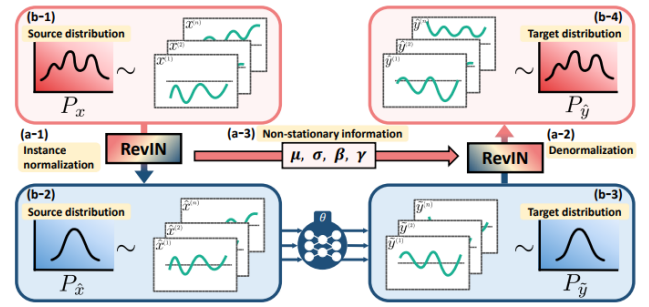
TABEL II. CUPLIKAN DATASET AKHIR

ds	unique_id	is_holiday	prev_holiday	next_holiday	y
2022-01-01	Aceh/Bawang Merah	1	0	0	28,970
2022-01-02	Aceh/Bawang Merah	0	1	0	29,900
2022-01-03	Aceh/Bawang Merah	0	0	0	28,970

### D. Reversible Instance Normalisation (RevIN)

Salah satu permasalahan utama model dalam melakukan *forecasting* adalah adanya *distribution shift*, yaitu perubahan distribusi data, termasuk rata-rata dan variansi, seiring waktu. *Distribution shift* perlu diatasi sebab *training* dan *testing set* dalam *forecasting* diambil dari rentang waktu yang berbeda sehingga distribusi antarkeduanya menjadi tidak sama dan dapat mengurangi performa model dalam memprediksi data masa depan.

RevIN, diperkenalkan oleh Kim et al. [22], menjadi salah satu solusi dari permasalahan ini dengan melakukan normalisasi serta denormalisasi data yang dapat meningkatkan performa model *forecasting*.



Gambar 2. Arsitektur RevIN [22]

Pada tahap normalisasi, informasi nonstasioner dari setiap *instance* (data satu waktu) pada *dataset* akan dihilangkan untuk meningkatkan similaritas distribusi antar *training* dan *testing set*. Kemudian, model melakukan *forecast* dari hasil *training* tersebut. Lalu, hasil *forecast* di-denormalisasi kembali (mengembalikan informasi yang dihilangkan) agar hasil prediksi tetap sesuai dengan distribusi awal data. Proses secara lengkap terlihat pada Gambar 2.

Setiap *instance*  $x_i$  dari total  $n$  *instance*, dinormalisasi menjadi  $\hat{x}_i$  dengan mengaplikasikan rumus berikut:

$$\hat{x}_i = \gamma \cdot \left( \frac{x_i - \mu}{\sigma} \right) + \beta, \forall i \in \{1, 2, \dots, n\} \quad (1)$$

Kemudian, setiap prediksi akhir  $y_i$  didenormalisasi kembali dengan rumus berikut:

$$\hat{y}_i = \sigma \cdot \left( \frac{y_i - \beta}{\gamma} \right) + \mu, \forall i \in \{1, 2, \dots, n\} \quad (2)$$

Di mana  $\sigma$  dan  $\mu$  adalah standar deviasi dan mean dari *training set*.  $\beta$  dan  $\gamma$  adalah *affine learnable parameter*, yakni parameter yang dapat dioptimalkan selama *training* model untuk menangkap *distribution shift* pada data dengan lebih baik.

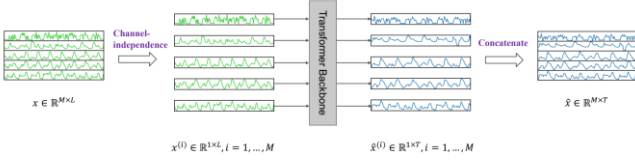
### E. Patch Time Series Transformer (PatchTST)

PatchTST merupakan model berbasis Transformer yang efisien untuk peramalan deret waktu multivariat dan *self-supervised representation learning*. Model ini didasarkan pada dua komponen utama: *patching* dan *channel-independence* [19].

Dalam peramalan deret waktu, memahami korelasi antar data pada berbagai langkah waktu merupakan aspek krusial. Namun, satu titik waktu tidak memiliki makna semantik seperti kata dalam kalimat, sehingga mengekstraksi informasi semantik lokal sangat penting dalam menganalisis koneksinya. PatchTST mengatasi keterbatasan ini dengan meningkatkan lokalitas serta menangkap informasi semantik yang lebih kaya melalui teknik *patching*, yaitu menggabungkan beberapa langkah waktu menjadi unit *patch* pada tingkat sub-deret. Pendekatan ini memungkinkan model untuk memahami pola jangka pendek dengan lebih baik dibandingkan pendekatan berbasis titik tunggal.

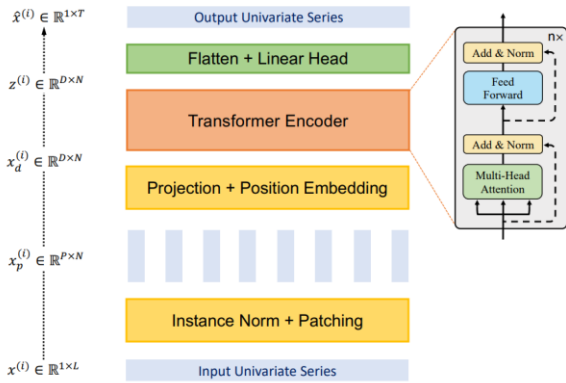
Selain itu, dalam deret waktu multivariat, setiap variabel dianggap sebagai sinyal *multi-channel*, di mana satu token masukan dalam *Transformer* dapat direpresentasikan oleh data dari satu atau beberapa *channel*. PatchTST menerapkan *channel-independence*, di mana setiap *channel* berisi satu

deret waktu univariat yang berbagi bobot input dan Transformer yang sama di semua deret. Metode tersebut terbukti bekerja dengan baik dengan CNN [23] dan model linier [15], tetapi belum diterapkan pada model berbasis *Transformer*.



Gambar 3. Arsitektur PatchTST [19]

Arsitektur model PatchTST diilustrasikan pada Gambar 3, di mana data deret waktu multivariat dipisahkan ke dalam beberapa channel. Setiap channel diproses secara independen menggunakan backbone *Transformer* yang sama, namun mekanisme *forward pass* tetap bersifat independen antar channel.

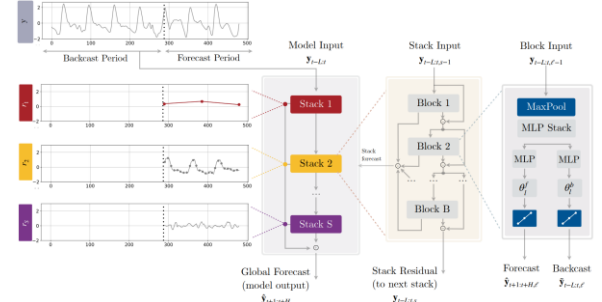


Gambar 4. Proses dalam setiap channel PatchTST [19]

Gambar 4 mengilustrasikan proses yang terjadi pada setiap *channel* deret univariat. Pada tahap awal, setiap deret waktu univariat terlebih dahulu dinormalisasi menggunakan operator *instance normalization* untuk mengurangi perbedaan skala antar variabel. Selanjutnya, deret  $x^{(i)}$  yang telah dinormalisasi menjadi beberapa segmen (*patch*). Dengan *patch length*  $P$  dan *stride*  $S$ , proses *patching* akan menghasilkan urutan patch  $x_p^{(i)} \in \mathbb{R}^{P \times N}$ , di mana  $N$  adalah jumlah *patch*,  $N = \frac{L-P}{S}$ . Dengan penggunaan *patch*, jumlah token input dapat berkurang dari  $L$  menjadi  $L/S$  yang mengakibatkan penggunaan memori dan kompleksitas komputasional dari *attention map* berkurang secara kuadrat dengan faktor  $S$ . *Patch-patch* ini kemudian diteruskan ke encoding *Transformer* melalui proyeksi linier dan *position encoding* yang dapat dipelajari. Terakhir, hasilnya dilewatkan melalui *flatten layer* dan lapisan linier untuk menghasilkan prediksi akhir.

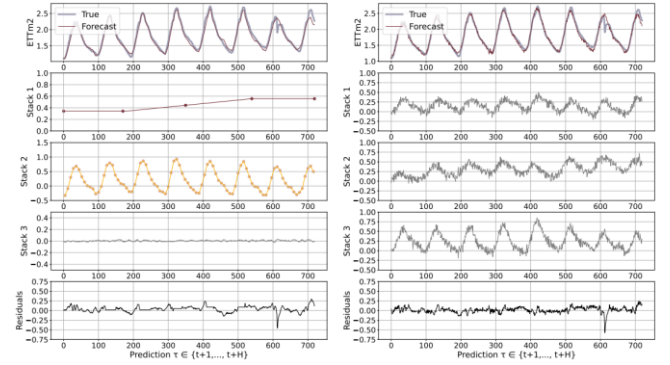
#### F. Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS)

N-HiTS merupakan model deret waktu berbasis MLP yang mengembangkan arsitektur N-BEATS dengan memperkenalkan *multi-rate sampling* untuk menguraikan sinyal input ke dalam berbagai frekuensi serta *hierarchical interpolation* guna meningkatkan *smoothness* prediksi [11]. Arsitektur N-HiTS terdiri dari  $S$  *stack*, di mana setiap *stack* terdiri dari  $B$  blok. Setiap blok berisi MLP yang bertugas memprediksi koefisien basis *forward* dan *backward*, yang diilustrasikan oleh Gambar 5.



Gambar 5. Arsitektur N-HiTS [11]

N-HiTS menggabungkan lapisan *sub-sampling* sebelum blok *fully connected* sehingga resolusi input untuk setiap blok berbeda. Hal tersebut dilakukan supaya setiap blok melihat pola yang terjadi pada resolusi waktu yang berbeda, seperti harian, mingguan, atau bulanan. Pendekatan ini meningkatkan ketepatan peramalan sekaligus mengurangi kebutuhan memori dan komputasi, karena model tidak perlu menganalisis seluruh jendela *lookback* dengan panjang  $H$ , tetapi hanya segmen yang lebih kecil seperti  $H/2$ ,  $H/4$ , dan seterusnya. Hal ini memungkinkan model menangkap pola multi-skala secara efisien tanpa mengorbankan kualitas prediksi.



Gambar 6. Perbandingan hasil peramalan menggunakan N-HiTS dengan hierarchical interpolation dan multi-rate sampling (kiri), dengan tanpa hierarchical interpolation dan multi-rate sampling (kanan). [11]

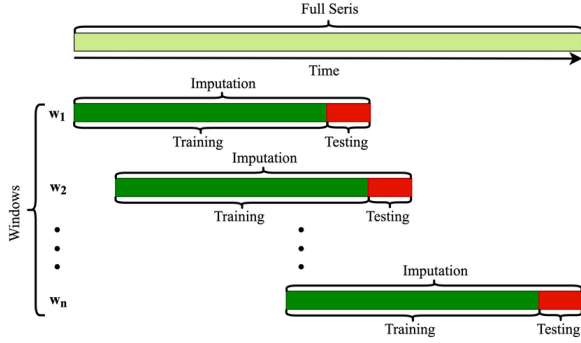
Dalam *multi-step forecasting*, model harus memprediksi  $H$  langkah ke depan. Seiring bertambahnya  $H$ , kebutuhan komputasi meningkat secara signifikan, menyebabkan ledakan *expressive power* yang dibutuhkan model. Melatih model dengan *expressive power* tinggi tanpa mengalami *overfitting* menjadi tantangan tersendiri. Untuk mengatasi masalah ini, N-HiTS memperkenalkan *temporal interpolation*, sebuah pendekatan yang mengendalikan dimensi output dengan menyesuaikan rasio ekspresif ( $\tau$ ).

*Hierarchical interpolation* diterapkan dengan mendistribusikan rasio ekspresif secara hierarkis di antara blok-blok model, yang disinkronkan dengan mekanisme *multi-rate sampling*. Blok yang lebih dekat ke input memiliki nilai  $\tau_1$  lebih kecil dan tingkat *sub-sampling*  $k_1$  lebih besar, sehingga memaksa model untuk menghasilkan sinyal granularitas rendah yang lebih halus dan melihat pola dengan resolusi lebih luas. Pendekatan ini menghasilkan prediksi akhir  $\hat{y}_{t+1:t+H}$  yang merupakan kombinasi interpolasi dari berbagai tingkat hierarki waktu. Gambar 6 mengilustrasikan perbedaan peramalan menggunakan N-HiTS secara penuh dengan N-HiTS tanpa *hierarchical interpolation* dan *multi-rate sampling*.



### G. Model Evaluation

Strategi evaluasi model yang digunakan dalam penelitian ini adalah *cross-validation* dalam deret waktu dengan metode *rolling forecasting*. Dalam pendekatan ini, data historis diperluas secara bertahap dengan langkah tetap (*stride*) di setiap iterasi, kecuali pada iterasi terakhir di mana data historis telah mencakup seluruh deret waktu. Model peramalan kemudian diterapkan pada setiap iterasi untuk menghasilkan prediksi pada *forecasting horizon* kedepan yang telah ditentukan, dan rata-rata metrik evaluasinya dihitung dari semua sampel (*windows*) pada setiap deret waktu.



Gambar 7. Rolling Forecasting [25]

Metrik evaluasi model yang digunakan adalah *Mean Absolute Percentage Error* (MAPE), yang mengukur rata-rata relatif dari error prediksi. MAPE dirumuskan sebagai berikut.

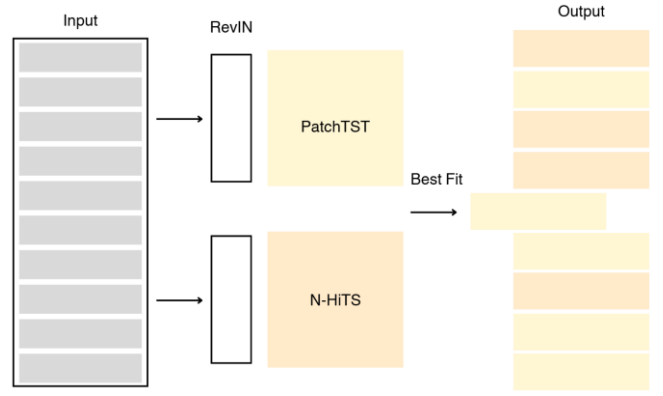
$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - x_i}{x_i} \right| \quad (3)$$

dengan  $y_i$  adalah nilai prediksi,  $x_i$  adalah nilai aktual dan  $n$  adalah jumlah data. Semakin kecil nilai MAPE, semakin akurat model dalam melakukan prediksi.

### H. Model Implementation

Penelitian ini menggunakan model PatchTST dan N-HiTS karena kemampuannya dalam menangkap pola kompleks dalam data deret waktu serta statusnya sebagai metode *state-of-the-art* dalam peramalan jangka panjang. Implementasi model dilakukan menggunakan NeuralForecast [24], sebuah *library* Python yang dirancang untuk menjalankan model peramalan berbasis neural secara efisien dan mudah digunakan.

Sebelum data dimasukkan ke dalam model, *Reversible Instance Normalization* (RevIN) diterapkan untuk menormalisasi data. RevIN digunakan untuk mengatasi perbedaan skala antar waktu, sehingga meningkatkan stabilitas prediksi dan performa model. Selanjutnya, model PatchTST dan N-HiTS dilatih dan divalidasi pada dataset train. Model terbaik untuk setiap deret waktu dipilih berdasarkan hasil evaluasi performa selama validasi. Hasil peramalan akhir untuk setiap deret waktu akan diperoleh dari model dengan performa terbaik tersebut.



Gambar 8. Implementasi model peramalan harga bahan pangan

*Hyperparameter* yang digunakan dalam penelitian ini dipilih berdasarkan implementasi resmi PatchTST dan N-HiTS [11][19], serta hasil public score pada Kaggle. Pada model N-HiTS, variabel *exogenous* yang digunakan adalah hari libur nasional. Rincian lengkap *hyperparameter* yang digunakan disajikan dalam Tabel 3 dan 4.

TABLE III. HYPERPARAMETER PATCHTST

Hyperparameter	Nilai	Penjelasan
h	92	<i>forecast horizon</i>
input_size	180	jumlah titik data historis sebagai input (6 bulan)
patch_len	16	panjang patch
stride	8	jarak pergeseran antar patch saat membagi data deret waktu
revin	True	teknik normalisasi
hidden_size	128	lapisan tersembunyi <i>transformer</i>
n_heads	4	jumlah <i>head</i> dalam <i>transformer</i>
scaler_type	'robust'	jenis normalisasi
learning_rate	1e-3	Tingkat pembaruan bobot

TABLE IV. HYPERPARAMETER N-HITS

Hyperparameter	Nilai	Penjelasan
h	92	<i>forecast horizon</i>
input_size	180	jumlah titik data historis sebagai input (6 bulan)
futr_exog_list	['is_holiday', 'prev_day_holiday', 'next_day_holiday']	variabel <i>exogenous</i> untuk periode mendatang
scaler_type	'robust'	jenis normalisasi
learning_rate	1e-3	tingkat pembaruan bobot
n_blocks:list	[1,1,1]	jumlah blok arsitektur N-HiTS
mlp_units:list	[[512, 512], [512, 512], [512, 512]]	jumlah unit neuron dalam MLP

## III. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan model utama N-HiTS dan PatchTST serta model *baseline* TimesNet, iTransformer, RmoK dengan normalisasi RevIN. Setiap model dievaluasi

menggunakan teknik *rolling forecasting* dengan 3 *windows* (dimulai dari Mei 2023) yang masing-masing memprediksi nilai 92 hari ke depan (*forecasting horizon*) dengan *stride* 30 hari.

Dari hasil *rolling forecasting*, untuk setiap ‘unique\_id’ (pangan per daerah), nilai MAPE dari setiap *windows* dirata-ratakan untuk mendapatkan nilai MAPE akhir yang digunakan untuk pemilihan model terbaik. Setiap ‘unique\_id’ akan memiliki 1 model dengan MAPE terendah (performa terbaik). Menggunakan teknik *best fit*, model dengan performa terbaik akan dipilih sebagai model akhir yang digunakan untuk memprediksi ‘unique\_id’ tersebut.

TABEL V. HASIL EVALUASI TEKNIK *ROLLING FORECASTING*

	Times Net	iTrans former	RMoK	N-HITS	Patch TST
MAPE	0.0964	0.0995	0.0959	0.0807	0.0755

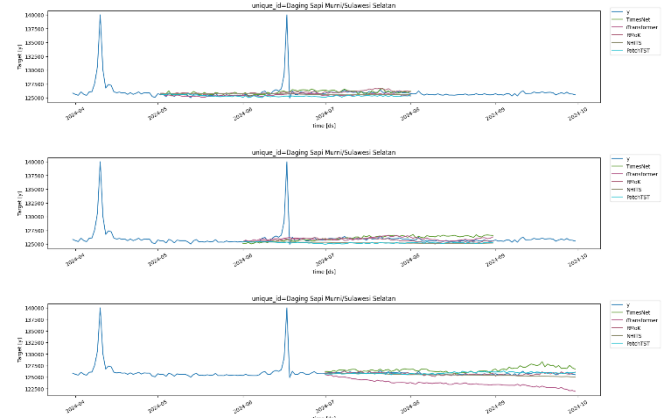
Hasil evaluasi seluruh model terlihat pada Tabel 5. Model utama N-HITS dan PatchTST terlihat jauh mengungguli seluruh model *baseline*. Oleh sebab itu, untuk memaksimalkan kembali hasil prediksi akhir, akan dipilih PatchTST dan N-HITS sebagai model akhir yang akan digunakan untuk *forecast testing set*.

#### A. Analisis Hasil Terbaik

TABEL VI. CUPLIKAN HASIL EVALUASI TEKNIK *ROLLING FORECASTING* TERBAIK

unique_id	Times Net	iTrans former	RMoK	N-HITS	Patch TST	best_model
Daging Sapi Murni/ Sulawesi Selatan	0.0053	0.0081	0.0028	0.0042	0.0046	RMoK
Daging Sapi Murni/ Sumatera Barat	0.0055	0.0069	0.0047	0.0047	0.0036	PatchTST
Daging Sapi Murni/ Jawa Timur	0.0093	0.0070	0.0078	0.0038	0.0040	N-HITS
Daging Sapi Murni/ Nusa Tenggara Barat	0.0066	0.0038	0.0055	0.0055	0.0043	iTransf ormer
Daging Sapi Murni/ Jambi	0.0094	0.0093	0.0085	0.0045	0.0038	PatchTST

Berdasarkan hasil evaluasi pada Tabel 6, daging sapi murni menjadi bahan pangan dengan hasil prediksi dari seluruh model, terutama model utama, yang paling menyerupai distribusi asli *training set*. Melihat visualisasi hasil prediksi pada Gambar 9, hal ini dipengaruhi oleh pola harga daging sapi murni yang cukup stabil seiring waktu.



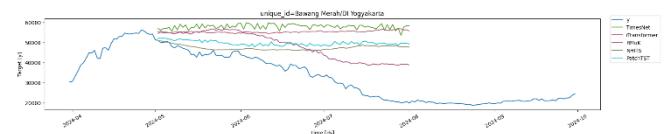
Gambar 9. Cuplikan Hasil *Rolling Forecasting* dengan MAPE terbaik

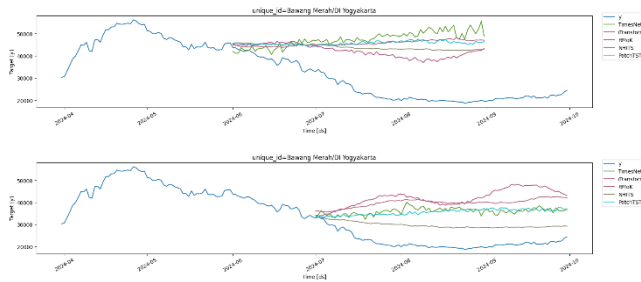
#### B. Analisis Hasil Terburuk

TABEL VII. CUPLIKAN HASIL EVALUASI TEKNIK *ROLLING FORECASTING* TERBURUK

unique_id	Times Net	iTrans former	RMoK	N-HITS	Patch TST	best_model
Bawang Merah/ Nusa Tenggara Timur	0.5437	0.5585	0.5332	0.5002	0.5092	N-HITS
Bawang Merah/ DI Yogyakarta	0.7568	0.6493	0.7601	0.4925	0.6368	N-HITS
Bawang Merah/ Lampung	0.7387	0.6197	0.7643	0.4850	0.5958	N-HITS
Bawang Merah/ Bengkulu	0.578056	0.562350	0.606076	0.467653	0.493912	N-HITS
Bawang Merah/ Aceh	0.576415	0.516405	0.601587	0.456554	0.483304	N-HITS

Berdasarkan hasil evaluasi pada Tabel 7, bawang merah menjadi bahan pangan dengan hasil prediksi dari seluruh model yang paling jauh dari distribusi asli *training set*. Melihat visualisasi hasil prediksi pada Gambar 10, hal ini dipengaruhi oleh pola harga bawang merah yang sangat fluktuatif seiring waktu.





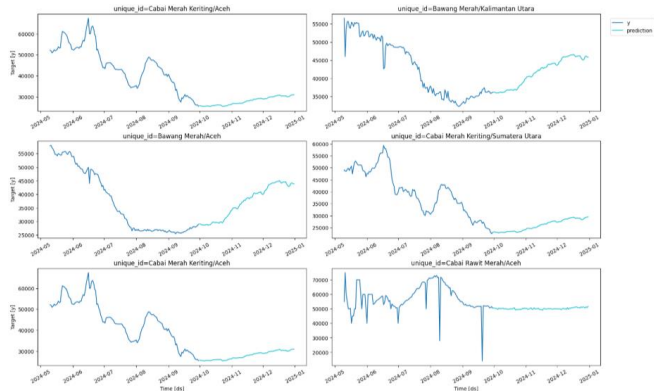
Gambar 10. Cuplikan Hasil *Rolling Forecasting* dengan MAPE terburuk

### C. Analisis Prediksi Akhir

TABEL VIII. CUPLIKAN HASIL EVALUASI TEKNIK *ROLLING FORECASTING* MODEL GABUNGAN

	N-HITS	PatchTST	N-HITS+PatchTST
MAPE	0.0807	0.0755	0.0694

Hasil evaluasi dari model akhir PatchTST dan N-HITS terlihat pada Tabel 8. Dengan menggunakan teknik *best fit*, model akhir yang digunakan untuk memprediksi 'unique\_id' adalah model dengan MAPE terkecil, yaitu model gabungan N-HITS+PatchTST. Dengan begitu, hasil prediksi akhir akan menjadi optimal karena dapat semakin mendekati distribusi aktual dari *testing set*. Visualisasi hasil prediksi akhir terlihat pada Gambar 11.



Gambar 11. Cuplikan Hasil Prediksi Akhir

## IV. KESIMPULAN

Berdasarkan hasil validasi *rolling forecast* harga bahan pangan menggunakan lima model peramalan, PatchTST dan N-HITS menunjukkan performa terbaik dengan nilai MAPE sebesar 0.0755 dan 0.0807, mengungguli model TimesNet, iTransformer, dan RMoK. Oleh karena itu, prediksi dari kedua model dikombinasikan menggunakan metode *best fit*, yang berhasil menurunkan MAPE menjadi 0.0694, serta meningkatkan *public score* pada Kaggle dibandingkan dengan penggunaan model PatchTST atau N-HITS secara terpisah.

Hasil prediksi menunjukkan bahwa model mampu memprediksi harga daging sapi murni dengan baik, diduga karena harga yang relatif stabil. Sebaliknya, prediksi harga bawang merah mengalami penyimpangan yang cukup besar dari distribusi data historis. Visualisasi hasil menunjukkan bahwa kedua model gagal menangkap fluktuasi harga yang tinggi pada komoditas tersebut, mengindikasikan perlunya

pendekatan tambahan untuk meningkatkan akurasi pada bahan pangan dengan volatilitas tinggi.

Untuk penelitian selanjutnya, akurasi peramalan dapat lebih ditingkatkan dengan mengeksplorasi variabel *exogenous* lainnya seperti variabel cuaca, pola musiman, serta data terkait kebijakan perdagangan yang dapat mempengaruhi harga bahan pangan sehingga model dapat menangkap fluktuasi harga bahan pangan. Selain itu, dalam penanganan *missing value*, pendekatan interpolasi berbasis model seperti *deep learning* imputation dapat diuji untuk memastikan data hasil imputasi seragam dengan distribusi awal data. Terakhir, pendekatan *ensemble learning* yang lebih kompleks atau metode adaptasi model terhadap volatilitas tinggi juga dapat dipertimbangkan untuk meningkatkan akurasi prediksi pada komoditas dengan fluktuasi harga yang tajam.

## REFERENCES

- [1] Y. Purwaningsih, "KETAHANAN PANGAN: SITUASI, PERMASALAHAN, KEBIJAKAN, DAN PEMBERDAYAAN MASYARAKAT," J. Ekon. Pembang. Kaji. Masal. Ekon. dan Pembang., vol. 9, no. 1, p. 1, Jun. 2008, doi: 10.23917/JEP.V9I1.1028.
- [2] M. F. F. Mardianto, Sediono, I. Syahzaqi, S. A. D. Safitri, and N. Afifah, "Prediction of Indonesia Strategic Commodity Prices during the COVID-19 Pandemic based on a Simultaneous Comparison of Kernel and Fourier Series Estimator," J. Southwest Jiaotong Univ., vol. 55, no. 6, 2020, doi: 10.35741/issn.0258-2724.55.6.43..
- [3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.
- [4] Muhardi, D. Evansyah, A. Made, and Effendy, "Forecasting Model of Production and Price of Grains Commodity in Central Sulawesi," Int. J. Econ. Stat., vol. 9, pp. 47–51, 2021, doi: 10.46300/9103.2021.9.8.
- [5] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, dan J. Wang, "Deep Time Series Models: A Comprehensive Survey and Benchmark," 18 Juli 2024, arXiv: arXiv:2407.13278. doi: 10.48550/arXiv.2407.13278.
- [6] X. Qiu dkk., "TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods," Proc. VLDB Endow., vol. 17, no. 9, hlm. 2363–2377, Mei 2024, doi: 10.14778/3665844.3665863.
- [7] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis," Apr. 12, 2023, arXiv: arXiv:2210.02186. doi: 10.48550/arXiv.2210.02186.
- [8] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," Aug. 29, 2016, arXiv: arXiv:1608.08242. doi: 10.48550/arXiv.1608.08242.
- [9] X. Han, X. Zhang, Y. Wu, Z. Zhang, and Z. Wu, "Are KANs Effective for Multivariate Time Series Forecasting?," Feb. 11, 2025, arXiv: arXiv:2408.11306. doi: 10.48550/arXiv.2408.11306.
- [10] M. Alharthi and A. Mahmood, "xLSTMTIME : Long-term Time Series Forecasting With xLSTM," Aug. 12, 2024, arXiv: arXiv:2407.10240. doi: 10.48550/arXiv.2407.10240.
- [11] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler-Cansco, and A. Dubrawski, "N-HITS: Neural Hierarchical Interpolation for Time Series Forecasting," Nov. 29, 2022, arXiv: arXiv:2201.12886. doi: 10.48550/arXiv.2201.12886.
- [12] H. Zhou et al., "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," Mar. 28, 2021, arXiv: arXiv:2012.07436. doi: 10.48550/arXiv.2012.07436.
- [13] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," Jan. 07, 2022, arXiv: arXiv:2106.13008. doi: 10.48550/arXiv.2106.13008.
- [14] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," Jun. 16, 2022, arXiv: arXiv:2201.12740. doi: 10.48550/arXiv.2201.12740.
- [15] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," Aug. 17, 2022, arXiv: arXiv:2205.13504. doi: 10.48550/arXiv.2205.13504.

- [16] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Jun. 03, 2021, *arXiv*: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.
- [17] H. Bao, L. Dong, S. Piao, and F. Wei, “BEiT: BERT Pre-Training of Image Transformers,” Sep. 03, 2022, *arXiv*: arXiv:2106.08254. doi: 10.48550/arXiv.2106.08254.
- [18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” Dec. 19, 2021, *arXiv*: arXiv:2111.06377. doi: 10.48550/arXiv.2111.06377.
- [19] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A Time Series is Worth 64 Words: Long-term Forecasting with Transformers,” Mar. 05, 2023, arXiv: arXiv:2211.14730. doi: 10.48550/arXiv.2211.14730.
- [20] Suselo, Sri Liani, et al. “Pengaruh Hari Besar Pada Komoditas Utama Inflasi Di Indonesia.” Working Papers, 2013.
- [21] “Beranda | Kementerian Koordinator Bidang Pembangunan Manusia dan Kebudayaan.” Accessed: Mar. 15, 2025. [Online]. Available: <https://kemenkopmk.go.id/>
- [22] Kim, Taesung, et al. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. 2021. openreview.net, <https://openreview.net/forum?id=cGDAkQo1C0p>.
- [23] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks,” in Web-Age Information Management, F. Li, G. Li, S. Hwang, B. Yao, and Z. Zhang, Eds., Cham: Springer International Publishing, 2014, pp. 298–310. doi: 10.1007/978-3-319-08010-9\_33.
- [24] K. G. Olivares, C. Challú, A. Garza, M. M. Canseco, and A. Dubrawski, “NeuralForecast: User friendly state-of-the-art neural forecasting models,” PyCon, Salt Lake City, Utah, USA, 2022. [Online]. Available: <https://github.com/Nixtla/neuralforecast>
- [25] A. Uddin, X. Tao, C.-C. Chou, and D. Yu, “Nonlinear Tensor Completion Using Domain Knowledge: An Application in Analysts’ Earnings Forecast,” in 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy: IEEE, Nov. 2020, pp. 377–384. doi: 10.1109/ICDMW51313.2020.00059.