

# Real-time Deep Learning Algoritmes Voor Optimale Verkeersdoorstroming Bij Brugopening

Daniël R.L. Overdeest<sup>1,2</sup>, Michael R. de Winter<sup>1</sup>, Dennis van Muijen<sup>1,3</sup> en Joana F.M.F. Cardoso<sup>1\*</sup>

## Samenvatting

De provincie Zuid-Holland heeft een belangrijke rol in de verkeersdoorstroming binnen de regio door het bedienen en onderhouden van meer dan 100 bruggen. Slimme ICT-oplossingen bieden nieuwe kansen om via sensoren en algoritmes haarfijn actuele verkeersinformatie te koppelen aan live scheepvaart informatie. In het project 'Impactmonitor Brugopeningen' is hierin een wezenlijke stap gezet in het voorspellen van verkeersintensiteiten rondom een aantal belangrijke bruggen in de provincie Zuid-Holland. De gebruikte Long Short-Term Memory neurale netwerken kunnen met een correlatiecoëfficiënt nauwkeurigheid van 75-95% voorspellingen doen tot 21 minuten vooruit. De ontwikkelde algoritmes zijn uitgerold binnen een Azure cloud omgeving en opgeleverd als gebruiksvriendelijke webapplicatie waarin live scheepvaart en verkeersinformatie gepresenteerd wordt. Deze informatie wordt door de applicatie gebruikt om het optimale moment voor een brugopening te adviseren aan een brugbediener. In een volgende fase kan ook deze informatie direct teruggevoerd worden aan het verkeersnetwerk waarmee bijvoorbeeld navigatie software rekening kan gaan houden met geplande brugopeningen. Samenvattend, de ontwikkelde webapplicatie gebruikt recente technologie op het gebied van kunstmatige intelligentie waarmee brugbedieners vakkennis kunnen combineren met de laatste technologie.

## Keywords

Brug monitoringsysteem (BMS), Verkeersdoorstroming Voorspellen, Deep learning, Long Short-Term Memory (LSTM), Verkeerscongestiemodel, Data science, Prognosemodel, Real-time, API

<sup>1</sup> Provincie Zuid-Holland, Programma Transparante Open Provincie (TOP) - Opgave Vernieuwing Datawarehouse, Afdeling I & A, Zuid-Hollandplein 1, 2509 LP Den Haag

<sup>2</sup> DO IT analytics: daniel@doit-analytics.nl

<sup>3</sup> Affine Solutions: d@affine.nl

\* Correspondentie: j.cardoso@pzh.nl

## Inhoudsopgave

|   |          |  |          |
|---|----------|--|----------|
| <b>1 Inleiding</b>                                      | <b>1</b> | <b>5.4 Performance</b>                 | <b>6</b> |
| <b>2 Dataset voorbereidingen</b>                        | <b>2</b> | <b>6 Conclusie en vervolgonderzoek</b> | <b>7</b> |
| 2.1 Gebruikte datasets                                  | 2        | <b>Acknowledgments</b>                 | <b>7</b> |
| 2.2 Datakwaliteit en opschoonacties                     | 2        | <b>Referenties</b>                     | <b>7</b> |
| 2.3 Gebruikte tools & technieken                        | 3        |  |          |
| <b>3 Modeleren en voorspellen verkeersintensiteiten</b> | <b>3</b> |  |          |
| 3.1 LSTM neuraal netwerken                              | 3        |  |          |
| 3.2 Geselecteerde modellen                              | 3        |  |          |
| 3.3 Parameter optimalisatie                             | 4        |  |          |
| <b>4 Model evaluatie</b>                                | <b>4</b> |  |          |
| 4.1 Baseline model                                      | 4        |  |          |
| 4.2 Resultaten  | 4        |  |          |
| <b>5 Deployment</b>                                     | <b>5</b> |  |          |
| 5.1 Productiewaardig maken                              | 5        |  |          |
| 5.2 Real-time databronnen                               | 5        |  |          |
| 5.3 Prognose API  | 5        |  |          |

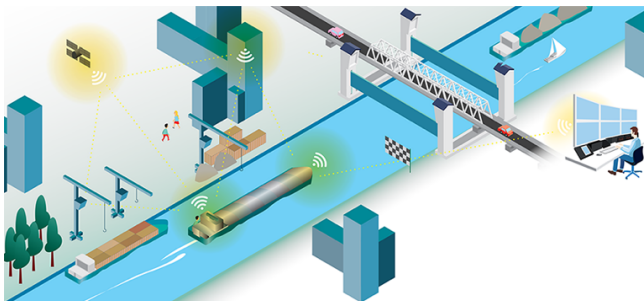
## 1. Inleiding

In de waterrijke en dichtbevolkte provincie Zuid-Holland is een vloeiende doorstroming van het verkeer over water en land van essentieel belang. De provincie Zuid-Holland heeft hierin een belangrijke rol als verantwoordelijke voor het bedienen en onderhouden van meer dan 100 bruggen. Van deze bruggen zijn er 55 die sensoren hebben waarmee gemonitord wordt of de brug open of dicht is. Voor een steeds groter wordend aantal van deze bruggen staat de provincie voor een uitdaging om de verkeersstroom optimaal af te stemmen met de scheepvaart en een toenemende verkeersdruk. De provincie zet zich in voor een slimmere, veiligere en efficiëntere doorvaart om deze uitdagingen te verwezenlijken. Dit doet zij samen met haar partners in de regio – o.a. wegbeheerders binnen Bereik! [1] – het rijk zoals Blauwe Golf Verbindend [2] en het Smart

Shipping programma vanuit Rijkswaterstaat [3].

Bij de Provincie is er in projectvorm, samen met Dienst Beheer Infrastructuur (DBI) en Afdeling Informatisering & Automatisering (I&A), gewerkt aan een innovatieve ICT-oplossing: de Impactmonitor Brugopeningen. Brugbedieners worden in deze applicatie van real-time informatie voorzien om het meest efficiënte openingsmoment voor een brug te bepalen. Hierbij wordt gekeken naar de impact van een brugopening op doorstroming en uitstoot. Goederenvervoer over water staat hoog op de agenda van de provincie als duurzame vervoers-modaliteit. Tegelijkertijd veroorzaakt een brugopening overlast op de wegen; het zorgt voor file en wachtrijen voor de brug. De Impactmonitor Brugopeningen is gebouwd om brugbedieners te ondersteunen in het kiezen van het optimale moment voor een brugopening. De applicatie is uitgerust om de verkeersdrukte tot 21 minuten vooruit te voorspellen in combinatie met actuele scheepsverkeer informatie. Er zijn op dit moment drie bruggen opgenomen in de Impactmonitor Brugopeningen: de Coenecoopbrug (Waddinxveen), de Lammebroeg (Leiden) en de Kruithuisbrug (Delft). De Impactmonitor Brugopeningen is een beslissings-ondersteunende webapplicatie die helemaal is aangepast op het gedrag en werkproces van brugbedieners. Uiteindelijk biedt de applicatie naast real-time informatie ook adviezen voor het beste openings-moment waarmee zo min mogelijk hinder wordt veroorzaakt voor het wegverkeer met de laagste CO2 voetafdruk. De applicatie wordt momenteel met een selecte groep getest waarna deze na verdere evaluatie gefaseerd zal worden ingevoerd in de bedienentrale.

Op technisch vlak komen in de Impactmonitor Brugopeningen moderne ontwikkelingen op het gebied van big data, kunstmatige intelligentie en applicatieontwikkeling bijeen (figuur 1). Van het uitlezen en verwerken van real-time big data in de vorm van verkeersdata uit meetlussen, het verwerken van deze data door neurale netwerken, een prognose over de verwachte verkeersdrukte tot het uitlezen van diverse API's. Dit alles samengebracht in een moderne webapplicatie. In deze paper beschrijven wij zowel de benodigde technische aspecten om de data te verwerken, de gebruikte gereedschappen en aanpak op het gebied van machine learning en de uiteindelijke presentatie-vorm in een gebruiksvriendelijke userinterface.



**Figuur 1.** Ideale situatie waar de schepen, wegverkeer en brugbediening met elkaar in verbinding staan via slimme sensoren

## 2. Dataset voorbereidingen

### 2.1 Gebruikte datasets

**HIG meetlusgegevens** In de provinciale wegen liggen meetlussen die de doorstroom van het verkeer in aantal voertuigen, soort voertuig en snelheid per voertuig registreert. Slechts een subset van deze lussen wordt real-time ontsloten aan het NDW (Nationale Databank Wegverkeersgegevens) – de rest wordt gebruikt voor monitoring en evaluatie. Omdat niet alle voor het project relevante meetlussen beschikbaar waren in het NDW is bij de leverancier een ruwe dataset opgevraagd voor de drie in de inleiding genoemde bruggen. Hierin stonden de meest gedetailleerde data van alle voertuigen die over een meetlus rijden, waarbij elke regel in de dataset één voertuig is. Voor het verdere gebruik van deze data zijn de individuele regels geaggregeerd per minuut met de som van aantal voertuigen, gemiddelde snelheid en gemiddelde lengte en aantallen per voertuigclasses. Hierbij is tussen de één en twee jaar aan historische data gebruikt voor het maken van het voorspelmodel afhankelijk van beschikbaarheid per meetlus.

**NDW real-time wegverkeersintensiteiten** NDW heeft actuele verkeersgegevens van het aantal voertuigen dat per minuut een meetpunt passeert. Deze real-time data wordt gebruikt voor de uiteindelijke real-time voorspelling.

**Centrale Object Bediening Analyse Log Data (COBALD) brugopening data** COBALD is een voor de Provincie ontwikkeld systeem dat gekoppeld staat aan de bedienentrale. Alle historische brugopeningen met daarin het moment (start datum/tijd) en de duur van de brugopening staan hierin. Deze historische data is gebruikt voor het prepareren en analyseren van de meetlusgegevens.

**Bridge Sense real-time data** Op de meeste bruggen zijn extra sensoren geplaatst die real-time brugopeningen detecteert en doorstuurt aan de Blauwe Golf Verbindend en het NDW – die het beschikbaar stellen als Open Data. Deze real-time data wordt gebruikt voor de actuele brugstatus in de ontwikkelde webapplicatie.

### 2.2 Datakwaliteit en opschoonacties

Sensordata is vaak gevoelig voor meetfouten waardoor er een aantal data kwaliteitscontroles en opschoonacties benodigd waren. De opschoonacties zijn gebaseerd op beschrijvende statistiek en uitbijters. De volgende opschoonregels zijn toegepast:

- Meetlus: Voertuig lengte langer dan 18.7 meter zijn verwijderd.
- Meetlus: Voertuig snelheid groter dan 120 km/u zijn verwijderd.
- Brugopening: Brugopeningen korter dan 3 minuten en langer dan 14 minuten zijn verwijderd.

Voor het model willen we de te verwachten verkeersdrukte weten. Gebeurtenissen op de weg en wegopbrekingen (zoals een brugopening) geven een vertekenend beeld van de

te verwachte intensiteit op een wegdeel. Ten tijde van een brugopening geeft de meetlus veelal een intensiteit van 0 voertuigen per minuut aan. Hiertoe is besloten om de verkeersintensiteit ten tijde van de werkelijke brugopening en een periode van 5 minuten ervoor en erna te corrigeren door de laatst bekende intensiteit door te trekken (forward fill).

### 2.3 Gebruikte tools & technieken

**Verkeersintensiteitenmodel ontwikkeling** Deep learning modellen zijn ontwikkeld in de Keras library (versie 2.2.4), een open-source neural network pakket geschreven in Python [4]. Keras stelt onderzoekers in staat om snel te experimenteren met deep neural networks [5]. Keras was gebruikt in samenhang met TensorFlow (versie 1.5.0) als backend [6] waarbij in de experimentele fase de GPU variant van het gekozen model (sectie 3) is gebruikt. Een grid-search was toegepast ontleend aan de Scikit-learn machine learning library [7] voor Python. Alle analyses voor het opschonen van data en het modelleren van verkeersintensiteiten zijn uitgevoerd in Python.

**Model evaluatie** Tijdens het trainen en selecteren van de modellen zijn verschillende foutstatistieken gebruikt om de nauwkeurigheid van de modellen te vergelijken op data uit tijdsreeksen waar het model geen gegevens van heeft gezien tijdens het trainen. Hiermee kan het model getoetst worden aan een situatie die vergelijkbaar is in de praktijk. Een aantal foutstatistieken kunnen gebruikt worden om de nauwkeurigheid van het resulterende regressie model. In het huidige werk zijn de volgende foutstatistieken gebruikt:

- R-kwadraat ( $R^2$ ): - Het kwadraat van de correlatie - coëfficiënt tussen geobserveerde waarden en voorspelde waarden door het model. Hoe dichterbij 1, hoe beter het model.
- Gemiddelde gekwadreeerde fout, Mean Squared Error (MSE) - Het gemiddelde verschil tussen geobserveerde en voorspelde waarden in het kwadraat. Hoe lager de MSE, hoe nauwkeuriger het model.
- Gemiddeld absolute percentage fout, Mean Absolute Percentage Error (MAPE) - Het gemiddelde absoluut percentage afwijking tussen observaties en voorspelde waarden. Des te kleiner MAPE, des te nauwkeuriger het model.

Om de nauwkeurigheid van de getrainde modellen te vergelijken is het belangrijk deze te testen op data die het model nog niet eerder heeft gezien. Machine learning en zeker deep learning modellen zijn dermate complex dat deze zeer vatbaar zijn voor overfitting, d.w.z. niet patronen zijn geleerd, maar het algoritme heeft de dataset leren kennen. Kruisvalidatie is een techniek om de nauwkeurigheid en mate van overfitting te kwantificeren voor machine learning modellen. In deze dataset is hiervoor gebruikt gemaakt door een model te trainen op historische data uit bijvoorbeeld 2017 en deze te testen op data uit 2018. Alle boven geschreven foutstatistieken zijn gebruikt voor modelevaluatie.

## 3. Modeleren en voorspellen verkeersintensiteiten

### 3.1 LSTM neurale netwerken

Artificiële Neuronale Netwerken (ANN) zijn een set aan algoritmes geïnspireerd door de enorm efficiënte werking van het menselijk brein om patronen te herkennen [8][9][10]. Een ANN bestaat uit artificiële neuronen en kan in recente jaren tot in vergaande diepte en complexiteit efficiënt getraind worden door zogenoemde deep learning algoritmes [11]. Voor tijdseries data zoals in deze casus zijn Recurrent (Terugkerend) Neuronale Netwerken (RNN) vaak een geschikte oplossing omdat hierbij neuronen niet alleen informatie uit vorige lagen van het netwerk meewegen, maar ook informatie uit vorige tijdstippen meenemen [12]. Een krachtige variant op RNN's is het Long Short-Term Memory (LSTM) netwerk die speciaal is ontwikkeld om lange tijdsreeksen te modelleren [13][14][15]. In het hier gepresenteerde werk zijn LSTM gekozen als uiteindelijk algoritme om real-time verkeersintensiteiten te voorspellen voor drie bruggen.

### 3.2 Geselecteerde modellen

Een univariaat LSTM model volgens Brownlee [16] in Keras waarbij meerdere stappen vooruit voorspeld worden was getraind voor elke meetlus. Twee verschillende LSTM modellen zijn getest per meetlus en vergeleken op de foutstatistieken beschreven in sectie 2. Als eerste is een LSTM model getraind op historische verkeersdata waarbij een tijdsreeks aan input gebruikt is om een vector aan stappen vooruit te voorspellen (script 1).

In het tweede geval is een encoder-decoder LSTM gebruikt (script 2) waarbij de invoer tijdsreeks aan data gecodeerd wordt (encode). De decoder gebruikt deze gecodeerde data en maakt een voorspelling een stap vooruit voor elk element in de output reeks [16]. Het verschil is klein, aangezien in de praktijk beide modellen een reeks vooruit voorspellen. Het belangrijkste verschil is dat een LSTM model encoder-decoder model informatie heeft over zowel de voorgaande stap die voorspeld is als ook de geaccumuleerde informatie uit de encoder stap wordt meegewogen in het voorspellen van de output sequentie.

**Script 1.** Python code voor trainen LSTM model met vector output

```
model = Sequential()
model.add(LSTM(n_layers, activation=activation,
               kernel_initializer=kernel_initializer,
               input_shape=(n_steps_in, features)))
model.add(Dense(n_steps_out))

optimizer = optimizer(lr=lr, beta_1=beta_1,
                     beta_2=beta_2,
                     decay=decay, epsilon=epsilon)
model.compile(optimizer=optimizer, loss=loss,
              metrics=['accuracy', 'mae'])
```

Voor zowel het vector output model en het encoder-decoder model zijn de verkeersintensiteit gegevens over de afgelopen

21 minuten gebruikt als input (*n\_steps\_in*) voor de aankomende 21 minuten (*n\_steps\_out*). Aangezien hier gebruikt wordt gemaakt van een univariaat model is het aantal features per definitie 1.

### Script 2. Python code voor trainen van Encoder-decoder LSTM model

```
model = Sequential()
model.add(LSTM(n_layers, activation=activation,
               kernel_initializer=kernel_initializer,
               input_shape=(n_steps_in, features)))
model.add(RepeatVector(n_steps_out))
model.add(LSTM(n_layers, activation=activation,
               return_sequences=True))
model.add(TimeDistributed(Dense(dense2,
                                activation=activation2,
                                kernel_initializer=kernel_initializer)))
model.add(TimeDistributed(Dense(dense)))

optimizer = optimizer(lr=lr,
                      beta_1=beta_1, beta_2=beta_2,
                      decay=decay, psilon=epsilon)
model.compile(optimizer=optimizer, loss=loss,
              metrics=['accuracy', 'mae'])
```

### 3.3 Parameter optimalisatie

Hyperparameter optimalisatie is een belangrijke stap in ontwikkelen van een deep learning model. Neurale netwerken zijn complex en lastig te configureren door de vele parameters die verfijnd en getest kunnen worden. Deze techniek is beschikbaar binnen de GridSearchCV klasse [13], zie sectie 2.

**Tabel 1.** Gekozen hyperparameters

| Parameter          | Value          |
|--------------------|----------------|
| epochs             | 30             |
| batch size         | 40             |
| epochs             | 30             |
| batch size         | 40             |
| n_layers           | 85             |
| optimizer          | Adam           |
| learning_rate (lr) | 0.01           |
| beta_1             | 0.9            |
| beta_2             | 0.9            |
| decay              | 0              |
| epsilon            | 0.7            |
| activation         | sigmoid        |
| activation2        | hard_sigmoid   |
| kernel_initializer | glorot_uniform |
| loss               | mse            |
| dense1             | 1              |
| dense2             | 1024           |

De hyperparameter optimalisatie was uitgevoerd op de Coenecoopbrug data (tabel 1). De geselecteerde hyperparameters zijn vervolgens ook gebruikt voor de overige bruggen waarvoor een neural netwerk model is ontwikkeld (Kruithuisbrug, Lammebrug). Het getrainde model is geëxporteerd als een h5 bestand (figuur 5).

## 4. Model evaluatie

### 4.1 Baseline model

Omdat de nauwkeurigheid van een machine learning model relatief is, is het essentieel om een robuust relatief simpel referentie te hebben als nulmeting. Vanuit deze referentie kunnen opeenvolgende modellen vergeleken worden. Als referentiemodel is een mediaan verkeersintensiteit model gebruikt waarbij per minuut per weekdag de mediaan is genomen als waarde.

Naast het mediaan model is er geëxperimenteerd met ARIMA (Autoregressive Integrated Moving Average) en gradient boosting. Hoewel beide methodes veelbelovende resultaten lieten zien, waren deze niet goed in staat om meerde stappen vooruit de voorspellen. Omdat in de eindapplicatie nauwkeurigheid in 1 minuut vooruit voorspellen even belangrijk is als 21 minuten vooruit is gekozen om de focus te verleggen naar LSTM modellen. Alle in sectie 2 genoemde foutstatistiek ( $R^2$ , MSE, MAE and MAPE) zijn gebruikt om de LSTM modellen af te vergelijken met het mediaan model.

### 4.2 Resultaten

Vector output en encoder-decoder LSTM modellen zijn getraind voor elke meetlus per rijrichting. Voorspelde waardes zijn vergeleken met daadwerkelijk observaties uit de test set. Hierbij presteerden de LSTM modellen over de gehele linie aanzienlijk beter dan het mediaan model, gemiddeld 20% beter op basis van de gekwadrateerde correlatiecoëfficiënt (tabel 2).

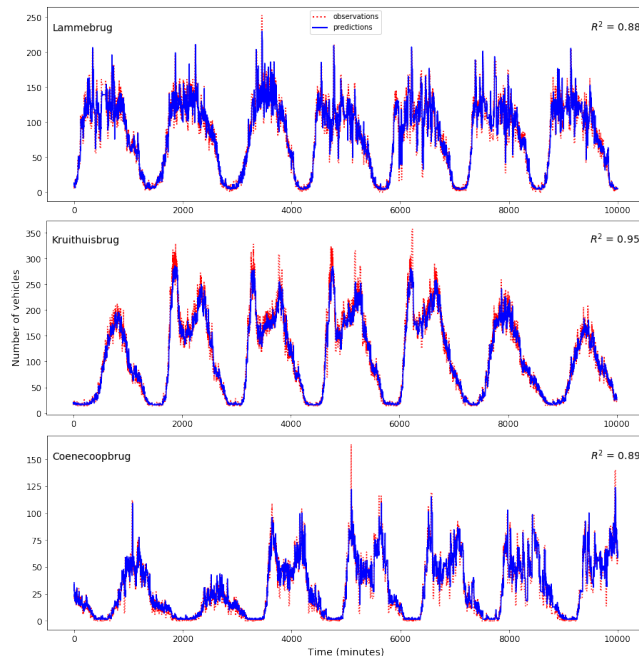
**Tabel 2.** Foutstatistieken voor de verschillende modellen

| Brug          | Meetlus | Rijbaan | Model         | MSE   | MAE  | MAPE  | R2   |
|---------------|---------|---------|---------------|-------|------|-------|------|
| Lammebrug     | 7       | 1       | mediaan       | 39.92 | 4.36 | 39.79 | 0.66 |
| Lammebrug     | 7       | 1       | vector output | 20.93 | 3.31 | 31.08 | 0.83 |
| Lammebrug     | 7       | 1       | encode-decode | 20.37 | 3.28 | 32.95 | 0.83 |
| Lammebrug     | 8       | 3       | mediaan       | 32.41 | 4.03 | 55.62 | 0.46 |
| Lammebrug     | 8       | 3       | vector output | 21.09 | 3.34 | 49.05 | 0.65 |
| Lammebrug     | 8       | 3       | encode-decode | 20.45 | 3.26 | 48.16 | 0.66 |
| Lammebrug     | 5       | 5-6     | mediaan       | 10.13 | 2.72 | 59.21 | 0.39 |
| Lammebrug     | 5       | 5-6     | vector output | 7.00  | 1.93 | 50.76 | 0.59 |
| Lammebrug     | 5       | 5-6     | encode-decode | 6.91  | 1.91 | 49.85 | 0.59 |
| Lammebrug     | 6       | 3-4     | mediaan       | 36.07 | 3.90 | 55.19 | 0.41 |
| Lammebrug     | 6       | 3-4     | vector output | 26.55 | 3.38 | 53.21 | 0.58 |
| Lammebrug     | 6       | 3-4     | encode-decode | 25.64 | 3.29 | 49.76 | 0.59 |
| Lammebrug     | 9       | 3-4     | mediaan       | 38.13 | 4.43 | 56.85 | 0.24 |
| Lammebrug     | 9       | 3-4     | vector output | 25.25 | 3.57 | 46.58 | 0.62 |
| Lammebrug     | 9       | 3-4     | encode-decode | 24.34 | 3.50 | 49.80 | 0.63 |
| Kruithuisbrug | 860     | 0-1     | mediaan       | 21.16 | 3.06 | 29.30 | 0.74 |
| Kruithuisbrug | 860     | 0-1     | vector output | 18.57 | 2.98 | 30.77 | 0.77 |
| Kruithuisbrug | 860     | 0-1     | encode-decode | 18.54 | 2.97 | 30.20 | 0.79 |
| Kruithuisbrug | 860     | 2       | mediaan       | 4.50  | 1.31 | 36.5  | 0.61 |
| Kruithuisbrug | 860     | 2       | vector output | 4.17  | 1.35 | 43.00 | 0.64 |
| Kruithuisbrug | 860     | 2       | encode-decode | 4.20  | 1.32 | 38.81 | 0.63 |
| Kruithuisbrug | 861     | 0-1     | mediaan       | 39.29 | 4.13 | 30.6  | 0.71 |
| Kruithuisbrug | 861     | 0-1     | vector output | 33.74 | 3.95 | 31.60 | 0.75 |
| Kruithuisbrug | 861     | 0-1     | encode-decode | 34.02 | 4.00 | 32.66 | 0.75 |
| Coenecoopbrug | 31      | 0       | mediaan       | 14.27 | 2.35 | 59.93 | 0.32 |
| Coenecoopbrug | 31      | 0       | vector output | 6.45  | 1.69 | 52.20 | 0.69 |
| Coenecoopbrug | 31      | 0       | encode-decode | 6.30  | 1.66 | 50.06 | 0.70 |
| Coenecoopbrug | 31      | 1       | mediaan       | 16.57 | 2.50 | 61.37 | 0.27 |
| Coenecoopbrug | 31      | 1       | vector output | 7.53  | 1.78 | 53.42 | 0.66 |
| Coenecoopbrug | 31      | 1       | encode-decode | 7.47  | 1.78 | 53.10 | 0.67 |

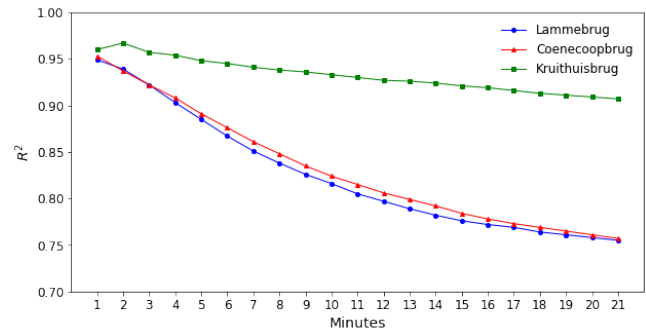
Het uiteindelijke model per brug om in real-time de verkeersintensiteit te voorspellen was geselecteerd op basis van de verschillende foutstatistieken. Voor de Coenecoopbrug en de Lammebrug was de encoder-decoder LSTM gekozen omdat de foutstatistieken over vrijwel de hele linie beter waren. Voor de Kruithuisbrug was het encoder-decoder maar



beter voor 1 van de meetlussen (860, laan 0-1). In dit geval is gekozen voor het vector output model. Hierbij heeft het model geleerd weekdagen van weekenddagen uit de univariate tijdseries te onderscheiden en past het model de voorspelling aan op basis van voorafgaande verkeersdrukte. In de Impactmonitor Brugopeningen worden drie scenario's doorge-rekend en gepresenteerd aan de brugbediener: de cumulatieve verkeersintensiteit over de aankomende 1 tot 7 minuten, het cumulatieve verkeer 8 tot 14 minuten vooruit en 15 tot 21 minuten. Voor het eerste scenario van 1 tot 7 minuten vooruit zijn de betrouwbaarheidsscores weergegeven in figuur 2. Hierbij is de prognose en de daadwerkelijke geobserveerde verkeersintensiteit afgezet tegen de tijd. De voorspellingen 1 minuut vooruit zijn zeer betrouwbaar ( $R^2 > 0.94$ ) (figuur 3). Binnen deze drie scenarios valt op dat de betrouwbaarheid van de voorspellingen afneemt wanneer er verder in de toekomst wordt voorspeld, maar de  $R^2$  blijft ruim boven 0.75. De hogere betrouwbaarheid van de Kruithuisbrug ( $R^2=0.91$  tot 21 minuten vooruit) komt mogelijk doordat het verkeer zich rondom deze brug uniformer over de meetlus beweegt en daardoor minder uitschieters bevat.



**Figuur 2.** – Werkelijke observaties en voorspelde verkeersintensiteiten van het aantal voertuigen voor het eerste blok van 7 minuten op basis van de – door het model nog ongeziene - test dataset. Data: Lammebrug meetlus 9 (rijbaan 3-4), Kruithuisbrug meetlus 861 (rijbaan 0-1) en Coenecoopbrug meetlus 31 (rijbaan 1).

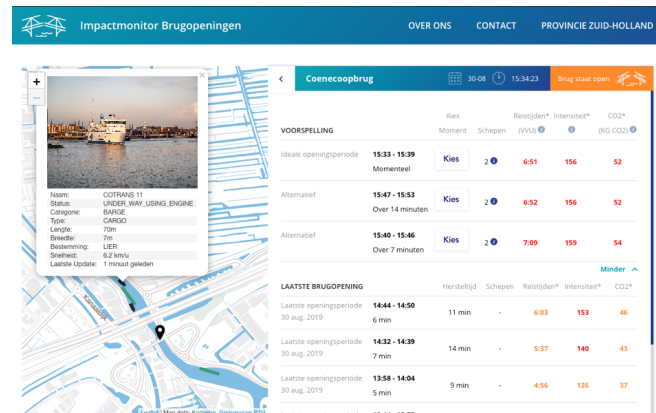


**Figuur 3.** – Betrouwbaarheid van de voorspellingen gedurende 21 minuten.

## 5. Deployment

### 5.1 Productiewaardig maken

Om het voorspelmodel in de praktijk te kunnen gebruiken moet het operationeel gemaakt worden. De input variabelen van het model worden real-time uitgelezen vanuit het NDW. De berekende uitkomsten van het model zijn vervolgens benaderbaar voor de webapplicatie middels een API. De data uit de API wordt opgehaald en getoond in de uiteindelijke webapplicatie (figuur ??).



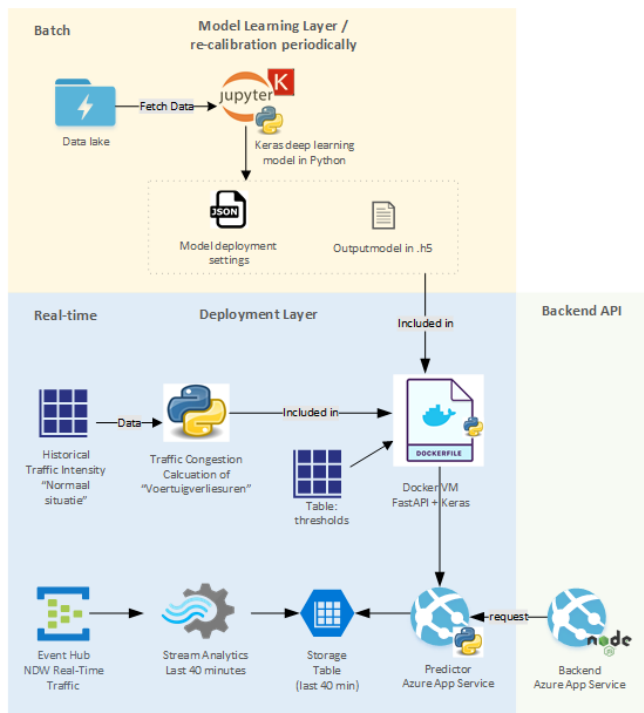
**Figuur 4.** Screenshot van detailpagina van één brug uit de Impactmonitor Brugopeningen

### 5.2 Real-time databronnen

De input voor het model is de NDW meetlusinformatie met daarin de intensiteit per minuut (figuur 5). Deze data moet real-time beschikbaar zijn. Dit is gerealiseerd door elke minuut de Open Data-set bij het NDW op te halen via FTP en op te slaan in een Service Bus (Azure EventHub). Deze zet vervolgens per meetlus de data klaar in sets van 30 minuten in een tijdelijke Azure Table omwille van de performance.

### 5.3 Prognose API

De voorspelling is beschikbaar gesteld middels een zelf ontwikkelde API in de programmeertaal Python [4] gebaseerd op FastAPI framework [17]. Er is gekozen voor een zelf ontwikkelde API omdat performance erg belangrijk is en er geen



**Figuur 5.** Onderzoeksfase met als resultaat een model uitkomst (.h5 bestand) die wordt gebruikt in de real-time voorspel API.

standaardoplossingen zijn om een API te maken voor een Keras model. De documentatie van de API is gedaan middels OpenAPI standaarden.

De API voert de volgende stappen uit om tot het resultaat te komen bij het op vragen van de voorspel end-point /predict/{isrs}:

1. Configuratie van brug ophalen met daarin vooraf ingeladen brug specifieke LSTM model (.h5 files) & verkeerscongestierekenmodel (in-memory). Het verkeerscongestierekenmodel is ontwikkeld door Arcadis en rekent de voertuigverliesuren en hersteltijd van een brug uit op basis van de schokgolftheorie.
2. NDW real-time dataset ophalen met verkeersintensiteiten en verhouding soort verkeer.
3. LSTM model laten voorspellen met NDW set als input.
4. Output van model zijn voorspelde intensiteiten voor aankomende 21 minuten die worden “ingeladen” in het verkeerscongestierekenmodel.
5. Verkeerscongestierekenmodel rekent voertuigverliesuren (VUU) en hersteltijd van brug uit.
6. CO2 uitstoot indicator wordt uitgerekend aan de hand van Voertuigverliesuren en verhouding van het soort verkeer.

7. Resultaten worden samengevat in 3 tijdsblokken van 7 minuten (in de toekomst) – dat de tijdsblokken zijn waaruit de brugbediener kan kiezen.

De verkeersintensiteiten prognose API is technisch beschikbaar gesteld via Azure App Services door de logica, verkeerscongestiemodel en FastAPI in te pakken in een Docker container waar alle afhankelijkheden in zitten. Het voorspelmodel is te benaderen via een webadres(<https://domeinnaam.nl/predict/isrs>) en geeft 3 tijdsblokken terug als uitkomst. Een voorbeeld hiervan is te zien in figuur 6.

```

1- {
2-   "predictions": [
3-     {
4-       "forecastText": "Ideale openingsperiode",
5-       "row": 1,
6-       "delay": 0,
7-       "offset_start": "13:13",
8-       "offset_end": "13:19",
9-       "forecast_voertuigverliesuren": 3.450309388851006,
10-      "forecast_voertuigverliesuren_color": "green",
11-      "forecast_TotalRecords_norm": 94.21042148272196,
12-      "forecast_CO2kilogram_VVU": 26.153345167490624,
13-      "forecast_CO2kilogram_VVU_color": "green",
14-      "hersteltijd": 7.857142857142857,
15-      "sort": 3.450309388851006
16-    },
17-     {
18-       "forecastText": "Alternatief",
19-       "row": 2,
20-       "delay": 7,
21-       "offset_start": "13:20",
22-       "offset_end": "13:26",
23-       "forecast_voertuigverliesuren": 3.5436123423116266,
24-       "forecast_voertuigverliesuren_color": "green",
25-       "forecast_TotalRecords_norm": 96.28804349899292,
26-       "forecast_CO2kilogram_VVU": 26.86058155472213,
27-       "forecast_CO2kilogram_VVU_color": "green",
28-       "hersteltijd": 8,
29-       "sort": 3.5436123423116266
30-     },
31-     {
32-       "forecastText": "Alternatief",
33-       "row": 3,
34-       "delay": 14,
35-       "offset_start": "13:27",
36-       "offset_end": "13:33",
37-       "forecast_voertuigverliesuren": 3.644427254517559,
38-       "forecast_voertuigverliesuren_color": "green",
39-       "forecast_TotalRecords_norm": 98.47048182920489,
40-       "forecast_CO2kilogram_VVU": 27.624758589243097,
41-       "forecast_CO2kilogram_VVU_color": "green",
42-       "hersteltijd": 7.857142857142857,
43-       "sort": 3.644427254517559
44-     }
45-   ],
46-   "trafficFlow": [
47-     {
48-       "roadName": "N207",
49-       "roadLaneDirection": "W-0",
50-       "threshold_color": "green",
51-       "meetus_id": "P21091MST_0031_01",
52-       "meetus_index": [1,2,3,4],
53-       "input": [6, 8, 10, 10, 8, 5, 12, 3, 3, 2, 9, 8, 6, 6, 5, 5, 8, 6, 5, 5, 7],
54-       "output": [5.9, 6.1, 6.2, 6.2, 6.3, 6.3, 6.3, 6.3, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.4, 6.3]
55-     }
56-   ]
57- }
58-
59-
60-

```

**Figuur 6.** Voorbeeld uitkomst van de voorspel API van de Coenecoopbrug – opgevraagd op zaterdag 31 augustus 2019 om 13:13, waarbij in onderdeel predictions de 3 voorspel blokken zichtbaar zijn en in het onderdeel trafficFlow de gebruikte verkeersintensiteit als input en de voorspelde intensiteit als output.

## 5.4 Performance

Performance is van enorm belang – gezien de brugbedieners niet willen wachten op de uitkomst van het model. De Non Functional Requirement (NFR) is dat de voorspelling binnen 3 seconden op het scherm te zien moet zijn. De uiteindelijk oplossing biedt een responsetijd tussen de 0,5 en 2 seconden.

## 6. Conclusie en vervolgonderzoek

Uit de analyse van verkeersintensiteiten bleek dat de verkeersdrukte rondom een aantal bruggen van de provincie uit verschillende componenten bestaat. Als eerste een cyclische dagelijkse dag en spits component. Ten tweede een wisselende component waarbij de direct voorgaande verkeersdrukte voorspellend is voor wat daarna volgt en ten derde een stochastische component waarin verkeersintensiteit schommelingen in korte uitbarstingen voorkomen. Ten opzichte van het mediaan model waarin informatie zoals tijdstip en dag is verwerkt kan het LSTM zonder deze informatie toch aanzienlijk betere voorspellingen maken. Dit geeft aan dat zowel de direct voorgaande druke als statische factoren een belangrijke rol spelen in de verkeersintensiteit.

De LSTM modellen bieden verder een nauwkeurige mogelijkheid om meerdere stappen in de toekomst vooruit te voorspellen. Hiermee kan een brugbediener tot 21 minuten vooruit een prognose krijgen over de verwachte verkeersdrukte en de berekende voertuigverliesuren, extra uitstoot en hersteltijd van verkeer rondom een brug veroorzaakt door een brugopening. Ook zijn er nog ruime mogelijkheden tot het verder optimaliseren van de verkeersintensiteit prognoses zoals het bouwen van een multivariate LSTM model waarin tijd en weergegevens worden meegewogen naast het trainen op grotere historische datasets.

Daarnaast zijn er optimalisaties mogelijk door data van de scheepvaart verder mee te wegen in het advies dat de Impactmonitor Brugopeningen geeft. Het ontwikkelde model is opgeleverd als gebruiksvriendelijke webapplicatie waarin big data concepten en recente ontwikkelingen op het gebied van machine learning zijn weg naar de praktijk vinden. De applicatie is in een fase waarin deze door de eerste gebruikers in gebruik kan worden genomen.

## Acknowledgements

Dit project was niet mogelijk geweest zonder de kennis uit de praktijk en support vanuit de afdeling Dienst Beheer Infrastructuur, in het bijzonder Ellen van der Knaap, Tim Blanken, Tino van As en Jillis Mani. Het ontwikkelen, opleveren en stroomlijnen van de applicatie was niet gelukt zonder de inbreng van de vele collega's vanuit het vernieuwd datawarehouse van de provincie Zuid-Holland, bovenal Bart Witteveen, Firuze Kuru, Dennis Weijnsfeld en Kevin Otjes.

## Referenties

- [1] <https://bereiknu.nl/> BEREIK! is hét samenwerkingsplatform van Rijkswaterstaat, Provincie Zuid Holland, Metropoolregio Rotterdam Den Haag, Gemeenten Rotterdam en Den Haag en Havenbedrijf Rotterdam voor regionale bereikbaarheidsvraagstukken.
- [2] <https://blauwegolfverbindend.nl/>, Blauwe Golf Verbindend ontsluit real-time gegevens over geopende bruggen en beschikbare ligplaatsen in havens

- [3] [https://www.zuid-holland.nl/@21238/smart-shipping/Partnershipovereenkomst Smart Shipping](https://www.zuid-holland.nl/@21238/smart-shipping/Partnershipovereenkomst%20Smart%20Shipping).
- [4] Van Rossum, G., Drake Jr, F.L., 1995. Python tutorial. Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands. <https://www.python.org/downloads/release/python-367>
- [5] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). 2016.
- [6] Chollet F. , 2015. Keras, GitHub. <https://github.com/fchollet/keras>
- [7] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.
- [8] Zurada J.M., 1992. Introduction to artificial neural systems vol. 8. West publishing company, St.Paul.
- [9] Jain A.K., Mao J., Mohiuddin K.M., 1996. Artificial neural networks: A tutorial. Computer 29, 31-44.
- [10] Zhang G., Patuwo B.E., Hu M.Y., 1998. Forecasting with artificial neural networks: The state of the art. International journal of forecasting 14, 35-62.
- [11] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.
- [12] Mandic D.P., Chambers J., 2001. Recurrent neural networks for prediction: learning algorithms, architectures and stability. John Wiley & Sons, Inc.
- [13] Hochreiter S., Schmidhuber J., 1997. Long short-term memory. Neural computation 9, 1735-1780.
- [14] Gers F.A., Eck D., Schmidhuber J., 2001. Applying LSTM to time series predictable through timewindow approaches. In: Dorffner G., Bischof H., Hornik K. (eds) Artificial Neural Networks. ICANN 2001, Lecture Notes in Computer Science, vol 2130. Springer, Berlin, Heidelberg.
- [15] Sak H., Senior A., Beaufays F., 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. INTERSPEECH 2014, 14-18 September, Singapore, 338-342.
- [16] Brownlee, Jason. "Text Generation With LSTM Recurrent Neural Networks in Python with Keras." (2018).
- [17] <https://fastapi.tiangolo.com/> FastAPI framework, high performance, easy to learn, fast to code, ready for production
- [18] Newell, Gordon F. "A simplified theory of kinematic waves in highway traffic, part I: General theory." Transportation Research Part B: Methodological 27.4 (1993): 281-287.