

POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Technologie informacyjne w systemach automatyki (ART)

PRACA DYPLOMOWA MAGISTERSKA

Algorytmy obliczania macierzy przepływu ruchu
w aglomeracji miejskiej wykorzystujące metody
uczenia maszynowego

The algorithms for calculation of the origination
destination matrix using machine learning
methods

AUTOR:
Robert Jan Czwartosz

PROWADZĄCY PRACĘ:
dr Marek Bazan

KONSULTANT:
prof. dr hab. inż. Ewa Skubalska-Rafajłowicz

Spis treści

1	Wstęp	2
2	Przegląd literaturowy	3
3	Cel i założenia pracy	8
4	Metoda rozwiązywania problemu	11
5	Generowanie danych	17
6	Model naśladowy symulator SUMO	21
7	Algorytmy genetyczne	26
8	Wyniki numeryczne	35
9	Perspektywy rozwoju	51
10	Podsumowanie i wnioski	53
	Bibliografia	53

Rozdział 1

Wstęp

W godzinach szczytu często dochodzi do zakorkowania ulic miast. To zatłoczenie stanowi poważny problem, ponieważ wydłuża czas jazdy oraz zanieczyszcza powietrze spalinami. Tworzenie nowych dróg i ich rozbudowa pozwala zmniejszyć korki. Jednak jest to dość kosztowne rozwiązanie, ponieważ koszt jednego kilometra drogi jest rzędu kilkudziesięciu milionów złotych. Zatem każda optymalizacja ruchu (np. wyznaczanie optymalnej trasy, regulacja świateł drogowych), która choć trochę zmniejszyłaby zatłoczenie w miastach, jest cenna. Do sprawdzania efektów różnych sposobów optymalizacji ruchu w mieście potrzebna jest symulacja danego miasta.

Symulowanie ruchu drogowego daje dużo możliwości. Symulacja ruchu miejskiego pozwala na predykcję natężenia ruchu w mieście, przewidywanie czasu dojazdu, wyznaczanie optymalnej trasy oraz regulację świateł drogowych. Symulacja miasta pozwala testować rozwiązania, które mogą być stosowane w przyszłości. Dzięki zastosowaniu takich rozwiązań czas podróży będzie krótszy, powietrze będzie czystsze oraz planowanie transportu będzie ułatwione.

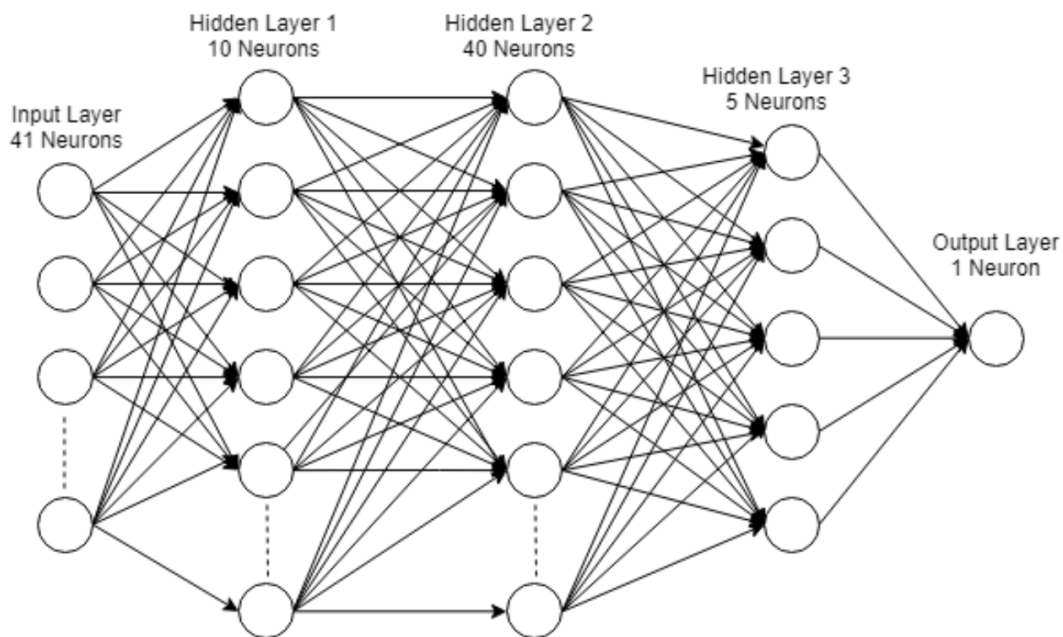
Do stworzenia symulacji miasta wymagana jest znajomość dynamicznej macierzy przepływów. Przepływ informuje o liczbie pojazdów poruszających się z węzła A (początek trasy) do węzła B (koniec trasy) w pewnym przedziale czasowym (np. od godz. 6:00 do godz. 7:00)[29]. Macierz przepływów zawiera przepływy dla każdej możliwej pary węzłów. Natomiast dynamiczna macierz przepływów zawiera macierze przepływów dla ciągu kolejnych przedziałów czasowych (np. ciągu składającego się z trzech przedziałów: 6:00 - 7:00, 7:00 - 8:00 oraz 8:00 - 9:00). W niniejszej pracy przedstawiono badania nad metodą estymacji dynamicznej macierzy przepływów opisanej w pracy [24]. Dane dla obliczania dynamicznej macierzy przepływów zostały wygenerowane za pomocą symulatora SUMO (*Simulation of Urban MObility*)[16]. Rozpatrzono ulicę Legnicką we Wrocławiu oraz część miasta Sioux Falls[32].

Rozdział 2. przedstawia przegląd publikacji na temat estymacji macierzy przepływów przy użyciu sieci neuronowych. W rozdziale 3. określono cel i założenia pracy. W rozdziale 4. opisane zostało działanie algorytmu rozwiązującego problem. Rozdziały 5, 6 i 7 opisują kolejne etapy rozwiązywania problemu: generowanie danych, tworzenie modelu naśladującego symulator ruchu miejskiego oraz poszukiwanie dynamicznej macierzy przepływu za pomocą algorytmów genetycznych ([18], [1]). Rozdział 8. przedstawia wyniki. W rozdziale 9. opisano możliwości rozwoju oprogramowania obliczającego dynamiczną macierz przepływów. Rozdział 10. zawiera podsumowanie i wnioski wyciągnięte z pracy nad problemem estymacji dynamicznej macierzy przepływów.

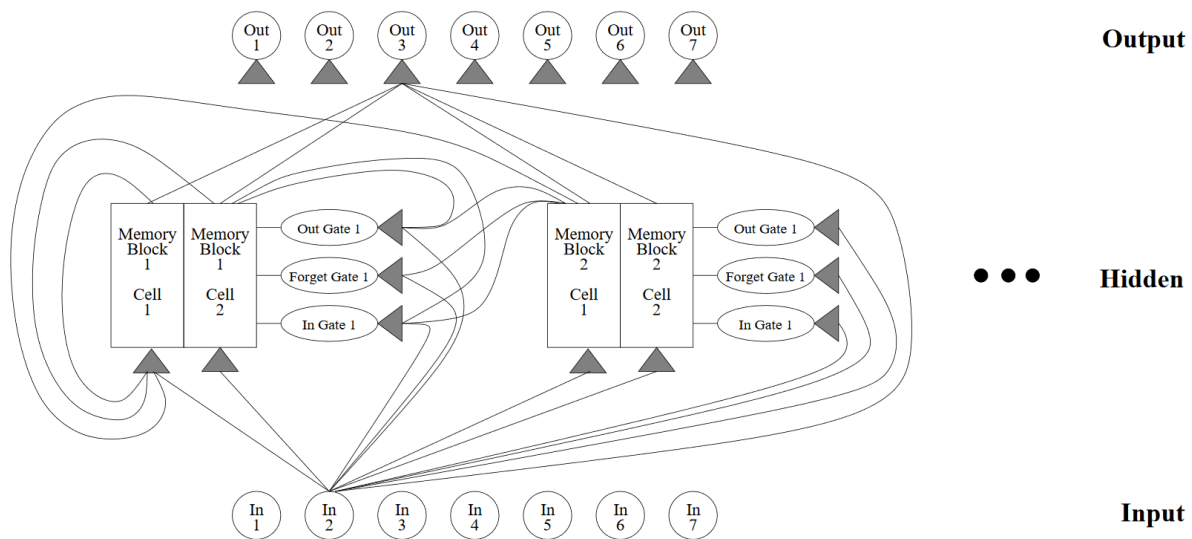
Rozdział 2

Przegląd literaturowy

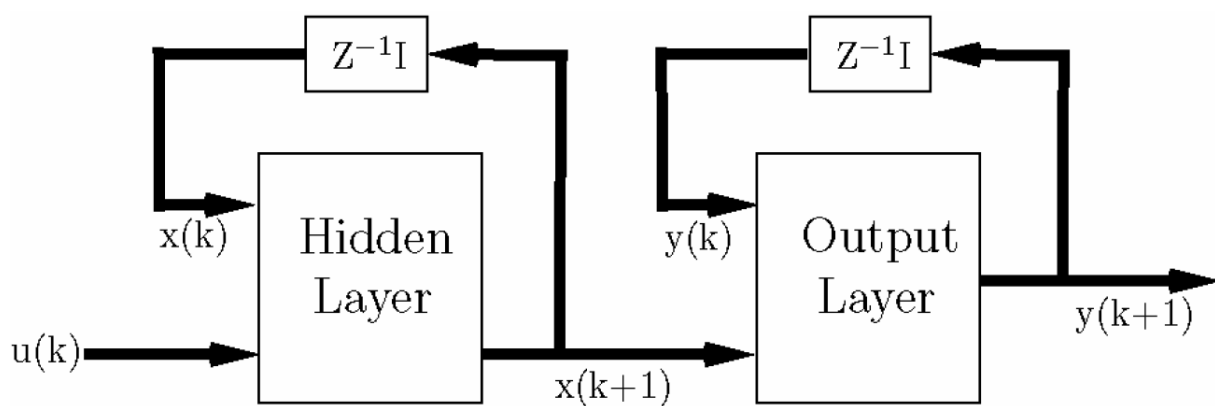
W literaturze dotyczącej użycia sieci neuronowych do obliczania macierzy przepływów omawiane są trzy zagadnienia tzn.: estymacja macierzy przepływów (OD - Origin - Destination), estymacja dynamicznej macierzy OD oraz predykcja macierzy OD. Estymacja polega na obliczaniu macierzy OD opisującej ruch w teraźniejszości lub w przeszłości, natomiast predykcja polega na obliczaniu macierzy OD opisującej ruch w przyszłości. Sieciami neuronowymi, używanymi do rozwiązywania tych problemów, są: perceptron wielowarstwowy (MLP - Multilayer Perceptron)[Rysunek 2.1], LSTM - Long Short Term Memory[Rysunek 2.2], rekurencyjna sieć neuronowa (RNN - Recurrent Neural Network)[Rysunek 2.3], konwolucyjne sieci neuronowe (CNN - Convolutional Neural Network)[Rysunek 2.4], sieci radialne (RBF - Radial Basis Function)[Rysunek 2.5] oraz sieci Hopfield'a (HNN - Hopfield Neural Network)[Rysunek 2.6].



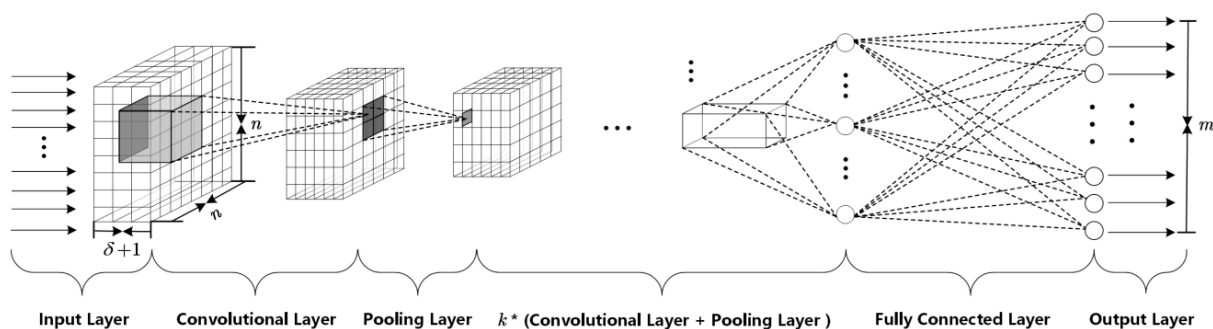
Rysunek 2.1: Przykład struktury sieci MLP[8]. Sieć składa się z warstwy neuronów wejściowych, trzech warstw ukrytych oraz warstwy wyjściowej. Wyjściem neuronu jest wartość funkcji aktywacji z sumy ważonej wejść. Neurony połączone w ten sposób tworzą sieć MLP. W przedstawionym modelu na wejściu sieci jest wektor liczb, natomiast na wyjściu jest jedna liczba.



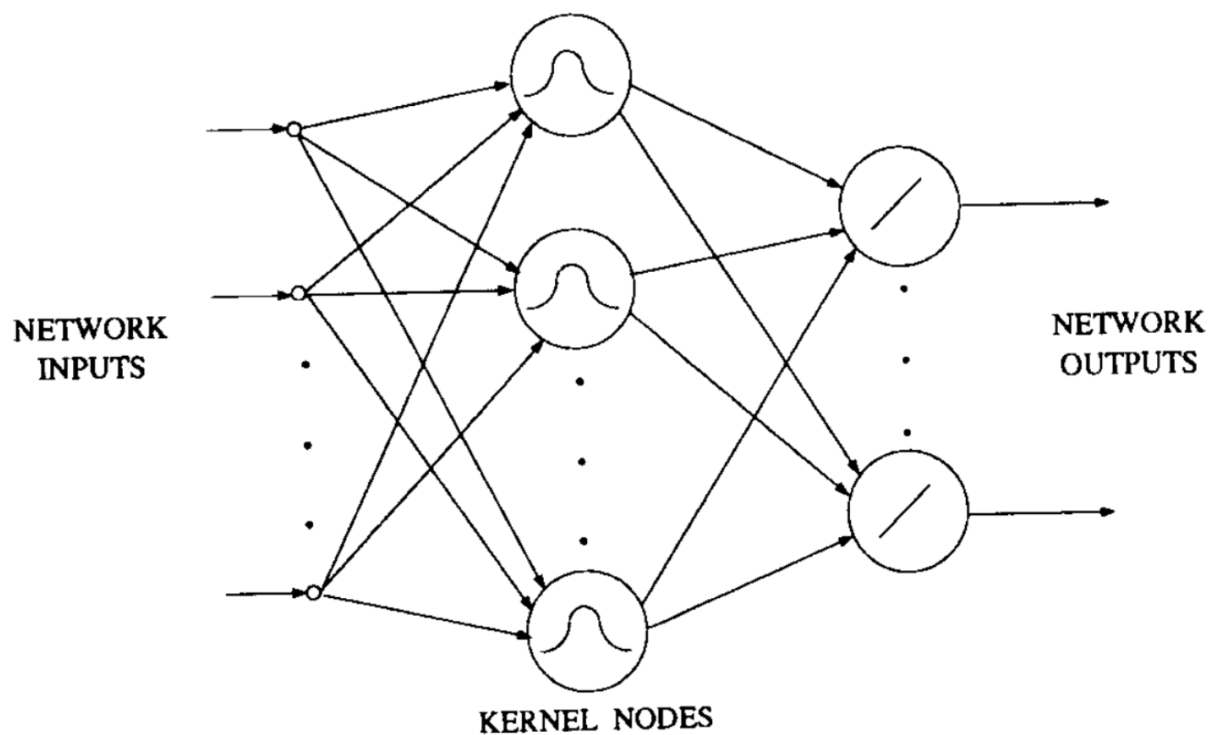
Rysunek 2.2: Przykład struktury sieci LSTM[4]. Sieć składa się z warstwy wejściowej, warstwy ukrytej LSTM oraz z warstwy wyjściowej. Warstwa ukryta składa się z bloków pamięci. Każdy blok pamięci składa się z dwóch komórek LSTM. Przed każdym blokiem pamięci znajdują się 3 bramki: bramka wejściowa, bramka zapominająca i bramka wyjściowa. W przedstawionym modelu na wejściu sieci jest wartość zmiennej objaśniającej w chwili t , natomiast na wyjściu jest przewidywana wartość zmiennej objaśnianej w chwili $t + 1$.



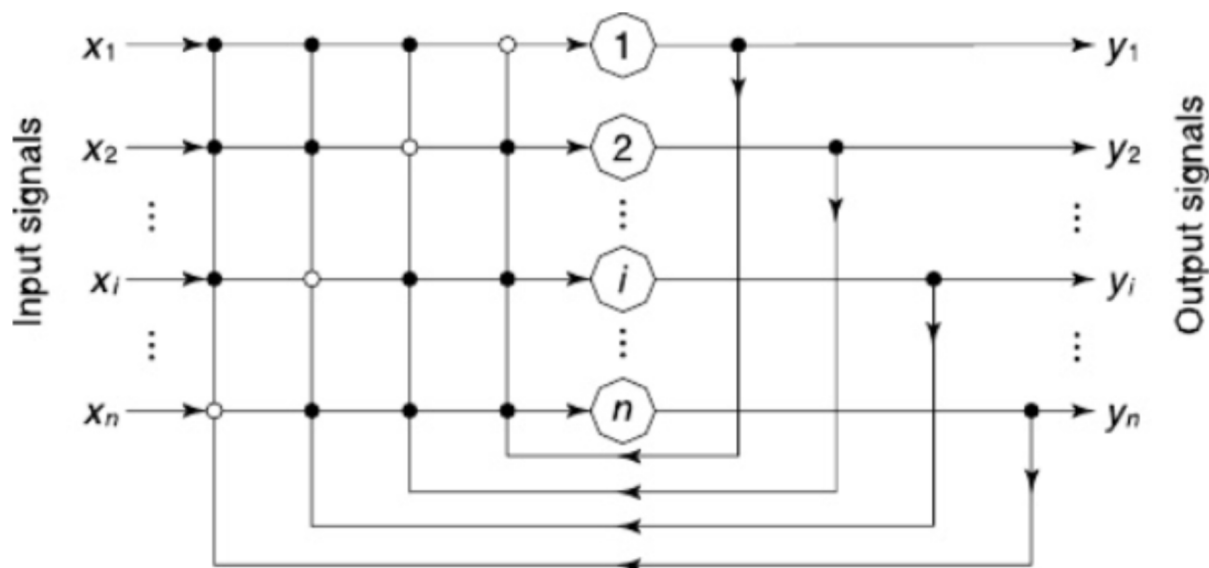
Rysunek 2.3: Struktura sieci RNN[10]. Na wejściu i na wyjściu sieci rekurencyjnej jest wektor liczb.



Rysunek 2.4: Struktura sieci CNN[24]. Na wejściu sieci CNN jest macierz n na n na $\delta + 1$. Na wyjściu sieci jest wektor m liczb.



Rysunek 2.5: Przykład struktury sieci RBF[2]. Neurony w warstwie ukrytej obliczają wartość funkcji Gaussa z normy różnicy między wejściem, a centrum. Na wejściu i na wyjściu sieci rekurencyjnej jest wektor liczb.



Rysunek 2.6: Struktura sieci HNN[22]. Wyjście każdego neuronu jest kierowane na wejście pozostałych. Neurony obliczają wartość funkcji aktywacji z sumy ważonej wejść. W pierwszej iteracji na wejście neuronów podawany jest sygnał wejściowy, a w następnych iteracjach na wejście neuronów jest podawane uzyskane poprzednio wyjście. Proces jest kontynuowany do momentu ustabilizowania się wyjścia. Na wejściu i na wyjściu sieci jest wektor n liczb.

Problem estymacji statycznej macierzy OD jest rozwiązywany przy użyciu sieci MLP ([21], [26], [17], [13], [12], [3], [23], [9], [8]), sieci rekurencyjnych ([25], [10]), sieci radialnych [11], oraz sieci Hopfield'a [5]. Publikacjami wydanymi przed 2010 rokiem są: [13], [12], [3], [9], [25], [10], [11] i [5]. Publikacjami wydanymi po 2010 roku są: [21], [26], [17], [23] i [8]. Artykułami opisującymi zastosowanie sieci MLP do estymacji macierzy OD opisujące ruch w mieście są: [21], [26], [17], [13], [12] i [3]. Artykuł [21] przedstawia rozwiązanie, w którym sieć MLP jest uczona algorytmem gradientów sprzężonych[19]. Sieć MLP jest uczona na danych zabranych z okolic autostrady M42 w Wielkiej Brytanii[21]. Publikacja [26] opisuje rozwiązanie, w którym sieć MLP jest uczona algorytmem Levenberg'a-Marquardt'a[20]. Dane do nauki sieci są wygenerowane dla stworzonej sieci przez autora artykułu [26]. W artykule [13] opisano rozwiązanie, w którym sieć MLP jest trenowana algorytmem gradientów sprzężonych[19]. Dane uczące są wygenerowane na podstawie sieci transportowej miasta Sioux Falls[13]. W artykułach [8], [23] i [9] przedstawiono zastosowanie sieci MLP do estymacji macierzy OD w sieciach komputerowych. W wymienionych publikacjach przedstawiono rozwiązania oparte na sieciach neuronowych uczonych na danych uzyskanych z sieci komputerowej Abilene. W artykule [9] sieć MLP jest uczona algorytmem IPFP - iterative proportional fitting procedure[9]. Natomiast w publikacjach [23] i [8] przedstawiono rozwiązania, w których sieć MLP jest uczona algorytmem Levenberg'a-Marquardt'a[20]. Ponadto w sieciach komputerowych macierz przepływu jest estymowana przy użyciu sieci rekurencyjnych ([25], [10]) oraz sieci RBF [11]. W wymienionych publikacjach przedstawiono rozwiązania oparte na sieciach neuronowych uczonych na danych uzyskanych z sieci komputerowej Abilene. W publikacji [25] opisano rozwiązanie, w którym sieć RNN jest uczona algorytmem propagacji wstecznej w czasie[34]. Artykuł [11] przedstawia rozwiązanie oparte na sieciach RBF uczonych algorytmem IPFP[11].

Problem estymacji dynamicznej macierzy OD jest rozwiązywany przy użyciu sieci MLP [6] (1998 r.) oraz sieci CNN [24] (2019 r.). Rozwiązanie w artykule [6] polega na

nauczeniu sieci MLP opisywania ruchu na skrzyżowaniach w poszczególnych przedziałach czasowych. Na podstawie zachowania pojazdów na skrzyżowaniach w każdym przedziale czasowym obliczana jest dynamiczna macierz przepływu [6]. Sieć MLP jest uczona algorytmem propagacji wstecznej[28]. Dane są uzyskane z autostrad w okręgu przemysłowym Hanshin w mieście Osaka. W publikacji [24] zastosowano sieć CNN do odwzorowywania zachowania symulatora ruchu drogowego, natomiast algorytmy genetyczne ([18], [1]) użyte są do obliczenia dynamicznej macierzy OD. Sieć CNN jest uczona algorytmem ADAM - Adaptive Moment Estimation[14]. Dane są uzyskane z miasta Kunshan w Chinach.

Problem predykcji macierzy OD jest rozwiązywany przy użyciu sieci LSTM ([31], [36]), sieci MLP ([35], [33]) oraz sieci CNN [15]. Publikacją wydaną przed 2010 rokiem jest [33]. Publikacjami wydanymi po 2010 roku są: [31], [36], [35] i [15]. W artykule [35] opisano zastosowanie sieci MLP do przewidywania macierzy OD na podstawie ilości pojazdów wyjeżdżających i wjeżdżających do węzłów oraz na podstawie odległości między węzłami. Ponadto w publikacji [35] znajduje się porównanie metod uczenia sieci MLP: metoda propagacji wstecznej (BP - Back Propagation)[28], metoda ze zmiennym współczynnikiem uczenia (VLR - Variable Learning Rate) oraz metoda Levenberg'a - Marquardt'a[20]. Publikacja [33] przedstawia rozwiązanie problemu predykcji macierzy OD w transporcie publicznym oparte o sieć MLP. W artykule sprawdzono algorytmy uczenia sieci takie jak: metoda największego spadku, metoda największego spadku z adaptacyjnym współczynnikiem uczenia i metoda gradientów sprzężonych[19]. Dane uczące są uzyskane z Aleji Aleksandry w Atenach w Grecji. W artykule [31] opisano rozwiązanie stosujące sieci LSTM. Dane uczące są zebrane z miasta Rennes we Francji. Artykuł [15] przedstawia rozwiązanie problemu predykcji macierzy OD opisującej ruch taksówek w okolicach Manhattan'u w Nowym Jorku. Problem jest rozwiązany przy pomocy sieci CNN oraz sieci kontekstowych[15]. W publikacji [36] przedstawiono rozwiązanie problemu predykcji macierzy OD w sieci komputerowej, w którym sieć LSTM jest uczona algorytmem RMSProp[30]. Dane są uzyskane z sieci komputerowej Abilene.

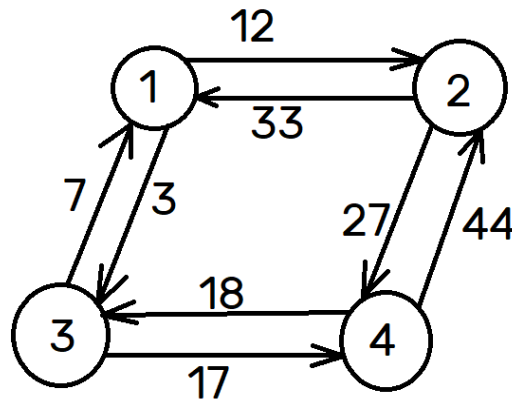
Jest mało publikacji, w których używa się sieci CNN do estymacji lub predykcji macierzy OD. W publikacji [15] sieci konwolucyjne występują w połączeniu z sieciami kontekstowymi. Takie połączenie jest używane do rozwiązania problemu predykcji macierzy OD. Natomiast w artykule [24] sieci CNN występują w połączeniu z algorytmami genetycznymi. Takie połączenie jest używane do rozwiązania problemu estymacji dynamicznej macierzy OD.

W niniejszej pracy przedstawiono badania nad metodą estymacji dynamicznej macierzy OD przy użyciu sieci CNN [24] dla danych z ulicy Legnickiej we Wrocławiu[29] oraz z miasta Sioux Falls[32]. Sieć CNN jest uczona algorytmem ADAM[14].

Rozdział 3

Cel i założenia pracy

Celem pracy było przeprowadzenie badań nad metodą ([24]) obliczania dynamicznej macierzy przepływów (*Origin-Destination matrix*) na podstawie natężeń (wektora natężeń) ruchu. Natężenie ruchu jest definiowane jako liczba pojazdów poruszająca się między sąsiednimi węzłami w danym przedziale czasowym (przykład przedstawiono na rysunku 3.1).



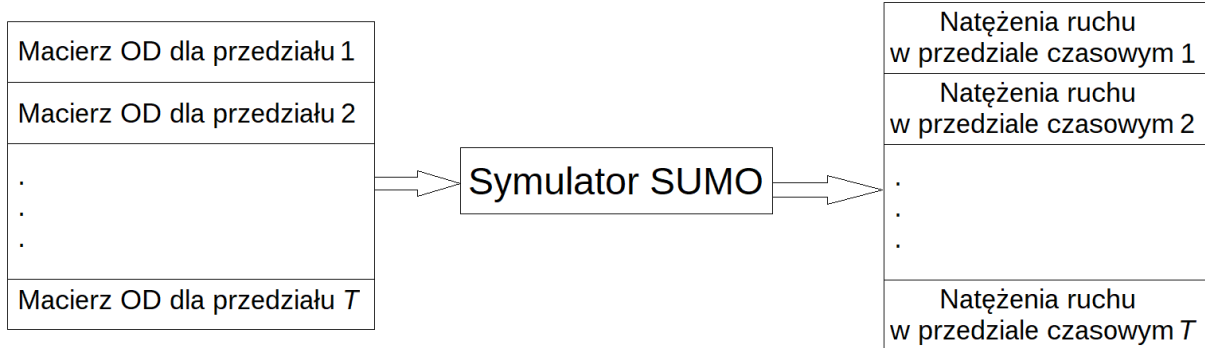
Rysunek 3.1: W pewnym przedziale czasowym (np. od 8:00 do 9:00) z węzła nr 3 do węzła nr 1 przejechało 7 pojazdów. Zatem w tym przedziale czasowym natężenie ruchu z węzła nr 3 do węzła nr 1 wynosi 7. Wektor natężeń ruchu w tej sieci w przedziale czasowym 8:00 - 9:00 może być zapisany następująco: [12, 33, 27, 44, 18, 17, 7, 3].

Dynamiczna macierz przepływów jest ciągiem macierzy przepływów. Przykładowo można zdefiniować dynamiczną macierz przepływów opisującą ruch miejski od 6:00 do 9:00, która składa się z 3 macierzy opisujących ruch w przedziałach czasowych trwających godzinę (rysunek 3.2).

6:00 - 7:00					7:00 - 8:00					8:00 - 9:00				
-	(1)	(2)	(3)	(4)	-	(1)	(2)	(3)	(4)	-	(1)	(2)	(3)	(4)
(1)	-	110	40	40	(1)	-	50	20	20	(1)	-	14	11	3
(2)	50	-	200	55	(2)	50	-	80	35	(2)	15	-	26	8
(3)	0	0	-	10	(3)	0	0	-	20	(3)	17	11	-	9
(4)	200	30	40	-	(4)	110	30	25	-	(4)	34	2	12	-

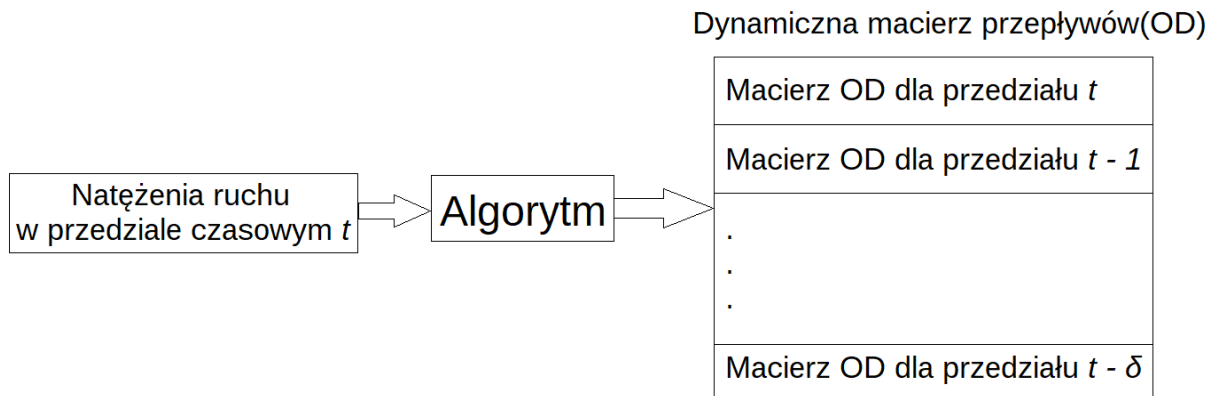
Rysunek 3.2: Ciąg trzech macierzy OD opisuje ruch od 6:00 do 9:00. Z dynamicznej macierzy OD można odczytać że np. w godzinach od 6:00 do 7:00 200 pojazdów rozpoczęło podróż od węzła nr 2 do węzła nr 3

Dane dla obliczania dynamicznej macierzy przepływów zostały wygenerowane za pomocą symulatora SUMO (*Simulation of Urban MObility*)[16]. Rozpatrzono ulicę Legnicką we Wrocławiu oraz część miasta Sioux Falls[32]. Symulator SUMO pozwala generować ciągi wektorów natężeń ruchu na podstawie ciągu macierzy OD (Origin-Destination). Wygenerowane dane zostały użyte w algorytmie estymacji dynamicznej macierzy OD. Sposób generowania danych przedstawiono na rysunku 3.3.



Rysunek 3.3: Symulator SUMO był używany do generowania natężeń ruchu na podstawie ciągu T macierzy OD. Zatem na wejściu symulatora był ciąg T macierzy OD, a na wyjściu ciąg T wektorów natężeń ruchu. Dla symulacji ulicy Legnickiej $T = 24$ (24 przedziały po 15 min, łącznie 6h), natomiast dla Sioux Falls $T = 4$ (4 przedziały po 40 min, łącznie 2h 40 min).

Na wejściu algorytmu były natężenia ruchu w przedziale czasowym t , a na wyjściu była dynamiczna macierz przepływu (rysunek 3.4). Ciąg macierzy OD opisywał ruch odbywający się w przedziale czasowym t oraz w δ poprzednich przedziałach czasowych. Algorytm rozwiązujący problem był oparty na konwolucyjnych sieciach neuronowych (*CNN - Convolutional Neural Network*) oraz algorytmach genetycznych (*GA - Genetic Algorithm*) z pracy [24].



Rysunek 3.4: Wejściem algorytmu są natężenia (wektor natężeń) ruchu w przedziale czasowym t (np. od 9:00 do 9:15). Wyjściem algorytmu jest dynamiczna macierz przepływu, która jest ciągiem macierzy OD opisujących ruch w przedziale t (od 9:00 do 9:15) i w δ (np. $\delta = 4$) poprzednich przedziałach (8:00 - 8:15, 8:15 - 8:30, 8:30 - 8:45 i 8:45 - 9:00).

Działanie całego algorytmu oraz działanie sieci CNN były oceniane za pomocą średniego błędu bezwzględnego wyrażonego w procentach (MAPE - *Mean Absolute Percentage Error*). Wynik był również oceniany za pomocą średniego błędu bezwzględnego (MAE -

Mean Absolute Error) oraz wskaźnika determinacji R^2 . Ocena wyniku odbywała się przy użyciu następujących wzorów:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (3.1)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (3.3)$$

gdzie y_i jest i -tą obserwacją wyjścia, \hat{y}_i jest estymowaną wartością wyjścia i oraz $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ jest średnią arytmetyczną empirycznych wartości wyjścia. Wskaźnik R^2 przyjmuje wartości od 0.00 do 1.00. Im większa wartość R^2 , tym dokładniejsze wyniki.

Rozdział 4

Metoda rozwiązywania problemu

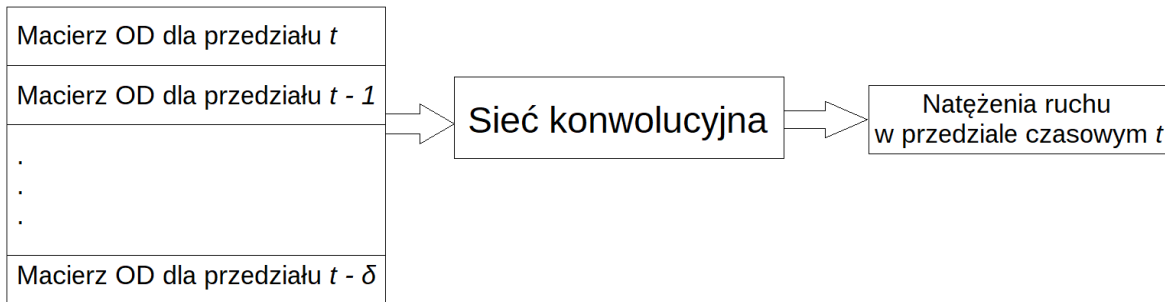
Rozwiązanie problemu polegało na implementacji algorytmu znajdującego się w pracy [24]. Algorytm składał się z czterech kroków:

1. wygenerowanie danych za pomocą symulatora,
2. stworzenie modelu naśladującego symulator,
3. poszukiwanie algorytmem genetycznym wzorcowej dynamicznej macierzy przepływów,
4. poszukiwanie (przy uwzględnieniu macierzy wzorcowej) algorytmem genetycznym właściwej dynamicznej macierzy przepływów.

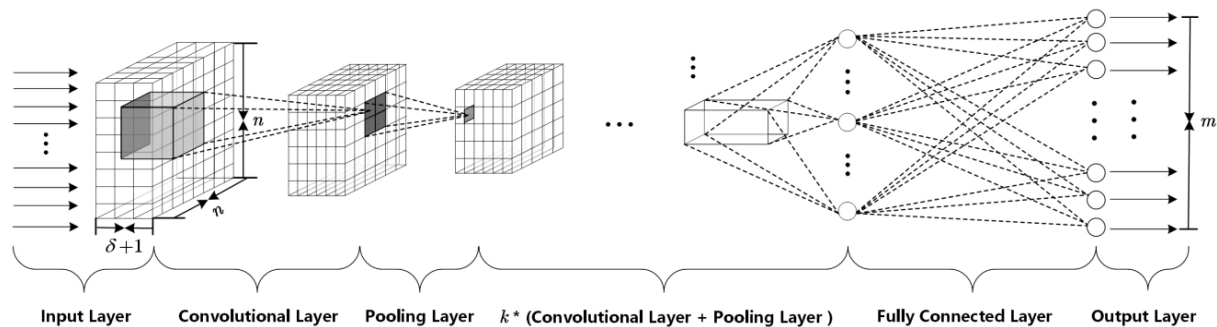
Najpierw należało wygenerować trzy zbiory danych: dane aprioryczne (*Prior Traffic Data*), dane z przeszłości (*Historical Traffic Data*) i dane w czasie rzeczywistym (*Real-time Traffic Data*) [24]. Na podstawie danych apriorycznych nauczony został model naśladujący symulator (krok 2. algorytmu). Dane z przeszłości były potrzebne do poszukiwania wzorcowej dynamicznej macierzy OD (krok 3. algorytmu). Przy użyciu danych w czasie rzeczywistym estymowana była dynamiczna macierz przepływów dla konkretnego przedziału czasowego (krok 4. algorytmu). Generowanie danych jest opisane w rozdziale 5.

Następnym krokiem było stworzenie modelu naśladującego symulator SUMO. Zadaniem tego modelu było jak najdokładniejsze odwzorowanie m -elementowego wektora natężeń ruchu na podstawie ciągu $\delta + 1$ macierzy OD. Parametr δ określa ilość poprzednich przedziałów czasowych, dla których estymowana jest dynamiczna macierz przepływu. Tym modelem jest sieć neuronowa składająca się z warstw konwolucyjnych i perceptronowych. Trening i testowanie sieci CNN (rysunki 4.1 i 4.2) odbywały się na zbiorze danych apriorycznych. Trening wraz z testowaniem modelu CNN jest opisany w rozdziale 6.

Dynamiczna macierz przepływów(OD)

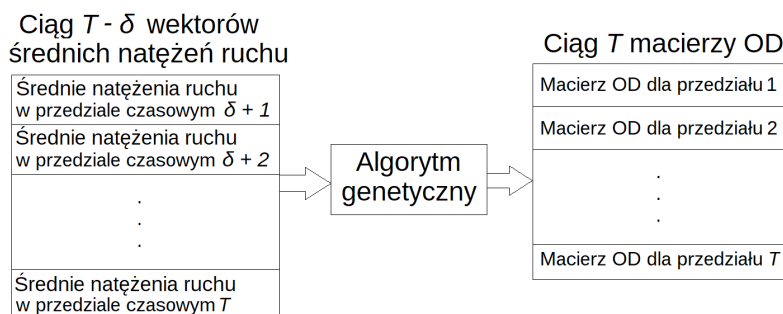


Rysunek 4.1: Wejściem sieci jest dynamiczna macierz przepływu, która opisuje ruch w przedziale czasowym t i w δ poprzednich przedziałach czasowych. Na wyjściu sieci jest m -elementowy wektor napięć ruchu w przedziale czasowym t .

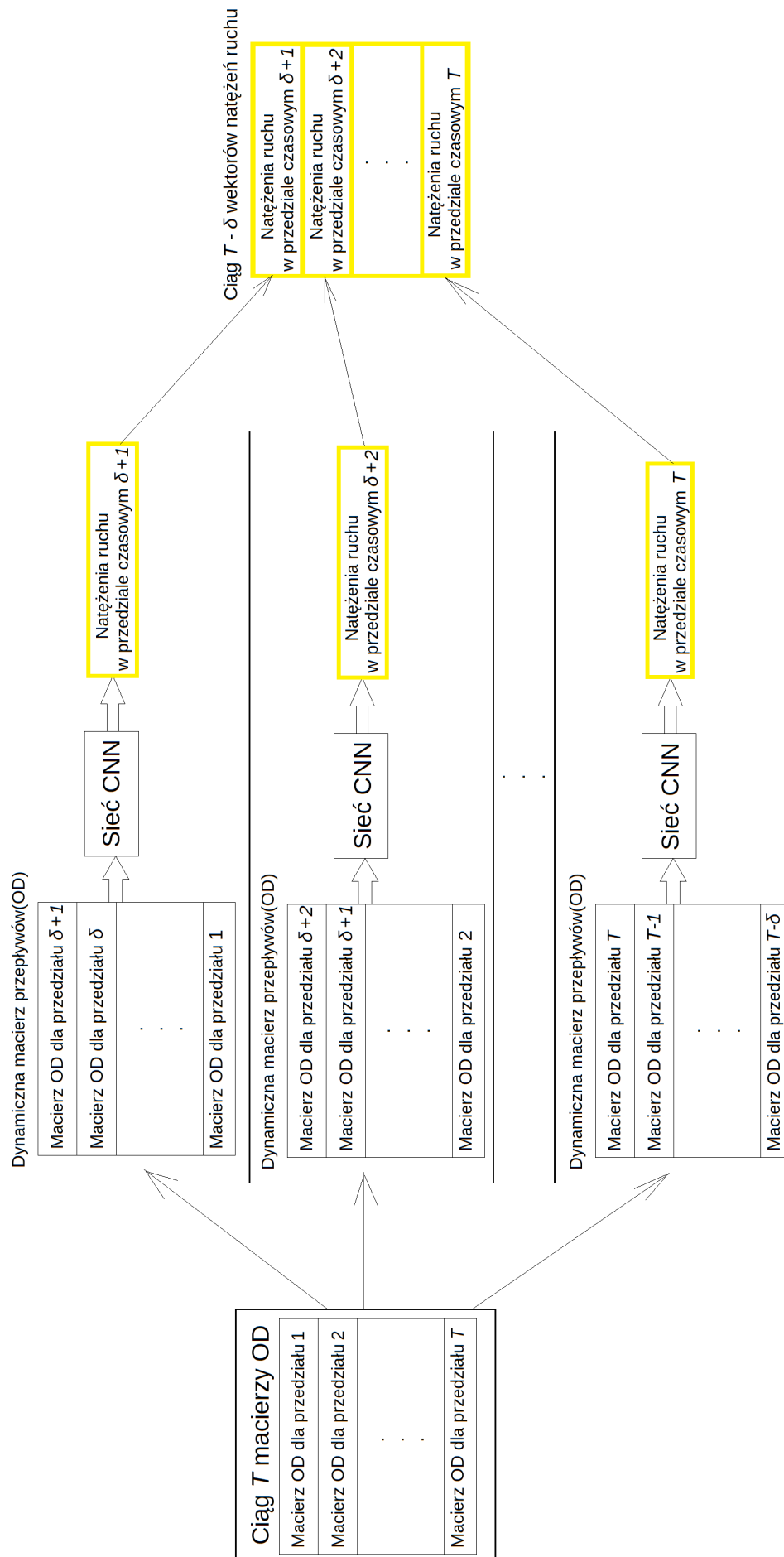


Rysunek 4.2: Struktura sieci konwulcyjnej z pracy [24]. Sieć składa się z $k + 1$ warstw konwulcyjnych, $k + 1$ warstw wykonujących operację *max pooling*'u oraz jednej warstwy perceptronowej.

Kolejnym krokiem algorytmu było poszukiwanie algorytmem genetycznym wzorcowego ciągu T macierzy OD na podstawie ciągu $T - \delta$ wektorów uśrednionych napięć ruchu (rysunek 4.3). Algorytm genetyczny szukał takiego wzorcowego ciągu macierzy OD, dla którego wartość funkcji przystosowania była jak największa. Przy obliczaniu wartości funkcji przystosowania (wzór opisujący funkcję znajduje się w rozdziale 7.) algorytm genetyczny korzystał z danych z przeszłości (*Historical Traffic Data*), w których znajdowały się ciągi wektorów napięć ruchu. Ciąg T macierzy OD był oceniany poprzez porównanie wyjścia sieci CNN (zaznaczonego kolorem żółtym na rysunku 4.4) z ciągiem wektorów uśrednionych napięć ze zbioru danych. Działanie algorytmu genetycznego wraz ze sposobem oceny ciągu T macierzy OD jest szczegółowo opisane w rozdziale 7.

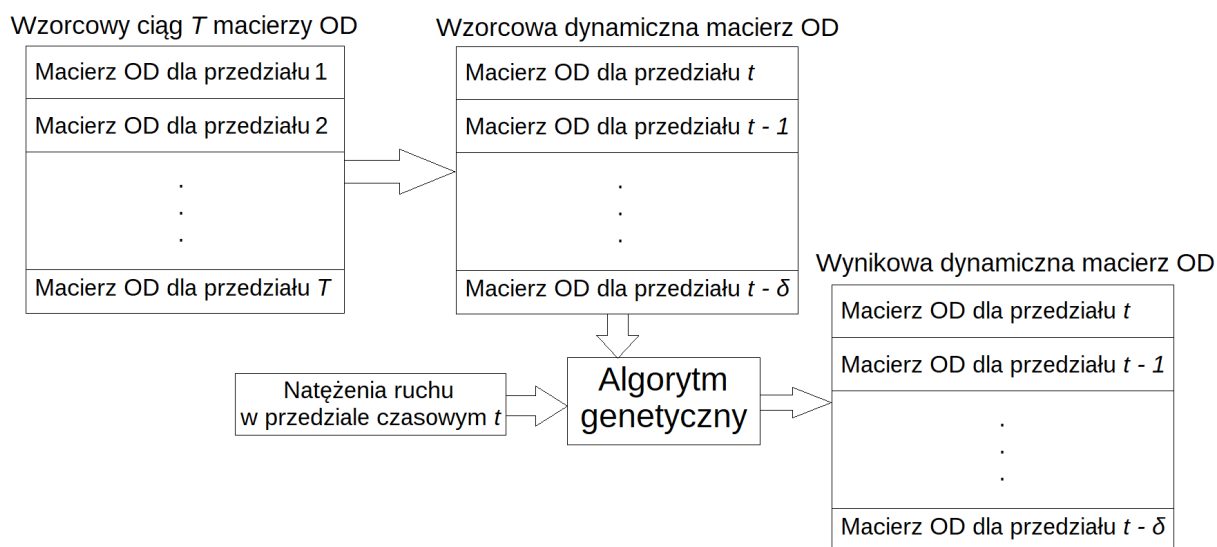


Rysunek 4.3: Na wejściu algorytmu genetycznego jest ciąg $T - \delta$ wektorów uśrednionych napięć ruchu, natomiast na wyjściu jest ciąg T macierzy OD.



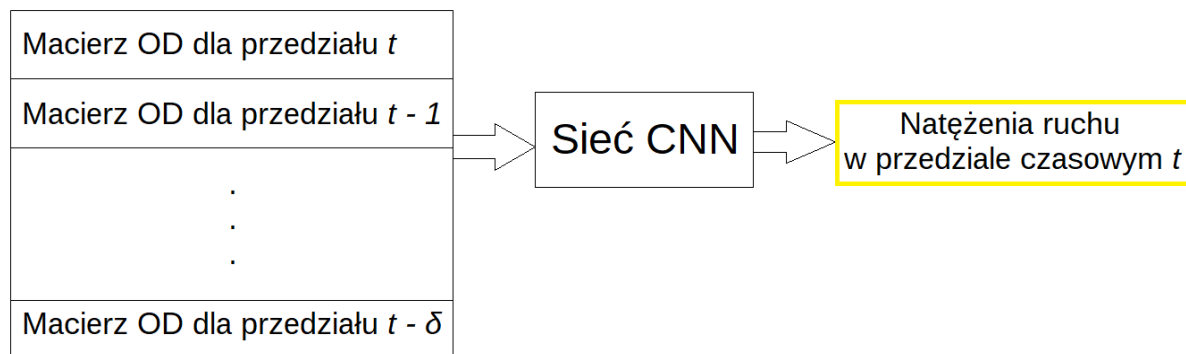
Rysunek 4.4: Z ciągu T macierzy OD są tworzone $T - \delta$ dynamiczne macierze OD. Po podaniu odpowiednich ciągów macierzy OD na wejście sieci CNN, otrzymano ciąg $T - \delta$ wektorów natężeń ruchu. Na podstawie ciągu $T - \delta$ wektorów natężeń ruchu oraz danych z przeszłości obliczana była funkcja przystosowania ciągu T macierzy OD.

Ostatnim etapem algorytmu było poszukiwanie (rysunek 4.5) algorytmem genetycznym dynamicznej macierzy OD, która była najlepiej dopasowana do danych w czasie rzeczywistym (*Real-time Traffic Data*). Populacja początkowa była określana na podstawie wzorcowej macierzy przepływów, która była wybierana z wzorcowego ciągu T macierzy OD. Algorytm genetyczny poszukiwał takiej dynamicznej macierzy OD, dla której wartość funkcji przystosowania (wzór opisujący funkcję znajduje się w rozdziale 7.) była jak największa. Przy obliczaniu wartości funkcji przystosowania algorytm genetyczny korzystał z danych w czasie rzeczywistym, w których znajdowały się wektory natężeń ruchu. Dynamiczna macierz przepływu była oceniana poprzez porównanie wyjścia sieci CNN (zaznaczonego kolorem żółtym na rysunku 4.6) z wektorem natężeń ze zbioru danych. Działanie algorytmu genetycznego wraz ze sposobem oceny dynamicznej macierzy OD jest szczegółowo opisane w rozdziale 7.



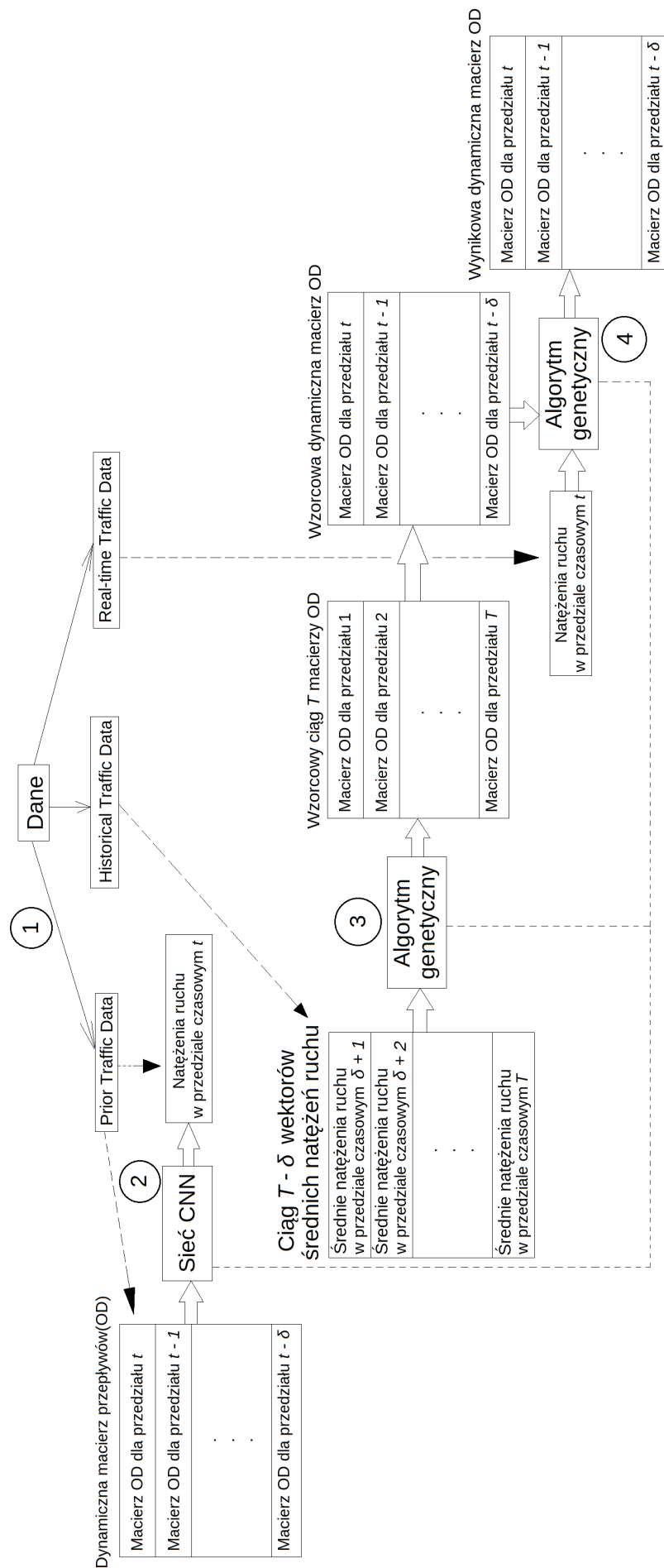
Rysunek 4.5: Z uzyskanego w poprzednim etapie wzorcowego ciągu T macierzy OD wybierana jest macierz przepływu dla przedziału t oraz δ macierzy OD z poprzednich przedziałów czasowych. Na wejściu algorytmu genetycznego jest wektor natężenia ruchu w przedziale czasowym t , natomiast na wyjściu jest ciąg $\delta + 1$ macierzy OD (wynikowa dynamiczna macierz przepływu). Populacja początkowa algorytmu genetycznego jest określana na podstawie wzorcowej dynamicznej macierzy OD.

Dynamiczna macierz przepływów(OD)



Rysunek 4.6: Po podaniu dynamicznej macierzy OD na wejście sieci CNN, otrzymano wektor natężeń ruchu w przedziale czasowym t . Na podstawie danych w czasie rzeczywistym (*Real-time Traffic Data*) i otrzymanego wektora natężeń obliczana była funkcja przystosowania dynamicznej macierzy OD.

Algorytm rozwiązania problemu składał się z czterech kroków: przygotowanie danych, trening modelu CNN, poszukiwanie algorytmem genetycznym wzorcowego ciągu T macierzy OD oraz poszukiwanie algorytmem genetycznym dynamicznej macierzy OD. Najpierw wygenerowano dane i podzielono je na trzy zbiory, które były używane w trzech kolejnych etapach. Następnie sieć CNN była trenowana i testowana na danych apriorycznych (*Prior Traffic Data*). Potem algorytm genetyczny poszukiwał wzorcowego ciągu T macierzy OD na podstawie danych historycznych (*Historical Traffic Data*). Przy obliczaniu funkcji przystosowania używana była wytrenowana wcześniej sieć CNN. Na koniec algorytm genetyczny poszukiwał dynamicznej macierzy OD na podstawie danych w czasie rzeczywistym. Funkcja przystosowania była obliczana przy użyciu sieci CNN. Populacja początkowa była określana na podstawie wcześniej obliczonej wzorcowej dynamicznej macierzy OD. Schemat działania algorytmu przedstawiono na rysunku 4.7.



Rysunek 4.7: Schemat przedstawiający działanie algorytmu rozwiązyującego problem estymacji dynamicznej macierzy przepływów.

Rozdział 5

Generowanie danych

Ciąg macierzy OD, opisujący dynamikę ruchu drogowego na ul Legnickiej, został wygenerowany na podstawie macierzy OD z pracy [29]. Ta macierz została obliczona przy pomocy algorytmu gradientowego[27] na podstawie danych o natężeniu ruchu we Wrocławiu pochodzących z repozytorium systemu ITS (*Intelligent Transportation System*). Obliczona macierz przepływu opisuje ruch na ulicy Legnickiej od godziny 6:00 do godziny 12:00. Założono podział tego przedziału czasowego na 24 ($T = 24$) mniejsze przedziały. Każdy przedział obejmował $d_T = 15$ minut czasu. Skoro przedział został podzielony na T , to również podzielono wartości macierzy OD z pracy przez T . Otrzymana macierz przepływu była macierzą bazową. Bazowy ciąg macierzy OD był ciągiem T macierzy bazowych. Na podstawie ciągu bazowego losowano z rozkładu jednostajnego ciągi T macierzy OD. Dane aprioryczne były generowane poprzez losowanie z zakresu od $\tau_{amin} = 0.75$ bazowego ciągu macierzy OD do $\tau_{amax} = 1.25$ bazowego ciągu macierzy OD. Natomiast dane historyczne oraz dane w czasie rzeczywistym generowane były poprzez losowanie z zakresu $\tau_{rmin} = 0.9$ do $\tau_{rmin} = 1.1$ bazowego ciągu macierzy OD. Kolejne etapy generowania ciągu macierzy OD przedstawiono na rysunkach 5.1, 5.2, 5.3, 5.4 i 5.5.

0	1301	...
1044	0	...
...

Rysunek 5.1: Pierwotna macierz przepływu opisująca ruch na ulicy Legnickiej we Wrocławiu od 6:00 do 12:00 (przedział obejmujący 6h). Pierwotna macierz przepływu dla ul. Legnickiej została obliczona algorytmem gradientowym w pracy [29]. Macierz ma wymiary n na n ($n = 15$).

6:00 - 6:15			6:15 - 6:30			...			11:45 - 12:00		
0	54	...	0	54	...				0	54	...
43	0	...	43	0	...				43	0	...
...

Rysunek 5.2: Bazowy ciąg macierzy OD składa się z T bazowych macierzy OD. Bazowa macierz przepływu, opisująca ruch w 15-minutowym przedziale czasowym, powstała poprzez podzielenie wartości pierwotnej macierzy OD przez T . Każda macierz ciągu ma wymiary n na n ($n = 15$).

6:00 - 6:15			6:15 - 6:30			11:45 - 12:00		
0	40	...	0	40	...	0	40	...
32	0	...	32	0	...	32	0	...
...

Rysunek 5.3: Ciąg macierzy OD zawierających minimalne wartości zakresu z którego losowany był ciąg macierzy OD. Minimalny ciąg macierzy OD otrzymano poprzez pomnożenie bazowego ciągu macierzy przez τ_{amin} . Każda macierz ciągu ma wymiary n na n ($n = 15$).

6:00 - 6:15			6:15 - 6:30			11:45 - 12:00		
0	67	...	0	67	...	0	67	...
53	0	...	53	0	...	53	0	...
...

Rysunek 5.4: Ciąg macierzy OD zawierających maksymalne wartości zakresu z którego losowany był ciąg macierzy OD. Maksymalny ciąg macierzy OD otrzymano poprzez pomnożenie bazowego ciągu macierzy przez τ_{amax} . Każda macierz ciągu ma wymiary n na n ($n = 15$).

6:00 - 6:15			6:15 - 6:30			11:45 - 12:00		
0	47	...	0	41	...	0	65	...
39	0	...	51	0	...	33	0	...
...

Rysunek 5.5: Losowy ciąg T macierzy OD. Każda macierz przepływu tego ciągu jest losowana według rozkładu jednostajnego na przedziale określonym przez minimalny ciąg macierzy OD i maksymalny ciąg macierzy OD. Każda macierz ciągu ma wymiary n na n ($n = 15$).

Ciąg macierzy OD, opisujący ruch w części miasta Sioux Falls, został wygenerowany na podstawie ciągu macierzy OD ze strony [32]. Ciąg czterech macierzy OD ($T = 4$) opisywał ruch w Sioux Falls w przedziale czasowym obejmującym 2 godziny i 40 minut. Każda z macierzy w tym ciągu opisywała ruch w przedziale czasu obejmującym 40 minut. Ciąg macierzy OD był bazowym ciągiem macierzy OD, na podstawie którego losowano z rozkładu jednostajnego ciągu T macierzy OD. Dane aprioryczne były generowane poprzez losowanie z zakresu od $\tau_{amin} = 0.75$ do $\tau_{amax} = 1.25$ bazowego ciągu macierzy OD. Natomiast dane historyczne oraz dane w czasie rzeczywistym generowane były poprzez losowanie z zakresu $\tau_{rmin} = 0.9$ do $\tau_{rmax} = 1.1$ bazowego ciągu macierzy OD. Kolejne etapy generowania ciągu macierzy OD przedstawiono na rysunkach 5.6, 5.7, 5.8 i 5.9.

7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	10	...	0	12	...	0	15	...	0	7	...
10	0	...	12	0	...	15	0	...	7	0	...
...

Rysunek 5.6: Bazowy ciąg T macierzy OD opisujący ruch w przedziale czasowym obejmującym 2h 40 min. Każda macierz w ciągu opisuje ruch w Sioux Falls odbywający się w 40-minutowym przedziale czasowym. Bazowy ciąg macierzy OD został pobrany ze strony [32]. Każda macierz ciągu ma wymiary n na n ($n = 24$).

7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	7	...	0	9	...	0	11	...	0	5	...
7	0	...	9	0	...	11	0	...	5	0	...
...

Rysunek 5.7: Ciąg macierzy OD zawierających minimalne wartości zakresu z którego losowany był ciąg macierzy OD. Minimalny ciąg macierzy OD otrzymano poprzez pomnożenie bazowego ciągu macierzy przez $\tau_{amin} = 0.75$. Każda macierz ciągu ma wymiary n na n ($n = 24$).

7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	12	...	0	15	...	0	18	...	0	8	...
12	0	...	15	0	...	18	0	...	8	0	...
...

Rysunek 5.8: Ciąg macierzy OD zawierających maksymalne wartości zakresu z którego losowany był ciąg macierzy OD. Maksymalny ciąg macierzy OD otrzymano poprzez pomnożenie bazowego ciągu macierzy przez $\tau_{amax} = 1.25$. Każda macierz ciągu ma wymiary n na n ($n = 24$).

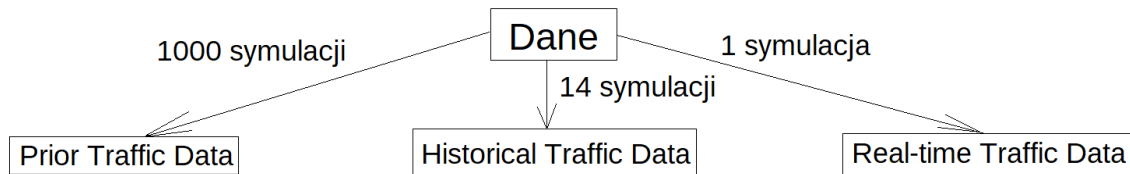
7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	9	...	0	13	...	0	12	...	0	7	...
11	0	...	15	0	...	11	0	...	8	0	...
...

Rysunek 5.9: Losowy ciąg T macierzy OD. Ciąg macierzy OD był losowany według rozkładu jednostajnego na przedziale określonym przez minimalny ciąg macierzy OD i maksymalny ciąg macierzy OD. Każda macierz ciągu ma wymiary n na n ($n = 24$).

Po wygenerowaniu ciągu T macierzy OD, podawano ten ciąg na wejście symulatora SUMO. W symulacji były ustawione detektory, które zliczały liczbę pojazdów przejeżdżających pomiędzy sąsiednimi węzłami w poszczególnych przedziałach czasowych. Na podstawie danych z detektorów powstał ciąg wektorów natężeń ruchu. Dla ulicy Legnickiej we Wrocławiu oraz dla miasta Sioux Falls na podstawie ciągu T macierzy OD uzyskiwany jest ciąg T wektorów natężeń ruchu. Ciąg macierzy OD oraz ciąg wektorów natężeń ruchu

są zapisywane do osobnych plików. W wyniku jednej symulacji uzyskiwano jedną parę (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu).

Dla ul. Legnickiej oraz Sioux Falls wygenerowano (rysunek 5.10) po 3 zbiory danych (dane aprioryczne, dane historyczne i dane w czasie rzeczywistym). W zbiorze danych apriorycznych znajdowało się $N_{asym} = 1000$ par, w zbiorze danych historycznych znajdowało się $N_{hsym} = 14$ par, natomiast w zbiorze danych w czasie rzeczywistym znajdowała się jedna para (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu). Ilości par są takie same jak w pracy [24]. W zbiorze apriorycznym potrzeba było znacznie większej ilości par niż w pozostałych zbiorach, ponieważ ten zbiór użyto do trenowania sieci konwolucyjnej. W celu wygenerowania wszystkich zbiorów danych dla ul. Legnickiej, symulowano $N_{asym} + N_{hsym} + 1 = 1015$ razy 6h ruchu na ul. Legnickiej. W celu wygenerowania wszystkich zbiorów danych dla Sioux Falls, symulowano $N_{asym} + N_{hsym} + 1 = 1015$ razy 2h 40 min ruchu w Sioux Falls.

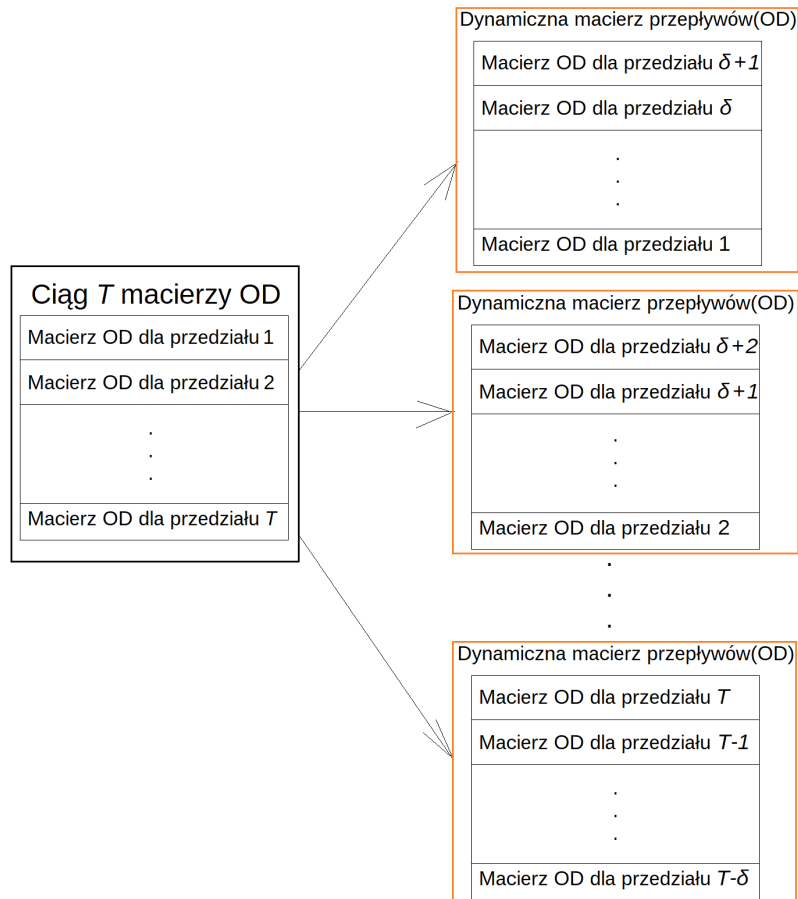


Rysunek 5.10: Wygenerowane zbiory danych.

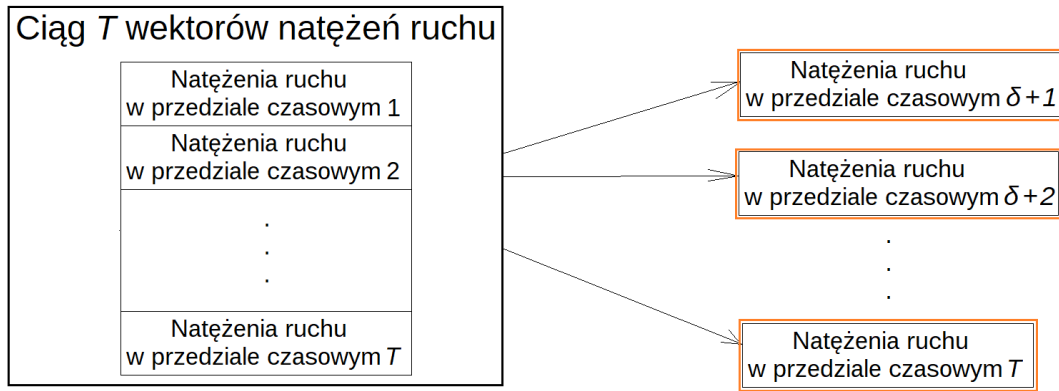
Rozdział 6

Model naśladowujący symulator SUMO

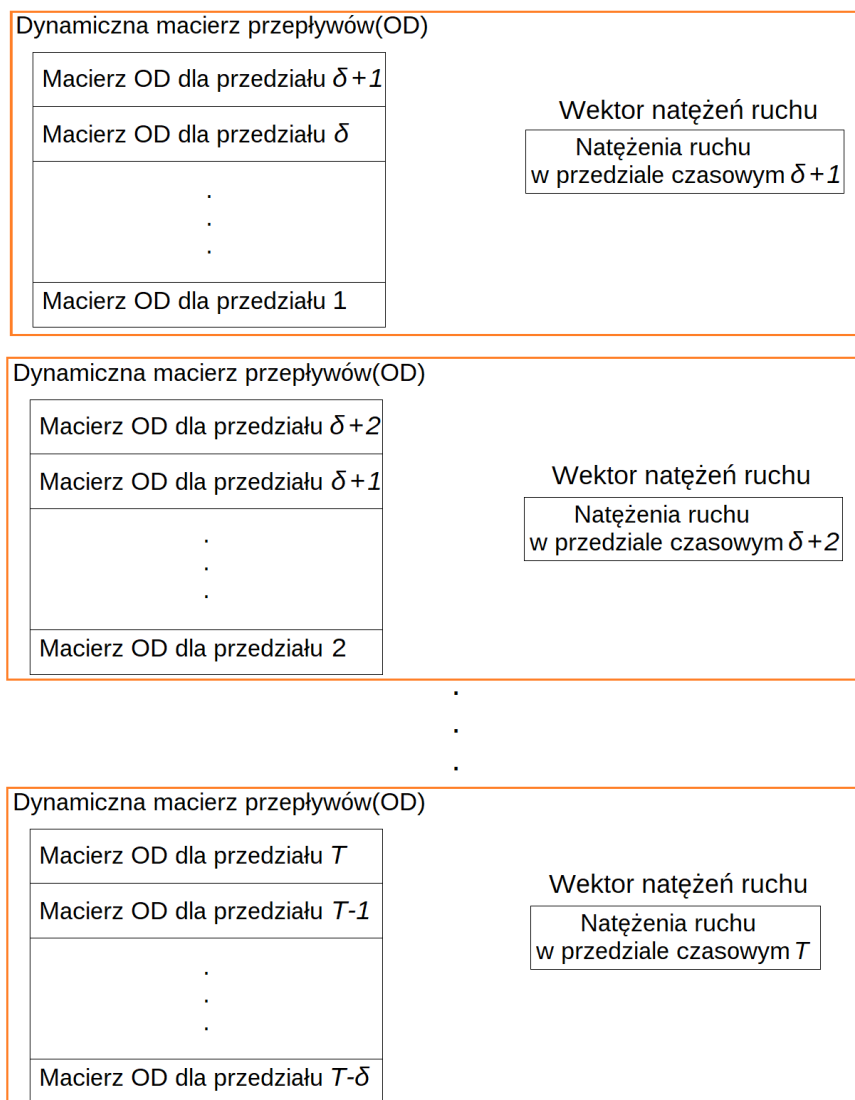
Przed rozpoczęciem treningu sieci neuronowej odpowiednio przetworzono zbiór danych apriorycznych. Każda para (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu) została przetworzona w taki sam sposób. Z ciągu macierzy OD były tworzone $T - \delta$ dynamiczne macierze OD, będące ciągami $\delta + 1$ macierzy OD. Następnie utworzono $T - \delta$ par (dynamiczna macierz przepływu, wektor natężeń ruchu). Sposób przetworzenia pary (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu) na $T - \delta$ par (dynamiczna macierz przepływu, wektor natężeń ruchu) przedstawiono na rysunkach 6.1, 6.2, 6.3.



Rysunek 6.1: Przekształcenie ciągu T macierzy OD na $T - \delta$ dynamicznych macierzy OD. Każda dynamiczna macierz przepływu opisuje ruch pojazdów w przedziale czasowym t oraz w δ poprzednich przedziałach czasowych. Po przekształceniu uzyskano dynamiczne macierze OD dla $t = \{\delta + 1, \delta + 2, \dots, T\}$.



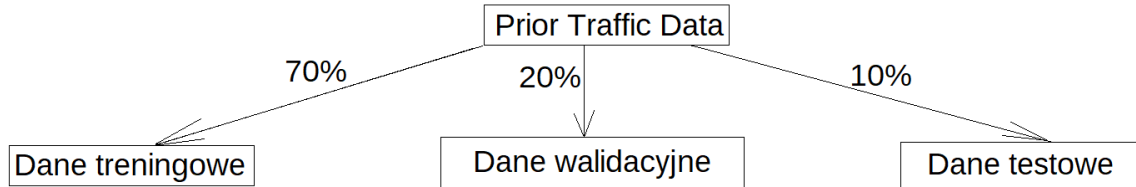
Rysunek 6.2: Wybranie z ciągu T wektorów OD $T - \delta$ wektorów natężeń ruchu, zaczynając od wektora natężeń ruchu w przedziale $T - \delta$. W ten sposób uzyskano natężenia ruchu dla $t = \{\delta + 1, \delta + 2, \dots, T\}$.



Rysunek 6.3: $T - \delta$ par (dynamiczna macierz przepływu, wektor natężeń ruchu).

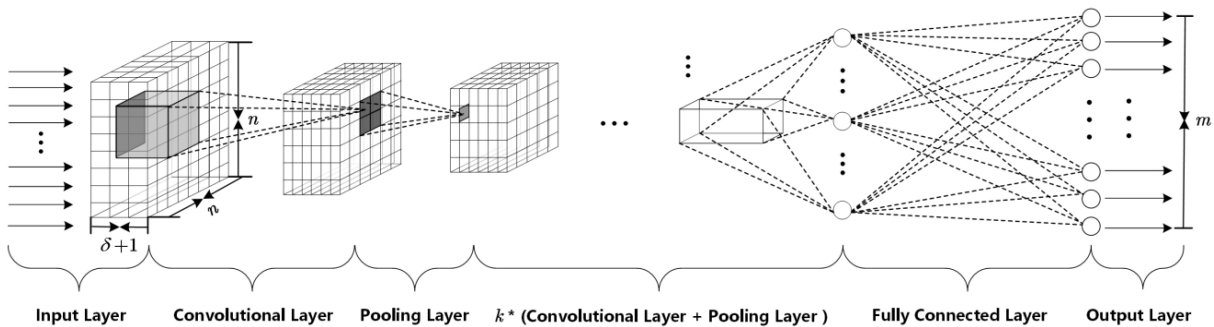
W zbiorze danych apriorycznych było 1000 par (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu). Każda para była przekształcana na $T - \delta$ par (dynamiczna macierz

przepływu, wektor natężeń ruchu). Zatem zbiór danych używanych do treningu i testowania sieci CNN składał się z $1000 * (T - \delta)$ par (dynamiczna macierz przepływu, wektor natężeń ruchu). Zadaniem modelu było obliczenie wektora natężeń ruchu na podstawie dynamicznej macierzy OD. Zbiór danych podzielono na 3 części (rysunek 6.4): zbiór treningowy, zbiór walidacyjny oraz zbiór testowy. 70% zbioru danych ($0.70 * 1000 * (T - \delta)$ par) stanowił zbiór treningowy, 20% zbioru danych ($0.20 * 1000 * (T - \delta)$ par) stanowił zbiór walidacyjny, natomiast 10% zbioru danych ($0.10 * 1000 * (T - \delta)$ par) stanowił zbiór testowy.



Rysunek 6.4: Podział danych na zbiór treningowy, walidacyjny i testowy.

Wejściem sieci była dynamiczna macierz przepływu, która miała wymiary n na n na $(\delta + 1)$ (n - liczba węzłów badanej sieci miejskiej, δ - ilość poprzednich przedziałów czasowych uwzględnionych w dynamicznej macierzy OD)[24]. Natomiast wyjściem sieci był m -elementowy wektor natężeń ruchu. Sieć konwolucyjna (rysunek 6.5) składała się z trzech warstw konwolucyjnych, trzech warstw wykonujących operację *max pooling*'u oraz jednej warstwy perceptronowej.



Rysunek 6.5: Struktura sieci konwolucyjnej.

Każda warstwa konwolucyjna miała 2 filtry, których jądro miało wymiary 3 na 3. Przy filtracji zastosowano wypełnienie zerami na brzegach (tzw. *zero padding*). Funkcją aktywacji w warstwach konwolucyjnych była funkcja rampowa (*rectified linear unit (ReLU)*). Wagi sieci konwolucyjnej były inicjalizowane poprzez losowanie z rozkładu Gaussa. Średnia rozkładu wynosiła 0, natomiast odchylenie standardowe $\sqrt{\frac{2}{n_l}}$ (n_l - liczba połączeń od poprzedniej warstwy). Wyrazy wolne (*biases*) były inicjalizowane zerami. Liczba połączeń od poprzedniej warstwy wyrażała się wzorem: $n_l = 2 * 3 * 3 * n^2(\delta + 1)$.

Każda warstwa wykonująca operację *max pooling*'u, filtrowała swoje wejście przy użyciu filtru maksymalnego. Maskę filtru maksymalnego miała wymiary 2 na 2.

Warstwa perceptronowa składa się z m neuronów, gdzie m jest ilością wyjść. Funkcją aktywacji warstwy perceptronowej była funkcja SAAF (*Smooth Adaptive Activation Function*). Funkcja SAAF jest wielomianem[24]. W ogólności funkcja SAAF wyraża się wzorem

$$f(x) = \sum_{c=0}^{\eta-1} \alpha_c p^c(x) + \sum_{l=0}^{\xi} \beta_l b_l^\eta(x)$$

, gdzie

$$p^c(x) = \frac{x^c}{c!},$$

$$b_l^{i+1}(x) = \int_0^x b_l^i(\gamma) d\gamma,$$

$$b_l^0(x) = I(a_l \leq x < a_{l+1}),$$

$$I(a_l \leq x < a_{l+1}) = \begin{cases} 1, & x \in [a_l, a_{l+1}] \\ 0, & x \notin [a_l, a_{l+1}] \end{cases} \quad \text{- funkcja charakterystyczna zbioru,}$$

$a_1, a_2, \dots, a_{\xi+1}$ - rosnący ciąg liczb rzeczywistych,

$$l \in \{1, 2, \dots, \xi\},$$

η - stopień wielomianu,

p^c i b_l^η - funkcje bazowe,

α_c i β_l - parametry wyznaczone podczas treningu sieci

Wartości $a_1, a_2, \dots, a_{\xi+1}$ zostały równomiernie rozłożone na przedziale $[-1.1, 1.1]$. W rozwiązaniu przyjęto $\eta = 2$ oraz $\xi = 25$.

Uczenie sieci polegało na dobraniu parametrów we wszystkich warstwach, tak aby średnia arytmetyczna wartości funkcji błędu była jak najmniejsza. Funkcją błędu była funkcja Hubera[7] opisana wzorem

$$L_\delta(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & |y_i - \hat{y}_i| \leq \delta \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2, & |y_i - \hat{y}_i| > \delta \end{cases}, \quad (6.1)$$

gdzie y_i jest obserwowaną wartością i -tego wyjścia ze zbioru treningowego, \hat{y}_i jest wartością i -tego wyjścia obliczonego przez sieć neuronową oraz δ jest parametrem. Minimalizacja średniej wartości funkcji Hubera jest lepszym rozwiązaniem niż minimalizacja błędu średniokwadratowego, ponieważ taka metoda nie wymaga założenia, że dane podlegają rozkładowi normalnemu. W rozwiązaniu przyjęto $\delta = 1$. Zatem podczas treningu minimalizowano wartość następującej funkcji $L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^N L(y_i, \hat{y}_i)$, gdzie N jest iloczynem rozmiaru wsadu (*batch size*) oraz m - liczby elementów wektora nateżeń ruchu. Do nauki sieci użyto optymalizatora ADAM (*Adaptive Moment Estimation* - adaptacyjna estymacja momentu). Algorytm aktualizuje wagi θ_s w kroku s według następujących reguł [24]:

$$\theta_s = \theta_{s-1} - \alpha * \frac{\omega_s}{(1-\beta_1^s)} / \left(\sqrt{\frac{v_s}{1-\beta_2^s}} + \epsilon \right)$$

$$\omega_s = \beta_1 \omega_{s-1} + (1 - \beta_1) g_s$$

$$v_s = \beta_2 v_{s-1} + (1 - \beta_2) g_s^2$$

$$g_s = \nabla_{\theta} L_s(\theta_{s-1})$$

g_s - gradient minimalizowanej funkcji względem wag θ w kroku s

ω_s - estymator pierwszego momentu (średnia) gradientu, $\omega_0 = 0$,

v_s - estymator drugiego momentu (średnia kwadratów) gradientu, $v_0 = 0$,

$\beta_1, \beta_2 \in [0, 1)$ - współczynniki wykładniczej szybkości zaniku dla estymatorów momentów, w rozwiązaniu ustawiono $\beta_1 = 0.9$, $\beta_2 = 0.999$,

α - współczynnik uczenia wag θ , w rozwiązaniu ustawiono $\alpha = 0.001$,

ϵ - mała stała, która jest dodawana w celu uniknięcia dzielenia przez 0, w rozwiązaniu jej wartość wynosi 10^{-7} .

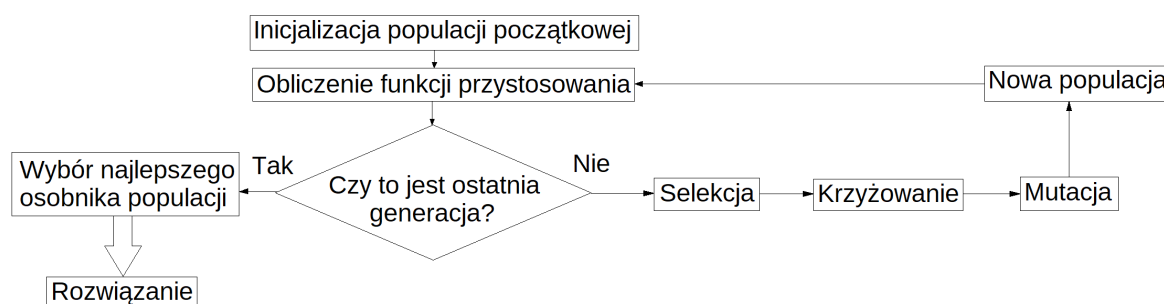
W każdym kroku następowała aktualizacja wag θ na podstawie pewnej ilości przykładów (par (dynamiczna macierz przepływu, wektor nateżeń ruchu)). Ilość przykładów była określona przez rozmiar wsadu (*batch size*), który wynosił 3000. Iteracja kończyła się, gdy przetworzono cały zbiór danych. Na podstawie zbioru walidacyjnego po każdej iteracji oceniana była jakość trenowanego modelu. W rozwiązaniu użyto mechanizmu *early stopping*, który polegał na zakończeniu treningu modelu w momencie gdy wartość minimalizowanej funkcji nie poprawiała się w ciągu 50 ostatnich iteracji.

Jakość wytrenowanego modelu była oceniana na podstawie średniego błędu bezwzględnego (MAE), średniego błędu bezwzględnego wyrażonego w procentach (MAPE) oraz wskaźnika determinacji R^2 . MAPE, MAE oraz R^2 wyrażają się następującymi wzorami: (3.1), (3.2) oraz (3.3).

Rozdział 7

Algorytmy genetyczne

Algorytm genetyczny poszukuje rozwiązania optymalnego za pomocą populacji osobników (rozwiązań) ([18], [1]). Osobnikiem jest możliwe rozwiązanie problemu, czyli np. w przypadku poszukiwania dynamicznej macierzy OD osobnikiem jest dowolna dynamiczna macierz przepływu. Na populacji wykonywane były następujące operacje: selekcja, krzyżowanie oraz mutacja. Selekcja decydowała o tym, które osobniki przetrwały do następnej generacji. Im lepsza wartość funkcji przystosowania, tym większa była szansa na przetrwanie osobnika (rozwiązania). Krzyżowanie oraz mutacja były operacjami niezależnymi od funkcji przystosowania rozwiązania. Algorytmy genetyczne były używane do poszukiwania wzorcowego ciągu macierzy OD dopasowanego do danych historycznych (krok 3. algorytmu [24] rozwiązującego problem estymacji dynamicznej macierzy OD) oraz do poszukiwania dynamicznej macierzy OD dopasowanej do danych w czasie rzeczywistym (krok 4. algorytmu). Działanie algorytmu genetycznego przedstawiono na rysunku 7.1.

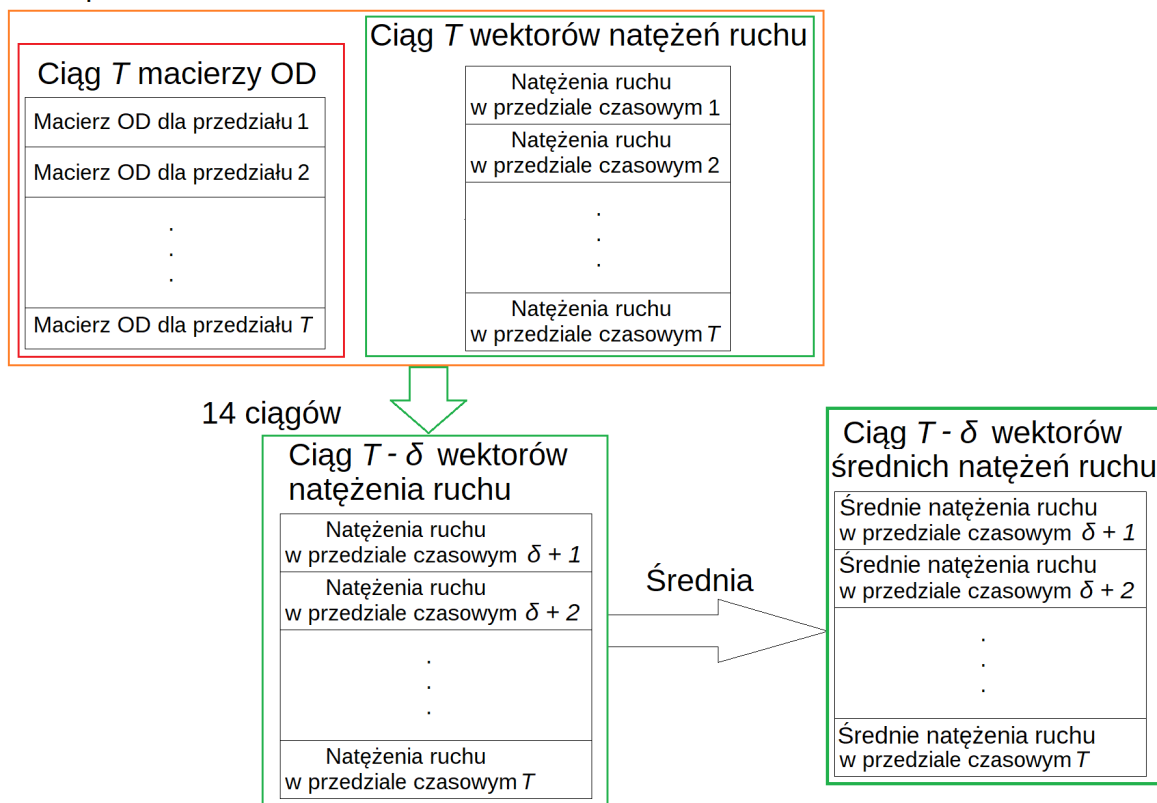


Rysunek 7.1: Schemat działania algorytmu genetycznego [24]

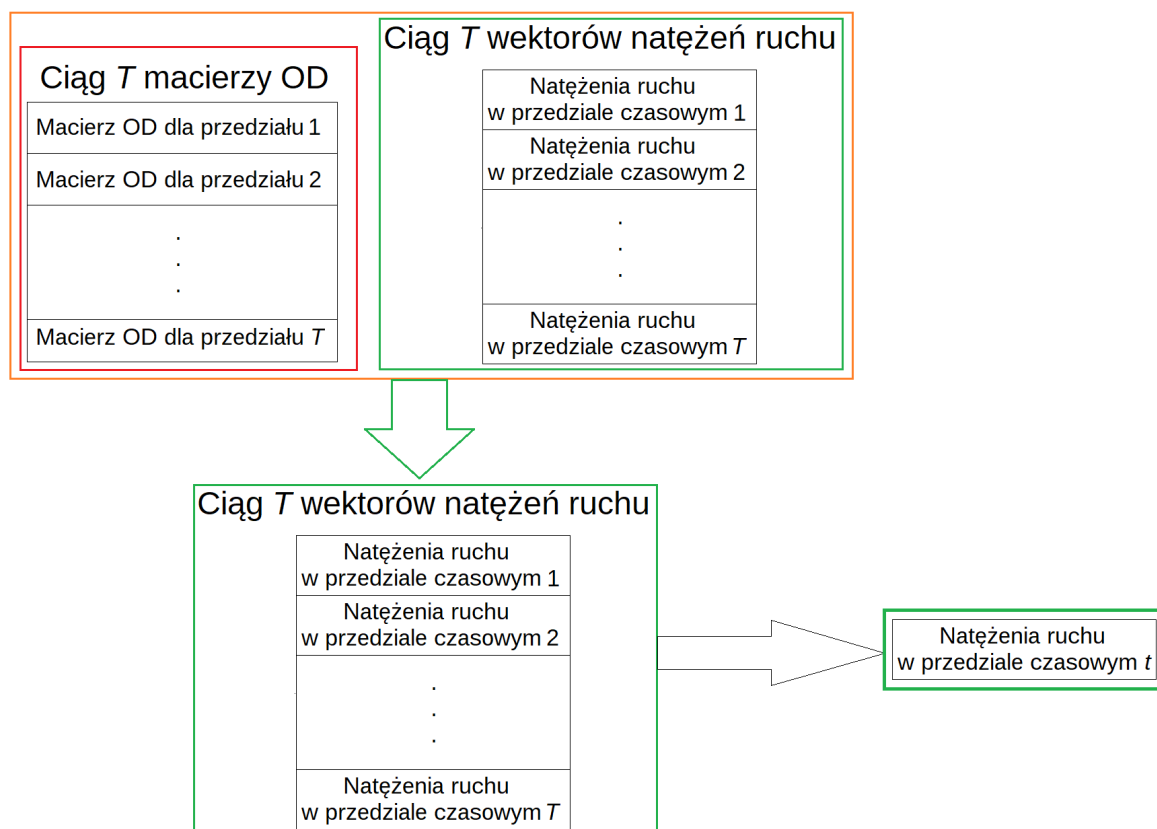
Przed uruchomieniem algorytmu genetycznego poszukującego wzorcowego ciągu macierzy OD, dane historyczne były odpowiednio przygotowane (rysunek 7.2). Dane historyczne zawierały 14 par (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu). Z każdego ciągu T wektorów wybrano ciąg $T - \delta$ wektorów. Otrzymano w ten sposób 14 ciągów $T - \delta$ wektorów natężeń ruchu. Średnia arytmetyczna 14 ciągów $T - \delta$ wektorów natężenia ruchu została użyta do obliczenia funkcji przystosowania osobników (ciągów T macierzy OD) w algorytmie genetycznym.

Analogicznie przed uruchomieniem algorytmu genetycznego poszukującego dynamicznej macierzy OD, dane w czasie rzeczywistym były odpowiednio przygotowane (rysunek 7.3). Dane w czasie rzeczywistym zawierały jedną parę (ciąg T macierzy OD, ciąg T wektorów natężeń ruchu). Wykorzystano tylko ciąg T wektorów natężeń ruchu. Z tego ciągu wybrano wektor natężeń ruchu w przedziale czasowym t . Ten wektor był użyty do oceny osobników (dynamicznych macierzy OD) w algorytmie genetycznym.

14 par



Rysunek 7.2: Przetworzenie danych historycznych.



Rysunek 7.3: Przetworzenie danych w czasie rzeczywistym.

Na 3. etapie (rysunek 4.7) rozwiązywania problemu estymacji dynamicznej macierzy OD, algorytm genetyczny poszukiwał wzorcowego ciągu T macierzy OD na podstawie przetworzonych danych historycznych. Pierwszym krokiem algorytmu genetycznego była inicjalizacja populacji początkowej $D = \{D_h | h \in \{1, 2, \dots, N\}\}$. N było rozmiarem populacji, który wynosił 800. Osobniki populacji początkowej w algorytmie poszukującym wzorcowego ciągu T dynamicznych macierzy OD były losowane ze zbioru możliwych rozwiązań R . Zbiór R był przestrzenią poszukiwań i był taki sam jak zbiór, z którego były losowane ciągi T macierzy OD w danych apriorycznych. Zbiór R był określany na podstawie minimalnego i maksymalnego ciągu T macierzy OD (rysunki 7.4 i 7.5). Przestrzeń poszukiwań wyrażała się wzorem[24]

$$R = \{\tau_c\}_{c=1}^{n*n*T} = \tau_1 \times \tau_2 \times \dots \times \tau_{n*n*T},$$

gdzie n było ilością węzłów, T było ilością przedziałów czasowych oraz τ_i jest przedziałem, określającym przestrzeń poszukiwań wartości i -tego elementu ciągu T macierzy przepływów. Zatem każdy osobnik (ciąg T macierzy OD) był przedstawiany jako wektor $n * n * T$ liczb całkowitych. Na przykład dla Sioux Falls zbiór R był następujący $R = [0, 0] \times [7, 12] \times \dots \times \tau_{n*n*T}$.

7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	7	...	0	9	...	0	11	...	0	5	...
7	0	...	9	0	...	11	0	...	5	0	...
...

Rysunek 7.4: Minimalny ciąg macierzy OD opisujący ruch w Sioux Falls.

7:00 - 7:40			7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	12	...	0	15	...	0	18	...	0	8	...
12	0	...	15	0	...	18	0	...	8	0	...
...

Rysunek 7.5: Maksymalny ciąg macierzy OD opisujący ruch w Sioux Falls.

Osobniki populacji były oceniane (sposób oceny przedstawiono na rysunku 7.6) za pomocą funkcji przystosowania, opisanej wzorem [24]

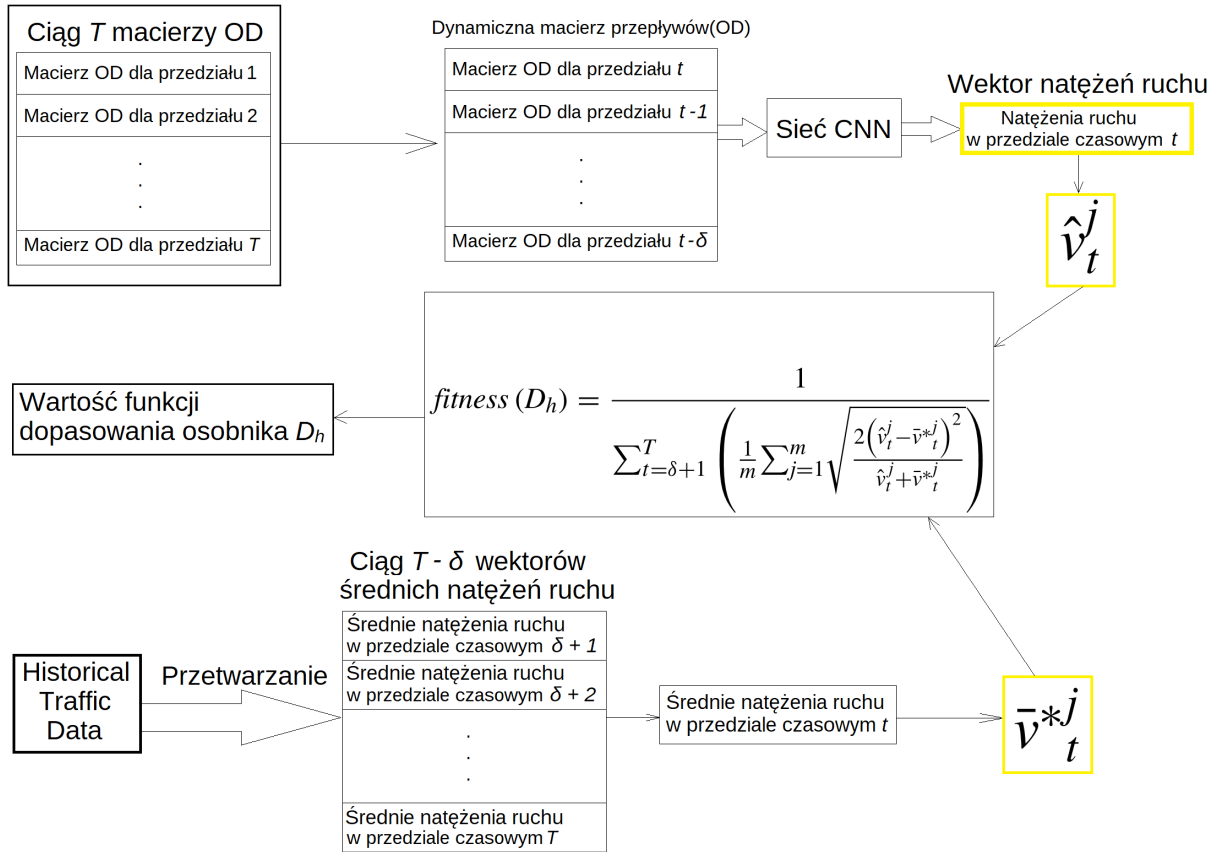
$$fitness(D_h) = \frac{1}{\sum_{t=\delta+1}^T \left(\frac{1}{m} \sum_{j=1}^m \sqrt{\frac{2(\hat{v}_t^j - \bar{v}_t^{*j})^2}{\hat{v}_t^j + \bar{v}_t^{*j}}} \right)},$$

gdzie

\hat{v}_t^j jest obliczonym przez sieć CNN j -tym elementem wektora natężeń ruchu w przedziale czasowym t ,

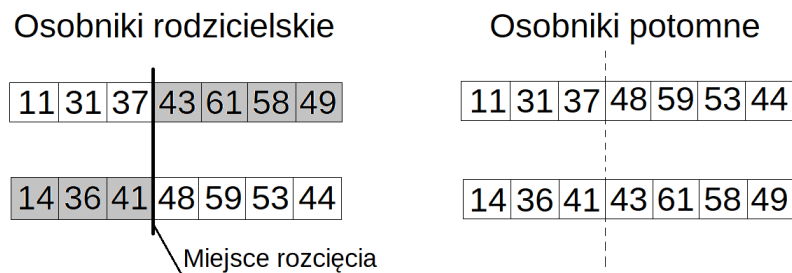
\bar{v}_t^{*j} jest j -tym elementem wektora średnich natężeń ruchu w przedziale czasowym t ,

m jest ilością obserwowanych natężeń ruchu.

Osobnik D_h populacji D **Rysunek 7.6:** Sposób oceny osobnika populacji, będącego ciągiem T macierzy OD [24]

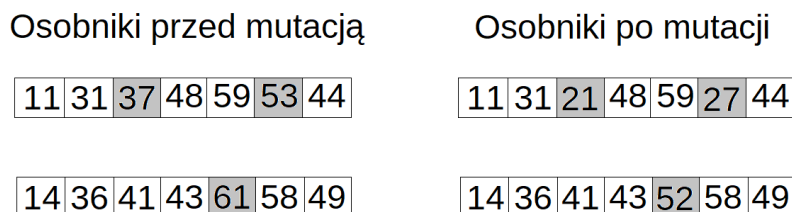
Po obliczeniu wartości funkcji przystosowania każdego osobnika populacji, następowała selekcja. Selekcja polegała na wylosowaniu N (rozmiar populacji) osobników spośród osobników całej populacji. Losowanie odbywało się ze zwracaniem, czyli ten sam osobnik mógł być wylosowany więcej niż raz. Prawdopodobieństwo $p(D_h)$ wylosowania każdego z osobników zależało od jego funkcji przystosowania i wyrażało się wzorem $p(D_h) = \frac{fitness(D_h)}{\sum_{i=1}^N fitness(D_i)}$ [24]. Taki sposób selekcji jest nazywany selekcją ruletkową [1].

Po selekcji następowało dobranie osobników nowej populacji w rozłączne pary ($N/2$ par) i podjęcie decyzji dla każdej pary o krzyżowaniu (rysunek 7.7). Jeśli decyzja była pozytywna, to następowało krzyżowanie, a powstałe osobniki potomne zastępowały swoich rodziców [1]. Jeśli decyzja była negatywna, osobniki pozostawały bez zmian. Prawdopodobieństwo krzyżowania μ było parametrem algorytmu. Operacja krzyżowania polegała na wyborze miejsca rozcięcia wektorów i zamianie odpowiednich fragmentów wektorów. Miejsce rozcięcia było wybierane losowo z rozkładem równomiernym.



Rysunek 7.7: Przykład operacji krzyżowania pary wektorów. Pierwszy osobnik potomny składa się z pierwszego fragmentu pierwszego rodzica i z drugiego fragmentu drugiego rodzica. Drugi osobnik potomny składa się z pierwszego fragmentu drugiego rodzica i drugiego fragmentu pierwszego rodzica. Rysunek wykonano na podstawie [1].

Mutacja polegała na zmianie wartości elementu c wektora (osobnika D_h) na wartość wylosowaną z przedziału τ_c . Dla każdego elementu c rozwiązania była podejmowana decyzja o mutacji. Jeśli decyzja była pozytywna, wartość elementu c była zmieniana na wartość losową. Jeśli decyzja była negatywna, wartość elementu c nie zmieniała się [24]. Prawdopodobieństwo mutacji λ było parametrem algorytmu. Na rysunku 7.8 przedstawiono przykład operacji mutacji przeprowadzonej na dwóch osobnikach.



Rysunek 7.8: Przykład operacji mutacji populacji składającej się z dwóch osobników. Każdy element wektorów był mutowany z prawdopodobieństwem $\lambda = 1/6$. Mutacji podlegały losowo wybrane elementy (3. i 6. element wektora pierwszego oraz 5. element wektora drugiego). Losowane były liczby całkowite z przedziałów: $\tau_3 = [20, 40]$, $\tau_5 = [50, 70]$, $\tau_6 = [20, 60]$.

Po wykonaniu selekcji, krzyżowania i mutacji otrzymywano nową populację. Każda nowa populacja jest nazywana generacją [1]. Jeśli osiągnięto maksymalną ilość generacji, to algorytm kończył działanie. Algorytm zwracał najlepsze (pod względem funkcji przystosowania) rozwiązanie z ostatniej generacji. Po zakończeniu działania algorytmu otrzymano wzorcowy ciąg T macierzy OD (rysunek 7.9).

Ciąg T - δ wektorów średnich natężeń ruchu

Średnie natężenia ruchu w przedziale czasowym $\delta + 1$
Średnie natężenia ruchu w przedziale czasowym $\delta + 2$
⋮
Średnie natężenia ruchu w przedziale czasowym T

Algorytm
genetyczny

Ciąg T macierzy OD

Macierz OD dla przedziału 1
Macierz OD dla przedziału 2
⋮
Macierz OD dla przedziału T

Rysunek 7.9: Algorytm genetyczny na podstawie ciągu T – δ wektorów uśrednionych natężeń ruchu znalazł wzorcowy ciąg T macierzy OD.

W ostatnim (krok 4.) etapie algorytmu (rysunek 4.7), rozwiązującego problem estymacji dynamicznej macierzy OD, poszukiwano algorytmem genetycznym dynamiczną macierz przepływu dopasowaną do natężeń ruchu w przedziale czasowym t . Do sprawdzania dopasowania użyto danych w czasie rzeczywistym (*Real-time Traffic Data*). Populacja początkowa była losowana na podstawie wzorcowego ciągu T macierzy OD. Z wzorcowego ciągu T macierzy OD wybrano (rysunek 7.10) odpowiedni podciąg $\delta + 1$ macierzy OD, będący wzorcową dynamiczną macierzą OD. Wzorcowa dynamiczna macierz przepływu składała się z macierzy OD dla przedziału t oraz macierzy OD dla δ poprzednich przedziałów czasowych. Na podstawie wzorcowej dynamicznej macierzy OD została wyznaczona (rysunki 7.11 i 7.12) maksymalna dynamiczna macierz przepływu (wartości wzorcowej dynamicznej macierzy OD pomnożone przez 1.5) oraz minimalna dynamiczna macierz przepływu (wartości wzorcowej dynamicznej macierzy OD pomnożone przez 0.5). Z rozkładu jednostajnego zostały wylosowane (rysunek 7.13) osobniki populacji D . Przedział, z którego losowano osobniki, był określony przez minimalną i maksymalną dynamiczną macierz przepływu. Przestrzeń poszukiwań R była taka sama jak zbiór z którego losowana była populacja początkowa. Przestrzeń poszukiwań opisywał wzór[24]

$$R = \{\tau_c\}_{c=1}^{n*n*(\delta+1)} = \tau_1 \times \tau_2 \times \dots \times \tau_{n*n*(\delta+1)},$$

gdzie n jest ilością węzłów, δ jest ilością poprzednich przedziałów czasowych uwzględnionych w dynamicznej macierzy OD oraz τ_i jest przedziałem, określającym przestrzeń poszukiwań wartości i -tego elementu dynamicznej macierzy przepływów. Zatem każdy osobnik (dynamiczna macierz przepływu) był przedstawiany jako wektor $n*n*(\delta+1)$ liczb całkowitych. Na przykład dla Sioux Falls zbiór R był następujący $R = [0, 0] \times [1, 3] \times \dots \times \tau_{n*n*(\delta+1)}$.

Wzorcowa dynamiczna macierz OD

7:00 - 7:40

0	1	...
1	0	...
...

7:40 - 8:20

0	2	...
2	0	...
...

8:20 - 9:00

0	2	...
3	0	...
...

9:00 - 9:40

0	0	...
1	0	...
...

Rysunek 7.10: Wzorcowy ciąg T macierzy OD dla Sioux Falls wraz z zaznaczoną wzorcową dynamiczną macierzą OD dla $t = 4$ i $\delta = 2$. Każda macierz ciągu ma wymiary 24 na 24.

7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	1	...	0	1	...	0	0	...
1	0	...	1	0	...	0	0	...
...

Rysunek 7.11: Minimalna dynamiczna macierz przepływu. Każda macierz ciągu ma wymiary 24 na 24.

7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	3	...	0	3	...	0	0	...
3	0	...	4	0	...	1	0	...
...

Rysunek 7.12: Maksymalna dynamiczna macierz przepływu. Każda macierz ciągu ma wymiary 24 na 24.

7:40 - 8:20			8:20 - 9:00			9:00 - 9:40		
0	3	...	0	2	...	0	0	...
2	0	...	4	0	...	0	0	...
...

Rysunek 7.13: Wylosowany osobnik D_h populacji D , będący dynamiczną macierzą OD.

Osobniki populacji były oceniane (sposób oceny przedstawiono na rysunku 7.14) za pomocą funkcji przystosowania, wyrażającej się wzorem [24]

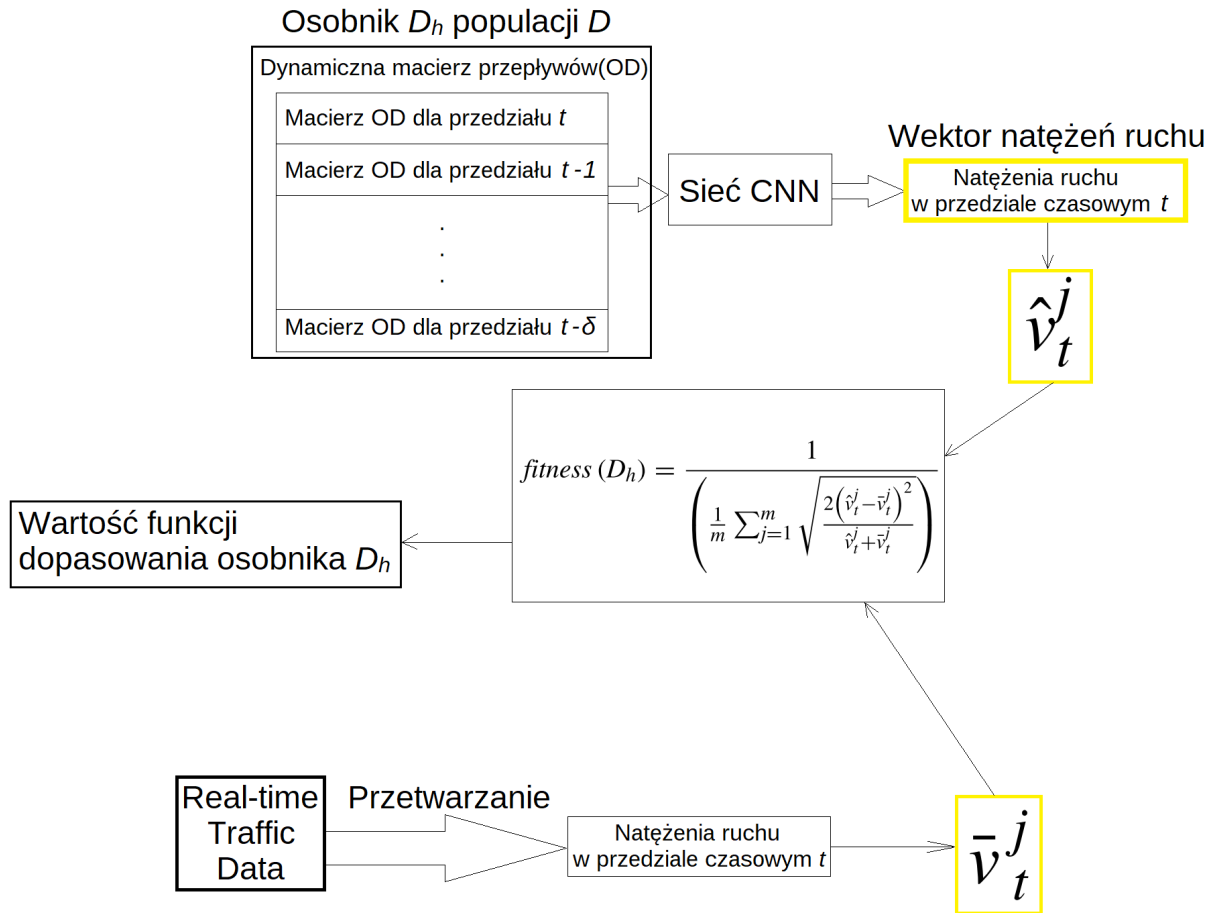
$$fitness(D_h) = \frac{1}{\frac{1}{m} \sum_{j=1}^m \sqrt{\frac{2(\hat{v}_t^j - \bar{v}_t^j)^2}{\hat{v}_t^j + \bar{v}_t^j}}},$$

gdzie

\hat{v}_t^j jest obliczonym przez sieć CNN j -tym elementem wektora natężeń ruchu w przedziale czasowym t ,

\bar{v}_t^j jest j -tym elementem wektora natężeń ruchu w przedziale czasowym t ,

m jest ilością obserwowanych natężeń ruchu.



Rysunek 7.14: Sposób oceny osobnika populacji, będącego dynamiczną macierzą OD [24]

Po obliczeniu wartości funkcji przystosowania każdego osobnika populacji, następowała selekcja. Selekcja była wykonywana w taki sam sposób jak poprzednio tzn. metodą ruletki. Krzyżowanie i mutacja również odbywały się tak samo jak wcześniej. Po wykonaniu selekcji, krzyżowania i mutacji otrzymywano nową populację. Jeśli osiągnięto maksymalną ilość generacji, to algorytm kończył działanie. Algorytm zwracał najlepsze (pod względem funkcji przystosowania) rozwiązanie z ostatniej generacji. Po zakończeniu działania algorytmu (rysunek 7.15) otrzymano dynamiczną macierz przepływu dopasowaną do natężeń w przedziale czasowym t .

Wzorcowy ciąg T macierzy OD

Macierz OD dla przedziału 1
Macierz OD dla przedziału 2
⋮
Macierz OD dla przedziału T

Wzorcowa dynamiczna macierz OD

Macierz OD dla przedziału t
Macierz OD dla przedziału $t - 1$
⋮
Macierz OD dla przedziału $t - \delta$

Wynikowa dynamiczna macierz OD

Macierz OD dla przedziału t
Macierz OD dla przedziału $t - 1$
⋮
Macierz OD dla przedziału $t - \delta$

Nateżenia ruchu
w przedziale czasowym t Algorytm
genetyczny

Rysunek 7.15: Algorytm genetyczny na podstawie wektora nateżeń ruchu znalazł dynamiczną macierz przepływu. Populacja początkowa była inicjalizowana z wykorzystaniem wzorcowego ciągu T macierzy OD.

Rozdział 8

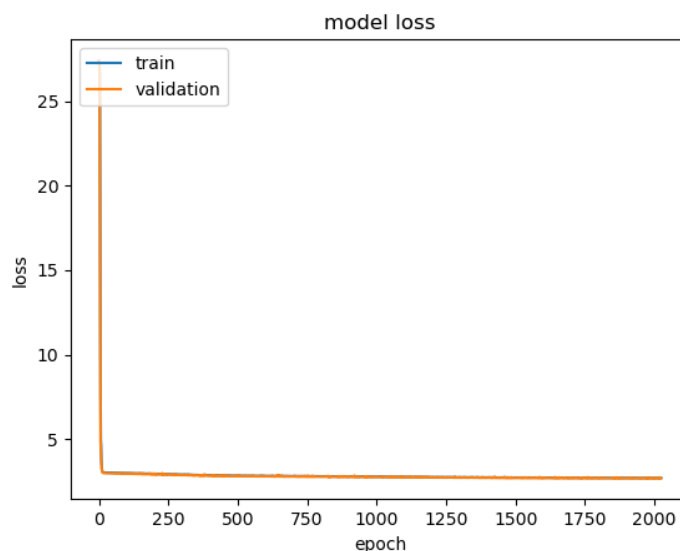
Wyniki numeryczne

Badania wykonano dla ulicy Legnickiej we Wrocławiu oraz dla miasta Sioux Falls. Algorytm rozwiązujący problem estymacji dynamicznej macierzy przepływów wyznacza ciąg $\delta + 1$ macierzy OD. Parametr δ w obu przypadkach (ul. Legnicka i Sioux Falls) jest równy 2. W algorytmach genetycznych rozmiar populacji wynosił 800, natomiast liczba generacji wynosiła 5000. Prawdopodobieństwo krzyżowania μ wynosiło 0.9. Algorytm poszukujący wzorcowego ciągu T macierzy OD miał prawdopodobieństwo mutacji $\lambda = 10^{-5}$. W algorytmie poszukującym dynamicznej macierzy OD prawdopodobieństwo mutacji λ wynosiło 10^{-4} .

Po wygenerowaniu danych dla ul. Legnickiej i Sioux Falls okazało się, że macierze OD mają zbyt duże wartości w porównaniu z natężeniami ruchu. Suma wszystkich żądań OD (Origin-Destination - źródło-cel) wynosiła 50 323 dla ul. Legnickiej i 159 815 dla Sioux Falls. Natomiast ilość pojazdów, które dojechały do celu w czasie całej symulacji, wynosiła 11 293 dla ul. Legnickiej i 29 351 dla Sioux Falls. Oznaczało to, że około 5 razy mniej pojazdów dojeżdżało do celu niż było to określone w macierzach OD. Dla późniejszych przedziałów czasowych kolejne macierze OD nie miały żadnego wpływu na dalszy ciąg symulacji, ponieważ wynikające z macierzy żądania OD nie były wykonywane w trakcie symulacji. W takiej sytuacji estymacja dynamicznej macierzy OD (która nie ma wpływu na natężenia ruchu) na podstawie natężeń ruchu byłaby bardzo niedokładna.

Rozwiązaniem problemu, które zastosowano, było pięciokrotne zmniejszenie (i zaokrąglenie w dół do liczby całkowitej) pierwotnych wartości macierzy OD i ponowne wygenerowanie danych. Po zastosowaniu tego rozwiązania suma wszystkich żądań OD wynosiła 8 423 dla ul. Legnickiej i 30 386 dla Sioux Falls. Natomiast ilość pojazdów, które dojechały do celu w czasie całej symulacji, wynosiła 8 115 dla ul. Legnickiej i 25 070 dla Sioux Falls. Wówczas zdecydowana większość pojazdów dojeżdżała do celu.

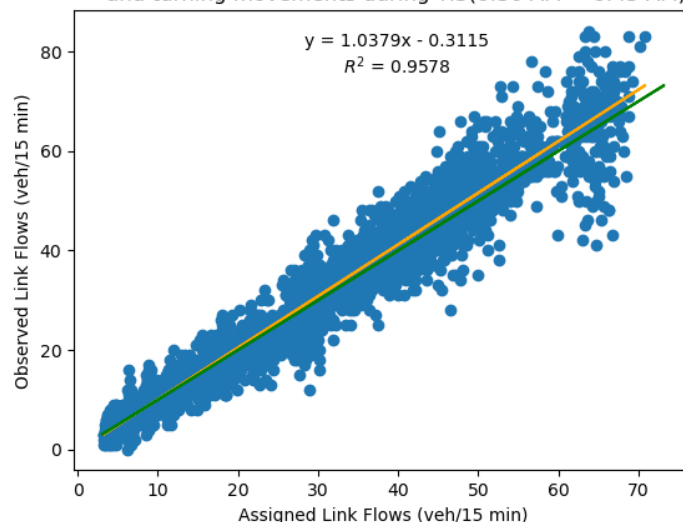
Jakość działania sieci CNN, dla danych z ul. Legnickiej, została oceniona poprzez wskaźniki MAE, MAPE oraz R^2 . Zbadano dokładność estymacji natężeń ruchu dla trzech wybranych przedziałów czasowych. Przebieg treningu sieci CNN dla ul. Legnickiej jest przedstawiony na wykresie (rysunek 8.1).



Rysunek 8.1: Przebieg treningu sieci CNN na danych z ul. Legnickiej. Funkcja błędu modelu wraz z kolejnymi epokami malała. Wartość funkcji strat na zbiorze treningowym była zbliżona do wartości funkcji strat na zbiorze walidacyjnym, dlatego wykresy (niebieski i pomarańczowy) się pokrywają. To świadczy o balansie pomiędzy wariancją i obciążeniem modelu.

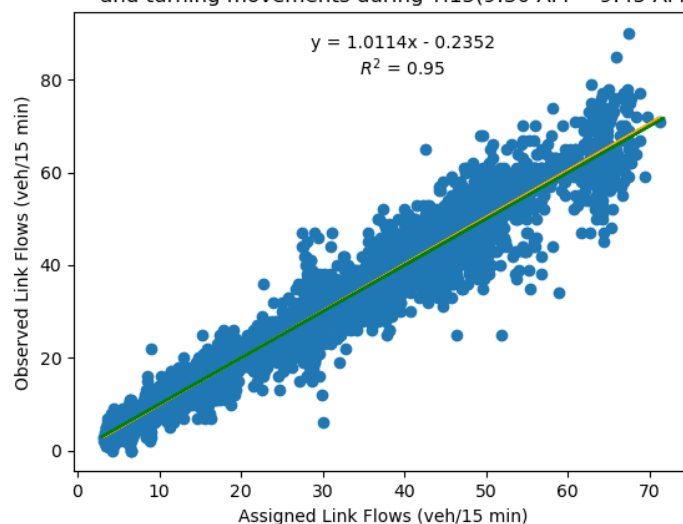
Zbadano 3 przedziały czasowe: TI3 (6:30 - 6:45), TI15 (9:30 - 9:45), TI23 (11:30 - 11:45). Dla każdego przedziału czasowego zostały narysowane wykresy (rysunki 8.2, 8.3 i 8.4) oraz obliczono wskaźniki (tabela 8.1) MAE, MAPE oraz R^2 . Na osi poziomej wykresów jest wartość natężenia przypisana przez sieć CNN, natomiast na osi pionowej znajduje się wartość natężenia obserwowanego (ze zbioru danych). Żółta prosta (jej równanie jest na wykresach) jest dopasowana do danych za pomocą regresji liniowej. Zielona prosta jest prostą o równaniu $y=x$. Im bliżej zielonej prostej znajdują się punkty tym model jest dokładniejszy. Wskaźnik MAE określa średnie odchylenie punktów od zielonej prostej. MAPE jest średnim błędem bezwzględnym wyrażonym w procentach. R^2 określa dopasowanie modelu do danych i jego wartość jest w zakresie od 0.0 do 1.0. Im większy jest współczynnik determinacji R^2 tym dokładniejszy jest model.

Comparisons of the assigned and observed flows of the monitored links and turning movements during TI3(6:30 AM ~ 6:45 AM)

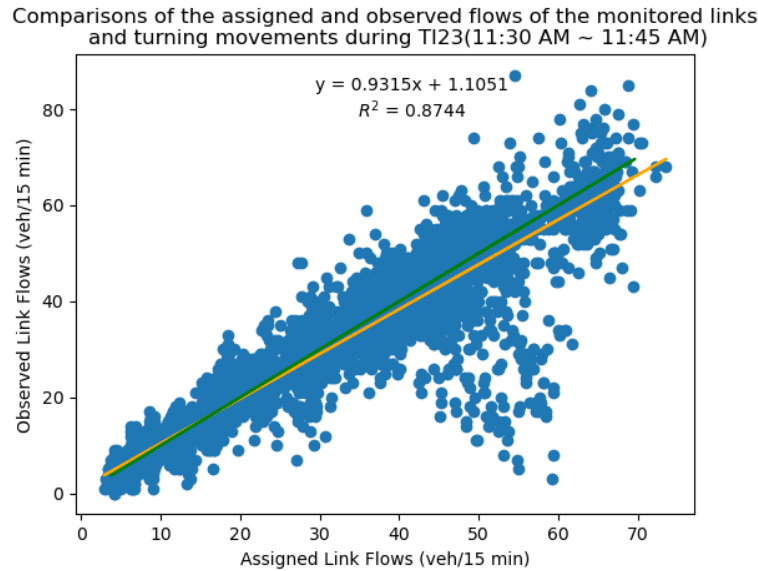


Rysunek 8.2: Porównanie przypisanych i obserwowanych natężeń ruchu w przedziale czasowym TI3 na ul. Legnickiej. Wyniki: MAE = 2.95, MAPE = 13.78%.

Comparisons of the assigned and observed flows of the monitored links and turning movements during TI15(9:30 AM ~ 9:45 AM)



Rysunek 8.3: Porównanie przypisanych i obserwowanych natężeń ruchu w przedziale czasowym TI15 na ul. Legnickiej. Wyniki: MAE = 3.04, MAPE = 14.31%.



Rysunek 8.4: Porównanie przypisanych i obserwowanych natężeń ruchu w przedziale czasowym TI23 na ul. Legnickiej. Wyniki: MAE = 3.90, MAPE = 19.66%.

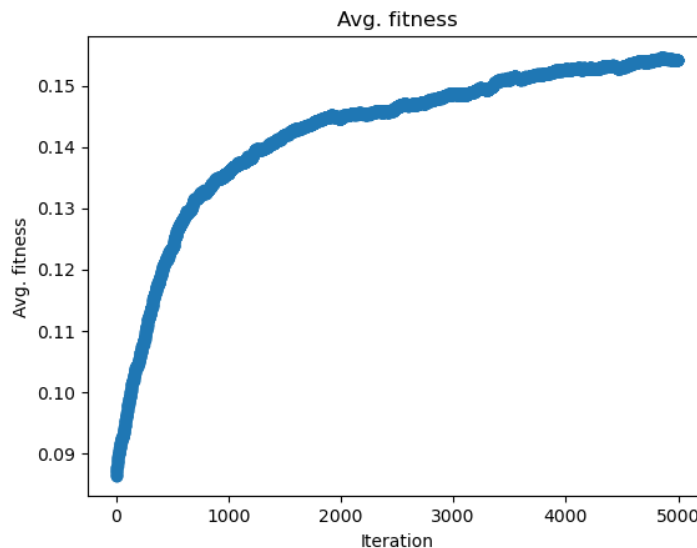
Tabela 8.1: Otrzymane wyniki w poszczególnych przedziałach czasowych dla ul. Legnickiej.

Przedział	MAE[pojazdy/15min]	MAPE[%]	R^2
TI3	2.95	13.8%	0.955
TI15	3.04	14.3%	0.950
TI23	3.90	16.7%	0.868

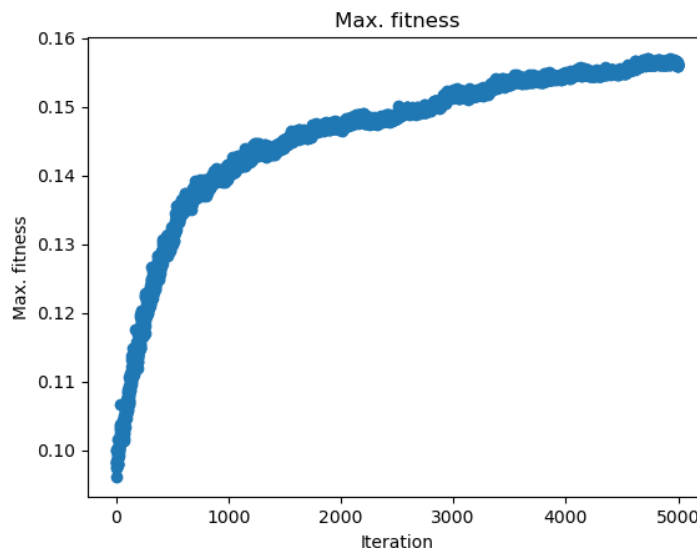
W kolejnych przedziałach czasowych zaobserwowano pogarszanie się wszystkich trzech wskaźników. W pracy [24] wykonano podobne badanie. Wartość współczynnika R^2 w pracy [24] (od 0.905 do 0.967) jest trochę większa niż dla ul. Legnickiej (od 0.874 do 0.957). Zatem sieć CNN była dokładniejsza dla danych z miasta z pracy [24] niż dla danych z ul. Legnickiej.

Przy uwzględnieniu wyników dla całego zbioru danych (22 przedziałów czasowych ($T - \delta = 24 - 2 = 22$)) wskaźniki są następujące: MAE = 3.14, MAPE = 15.08%, $R^2 = 0.932$.

Przebieg algorytmu genetycznego poszukującego wzorcowego ciągu T macierzy OD dla ul. Legnickiej został przedstawiony za pomocą średniej i maksymalnej wartości funkcji przystosowania osobników. Na wykresach (rysunki 8.5 i 8.6) przedstawiono średnią i maksymalną wartość funkcji przystosowania w kolejnych generacjach.



Rysunek 8.5: Średnia wartość funkcji przystosowania w kolejnych generacjach.



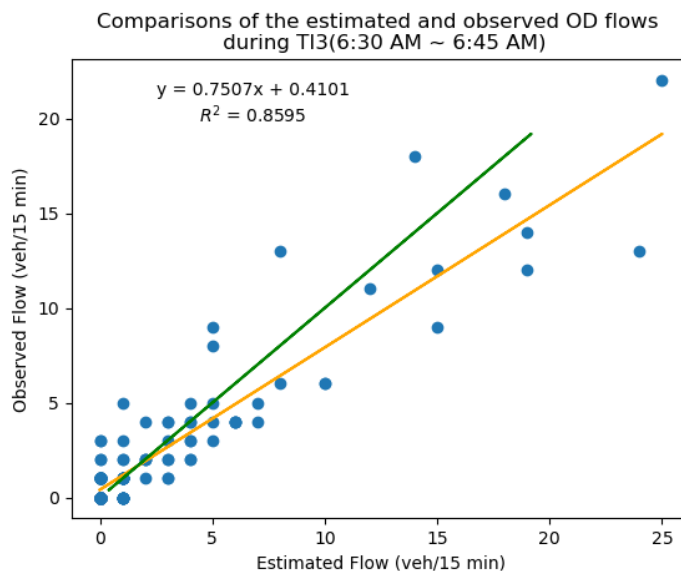
Rysunek 8.6: Wartość funkcji przystosowania najlepszego osobnika populacji w kolejnych generacjach.

Wyniki algorytmu genetycznego poszukującego dynamiczną macierz przepływu dla ul. Legnickiej zostały ocenione za pomocą wartości funkcji przystosowania, błędu MAE, błędu MAPE oraz współczynnika R^2 . Błąd MAE, błąd MAPE oraz R^2 były liczone na podstawie wybranej macierzy OD (dla przedziału czasowego t) ze znalezionego ciągu $\delta + 1$ macierzy OD. Wybrana macierz przepływu dla przedziału czasowego t była porównywana z macierzą OD dla przedziału t ze zbioru danych poprzez obliczenie błędu MAE, błędu MAPE oraz R^2 .

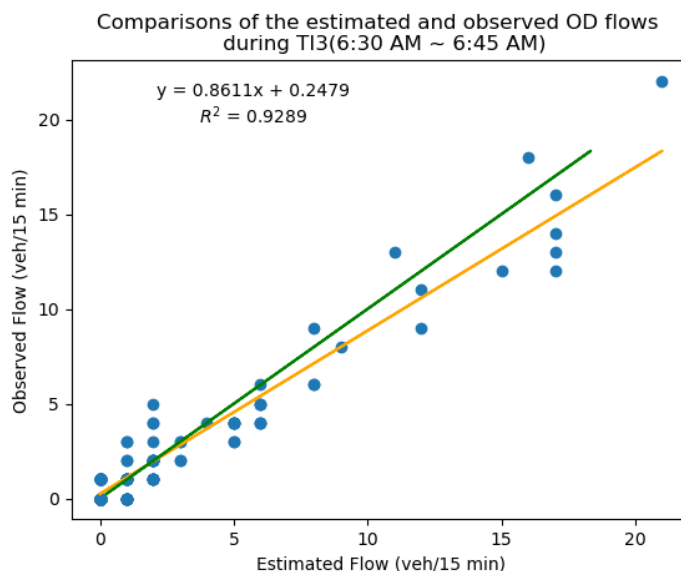
Zbadano wpływ przestrzeni poszukiwań na jakość wyników. Przestrzeń poszukiwań była przedziałem (w przestrzeni wielowymiarowej) zależnym od wzorcowej dynamicznej macierzy OD. Badania zostały wykonane dla przedziału czasowego $t = 3$. Sprawdzone następujące przedziały: od 0.5 do 1.5, od 0.75 do 1.25 oraz od 0.9 do 1.1 wzorcowej

dynamicznej macierzy OD. Wyniki przedstawiono w tabeli 8.2 oraz na wykresach (rysunki 8.7, 8.8 i 8.9).

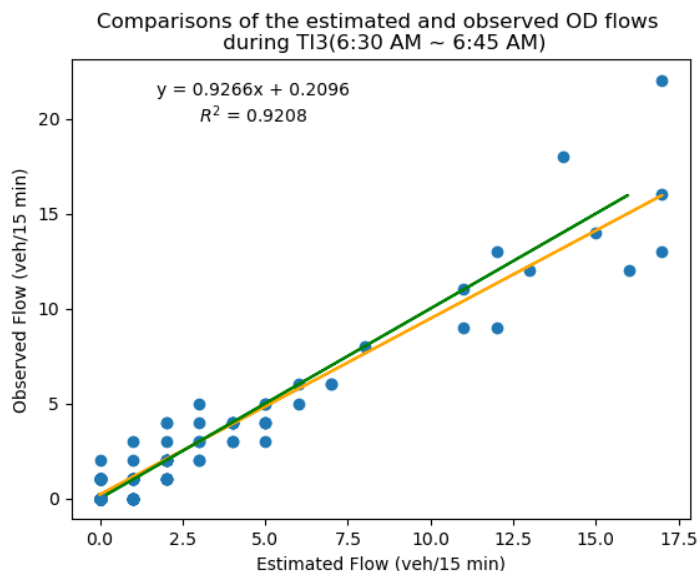
Na osi poziomej wykresów jest wartość przepływu znaleziona przez algorytm genetyczny, natomiast na osi pionowej znajduje się wartość przepływu obserwowanego (ze zbioru danych). Żółta prosta (jej równanie jest na wykresach) jest dopasowana do danych za pomocą regresji liniowej.



Rysunek 8.7: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.5 do 1.5 wzorcowej dynamicznej macierzy OD: MAE = 0.77, MAPE = 39%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 3.0.



Rysunek 8.8: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.75 do 1.25 wzorcowej dynamicznej macierzy OD: MAE = 0.59, MAPE = 35%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 2.5.



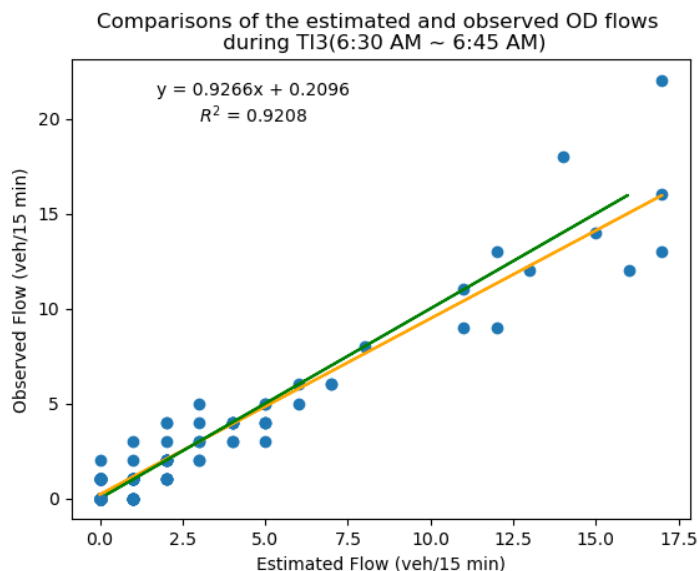
Rysunek 8.9: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.9 do 1.1 wzorcowej dynamicznej macierzy OD: MAE = 0.52, MAPE = 34%.

Tabela 8.2: Badanie wpływu rozmiaru przestrzeni poszukiwań na otrzymane wyniki.

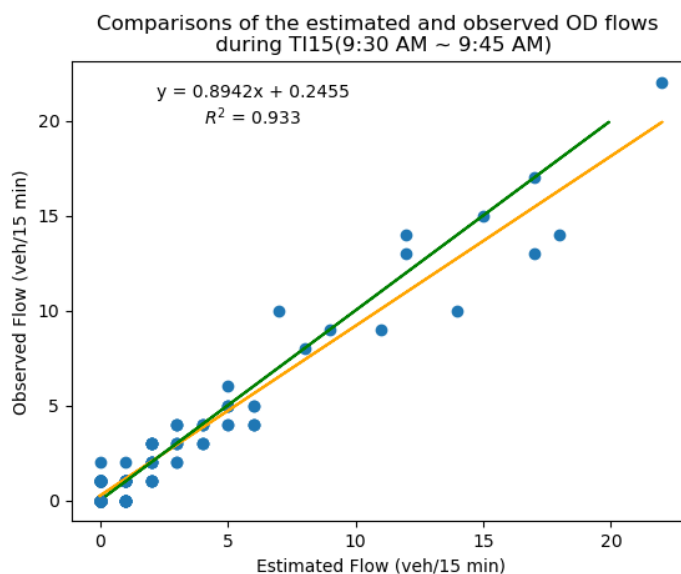
Przedział	F. przystosowania	MAE[pojazdy/15min]	MAPE[%]	R^2
od 0.5 do 1.5	około 3.0	0.77	39%	0.783
od 0.75 do 1.25	około 2.5	0.59	35%	0.906
od 0.9 do 1.1	około 2.0	0.52	34%	0.922

Wraz ze zmniejszaniem przestrzeni poszukiwań wartość funkcji przystosowania dynamicznej macierzy OD malała oraz MAE i MAPE również malały. Zatem im mniejsza była przestrzeń poszukiwań, tym mniejsza różnica była pomiędzy znalezioną dynamiczną macierzą OD a dynamiczną macierzą OD ze zbioru danych. Zmniejszanie przestrzeni poszukiwań powodowało znalezienie gorzej dopasowanej (wartość funkcji przystosowania) dynamicznej macierzy OD do natężeń ruchu ze zbioru danych. Przedstawione wyniki pokazują, że polepszenie wartości funkcji przystosowania nie jest równoważne ze zmniejszeniem błędu MAPE. Tak jest, ponieważ funkcja przystosowania opisuje dopasowanie znalezionej dynamicznej macierzy OD do natężeń ruchu, a błąd MAPE określa dopasowanie znalezionej dynamicznej macierzy OD do dynamicznej macierzy OD ze zbioru danych.

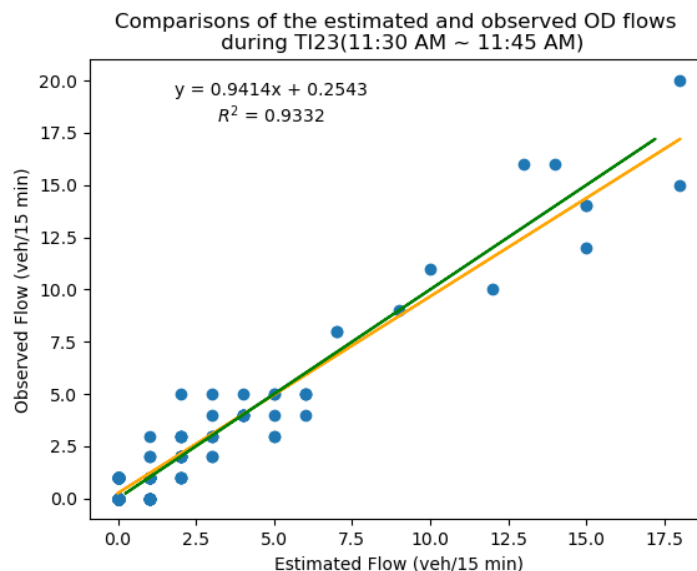
Sprawdzono również wyniki dla następujących przedziałów czasowych: TI3 (6:30 - 6:45), TI15 (9:30 - 9:45) i TI23 (11:30 - 11:45). Przestrzeń poszukiwań była przedziałem od 0.9 do 1.1 wzorcowej dynamicznej macierzy OD. Otrzymane wyniki porównano z wynikami z pracy [24]. Wyniki przedstawiono w tabeli 8.3 oraz na wykresach (rysunki 8.10, 8.11 i 8.12).



Rysunek 8.10: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3 na ul. Legnickiej. Wyniki: MAE = 0.52, MAPE = 34%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 2.0.



Rysunek 8.11: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI15 na ul. Legnickiej. Wyniki: MAE = 0.53, MAPE = 37%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 1.4.



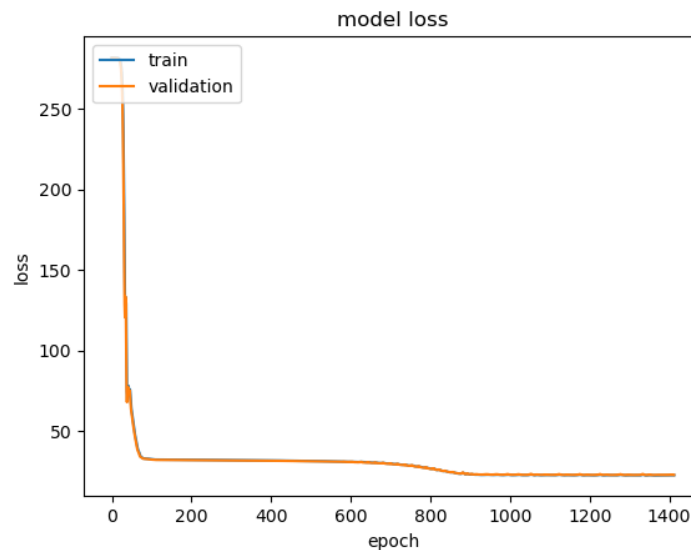
Rysunek 8.12: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI23 na ul. Legnickiej. Wyniki: MAE = 0.51, MAPE = 34.5%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 1.5.

Tabela 8.3: Otrzymane wyniki w poszczególnych przedziałach czasowych.

Przedział	F. przystosowania	MAE[pojazdy/15min]	MAPE[%]	R^2
TI3	około 2.0	0.52	34%	0.911
TI15	około 1.4	0.53	37%	0.910
TI23	około 1.5	0.51	34.5%	0.922

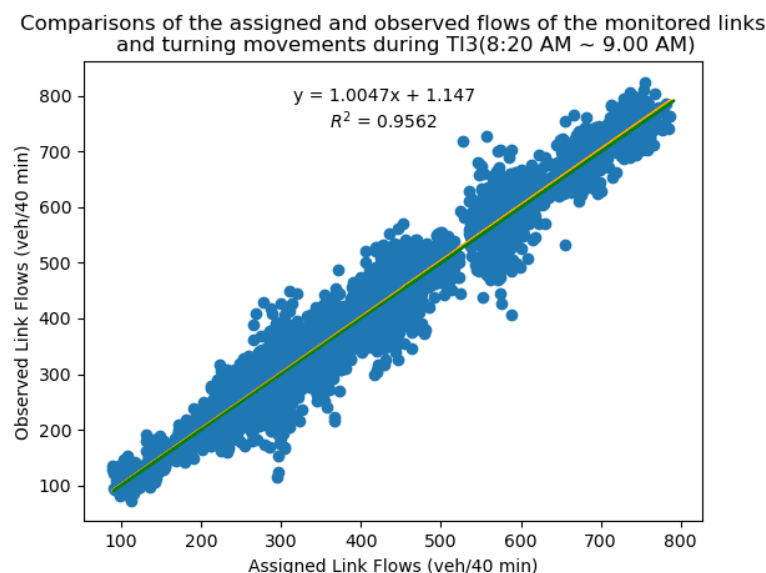
Błąd MAE, błąd MAPE i R^2 nieznacznie się zmieniały wraz ze zmianą przedziału czasowego. Wartość funkcji przystosowania wynosiła dla kolejnych przedziałów czasowych 2.0, 1.4 i 1.5, natomiast w pracy [24] wartość funkcji przystosowania w poszczególnych przedziałach czasowych była w zakresie od 0.8 do 1.5. Zatem algorytm znalazł dynamiczną macierz przepływu lepiej dopasowaną do natężeń ruchu dla ul. Legnickiej niż dla miasta w pracy [24]. R^2 mieściło się w zakresie od 0.91 do 0.93, natomiast w pracy [24] R^2 było większe niż 0.95. Zatem algorytm zwrócił dokładniejsze wyniki dla danych z miasta z pracy [24] niż dla danych z ul. Legnickiej.

Jakość działania sieci CNN, dla danych z Sioux Falls, została oceniona poprzez wskaźniki MAE, MAPE oraz R^2 . Zbadano dokładność estymacji natężeń ruchu dla dwóch przedziałów czasowych. Przebieg treningu sieci CNN jest przedstawiony na wykresie (rysunek 8.13).

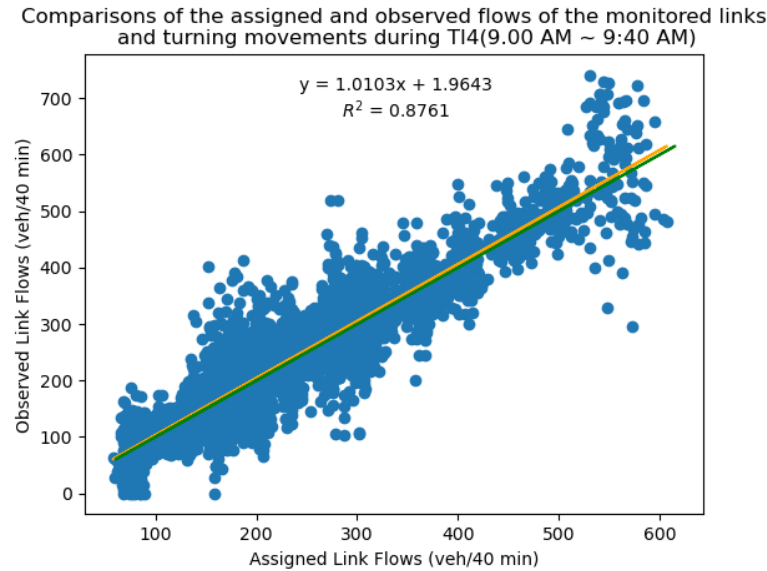


Rysunek 8.13: Przebieg treningu sieci CNN na danych z Sioux Falls. Funkcja błędu modelu wraz z kolejnymi epokami malała. Wartość funkcji strat na zbiorze treningowym była zbliżona do wartości funkcji strat na zbiorze walidacyjnym, dlatego wykresy (niebieski i pomarańczowy) się pokrywają. To świadczy o balansie pomiędzy wariancją i obciążeniem modelu.

Zbadano 2 przedziały czasowe: TI3 (8:20 - 9:00), TI4 (9:00 - 9:40). Dla każdego przedziału czasowego zostały narysowane wykresy (rysunki 8.14 i 8.15) oraz obliczono wskaźniki (tabela 8.4) MAE, MAPE oraz R^2 . Na osi poziomej wykresów jest wartość natężenia przypisana przez sieć CNN, natomiast na osi pionowej znajduje się wartość natężenia obserwowanego (ze zbioru danych). Żółta prosta (jej równanie jest na wykresach) jest dopasowana do danych za pomocą regresji liniowej. Zielona prosta jest prostą o równaniu $y=x$.



Rysunek 8.14: Porównanie przypisanych i obserwowanych natężeń ruchu dla przedziału TI3 w Sioux Falls. Wyniki: MAE = 21.4; MAPE = 6.49%.



Rysunek 8.15: Porównanie przypisanych i obserwowanych natężeń ruchu dla przedziału TI4 w Sioux Falls. Wyniki: MAE = 24.5, MAPE = 25.0%.

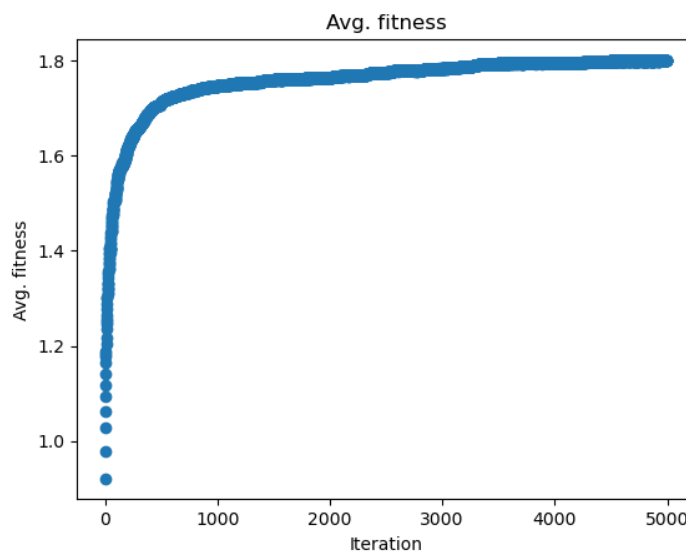
Tabela 8.4: Otrzymane wyniki w poszczególnych przedziałach czasowych.

Przedział	MAE[pojazdy/40min]	MAPE[%]	R^2
TI3	21.4	6.49%	0.956
TI4	24.5	25.0%	0.875

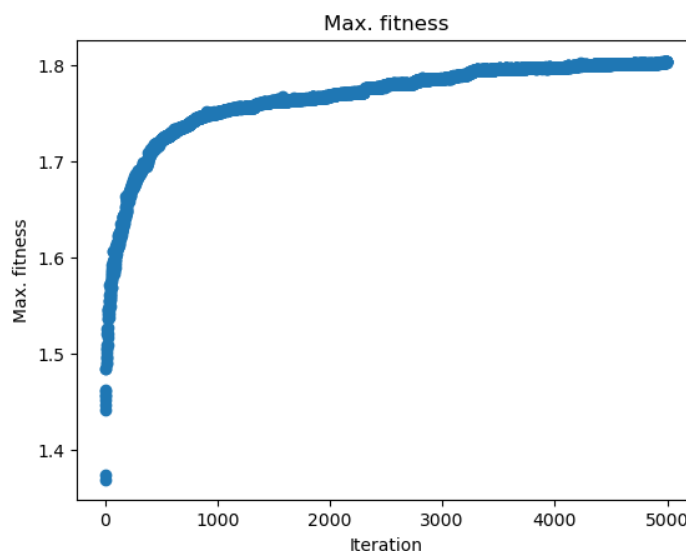
W przedziale TI4 zaobserwowano pogorszenie się wszystkich trzech wskaźników. W pracy [24] wykonano podobne badanie. Wartość współczynnika R^2 w pracy [24] (od 0.905 do 0.967) jest trochę większa niż dla Sioux Falls (0.957 i 0.874).

Przy uwzględnieniu wyników dla całego zbioru danych (2 przedziały czasowe ($T - \delta = 4 - 2 = 2$)) wskaźniki są następujące: MAE = 23.0, MAPE = 15.7%, $R^2 = 0.947$.

Przebieg algorytmu genetycznego poszukującego wzorcowego ciągu T macierzy OD dla Sioux Falls został przedstawiony za pomocą średniej i maksymalnej wartości funkcji przystosowania osobników. Na wykresach (rysunki 8.16 i 8.17) przedstawiono średnią i maksymalną wartość funkcji przystosowania w kolejnych generacjach.



Rysunek 8.16: Średnia wartość funkcji przystosowania w kolejnych generacjach.



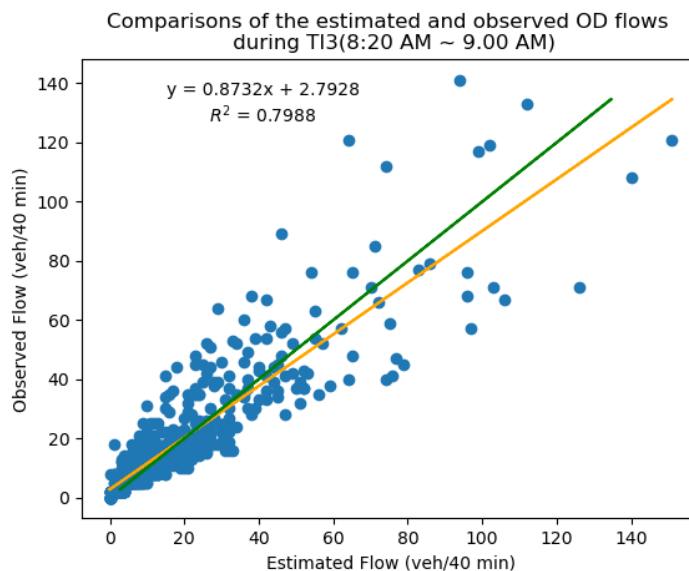
Rysunek 8.17: Wartość funkcji przystosowania najlepszego osobnika populacji w kolejnych generacjach.

Wyniki algorytmu poszukującego dynamiczną macierz przepływu dla Sioux Falls zostały ocenione za pomocą wartości funkcji przystosowania, błędu MAE, błędu MAPE oraz współczynnika R^2 . Błąd MAE, błąd MAPE oraz R^2 były liczone na podstawie wybranej macierzy OD (dla przedziału czasowego t) ze znalezionej ciągu $\delta + 1$ macierzy OD. Wybrana macierz przepływu dla przedziału czasowego t była porównywana z macierzą OD dla przedziału t ze zbioru danych poprzez obliczenie błędu MAE, błędu MAPE oraz R^2 .

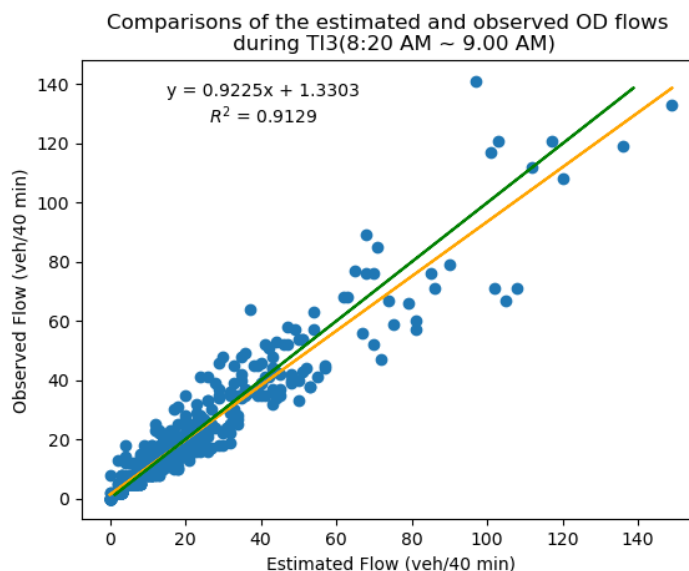
Zbadano wpływ przestrzeni poszukiwań na jakość wyników. Przestrzeń poszukiwań była przedziałem (w przestrzeni wielowymiarowej) zależnym od wzorcowej dynamicznej macierzy OD. Badania zostały wykonane dla przedziału czasowego $t = 3$. Sprawdzone następujące przedziały: od 0.5 do 1.5, od 0.75 do 1.25 oraz od 0.9 do 1.1 wzorcowej

dynamicznej macierzy OD. Wyniki przedstawiono w tabeli 8.5 oraz na wykresach (rysunki 8.18, 8.19 i 8.20).

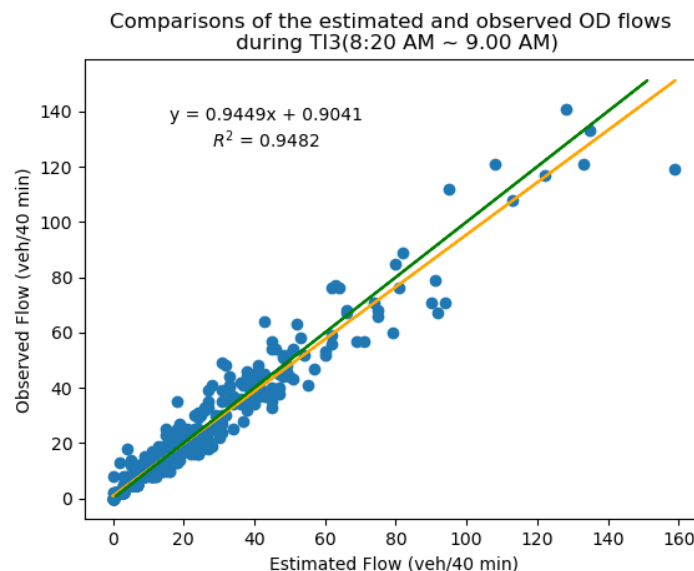
Na osi poziomej wykresów jest wartość przepływu znaleziona przez algorytm genetyczny, natomiast na osi pionowej znajduje się wartość przepływu obserwowanego (ze zbioru danych). Żółta prosta (jej równanie jest na wykresach) jest dopasowana do danych za pomocą regresji liniowej. Zielona prosta jest prostą o równaniu $y=x$.



Rysunek 8.18: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3 w Sioux Falls. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.5 do 1.5 wzorcowej dynamicznej macierzy OD: MAE = 5.76, MAPE = 33.9%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 2.2.



Rysunek 8.19: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3 w Sioux Falls. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.75 do 1.25 wzorcowej dynamicznej macierzy OD: MAE = 3.75, MAPE = 21.5%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 2.1.



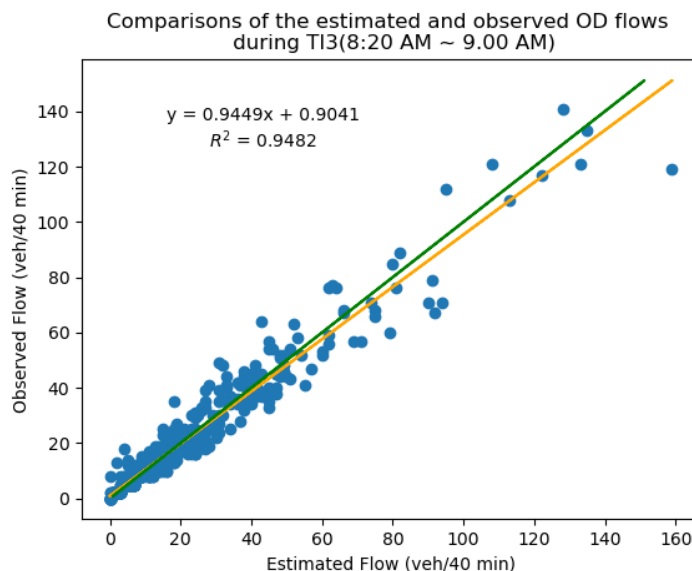
Rysunek 8.20: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3 w Sioux Falls. Wyniki dla przestrzeni poszukiwań będącej przedziałem od 0.9 do 1.1 wzorcowej dynamicznej macierzy OD: MAE = 2.91, MAPE = 18.3%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 1.9.

Tabela 8.5: Badanie wpływu rozmiaru przestrzeni poszukiwań na otrzymane wyniki.

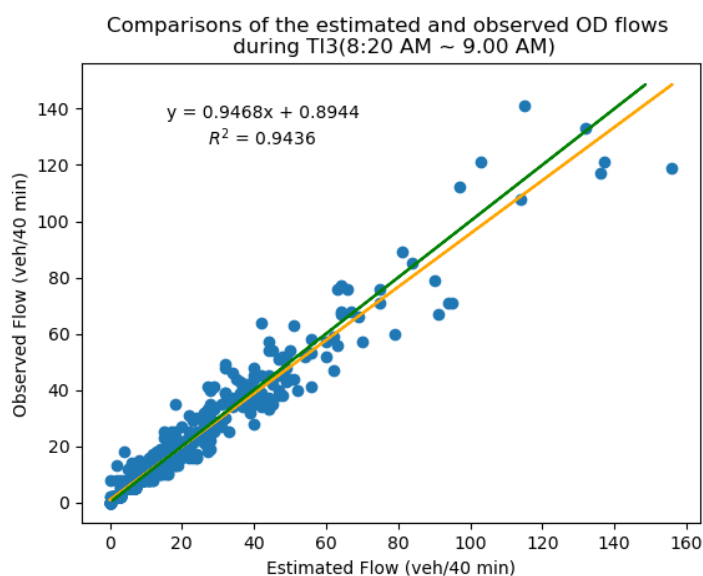
Przedział	F. przystosowania	MAE[pojazdy/40min]	MAPE[%]	R^2
od 0.5 do 1.5	około 2.2	5.76	33.9%	0.781
od 0.75 do 1.25	około 2.1	3.75	21.5%	0.906
od 0.9 do 1.1	około 1.9	2.91	18.3%	0.945

Wraz ze zmniejszaniem przestrzeni poszukiwań wartość funkcji przystosowania dynamicznej macierzy OD malała oraz MAE i MAPE również malały. Przedstawione wyniki pokazują, że polepszenie wartości funkcji przystosowania nie jest równoważne ze zmniejszeniem błędu MAPE. Tak jest, ponieważ funkcja przystosowania opisuje dopasowanie znalezionej dynamicznej macierzy OD do natężeń ruchu, a błąd MAPE określa dopasowanie znalezionej dynamicznej macierzy OD do dynamicznej macierzy OD ze zbioru danych.

Sprawdzono również wyniki dla następujących przedziałów czasowych: TI3 (8:20 - 9:00), TI4 (9:00 - 9:40). Przestrzeń poszukiwań była przedziałem od 0.9 do 1.1 wzorcowej dynamicznej macierzy OD. Otrzymane wyniki porównano z wynikami z pracy [24]. Wyniki przedstawiono w tabeli 8.6 oraz na wykresach (rysunki 8.21 i 8.22).



Rysunek 8.21: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI3 w Sioux Falls. Wyniki: MAE = 2.91, MAPE = 18.3%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 1.9.



Rysunek 8.22: Porównanie estymowanych i obserwowanych przepływów w przedziale czasowym TI4 w Sioux Falls. Wyniki: MAE = 2.94, MAPE = 18.4%. Wartość funkcji przystosowania dynamicznej macierzy OD wynosiła około 1.4.

Tabela 8.6: Otrzymane wyniki w poszczególnych przedziałach czasowych.

Przedział	F. przystosowania	MAE[pojazdy/40min]	MAPE[%]	R^2
TI3	około 1.9	2.91	18.3%	0.945
TI4	około 1.4	2.94	18.4%	0.941

Błąd MAE, błąd MAPE i R^2 nieznacznie się zmieniały wraz ze zmianą przedziału czasowego. Wartość funkcji przystosowania wynosiła dla kolejnych przedziałów czasowych 1.9 i 1.4, natomiast w pracy [24] wartość funkcji przystosowania w poszczególnych przedziałach czasowych była w zakresie od 0.8 do 1.5. Zatem algorytm znalazł dynamiczną macierz przepływu lepiej dopasowaną do natężeń ruchu dla Sioux Falls niż dla miasta w pracy [24]. R^2 dla Sioux Falls (0.945 i 0.941) było mniejsze niż w pracy [24] (większe niż 0.95).

Na zakończenie tego rozdziału warto byłoby podsumować wyniki osiągnięte przez sieć CNN i algorytm genetyczny. Im późniejszy przedział czasowy, tym gorsze wskaźniki (MAE, MAPE i R^2) miała sieć CNN. W pracy [24] ten problem również jest widoczny, ponieważ w początkowych przedziałach czasowych MAPE wynosił 6%, a w późniejszych około 10%. Powodem tego mógł być fakt, że natężenie ruchu zależało nie tylko od dynamicznej macierzy OD, lecz także od stanu początkowego zatłoczenia miasta. Sieć CNN była trochę dokładniejsza (pod względem współczynnika R^2) dla danych z pracy [32] niż dla ul. Legnickiej i Sioux Falls.

W obu przypadkach (ul. Legnicka i Sioux Falls), wraz ze zmniejszaniem przestrzeni poszukiwań wartość funkcji przystosowania dynamicznej macierzy OD malała oraz MAE i MAPE również malały. Zatem korzystne było zmniejszenie przestrzeni poszukiwań z zakresu od 0.5 do 1.5 wzorcowej dynamicznej macierzy OD do zakresu od 0.9 do 1.1 wzorcowej dynamicznej macierzy OD.

Algorytm genetyczny znalazł lepiej dopasowane dynamiczne macierze OD do natężeń ruchu dla Sioux Falls i ul. Legnickiej niż dla miasta z pracy [24]. Jednak znalezione dynamiczne macierze OD dla ul. Legnickiej i Sioux Falls trochę bardziej różniły się (mniejszy wskaźnik R^2) od dynamicznych macierzy OD ze zbioru danych niż dla miasta z pracy [24].

Rozdział 9

Perspektywy rozwoju

Pomimo dokładnego przebadania założonych obszarów zagadnienia estymacji dynamicznej macierzy przepływów, znajdują się możliwości rozbudowy oprogramowania wykonanego w ramach niniejszej pracy. Implementacja algorytmu estymującego dynamiczną macierz przepływu stwarza następujące możliwości:

- generowanie danych apriorycznych na podstawie rzeczywistego ciągu macierzy przepływów, który byłby wyznaczany poprzez śledzenie ruchu pojazdów np. na podstawie tablic rejestracyjnych;
- uzyskanie danych historycznych i danych w czasie rzeczywistym poprzez pomiar natężeń ruchu i śledzenie pojazdów (ciąg macierzy OD); ciąg macierzy OD byłby potrzebny do testowania oprogramowania, więc do zastosowań praktycznych nie byłby on wymagany;
- uwzględnienie w danych, obok informacji o natężeniu ruchu, informacji o ruchu na skrzyżowaniach np. o tym ile pojazdów skręciło z południowej części skrzyżowania w prawo; wówczas algorytm genetyczny wyznaczałby dynamiczną macierz przepływu na podstawie natężeń ruchu i informacji o ruchu na skrzyżowaniach;
- rozwiązanie problemu pogarszania się wskaźników sieci CNN wraz z kolejnymi przedziałami czasowymi np. poprzez dodanie do wejścia sieci wektora natężeń ruchu w przedziale czasowym $t - \delta$;
- przetestowanie estymacji natężeń ruchu przy użyciu innych metod uczenia maszynowego niż sieć konwolucyjna np. sieć RBF (*Radial Basis Function*), perceptron wielowarstwowy, drzewa decyzyjne, komitet sieci neuronowych;
- stworzenie oprogramowania estymującego ilość wolnych miejsc na parkingach;
Z macierzy OD można odczytać ile pojazdów rozpoczyna podróż z węzła A do węzła B. Na tej podstawie można oszacować ile miejsc na parkingach w danym rejonie się zwolniło. Znając czas podróży można też oszacować ile miejsc na parkingach w danym rejonie zostanie zajętych. Zatem przy użyciu programu obliczającego dynamiczną macierz OD można stworzyć oprogramowanie estymujące ilość wolnych miejsc na parkingach.
- stworzenie symulacji miasta, która mogłaby służyć do trenowania różnorodnych modeli np. przewidujących natężenie ruchu w mieście lub estymujących czas dojazdu z węzła A do węzła B;

-
- zastosowanie symulacji miasta do regulacji świateł drogowych lub stworzenia systemu wyznaczającego optymalną trasę dojazdu.

Rozdział 10

Podsumowanie i wnioski

Zrealizowano założone cele, którymi były napisanie oprogramowania estymującego dynamiczną macierz przepływów oraz zbadanie metody zaproponowanej w pracy [24]. Metoda zaproponowana w pracy [24] została zbadana na wygenerowanych danych z ul. Legnickiej i Sioux Falls. Podczas realizacji pracy zaimplementowano generator danych, sieć konwolucyjną (CNN) oraz algorytmy genetyczne. Z przeprowadzonych badań wynika, że metodę można usprawnić poprzez zmniejszenie przestrzeni poszukiwań. Poczynione obserwacje wskaźników sieci CNN pokazują, że wytrenowanie skutecznego modelu obliczającego (dla dowolnego t) wektor natężeń ruchu w przedziale czasowym t na podstawie dynamicznej macierzy OD sprawia problemy.

Stworzone w niniejszej pracy rozwiązanie można udoskonalić na kilka sposobów np. generując dane na podstawie rzeczywistego ciągu macierzy przepływów, uwzględniając w danych ruch na skrzyżowaniach lub rozwiązując problem zwiększającego się błędu sieci CNN wraz z kolejnymi przedziałami czasowymi. Również istnieją możliwości rozbudowy stworzonego oprogramowania np. stworzenie programu estymującego ilość wolnych miejsc na parkingach lub stworzenie symulacji miasta oraz zastosowanie jej w praktyce (np. do regulacji światła drogowych).

Podsumowując stwierdzam, że cel pracy został zrealizowany. Efekt pracy może być z powodzeniem wykorzystany w dalszych badaniach i aplikacjach usprawniających ruch pojazdów.

Bibliografia

- [1] J. Arabas. *Wykłady z algorytmów ewolucyjnych*. WNT, Warszawa, 2001.
- [2] T. Chen, H. Chen. Approximation capability to functions of several variables, non-linear functionals, and operators by radial basis function neural networks. *IEEE Transactions on Neural Networks*, 6(4):904–910, 1995.
- [3] M. De Carvalho, M. Dougherty, A. Fowkes, M. Wardman. Forecasting travel demand: a comparison of logit and artificial neural network methods. *Journal of the Operational Research Society*, 49(7):717–722, 1998.
- [4] F. A. Gers, J. Schmidhuber, F. Cummins. Learning to forget: Continual prediction with lstm. 1999.
- [5] Z. Gong. Estimating the urban od matrix: A neural network approach. *European Journal of operational research*, 106(1):108–115, 1998.
- [6] Y. Hai, T. Akiyama, T. Sasaki. Estimation of time-varying origin-destination flows from traffic counts: A neural network approach. *Mathematical and computer modeling*, 27(9-11):323–334, 1998.
- [7] P. J. Huber. Robust estimation of a location parameter. *Breakthroughs in statistics*, strony 492–518. Springer, 1992.
- [8] S. S. Hussain, M. A. Sultan, S. Qazi, M. Ameer. Intelligent traffic matrix estimation using levenberg-marquardt artificial neural network of large scale ip network. *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, strony 1–5. IEEE, 2019.
- [9] D. Jiang, G. Hu. Large-scale ip traffic matrix estimation based on backpropagation neural network. *2008 First International Conference on Intelligent Networks and Intelligent Systems*, strony 229–232. IEEE, 2008.
- [10] D. Jiang, G. Hu. Large-scale ip traffic matrix estimation based on the recurrent multi-layer perceptron network. *2008 IEEE International Conference on Communications*, strony 366–370. IEEE, 2008.
- [11] D. Jiang, G. Hu. A novel approach to large-scale ip traffic matrix estimation based on rbf neural network. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, strony 1–4. IEEE, 2008.
- [12] S. Kikuchi, M. Tanaka. Estimating an origin-destination table under repeated counts of in-out volumes at highway ramps: use of artificial neural networks. *Transportation research record*, 1739(1):59–66, 2000.

- [13] D. Kim, Y. Chang. Neural network-based od matrix estimation from link traffic counts. *International Journal of Urban Sciences*, 12(2):146–157, 2008.
- [14] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [15] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, L. Lin. Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3875–3887, 2019.
- [16] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner. Microscopic traffic simulation using sumo. *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [17] M. Lorenzo, M. Matteo. Od matrices network estimation from link counts by neural networks. *Journal of Transportation Systems Engineering and Information Technology*, 13(4):84–92, 2013.
- [18] Z. Michalewicz. *Algorytmy genetyczne+ struktury danych*. Wydawnictwa Naukowo-Techniczne, 2003.
- [19] M. F. Møller. *A scaled conjugate gradient algorithm for fast supervised learning*. Aarhus University, Computer Science Department, 1990.
- [20] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, strony 105–116. Springer, 1978.
- [21] L. Mussone, S. Grant-Muller, C. Haibo. A neural network approach to motorway od matrix estimation from loop counts. *Journal of Transportation Systems Engineering and Information Technology*, 10(1):88–98, 2010.
- [22] M. Negnevitsky. *Artificial intelligence: a guide to intelligent systems*. Pearson education, 2005.
- [23] A. Omidvar, H. Shahhoseini. Intelligent ip traffic matrix estimation by neural network and genetic algorithm. *2011 IEEE 7th International Symposium on Intelligent Signal Processing*, strony 1–6. IEEE, 2011.
- [24] J. Ou, J. Lu, J. Xia, C. An, Z. Lu. Learn, assign, and search: Real-time estimation of dynamic origin-destination flows using machine learning algorithms. *IEEE Access*, 7:26967–26983, 2019.
- [25] F. Qian, G. Hu, J. Xie. A recurrent neural network approach to traffic matrix tracking using partial measurements. *2008 3rd IEEE Conference on Industrial Electronics and Applications*, strony 1640–1643. IEEE, 2008.
- [26] K. Remya, S. Mathew. Od matrix estimation from link counts using artificial neural network. *International Conference on Innovations in Civil Engineering*, strony 287–290, 2013.
- [27] H. Spiess. A gradient approach for the od matrix adjustment problem. 1:2, 1990.

- [28] D. Svozil, V. Kvasnicka, J. Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.
- [29] A. Szymański. Metody wyznaczania macierzy przepływów dla aglomeracji miejskiej, 2014.
- [30] T. Tieleman, G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.
- [31] F. Toqué, E. Côme, M. K. El Mahrsi, L. Oukhellou. Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, strony 1071–1076. IEEE, 2016.
- [32] Transportation Networks for Research Core Team. Transportation networks for research. <https://github.com/bstabler/TransportationNetworks>, (accessed May 22, 2020).
- [33] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transportation Research Part C: Emerging Technologies*, 13(3):211–234, 2005.
- [34] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [35] G. Yaldi, M. A. Taylor, W. L. Yue. Forecasting origin-destination matrices by using neural network approach: A comparison of testing performance between back propagation, variable learning rate and levenberg-marquardt algorithms. *Australasian Transport Research Forum*, 2011.
- [36] J. Zhao, H. Qu, J. Zhao, D. Jiang. Towards traffic matrix prediction with lstm recurrent neural networks. *Electronics Letters*, 54(9):566–568, 2018.