



Politechnika Wrocławskie

**Wydział Informatyki i Zarządzania**

kierunek studiów: Informatyka

specjalność: Internet i Technologie Mobilne (ITM)

**Praca dyplomowa - magisterska**

**PREDYKCJA WYDAJNOŚCI STRON WEBOWYCH  
Z WYKORZYSTANIEM METOD EKSPLORACJI  
DANYCH**

Jacek Arciszewski

Słowa kluczowe:  
eksploracja danych  
wydajność Web  
predykcja

Krótkie streszczenie:

Celem pracy jest predykcja wydajności stron webowych poprzez zastosowanie technik eksploracji danych. Badany był wpływ różnych metod wykorzystywanych w eksploracji danych (w szczególności grupowania i klasyfikacji) pod kątem jakości uzyskiwanych predykcji.

opiekun pracy dyplomowej	.....	.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego	.....	.....	.....
	Tytuł/stopień naukowy/imię i nazwisko	ocena	podpis

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:\**

*a) kategorii A (akta wieczyste)*

*b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)*

*\* niepotrzebne skreślić*

pieczętka wydziałowa

Wrocław 2017

## Streszczenie pracy dyplomowej

Celem niniejszej pracy magisterskiej było zbadanie możliwości predykcji wydajności stron webowych poprzez zastosowanie technik eksploracji danych. Na potrzeby badań opracowano stanowisko, wykorzystujące system pomiarowy MWING<sup>1</sup>, służący do rejestrowania parametrów połączeń TCP. System ten używany był wcześniej w podobnych badaniach, wykonanych na kampusie Politechniki Wrocławskiej (PWr), co pozwoliło na porównanie zmian w wydajności lokalnej sieci w stosunku do lat poprzednich. Narzędzie MWING zostało zainstalowane na trzech maszynach. Pierwsza z nich znajduje się w laboratorium Pracowni Systemów Internetowych i Mobilnych w Katedrze Informatyki i Zarządzania PWr. Pozostałe dwie zarejestrowano w ramach Usług Kampusowych, których działanie oparte jest na ogólnopolskiej, szerokopasmowej sieci optycznej PIONIER.

Pomiary wykonano dla 87 serwerów udostępniających pliki systemu operacyjnego Debian. Dla każdego z punktów końcowych przeprowadzono analizę, mającą na celu wyciągnięcie jak największej ilości informacji, mogących być użytecznymi z perspektywy badania wydajności Web. Wykonano również szczegółową analizę systemów autonomicznych (AS), na której podstawie udało się m.in. opracować graf ścieżek wędrówki pakietów do każdego z wybranych kierunków.

Na bazie zebranych pomiarów przeprowadzono serię badań obliczeniowych w środowisku Matlab, których celem była analiza zaobserwowanych przebiegów parametrów wydajności sieci. Zaobserwowano silną zależność przepustowości TCP od czasu odpowiedzi serwera (RTT). Prawo potęgowe jest słuszne dla median tych dwóch wartości. Wykonano aproksymację, której wysokie dopasowanie sięgające  $R^2 = 0,95$  sugeruje, że wartość RTT może zostać wykorzystana w prognozowaniu przepustowości serwera. Silna zależność została również potwierdzona przez późniejsze badania z użyciem technik eksploracji danych.

Do badań w aspekcie eksploracji danych użyto pakiet SPSS Modeler 18 (na darmowej licencji), a także SPSS Statistics 24 (licencja Politechniki Wrocławskiej).

Ponieważ przestrzeń badawcza okazała się bardzo szeroka, metodologię eksploracji danych wykorzystano dla dwóch, określonych potrzebą biznesową problemów. Pierwszym z nich był przypadek predykcji działania wielu serwerów, z których użytkownik zainteresowany jest wyborem tego, działającego najwydajniej. Jakość predykcji uzyskana dla tego przypadku sięgała nawet 83% i osiągnięta została dla algorytmów drzew decyzyjnych, sieci neuronowych, a także metody K-najbliższych sąsiadów (zależnie od badanego przypadku).

Drugi problem badawczy dotyczył predykcji wydajności w obrębie jednego serwera. Na podstawie statystycznej analizy wybrano dwa hosty, których działanie powinno być najgorzej i najlepiej prognozowane. W przypadku najgorszym wartość predykcji była faktycznie równoznaczna z wyborem losowym. Dla przypadku potencjalnie najlepszego, nie przekroczone wartości 50% predykcji, co w porównaniu z wyborem losowym (33% dla wybranych założeń) było nieznaczną poprawą.

W wyniku przeprowadzonych badań pozostawiono infrastrukturę pomiarową. Może ona zostać użyta w przyszłości na potrzeby wykonania predykcji wydajności dla innych założeń badawczych bądź powtórzenia tych, przedstawionych w niniejszej pracy magisterskiej.

---

<sup>1</sup> Borzemski L. i Nowak Z., „WIN: A Web Probing, Visualization, and Performance Analysis Service,” w *Web Engineering. ICWE 2004. Lecture Notes in Computer Science, vol 3140*, Berlin, Springer, 2004, pp. 601-602.

## Abstract

The purpose of this thesis was to investigate the predictability of Web performance by applying data mining techniques. For research purposes, a station was constructed using the MWING measuring system<sup>1</sup> to record TCP connection parameters. This system was previously used in similar studies done at the Wrocław University of Technology campus (WUT), which allowed comparison of changes in local network performance compared to previous years. The MWING tool was installed on three machines. The first one is in the Laboratory of Internet and Mobile Systems in the WUT Faculty of Computer Science and Management. The other two were registered as part of the Campus Services, which are based on the PIONIER national wideband optical network.

Measurements were made for 87 servers that provided Debian operating system files. For each of the endpoints, an analysis was carried out to extract the maximum amount of information that could be useful from the Web performance perspective. A detailed analysis of autonomous systems (AS) was also carried out.

On the basis of the collected measurements, a series of computational tests were conducted in the Matlab environment, aimed at analyzing the observed performance parameters of the network. A strong dependency between TCP bandwidth and server response (RTT) has been observed. Power law is right for the medians of these two values. An approximation was made, the value of the coefficient of determination  $R^2 = 0.95$  suggests that the RTT value can be used to forecast server throughput. The strong dependence was also confirmed by subsequent studies using data mining techniques.

SPSS Modeler 18 (free license) and SPSS Statistics 24 (license of Wrocław University of Technology) were used for data mining.

Since the research space has proved to be very broad, the data mining methodology was used for two specific business problems. The first one was the case of the prediction of the operation of many servers from which the user is interested in choosing the one that works best. The predictive quality obtained for this case was as high as 83% and was achieved for decision tree algorithms, neural networks, and K-nearest neighbors (depending on the case).

The second research problem was the performance prediction within a single server. Based on the statistical analysis two hosts were chosen, whose performance should have the worst and best predictions. In the worst case, the value of the prediction was equivalent to a random choice. For the best case, the value of 50% prediction was not exceeded, which was slightly improved compared to the random selection (33% for selected assumptions).

As a result of the research, the measurement infrastructure was left behind. It may be used in the future for performance prediction for other research assumptions or for repetition of those presented in this thesis.

---

<sup>1</sup> Borzemski L. i Nowak Z., „WING: A Web Probing, Visualization, and Performance Analysis Service,” in *Web Engineering. ICWE 2004. Lecture Notes in Computer Science, vol 3140*, Berlin, Springer, 2004, pp. 601-602.

# Spis treści

1.	Wprowadzenie .....	5
1.1	Geneza tematu pracy .....	5
1.2	Cel i zakres pracy .....	5
2.	Analiza bieżącego stanu literatury .....	7
2.1	Problem wydajności w sieci .....	7
2.2	Czynniki wpływające na wydajność w sieci .....	8
2.3	Protokół komunikacyjny TCP .....	11
2.4	Protokół HTTP .....	14
2.5	Metryki wydajności Web.....	15
2.6	Problemy pomiarów sieci .....	16
2.7	Sposoby mierzenia ruchu w sieci .....	17
2.8	Badawcze infrastruktury sieci .....	18
2.9	Problem predykcji wydajności .....	19
2.10	Wybrane wyniki prognozowania przepustowości w sieci .....	20
3.	Technika Data Mining .....	24
3.1	Metodyka Web Performance Mining .....	25
3.2	Wybrane algorytmy Data Mining .....	26
3.2.1	Algorytmy klasteryzacji .....	26
3.2.2	Algorytmy klasyfikacji i regresji.....	27
4.	Opracowanie stanowiska badawczego .....	30
4.1	Wymagania dotyczące badań .....	30
4.2	Selekcja serwerów i określenie ich własności .....	30
4.2.1	Selekcja serwerów .....	30
4.2.2	Położenie geograficzne .....	31
4.2.3	Internet Service Provider (ISP) .....	33
4.2.4	Mechanizm CDN.....	34
4.2.5	Protokół HTTP .....	35
4.2.6	Dystans geograficzny .....	36
4.2.7	Długość ścieżki IP/AS.....	36
4.3	Selekcja plików wykorzystywanych w pomiarach .....	37
4.3.1	Wstępna analiza Web .....	37
4.3.2	Dokonanie wyboru plików .....	39
4.4	Wstępna analiza zachowania wybranych serwerów .....	40
4.4.1	Dostępność plików .....	40
4.4.2	Porównanie zawartości pobieranych plików .....	40

4.5	Stanowisko pomiarowe – System MWING .....	41
4.5.1	Opis systemu .....	41
4.5.2	Adaptacja systemu do celów badawczych .....	41
4.6	Hosting agentów .....	42
4.6.1	Analiza możliwości .....	42
4.6.2	Sieć PIONIER .....	42
4.6.3	Wybór agentów .....	43
4.6.4	Analiza wybranych agentów .....	44
4.7	Narzędzia wykorzystane w badaniach.....	44
5.	Szczegółowe badania dotyczące wybranych serwerów.....	45
5.1	Analiza Systemów Autonomicznych (AS).....	45
5.1.1	Wyciągnięcie charakterystyk AS .....	45
5.1.2	Tworzenie grafu ścieżek AS .....	47
5.1.3	Tworzenie geograficznej mapy ścieżek AS .....	48
5.2	Analiza najdłuższych ścieżek AS .....	52
6.	Przeprowadzenie badań pomiarowych w rzeczywistych warunkach Internetu w systemie MWING .....	53
6.1	Czas wykonywania badań .....	53
6.2	Stabilność serii pomiarowych.....	54
7.	Analiza zebranych pomiarów .....	57
7.1	Problem brakujących obserwacji.....	57
7.2	Zależność RTT i przepustowości TCP .....	58
7.3	Problem pomiarów na maszynach wirtualnych.....	63
7.4	Porównanie poprawy wydajności sieci na Politechnice Wrocławskiej .....	65
8.	Przygotowanie do badań Data Mining.....	67
8.1	Definicja zadania predykcji .....	67
8.2	Estymacja współczynnika Hursta przy wyborze serwerów.....	68
8.3	Wykonanie kolejnych etapów Data Mining .....	75
8.3.1	Selekcja danych.....	75
8.3.2	Czyszczenie danych .....	76
8.3.3	Transformacja danych .....	77
8.4	Opracowanie skryptu IBM SPSS Modeler .....	78
9.	Analiza wybranych wyników Data Mining .....	82
9.1	Agent Koral .....	82
9.2	Agent WCSS.....	93
9.3	Agent PCSS .....	97
10.	Podsumowanie .....	100

## **1. Wprowadzenie**

### **1.1 Geneza tematu pracy**

Internet stał się kluczową technologią w funkcjonowaniu współczesnej cywilizacji. Ciężko wyobrazić sobie życie codzienne bez możliwości przepływu informacji, jaką obecnie umożliwia ogólnoszczególna infrastruktura sieci.

Użytkownik końcowy zazwyczaj ma do czynienia z najwyższą warstwą działania sieci – warstwą aplikacji. To w niej znajdują się uruchamiane w przeglądarce strony WWW, jak również programy codziennego użytku, które do prawidłowego działania wymagają połączenia internetowego.

Użytkownicy sieci posiadają obecnie szeroką możliwość wyboru usług, dlatego usługodawcy muszą zadbać o to, by ich witryówki w postaci stron internetowych, były jak najlepsze. Rozwiązania niewydajne, wolne, mogą przełożyć się na utratę potencjalnych klientów. Trzy podstawowe oczekiwania użytkowników to *dostępność, szybkość i bezpieczeństwo*.

Poruszając się w Internecie, często pojawia się problem wyboru. Prostym przykładem jest wykorzystanie technik mirroringu, czyli przypadku, w którym użytkownik ma do wyboru jeden spośród kilku serwerów, udostępniających ten sam plik. Każda z możliwości wyboru charakteryzuje się innymi parametrami związanymi zarówno ze sprzętem komputera zdalnego, jak i z zainstalowanym na nim oprogramowaniem. Oczywistym jest, że użytkownik kieruje się logiką, w której wybiera serwer, pozwalający mu na jak najszybsze pobranie pliku.

Niestety w takich przypadkach, użytkownik jest zazwyczaj ograniczony do wyboru losowego. Wybrany temat pracy ma za zadanie pomóc użytkownikowi w dokonaniu właściwej selekcji. Z pomocą narzędzi, opracowanych na potrzeby przeprowadzonych badań, użytkownik powinien móc trafnie określić, który z przedstawionych mu serwerów należy wybrać, w celu jak najszybszego pobrania pliku.

Prognozowanie może być również wykonane w obrębie jednego serwera. Sytuacja taka może wydarzyć się, jeżeli użytkownikowi zależy na jak najszybszym poruszaniu się po określonej witrynie Web. W tym przypadku opracowane narzędzia powinny wskazać, czy podana przez użytkownika para dnia jest cechującą się dużą wydajnością danego serwera.

Predukcja wydajności sieci może być uzyskiwana na wiele sposobów. Jednym z nich jest użycie popularnych na przełomie ostatnich lat technik eksploracji danych (*Data Mining*). Z racji, że Internet zbiera ogromne ilości informacji (*Big Data*), przetwarzanie ich staje się działaniem zbyt kompleksowym dla wielu metod bazujących na klasycznej statystyce. Dzięki technikom *Data Mining*, pozyskiwanie wiedzy z takiej skali danych staje się możliwe. To właśnie ta metodologia została wybrana w temacie pracy magisterskiej.

### **1.2 Cel i zakres pracy**

Celem pracy jest predykcja wydajności stron webowych poprzez zastosowanie technik eksploracji danych. Przedmiotem badanym jest wpływ różnych metod wykorzystywanych w eksploracji danych (w szczególności grupowania i klasyfikacji) pod kątem jakości uzyskiwanych predykcji.

Układ pracy został przedstawiony poniżej.

Rozdział pierwszy to wprowadzenie do tematu pracy magisterskiej. Zawiera cel i zakres pracy, a także uzasadnienie wyboru tematu.

W rozdziale drugim przeprowadzona jest analiza obecnej literatury związanej z tematem wydajności sieci. W kolejnych podrozdziałach starano się omówić poszczególne zagadnienia dotyczące funkcjonowania sieci. Przytoczone są tutaj przykładowe prace badawcze, które poruszają problem predykcji wydajności w Internecie.

Rozdział trzeci jest krótkim wprowadzeniem do metodologii Data Mining. Przedstawia kolejne etapy, które należy wykonać w celu wyciągnięcia wiedzy z danych. Przytoczony zostaje tutaj temat metodyki Web Performance Mining, czyli eksploracji danych w przypadku sieci Web. W ostatnim podrozdziale przedstawione są wybrane techniki eksploracji danych, wykorzystywane w dalszej części pracy. Są one udostępniane przez oprogramowanie IBM SPSS Modeler 18.

Rozdział czwarty szczegółowo opisuje etap opracowywania stanowiska badawczego, a także wstępnych analiz, które należało wykonać w celu selekcji serwerów udostępniających taką samą zawartość. Rozdział ten opisuje również system pomiarowy MWING, skonfigurowany na potrzeby pracy magisterskiej.

Rozdział piąty przedstawia szczegółowe badania dotyczące wybranych wcześniej serwerów. Są one związane przede wszystkim z analizą tras systemów autonomicznych (AS), znajdującymi się między punktami pomiarowymi (agentami), a wybranymi serwerami zdalnymi.

Rozdział szósty opisuje pomiary przeprowadzone w wybranych agentach. W rozdziale tym przedstawione zostają parametry cechujące wykonywane pomiary na każdym z nich, takie jak czas wykonywania i ciągłość.

W rozdziale siódmym zaprezentowane są wyniki analizy wykonanych pomiarów. Poprowadzona zostaje krótka dyskusja na temat uzupełnienia braków obserwacji. Następnie, wyniki pomiarów zostają porównane z badaniami prowadzonymi wcześniej na terenie Politechniki Wrocławskiej. Na końcu przedstawione są nieoczekiwane wyniki badań, które wymuszają pytanie dotyczące wybranych agentów pomiarowych.

Z racji, że technika Data Mining charakteryzuje się kilkoma następującymi po sobie etapami, budowa rozdziału ósmego nawiązuje do każdego z nich w kontekście otrzymanych danych pomiarowych. W rozdziale tym zostaje zdefiniowane zadanie predykcji. Jednym z otrzymanych w ten sposób podzadań jest prognozowanie wydajności dla najmniej i najbardziej przewidywalnego serwera – w celu ich wyboru oszacowany zostaje współczynnik Hursta. Rozdział porusza metodologię estymacji tego parametru.

Rozdział dziewiąty przedstawia wybrane wyniki wykorzystanych technik eksploracji danych. Wyniki te podlegać będą analizie, w wyniku której wyciągnięte zostaną wnioski na temat działania zbudowanych modeli.

Rozdział dziesiąty to podsumowanie. Zawiera on zbiór najważniejszych wniosków otrzymanych w wyniku badań przeprowadzonych na potrzeby pracy magisterskiej.

## **2. Analiza bieżącego stanu literatury**

Internet był przedmiotem badań w wielu publikacjach na przestrzeni ostatnich czterdziestu lat. Ze względu na stale rosnącą złożoność sieci, zagadnienie to stało się bardzo szerokie, zmuszając autorów badań do analizy szczegółowych przypadków zachowania Internetu. Z tego względu również, w wykonanej analizie literatury należało ograniczyć przestrzeń badawczą przeglądanych prac jedynie do tych, które związane były bezpośrednio z tematem wydajności w sieci.

W kolejnych podrozdziałach opisane zostały zagadnienia, które były poruszane w przeglądanej literaturze. Na początku przedstawiony zostanie ogólny problem wydajności w sieci oraz czynniki mające wpływ na wydajność rozumianą z perspektywy użytkownika końcowego. Następnie opisane zostaną dwa podstawowe protokoły z wymiany danych z perspektywy przeglądarki Web – TCP i HTTP. Kolejne podrozdziały skupią się na temacie tworzenia specjalnie dostosowanych infrastruktur, uruchamianych na potrzeby ułatwienia badań w sieci. Przedstawione zostaną również problemy związane z wykonywaniem tego typu pomiarów.

W ostatnich podrozdziałach przedstawione będą różne możliwości mierzenia wydajności w sieci. Powołując się na kolejne prace badawcze, starano się pokazać wykorzystywane dotąd podejścia, których wyniki będzie można skonfrontować z własnymi, powstały mi na potrzeby pracy magisterskiej.

### **2.1 Problem wydajności w sieci**

Zależnie od przyjętego punktu widzenia, wydajność w sieci może posiadać różne znaczenia. Dla administratora sieci termin ten może oznaczać na przykład wydajność obsługi kolejkowania pakietów na routeraх. Z drugiej strony, dla właściciela serwera Web, może być to odpowiednie przetwarzanie nadchodzących żądań, wysyłanych przez użytkowników.

W przypadku poruszanego tematu pracy, wydajność jest określana jako doznanie użytkownika końcowego, wykonującego swoje działania w Internecie. Może być to na przykład uzyskiwanie dostępu do strony WWW, pobieranie żądanego pliku bądź wykorzystywanie aplikacji, która wymaga użycia łączą internetowego. Zważywszy na fakt, że podane we wcześniejszym akapicie przykłady wpływają na doznanie użytkownika końcowego, pojęcie wydajności w sieci jest często ograniczane do tego, rozumianego przez jego osobę [1]. Z tych względów do opisania wydajności bardzo często operuje się pojęciem *Quality of Service* (QoS). Według jednej z używanych definicji, QoS oznacza zapewnienie stałej, przewidywalnej usługi dostarczania danych – zaspokajającej wymagane życzenia klienta.

Z perspektywy użytkownika końcowego poruszającego się w serwisie WWW, wydajność sieci określana jest jako czas między kliknięciem linku, a całkowitym załadowaniem żądanej strony (tzw. *load time*). Czas załadowania strony rozumiany jest jako różnica między wykonaniem żądania do serwera Web, a otrzymaniem danych przez klienta, przedstawianych użytkownikowi na ekranie monitora [2].

Badania wykazały, że długi czas ładowania strony jest jedną z głównych przyczyn opuszczenia serwisu Web przez użytkownika. Tolerowany czas jego oczekiwania jest zależny od różnych czynników, takich jak doświadczenie, wiek, czy indywidualne cechy charakteru [3]. Administratorzy serwerów powinni dokonać wszelkich starań, by ten czas był jak najmniejszy, ponieważ przekłada się on bezpośrednio na uzyskiwany przez serwis dochód.

Pomimo niewątpliwej poprawy technologicznej w Internecie na przełomie ostatnich lat, problem wydajności pozostaje nadal aktualny. W kolejnym podrozdziale przedstawione zostaną czynniki, które wpływają na wydajność w sieci.

## 2.2 Czynniki wpływające na wydajność w sieci

Czynników wpływających na wydajność komunikacji w sieci mogą istnieć potencjalnie tysiące, począwszy od jakości sprzętu i oprogramowania używanego przez użytkownika końcowego, skończywszy na infrastrukturze docelowego serwera Web [1]. Przytoczone końce trasy poruszania się pakietów danych (*end-to-end*) mogą mieć jedynie częściowy wpływ na to, co dzieje się na etapach pośrednich.

W pracach badawczych starano się określić, które z czynników mogą być odpowiedzialne za wydajność end-to-end podczas pobierania strony WWW przez Internet. W publikacji [4] wylistowano kolejne czynniki mogące wpływać na wydajność użytkownika końcowego:

- *Opóźnienie sieci*

Wielkość oraz zmienność opóźnień między punktami końcowymi może mieć znaczący wpływ na całkowity czas odpowiedzi w przesyłaniu danych z serwera do klienta. Opóźnienie sieci bardzo często dzielone jest na osobne kategorie:

- *Opóźnienie przetwarzania* – czas potrzebny routерom na przetworzenie nagłówka pakietu.
- *Opóźnienie kolejkowania* – czas, podczas którego pakiet oczekuje w kolejkach routingu.
- *Opóźnienie transmisji* – czas potrzebny na przepchnięcie pakietu bitów do fizycznego medium.
- *Opóźnienie propagacji* – czas potrzebny sygnałowi na dotarcie do celu.

- *Obciążenie serwera*

Jest to aktualna wielkość przetwarzania wykonywana przez procesor zdalnego hosta. Opisywana jest skalarem lub w procentach. Idealną wartością obciążenia na pojedynczym procesorze jest maksymalnie 1. Oznacza to, że CPU wykonuje wszystkie zadania jedno po drugim. W takim przypadku nie ma procesów oczekujących w kolejce. Jeżeli wartość ta zostanie przekroczona, odpowiedź serwera będzie opóźniona, a w skrajnych przypadkach może nie nastąpić wskutek odrzucenia żądania. Przyczyny obciążenia serwera mogą być różne [5]:

- *Niewystarczające zasoby serwera* – wynikają często z nieprawidłowej estymacji wielkości obciążenia serwera przez klientów i niedostosowania parametrów wynajętej maszyny do faktycznej liczby użytkowników.
- *Ataki na serwer* – mogą pojawić się w przypadku, gdy nie był on wystarczająco dobrze chroniony. Jednym z przykładów takich działań może być odmowa usługi (*Denial of Service - DoS*).
- *Zadania wykonywane po stronie serwera* – związane z jego utrzymaniem. Mogą być to zadania takie jak tworzenie kopii zapasowej, aktualizacja dziennych statystyk, czy działania programów do harmonogramowania zadań w systemie.
- *W środowisku hostingu wirtualnego* znajdują się użytkownicy o różnych potrzebach. Indywidualni klienci najczęściej wynajmują maszyny na potrzeby prostych stron internetowych. Z drugiej strony znajdują się aplikacje e-Commerce, które są kosztowne zarówno pod względem użycia

procesora, jak i łącza internetowego. Jeżeli takie strony cechują się dużą liczbą odwiedzin, następuje duży wzrost obciążenia serwera hostującego maszyny wirtualne.

- *Liczba obiektów i całkowity rozmiar*

Czas załadowania strony Web jest zależny od liczby wbudowanych w niej obiektów oraz ich całkowitego rozmiaru. Wpływają one na ilość pakietów, które zdalny host musi wysłać do użytkownika końcowego.

- *Właściwości użytych protokołów*

Można zaliczyć do nich chociażby różnice między poszczególnymi wersjami protokołu HTTP używanymi na stronach internetowych. Wpływ na wydajność użytkownika końcowego może mieć również ustawienie odpowiednich opcji tego protokołu.

- *Pora dnia*

Pora dnia to czynnik, który bezpośrednio przekłada się na wielkość ruchu w sieci. W analizach należy brać pod uwagę zarówno porę dnia po stronie zdalnego hosta (większe obciążenie serwera w czasie wzmożonego ruchu), jak też klienta końcowego. Ruch w sieci jest również zależny od dnia tygodnia – zazwyczaj można zaobserwować jego wzrost w dniach roboczych.

- *Buforowanie (Caching)*

Buforowanie służy do przechowywania zasobu blisko klienta w celu zmniejszenia długiego czasu oczekiwania. W obecnym środowisku Web, do buforowania wykorzystuje się najczęściej serwer pośredniczący (*proxy*). Jeżeli użytkownik końcowy żąda zasób od serwera pośredniczącego, ten sprawdza czy taki posiada. Jeżeli tak, wysyła go do klienta. W wypadku, gdy zasób nie zostanie znaleziony lub dostępna jest jedynie jego nieaktualna wersja, serwer *proxy* wysyła żądanie zasobu do serwera zdalnego, a następnie wysyła pobraną z niego zawartość do klienta końcowego [6].

- *Zawartość strony na wielu serwerach*

Zamiast wysyłać użytkownikowi całą zawartość strony z serwera bazowego, serwisy Web udostępniają możliwość pobrania wybranych lub wszystkich wbudowanych w niej obiektów z różnych serwerów. Takie podejście zmniejsza obciążenie na serwerze bazowym.

Rozwiązaniem zapewniającym podobną funkcjonalność jest zastosowanie mechanizmu Content Delivery Network (CDN) [1]. Jest to system rozmieszczonych w wielu punktach serwerów, których celem jest dostarczanie użytkownikowi końcowemu zawartości (takiej jak obrazy wysokiej rozdzielczości i streaming filmów) na podstawie:

- *Geograficznego położenia klienta,*
- *Pochodzenia strony internetowej,*
- *Obecnej dostępności i obciążenia serwera CDN.*

Kiedy użytkownik wysyła żądanie zasobu, który jest częścią CDN, żądanie to jest przekierowywane z bazowego serwera do serwera znajdującego się w CDN, będącego bliżej użytkownika końcowego. Podobnie jak w przypadku *proxy*, jeżeli na tym

serwerze nie znajduje się żądanego zasobu, CDN połączy się z serwerem bazowym w celu jego pobrania, a następnie wyśle go do użytkownika.

Wysoka adaptowalność technologii CDN sprawiła, że stała się ona bazowym elementem infrastruktury stron Web o wysokiej wydajności [1]. Najczęściej używane obecnie CDN to m.in. Amazon CloudFront, Akamai, CDN77, MaxCDN, BitGravity, CDNetworks, CacheFly, ChinaCache.

- *Czas zapytania DNS*

System nazw domenowych (*Domain Name System – DNS*) to baza danych rozproszona w całym Internecie. Służy ona do tłumaczenia używanych przez użytkowników nazw domenowych na adresy IP, pozwalając na osiągnięcie właściwej strony po wpisywaniu jej adresu URL (*Uniform Resource Locator*). Przykładowo, strona Politechniki Wrocławskiej *pwr.edu.pl* jest tłumaczona na adres IP 156.17.16.240.

Szybkość i niezawodność DNS jest kluczowa dla wydajności działania stron internetowych. Jest to pierwsza interakcja, którą użytkownik wykonuje w celu połączenia się ze stroną Web.

DNS może być także używany w celu kierowania użytkowników końcowych do najlepszych lokalizacji serwerów (takich jak centra danych czy dostawcy usługi chmury) bazując na ich lokalizacji. Prawidłowe zdefiniowanie tego mechanizmu przekłada się na znaczną poprawę wydajności dla użytkownika końcowego [1].

Wiele serwisów WWW polega na darmowych albo niskobudżetowych serwisach DNS udostępnianych przez dostawców usług internetowych, usługi hostingu lub serwisy rejestrujące nazwy domenowe. Poprawa DNS polega na upewnieniu się, że używany serwis zapewnia m.in.:

- *Ciągłość dostępu,*
- *Globalny zasięg,*
- *Skalowalność,*
- *Bezpieczeństwo.*

- *Przekierowania*

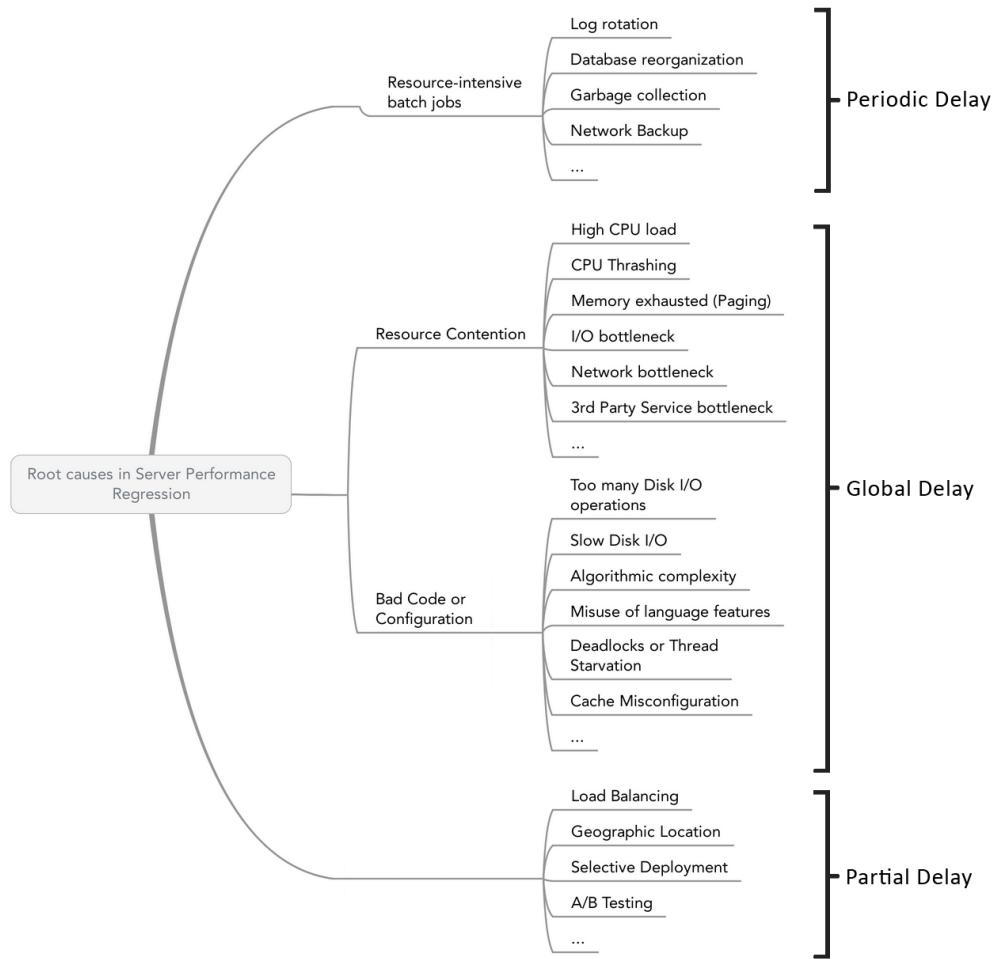
Niektóre strony internetowe używają przekierowań HTTP w celu przesłania żądań użytkownika na inny adres URL. Technika ta wykorzystywana jest m.in. do skracania adresów internetowych, a także przy przenoszeniu istniejącej strony internetowej na inny adres z jednoczesnym upewnieniem się, że stary adres nadal funkcjonuje.

- *Dynamiczna zawartość*

Czas, w którym serwer obsługuje żądanie obiektu statycznego powinien być mniejszy niż obsługa takiego samego obiektu, który jest generowany dynamicznie. Wynika to potrzeby przetworzenia żądania użytkownika przez serwis Web. Jest ściśle związane z wydajnością rozwiązań zaimplementowanych po stronie serwera.

Powyższe przykłady i wiele innych przedstawiają autorzy w publikacji [7], sprowadzając czynniki degradacji wydajności serwera do wykresu przedstawionego na Rysunku 2.1.

Jednym z przedstawionych wyżej czynników wpływających na wydajność w sieci było użycie odpowiednich protokołów. W kolejnych podrozdziałach przedstawione zostaną dwa podstawowe protokoły używane w transakcjach webowych.



Rysunek 2.1 Taksonomia przyczyn degradacji wydajności serwera  
 (źródło: [7])

### 2.3 Protokół komunikacyjny TCP

Podstawą działania Internetu jest zastosowanie dwóch protokołów: *Internet Protocol* (IP) oraz *Transmission Control Protocol* (TCP). Protokół IP jest fundamentalny dla adresacji i routingu w Internecie, podczas gdy TCP udostępnia niezawodny mechanizm transportowy, używany przez większość aplikacji internetowych, włączając w to transfer plików, czy uzyskiwanie dostępu do stron WWW [8].

Na podstawie działania tych dwóch protokołów stworzony został model TCP/IP, który z czasem stał się podstawą struktury Internetu. Głównym założeniem działania tego modelu jest podział komunikacji sieciowej na współpracujące ze sobą warstwy :

- *Warstwa aplikacji*

Jest to warstwa, w której mieszą się aplikacje wykorzystywane bezpośrednio przez użytkownika. Mogą być to na przykład przeglądarki internetowe czy serwer Web. Protokoły wykorzystywane przez tę warstwę to m.in. HTTP, FTP, SMTP czy DNS.

- *Warstwa transportowa (TCP)*

Warstwa ta gwarantuje niezawodność przesyłania danych. Zajmuje się również wysyłaniem informacji do znajdującej się wyżej warstwy aplikacji. Nawiązuje lub zrywa

połączenia między maszynami znajdującymi się w sieci. Dwa główne protokoły używane w tej warstwie to TCP i UDP.

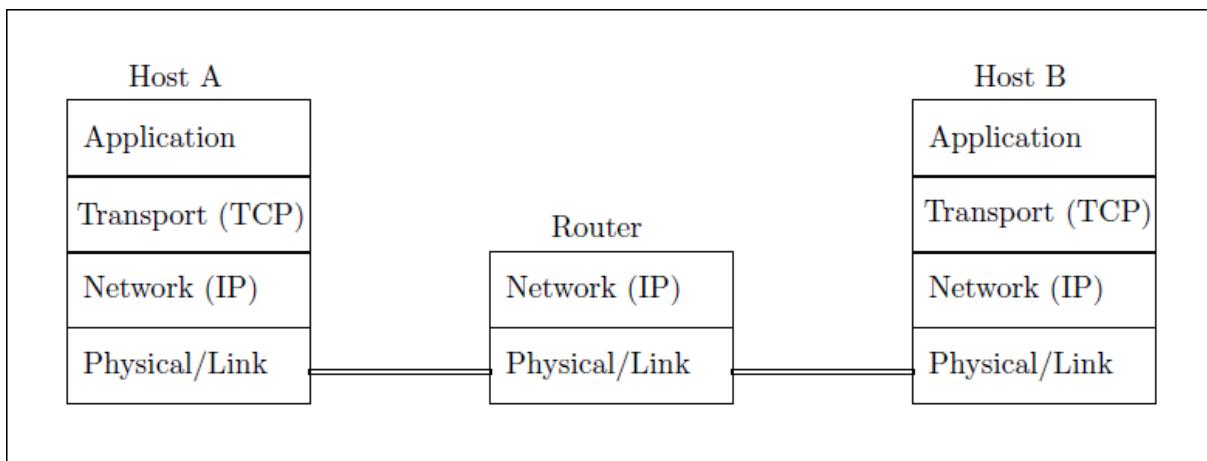
- *Warstwa Internetu (IP)*

Przetwarza datagramy zawierające adresy IP. Dzięki temu możliwe jest ustalenie odpowiedniej drogi do hosta docelowego, znajdującego się w sieci. W przypadku routerów, warstwa IP jest warstwą najwyższą. Routery to urządzenia używające odpowiednich protokołów trasowania w celu przekazania pakietów danych w kierunku miejsca destynacji.

- *Warstwa dostępu do sieci*

Jest to warstwa fizyczna. Zajmuje się odpowiednim przekazywaniem danych przez medium transmisyjne, łączące urządzenia sieciowe. Do urządzeń tych zaliczane są m.in. modemy oraz karty sieciowe.

Schemat modelu TCP/IP został przedstawiony na Rysunku 2.2.



Rysunek 2.2 Schemat modelu TCP/IP  
(źródło: [9])

Protokół TCP jest protokołem warstwy transportowej. To on zapewnia niezawodność transmisji w infrastrukturze Internetu. TCP ustanawia kanał komunikacyjny między procesami znajdującymi się na każdym z hostów. By to osiągnąć, sterowniki TCP dzielą ciąg danych sesji na segmenty, do których dołączają odpowiedni nagłówek. Po dołączeniu nagłówka IP, taki złożony pakiet jest następnie wysyłany do sieci.

Protokół TCP cechują następujące funkcjonalności [8]:

- *Transmisja unicast.* TCP bazuje na rodzaju transmisji typu *unicast* i wspiera wymianę danych między dokładnie dwoma maszynami.
- *Synchronizacja połączenia.* Zamiast narzucać stan w sieci do odpowiedniej obsługi połączenia, TCP używa synchronizacji między dwoma punktami końcowymi. Ten stan synchronizacji jest uzyskiwany za pomocą odpowiednich mechanizmów występujących przy nawiązywaniu połączenia między hostami. Synchronizacja ma zapewnić, że każde przejście stanu jednej maszyny będzie przekazywane i potwierdzane przez drugą stronę.
- *Niezawodność.* Zakłada, że strumień bitów przekazywany do sterownika TCP w jednym z punktów końcowych zostanie przesłany przez sieć i odebrany przez komputer zdalny jako niezmieniona sekwencja danych, identyczna z tą wyslaną

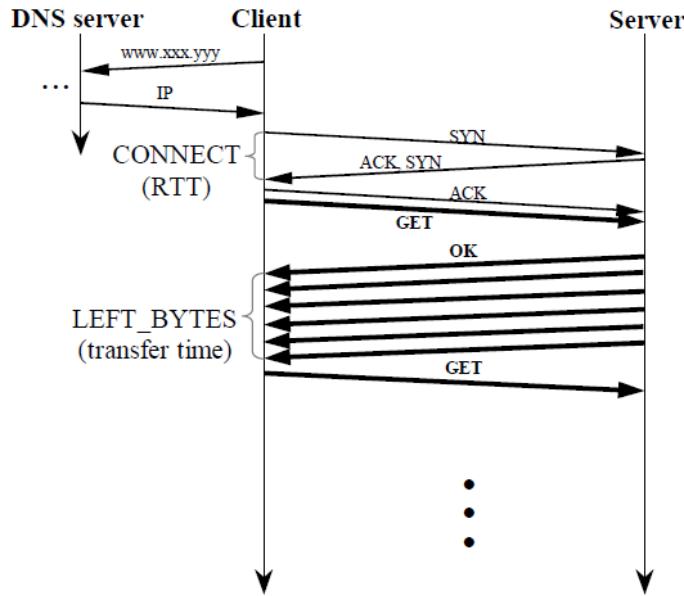
przez nadawcę. Protokół ma zapewnić detekcję segmentów, które zostały odrzucone, zduplikowane lub zmieniły swoją kolejność. Jeżeli jest to konieczne, pakiety takie należy wysłać ponownie.

- *Połny dupleks*. Informacje przesyłane są jednocześnie w obu kierunkach. Pozwala to punktom końcowym na wysyłanie i odbieranie pakietów w ramach pojedynczego połączenia TCP.
- *Streaming*. Pomimo, że TCP używa struktur pakietów do transmisji sieciowej, jest protokołem w pełni wspierającym streaming danych. Przykładowo, aplikacja TCP może wysłać kilka bloków danych, które zostaną odczytane przez zdalny komputer za pomocą pojedynczej operacji czytania. Wielkość segmentów danych używanych w sesji TCP jest ustalana na początku sesji. Wysyłający stara się użyć jak największego okna, dostosowując jego wielkość do ograniczeń narzuconych przez odbiorcę. Pod uwagę brany jest również rozmiar *MTU (Maximum Transmission Unit)*, czyli rozmiar największego datagramu, jaki można wysłać w warstwie protokołu komunikacyjnego. *MTU* jest aktualizowany, by dostosować się do zmian, które mogą wystąpić w sieci podczas nawiązanego połączenia TCP.  
W celu zapobiegania zatorom, stosowany jest mechanizm *Slow Start*. Polega on na tym, że w oknie TCP początkowo mieści się tylko jeden segment. Okno to jest zwiększane z każdym odebranym pakietem ACK.
- *Dostosowanie szybkości*. TCP jest protokołem, który zakłada dostosowanie szybkości transferu do warunków panujących w sieci. Jeżeli sieć wraz z odbiorcą posiada taką możliwość, nadawca TCP spróbuje wysłać większą ilość danych do sieci, by zająć to dostępne miejsce. Z drugiej strony, podczas przeciążenia, zmniejszy on szybkość wysyłania w celu polepszenia stanu sieci. Taka adaptacja pozwala na uzyskanie największej możliwej szybkości transferu danych bez powodowania utraty danych.

Pierwszą fazą sesji TCP jest ustanowienie połączenia. Wymaga ono użycia mechanizmu *three-way handshake*, który zapewnia, że obie strony mają jednoznaczne zrozumienie numerów sekwencji obowiązujących w komputerze zdalnym. Operacja ta jest następująca:

- Komputer lokalny wysyła do punktu zdalnego inicjalizujący numer sekwencyjny, używając pakietu SYN,
- Komputer zdalny odpowiada pakietem ACK o tym samym numerze sekwencyjnym, a także wysyła do komputera lokalnego swój własny numer sekwencyjny, używając pakietu SYN,
- Komputer lokalny odpowiada pakietem ACK o numerze sekwencyjnym otrzymanym z punktu zdalnego.

Po wykonaniu tych czynności, połączenie TCP jest otwarte. Można za jego pomocą wykonać na przykład operację pobrania zawartości strony Web. Ta operacja, uwzględniająca wysłanie zapytania DNS, została przedstawiona na schemacie widocznym na Rysunku 2.3.

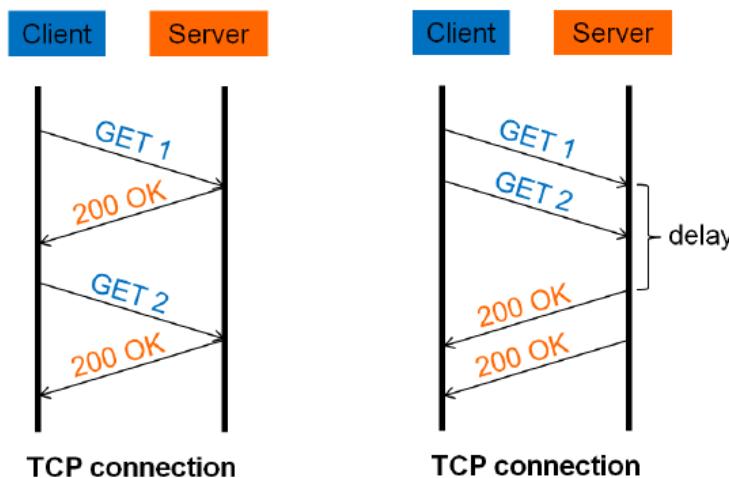


Rysunek 2.3 Prosta transakcja Web  
(źródło: [2])

#### 2.4 Protokół HTTP

Protokół Hyper-Text Transfer Protocol (HTTP) jest prawdopodobnie jednym z najbardziej znaczących protokołów używanych obecnie w Internecie. W przytoczonym wcześniej modelu TCP/IP znajduje się on w warstwie najbliższej użytkownikowi końcowemu – warstwie aplikacji. Protokół ten został stworzony w celu komunikacji między przeglądarkami i serwerami Web.

Obecnie najczęściej używaną wersją tego protokołu jest HTTP/1.1 [10]. Zazwyczaj działa on w oparciu o połączenie TCP. Na Rysunku 2.4 pokazano, jak przebiega przykładowe działanie protokołu HTTP/1.1. Kiedy klient i serwer nawiążą połączenie TCP, klient wysyła żądania GET do serwera w celu otrzymania zawartości. Jeżeli klient chce otrzymać kilka kawałków danych z serwera, muszą wykonać wiele takich żądań, w sposób sekwencyjny.



Rysunek 2.4 Działanie HTTP/1.1.  
Po lewej: Wersja podstawowa. Po prawej: Z użyciem mechanizmu pipelining.  
(źródło: [11])

Podstawową wadą takiego podejścia jest fakt, że żądania nie mogą być wysyłane przez klienta jednocześnie. Działanie to zakłada, że odpowiedzi otrzymywane z serwera będą ułożone w tej samej kolejności. Dlatego w tym przypadku, całkowity czas załadowania strony jest analogiczny do liczby żądań wykonywanych przez klienta.

Jednym z zaproponowanych rozwiązań ominienia powyższego problemu było wprowadzenie mechanizmu *pipelining*. Na powyższym rysunku można zauważyć jego działanie. Klient wysyła do serwera zdalnego dwa, następujące po sobie żądania, nie czekając na odpowiedź serwera na pierwsze z nich. Następnie otrzymuje odpowiedzi w tej samej kolejności, w której zostały wysłane żądania. Niestety, rozwiązanie to natknęło się na inny problem, jakim jest blokowanie *Head Of Line* (HOL). Jeżeli pierwsza odpowiedź jest znacznie większa od kolejnych (na przykład posiada w sobie duży plik), kolejne odpowiedzi serwera będą opóźnione [11].

Jednym ze sposobów ominienia blokowania HOL jest tworzenie kilku połączeń TCP z serwerem zdalnym. Dzięki temu, klient może pobierać jednocześnie różne elementy tej samej strony WWW. Niestety, takie działanie jest obciążające dla serwera, który musi utrzymać odpowiedni stan dla wszystkich nawiązanych połączeń.

Z racji przytoczonych wyżej wad, zdecydowano się na stworzenie kolejnej generacji protokołu HTTP, która miała je wyeliminować. Jest to protokół w wersji HTTP/2. Ulepszoną wydajność mają zapewnić: *binarność*, *kompresja danych*, *multipleksowanie*, *mechanizm priorytetów* oraz wysyłanie użytecznej zawartości przez serwer, która nie była wyraźnie żądana przez klienta [10].

## 2.5 Metryki wydajności Web

Podczas pomiarów wydajności w rozumieniu użytkownika końcowego, oczekującego jak najszybszego załadowania się strony WWW, należy zwrócić uwagę na poszczególne etapy połączenia oraz charakterystykę samej strony. Taka analiza wykorzystywana jest m.in. przez właścicieli serwisów Web w celu ulepszenia doświadczeń odwiedzających użytkowników (*user experience*).

Metryki, które sąbrane pod uwagę w takich analizach [12]:

- *Metryki dotyczące szybkości strony WWW*. Najkrócej można opisać je jako czas załadowania poszczególnych elementów strony internetowej, są to między innymi:
  - *Czas załadowania tytułu* – moment, w którym w zakładce przeglądarki widoczny staje się tytuł strony,
  - *Czas rozpoczęcia renderowania strony* – czas między wykonaniem żądania, a pojawiением się pierwszej zawartości strony,
  - *Czas interakcji* – czas między wykonaniem żądania, a momentem, w którym użytkownik może wchodzić w odnośniki, wpisywać dane w pola tekstowe i wykonywać temu podobne działania,
  - *Czas wyszukiwania DNS* – czas wyszukiwania IP podanego adresu domenowego w serwerach DNS,
  - *Czas nawiązywania połączenia* – czas potrzebny na nawiązanie połączenia między przeglądarką a serwerem,
  - *Czas do pierwszego bajta* – czas potrzebny pierwszemu bajtowi danych na osiągnięcie przeglądarki użytkownika,
  - *Czas do ostatniego bajta* – czas potrzebny na przesłanie ostatniego bajta danych od serwera zdalnego. Równa się z całkowitym czasem załadowania strony.

- *Metryki dotyczące złożoności strony WWW.* Związane są bezpośrednio z wydajnością rozwiązań zaimplementowanych na stronie, są to między innymi:
  - *Calkowita wielkość strony.* Liczba bajtów, które otrzymuje użytkownik przy żądaniu. Na wielkość tą składają się wszystkie obiekty na stronie, włącznie ze skryptami JavaScript i CSS.
  - *Calkowita liczba obiektów.* Wielkość i ilość obiektów należy odróżnić. Niezależnie od tego jak małe i skompresowane są elementy strony, wpływają one na czas jej ładowania.
  - *Obiekty firm trzecich.* Na stronie mogą istnieć elementy, które zależne są od działania zewnętrznych serwisów.
- *Metryki dotyczące zachowań użytkownika*

## 2.6 Problemy pomiarów sieci

Pomiar Internetu jest znacznym wyzwaniem dla badaczy. Oprócz przedstawionych we wcześniejszym podrozdziale czynników, należy brać pod uwagę wielkość obecnego Internetu, jego kompleksowość i ciągłą zmianę formy (choćby zmieniające się przewagi rozwiązań takich jak Web, e-Commerce czy peer-to-peer).

Internet nie został stworzony razem z infrastrukturą służącą do jego pomiarów. Dodatkowo, szczegółowe badania stają się znacznym problemem dla administratorów, z których większość nie chce dzielić się prywatnymi danymi na temat podlegających im sieci.

Problemów pomiarów sieci jest całe mnóstwo, przytoczone zostaną niektóre z nich [13]:

- *Slaba zwięzłość danych.* Zebrane dane na temat tej samej charakterystyki przy użyciu różnych metod nie muszą się zgadzać. Problem może pojawić się już przed wykonaniem pomiarów, w przypadku różnych perspektyw autorów na temat wartości mierzonych. Do słabej zwięzłości danych należy uwzględnić również problem synchronizacji zegarów, który mimo dostępnych rozwiązań jest trudny do osiągnięcia.
- *Niedokładne narzędzia.* Zależnie od przyjętej metodologii zbierania danych, narzędzia pomiarowe mogą być blokowane lub zachowywać się niekontrolowanie, co bezpośrednio wpływa na błąd pomiarowy. Problemem staje się tutaj też wspomniana wcześniej prywatność, która ogranicza możliwości walidacji wyników.
- *Reprezentatywność.* Ciągłe zmiany następujące w Internecie sprawiają, że wykonywane badania bardzo szybko tracą swoją reprezentatywność. Jednakże, są pewne własności, które wydają się niezmienne, np. samopodobieństwo lub długookonowość.
- *Reprodukcia wyników.* Społeczność internetowa nie wykształciła kultury reprodukcji badań. Narzędzia, które zostały opracowane na potrzeby pomiarowe zostają najczęściej w rękach badaczy eliminując możliwość porównania wyników przez innych użytkowników, w różnych warunkach sieci.
- *Ilość danych.* Problemem pomiarów w Internecie jest jego warstwowość. Pełne zrozumienie wszystkich jego aspektów wymaga przeprowadzenie badań w każdej z tych warstw. Za sprawą wielkości ruchu, który obecnie panuje w sieci, metody statystyczne są ograniczane przez ilość rekordów, znajdujących się w bazach danych (*Big Data*). By poradzić sobie z tym problemem, poszukuje się nowych możliwości (np. *Data Mining*) lub badania ograniczane są do zamkniętych środowisk.

## 2.7 Sposoby mierzenia ruchu w sieci

Narzędzia pomiarowe mierzące ruch w sieci mogą być zaklasyfikowane na różne sposoby [9]:

- *Narzędzia sprzętowe lub oparte na oprogramowaniu*

Jest to główny podział stosowany w klasyfikacji narzędzi pomiarowych. Narzędzia sprzętowe to urządzenia, stworzone specjalnie na potrzeby zbierania i analizy danych sieciowych. Często jest to rozwiążanie drogie, zależnie od liczby interfejsów sieciowych, typów kart sieciowych i pojemności.

Narzędzia oparte na programowaniu zazwyczaj bazują na modyfikacjach interfejsów sieciowych na poziomie jądra systemu w celu stworzenia maszyn do przechwytywania pakietów. Jednym z najczęściej używanych narzędzi w tym zakresie jest *tcpdump*, stworzone do przechwytywania pakietów TCP/IP. Innym sposobem opartym na oprogramowaniu jest zbieranie logów dostępu zapisanych przez serwery Web. Pliki te posiadają informacje na temat każdego żądania wysyłanego przez klienta. Przetwarzanie ich może dać wgląd na temat obciążień serwera, bez potrzeby szczegółowej analizy na poziomie pakietów danych.

- *Pomiary pasywne lub aktywne*

Pasywne narzędzia pomiarowe służą do obserwacji i zapisywania ruchu pakietów w działającej sieci, bez generowania dodatkowego ruchu. Takie narzędzie jest nieinwazyjne. Większość narzędzi pomiarowych w sieci należy do tej kategorii.

Pomiary aktywne używają pakietów generowanych przez narzędzie pomiarowe w celu badania Internetu. Przykładem takiego podejścia jest narzędzie *ping* używane do mierzenia opóźnienia sieci w wybranym kierunku. Innym narzędziem aktywnym jest *traceroute*, pozwalające na odczytanie ścieżek routingu dla wysłanego w kierunku zdalnego hosta pakietu danych.

- *Analiza Online lub Offline*

Niektóre narzędzia wykorzystywane do analizy wspierają analizę zbieranych danych w czasie rzeczywistym, zazwyczaj z odpowiednią wizualizacją danych. Większość narzędzi sprzętowych działa w sposób Online.

Inne narzędzia pomiarowe są stworzone z zamiarem zbierania danych i ich przechowywania. Analiza tych wartości jest wykonywana później, za pomocą specjalnie przygotowanego oprogramowania.

- *Pomiary LAN lub WAN*

Pomiary lokalne są łatwiejsze do zmierzenia, z wielu względów. Sieci te są zazwyczaj zarządzane przez jedną organizację, co przekłada się na wyeliminowanie problemów związanych z bezpieczeństwem. Dodatkowo, każdy pakiet przesyłany w sieci LAN jest widziany przez wszystkie hosty. Takie badania wykonywane były przede wszystkim w początkowych fazach literatury dotyczącej pomiarów ruchu w sieci.

Z czasem analizę rozszerzono na zbieranie danych i analizę środowiska sieci rozległej (WAN). Problemy pojawiające się w przypadku takich badań zostały przytoczone w podrozdziale wcześniejszym.

- *Zależnie od poziomu protokołu*

## 2.8 Badawcze infrastruktury sieci

W pracach badawczych bardzo często wykorzystywane są specjalnie przygotowane infrastruktury sieci. Zaletą takich środowisk są znacznie większe możliwości analityczne zebranych wyników, wynikające z dostępu do każdego ze skonfigurowanych agentów pomiarowych (*vantage points*). Większość z tych punktów zlokalizowana jest w środowisku użytkowników końcowych, w strategicznych lokalizacjach na mapie świata.

Dzięki ogromnej liczbie takich stanowisk, platformy te mogą zapewnić dane na temat kluczowych metryk wydajności z perspektywy użytkownika końcowego. Informacje te są przydatne zarówno operatorom sieci w celu ulepszenia ich usług, jak również ich użytkownikom.

Infrastruktury te mogą przybierać różne formy. Mogą być to rozwiązania sprzętowe i oparte na oprogramowaniu. Mogą używać zarówno pomiarów pasywnych, jak i aktywnych, a także być rozwiązaniami biznesowymi lub akademickimi. Poniżej zostaną przedstawione niektóre z nich [14]:

- *SamKnows*

Platforma ta rozpoczęła swoje działanie w roku 2008. W domach przystępujących do badania użytkowników rozmieszcza się sprzętowe sondy. Sondy te wykonują aktywne testy w trakcie dnia, kiedy łącze nie jest aktualnie wykorzystywane przez użytkownika. Użycie rozwiązania sprzętowego zapewnia dużą trafność wyników, ponieważ działanie sond nie jest uzależnione od specyfikacji maszyn uczestników.

Infrastruktura ta wspiera różne możliwości wykonywania testów, zarówno w warstwie sieciowej, jak i w warstwie aplikacji [15].

- *Bismark*

Jest platformą, która również bazuje swoje działanie na pomiarach sprzętowych. Celem tej infrastruktury jest badanie specyficznych zjawisk w sieci. Jednym z nich jest na przykład *bufferbloat*, czyli szkodliwe działanie, jaki mogą mieć na wydajność sieci zbyt duże wielkości buforów [16]. Pomimo, że używa on bardzo podobnych mechanizmów do *SamKnows*, nie jest nastawiony na konkurencję z innymi tego typu rozwiązaniami. Jego głównym zadaniem jest analiza wybranych zachowań sieci w maksymalnie szczegółowy sposób.

- *RIPE Atlas*

Rozwiązanie to również używa sprzętowych sond, jednak w odróżnieniu od dwóch powyższych przypadków, nie muszą być one instalowane bezpośrednio przy użytkowniku końcowym. Instalację takiego sprzętu można wykonać w dowolnym miejscu sieci dostawcy internetowego.

Przykładowymi możliwościami sond są m.in. badania czasów z narzędzia *ping* lub *traceroute* do wybranych przez uczestnika kierunków [17].

- *Ookla*

Platforma ta używa strony internetowej *speedtest.net*, pozwalającej użytkownikom na wykonanie testów sprawdzających wydajność ich połączenia internetowego. Wyniki tych testów są również zbierane po stronie serwera, a następnie udostępniane do dalszych badań. Wykonywane pomiary opierają się w pełni na oprogramowaniu, mogą więc być zniekształcone przez błędную konfigurację maszyny użytkownika bądź słabą wydajność sieci domowej.

- *PlanetLab*

Maszyny należące do tej platformy instalowane są przede wszystkim w instytucjach badawczych, lecz także w pobliżu centrów routingu. Celem PlanetLab jest rozmieszczenie węzłów łączących miejsca bliskie kręgosłupa sieci [18].

Wszystkie maszyny PlanetLab operują na pakiecie, który dystrybuowany jest na system operacyjny Linux. Posiada on potrzebne narzędzia do prawidłowego działania. Pakiet ten nazywa się *MyPLC*.

Kluczowym założeniem tego oprogramowania jest wsparcie do przydzielania tzw. *slice*, czyli zasobów do aplikacji. Pozwala to na uruchomienie aplikacji w większości maszyn rozproszonych po całym świecie. Usługi te to m.in.: udostępnianie plików, badania jakości obsługi (QoS) lub szeroko rozumiane pomiary sieci.

## 2.9 Problem predykcji wydajności

Predykcja wydajności jest jednym z kluczowych zagadnień w Internecie. Tak jak zostało wspomniane we wcześniejszych podrozdziałach, wydajność ta może być określana przez różne parametry, zależnie od przyjętego punktu widzenia. Wiele z prac rozumie ją przez opóźnienie transmisji (RTT) lub przepustowość (*throughput*) danych w protokole TCP [19].

Wcześniejszta znajomość wydajności sieci, rozumianej przez użytkownika końcowego jako czas transferu pliku, może przełożyć się na otrzymanie żądanej zawartości w sposób możliwie efektywny. Z sytuacją taką można się spotkać na przykład podczas używania mechanizmu mirroringu, w którym użytkownik wybiera jeden z zaproponowanych mu plików o takiej samej zawartości, znajdujących się na różnych serwerach zdalnych [20].

Predykcja czasu pobierania może okazać się kluczowa dla satysfakcji użytkownika końcowego. Szczególne znaczenie ma ona w przypadkach, gdy klient żąda transferu pliku o dużej zawartości. Może być to na przykład wysokiej rozdzielczości film, którego pobranie może zająć do kilkunastu godzin.

Predykcja może być krótkoterminowa lub długoterminowa. Krótkoterminowa prognoza wymaga ciągłych pomiarów wydajności na daną chwilę, co może przełożyć się na generowanie zbyt dużego ruchu w sieci. W przypadku prognoz długoterminowych, używany jest pewien zbiór historycznych danych, zebranych wcześniej w odpowiednich odstępach czasu.

Metody predykcji dzielimy na dwie grupy [21]:

- Analityczne (*formula-based*)

Metody analityczne używają wzorów matematycznych opracowanych na potrzeby badania danego rozumienia wydajności. W wyliczeniach wykorzystują niezależne zmienne. Zmiennymi tymi mogą być na przykład RTT, procent utraty pakietów danych czy wielkość okna MTU. Niestety, dynamika sieci oraz ograniczenia wynikające z używanego sprzętu komplikują zbieranie aktualnych i dokładnych pomiarów sprawiając, że modele analityczne muszą być na bieżąco sprawdzane.

- Historyczne (*history-based*)

Metody te używają wykonanych wcześniej serii pomiarowych w celu prognozowania przyszłych wartości wyjściowych. Wykorzystywane w predykcji pomiary mogą być zbierane zarówno w sposób aktywny, jak i pasywny. Przykładowym podejściem historycznym do predykcji wydajności jest użycie metodologii Data Mining (opisanej w kolejnym rozdziale).

Metody historyczne można podzielić na dwie podkategorie, w których używa się technik klasyfikacji lub regresji. Klasyfikacja prognozuje badany przypadek do jednej z ustalonych wcześniej klas badanego zjawiska, podczas gdy techniki regresyjne przewidują numeryczną wartość określonego współczynnika wydajności [19].

W ostatnim podrozdziale opisane zostaną wybrane podejścia wykonania predykcji wydajności sieci w literaturze.

## 2.10 Wybrane wyniki prognozowania przepustowości w sieci

Wykonanie wysokiej jakości predykcji jest pożądane z punktu widzenia zarówno biznesowego, jak i badawczego. Z tego powodu, w literaturze znalazło się bardzo dużo prac badawczych, starających się w jak najlepszy sposób wykonywać predykcję wydajności, rozumianej najczęściej jako wydajność Web – czas załadowania strony bądź pliku przez użytkownika.

Przykład użycia metod analitycznych przedstawiony został m.in. w publikacji [22]. Założony w pracy model posiada  $N+1$  węzłów, z czego jeden z nich jest węzłem wejściowym, który odbiera i wysyła żądania do węzłów w liczbie  $N$ . W publikacji tej założono, że odbierane pakiety danych mają rozkład Poissona, co przy wykonaniu odpowiednich założeń, zostało potwierdzone przez inne prace badawcze [23]. Węzeł wejściowy przedstawia element logiki biznesowej, odpowiedzialny za odbieranie żądań, zaś kolejne węzły przedstawiają do baz danych lub innych serwisów, które kolejno przetwarzają żądanie.

Przy użyciu powyższego modelu wyciągnięto wzory dla wyrażenia średniej czasu odpowiedzi oraz aproksymacje jego wariancji. W przypadku testów wykonanych na rzeczywistych serwerach Webowych otrzymano wyniki, których margines błędu w większości przypadków nie przekroczył 10%. Badania zostały wykonane w roku 2007 i niestety nie udało się znaleźć ich kontynuacji. Z racji przytoczonych wcześniej wad metod analitycznych, ich działanie w aktualnych warunkach Internetu może być znacznie gorsze.

Jedna z metod historycznych została przedstawiona w publikacji [24]. Zastosowana tam technika próbuje przychodzące segmenty protokołu TCP i wykorzystuje ekspercką wiedzę na temat wzorców w połączeniach TCP, w celu predykcji wartości przepustowości. Wybrane przez autorów podejście może okazać się wartościowe dla celów pomiarów sieci, w mniejszym stopniu dla użytkownika końcowego.

We wcześniejszym podrozdziale przytoczono przykład problemu użytkownika końcowego, tj. wybór jednego z wielu replikowanych plików znajdujących się na różnych serwerach. W publikacji [25] stworzono infrastrukturę i wykonano badania, których zadaniem była odpowiedź na przedstawiony problem – predykcja przepustowości end-to-end.

Na potrzeby pracy stworzono serwer GridFTP, który zapisywał wydajność otrzymaną przy każdym transferze wykonywanym z serwera zdalnego. Informacje te, razem z metadanymi na temat charakteru połączenia, były wartościami wejściowymi dla stworzonych predyktorów. Predyktory można było podzielić na trzy różne podejścia, zakładające prognozy przy użyciu podstawowych funkcji matematycznych: technik opartych na średniej oraz medianie, a także modelów autoregresji (dokładnie ARIMA).

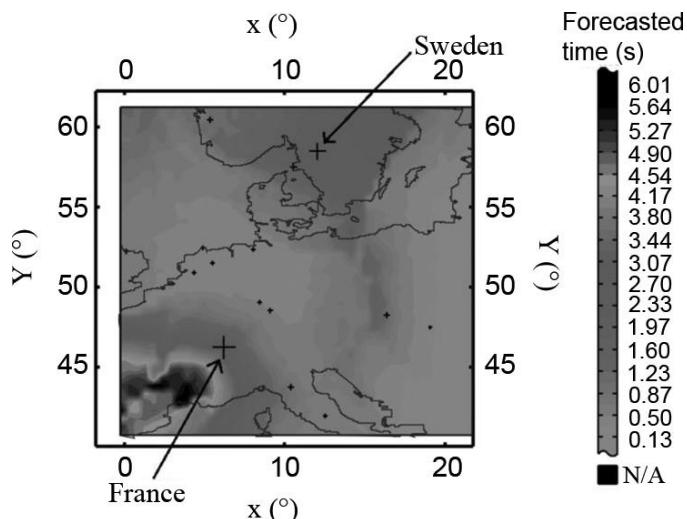
W publikacji wykonano również filtrację zebranych danych pomiarowych w celu sprawdzenia zmian jakości predykcji. Była to filtracja niezależna od kontekstu (na przykład użycie ostatnich pięciu pomiarów) lub zależna (choćżby wybieranie danych opisujących transfery plików o podobnej wielkości).

W wynikach przeprowadzonych badań zauważono, że nawet najprostsze, wykorzystane techniki predykcji posiadają względny błąd rzędu maksymalnie 25%. Stwierdzono również, że znaczną poprawę wyników uzyskano przez filtrację danych na podstawie ich wielkości, a także przy użyciu plików o wielkości przynajmniej 100 MB.

Najbardziej interesującą pozycją z perspektywy wykonywanej pracy magisterskiej, są badania przeprowadzone w warunkach sieci Politechniki Wrocławskiej [2] [19] [26] [27]. W publikacjach tych podjęto próby predykcji wydajności, rozumianej jako czas transferu zawartości ze strony Web.

Na potrzeby przytoczonych prac wykorzystano autorski system pomiarowy MWING, wykonujący aktywne pomiary w sieci. Agenci tego systemu zostali zainstalowani w środowiskach akademickich, trzech znajdujących się w Polsce – Wrocław, Gliwice i Gdańsk, a także w Las Vegas, w Stanach Zjednoczonych. Każdy z agentów używany był do pobierania tego samego pliku (*rfc1945.txt* o wielkości około 130 kB), z listy serwerów identycznej dla każdego węzła. Eksperymentalnie stwierdzono, że nienakładające się pomiary można było wykonywać co pół godziny. Stały odstęp między pomiarami pozwalał na wykorzystanie większej ilości technik związanych z seriami czasowymi.

W pierwszym z podejść predykcji zastosowano metodę *Turning Bands* (TB), używaną przede wszystkim w geostatystyce [19]. Jest to wielowymiarowy generator liczb losowych do symulacji przestrzennie skorelowanych pól losowych. Efekt końcowy takich predykcji został przedstawiony na mapie, która może zostać użyta do prognozowania wydajności serwerów, które nie tylko znajdują się w badanej próbce, a w całym badanym obszarze. (Rysunek 2.5).



Rysunek 2.5 Przykładowa mapa czasu pobierania w dniu 1 lipca 2008 r. o godzinie 6:00.  
(źródło [19])

Pomimo, że otrzymane w ten sposób predykcje cechują się stosunkowo wysoką wartością średniego błędu względnego, praca ta wskazuje na możliwości wykorzystania algorytmów stosowanych w innych dziedzinach do problemu związanego z prognozowaniem wydajności sieci.

Kolejne podejście zakładało użycie mechanizmów Data Mining, opisanych szczegółowo w kolejnym rozdziale pracy magisterskiej. W publikacji [2] zaproponowano dwuetapowe podejście do predykcji wydajności Web.

Pierwszy z etapów polegał na użyciu algorytmów klasteryzacji dla zebranych danych, pozwalających na wyciągnięcie wiedzy na temat różnych stanów danej sieci. Wielkości użyte do tworzenia tych klastrów to RTT, przepustowość, dzień tygodnia oraz pora dnia. Klastry te miały za zadanie wytypować charakterystyczne zachowania, którymi cechował się określony serwer. Przykład utworzonych klastrów został przedstawiony na Rysunku 2.6.

CLUSTER ID	Relative Cluster Size (%)	Cluster Description
6	17.73	DAY-OF-WEEK is predominantly 2, TIME-OF-DAY is predominantly 3, RTT is predominantly 1, and THROUGHPUT is high.
2	12.63	TIME-OF-DAY is predominantly 9, DAY-OF-WEEK is predominantly 7, RTT is predominantly 2, and THROUGHPUT is medium.
0	11.92	TIME-OF-DAY is predominantly 3, DAY-OF-WEEK is predominantly 7, RTT is predominantly 1, and THROUGHPUT is medium.
7	11.61	DAY-OF-WEEK is predominantly 1, TIME-OF-DAY is predominantly 6, RTT is predominantly 1, and THROUGHPUT is medium.
1	11.53	DAY-OF-WEEK is predominantly 7, TIME-OF-DAY is predominantly 4, RTT is predominantly 1, and THROUGHPUT is high.
5	9.73	RTT is predominantly 4, THROUGHPUT is low, TIME-OF-DAY is predominantly 7, and DAY-OF-WEEK is predominantly 4.
4	9.41	DAY-OF-WEEK is predominantly 4, TIME-OF-DAY is predominantly 5, RTT is predominantly 2, and THROUGHPUT is medium.
3	8.78	TIME-OF-DAY is predominantly 1, DAY-OF-WEEK is predominantly 4, RTT is predominantly 1, and THROUGHPUT is medium.
8	6.66	RTT is predominantly 3, DAY-OF-WEEK is predominantly 2, TIME-OF-DAY is predominantly 9 and THROUGHPUT is low.

Rysunek 2.6 Charakterystyki utworzonych klastrów  
(źródło [2])

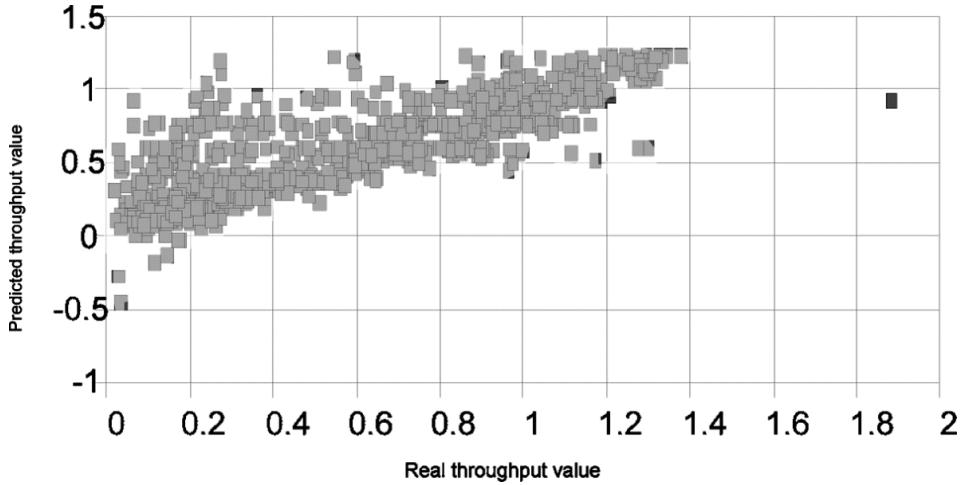
Na podstawie algorytmów klasyfikacji, w drugim etapie zaproponowanego podejścia, można było określić, do którego z wcześniej ustalonych klastrów należeć będzie pomiar, wykonywany w określonym dniu tygodnia, o określonej porze. W publikacji zastosowano algorytm drzewa decyzyjnego, którego wynik działania jest bardzo łatwy do wizualizacji poprzez zbiór kolejnych reguł, dzielących pierwotny zbiór danych w taki sposób, by jak najlepiej rozróżnić obserwacje pod względem wartości wyjściowej.

Przeprowadzone badania wykazały, że dla przypadku serwera wykazującego się największym samopodobieństwem (wybranego statystycznie, za pomocą współczynnika Hursta), zastosowana technika wykazała się dużą trafnością – około 80% poprawnie zaklasyfikowanych danych.

Trafność klasyfikacji została dodatkowo zweryfikowana przy użyciu różnych zbiorów danych, służących do uczenia algorytmów klasyfikacji. Wykorzystano mechanizmy zarówno rozszerzającego, jak i przesuwającego się okna czasowego. Wykazano, że dane zebrane z jednego tygodnia są niewystarczające do uzyskania zadowalających wyników predykcji. Pozostałe wyniki wskazywały, że znacznie lepszą wartością prognozowania cechowały się okna 6-tygodniowe i 20-tygodniowe. Te ostatnie były jednak niepraktyczne ze względu na dynamicznie zmieniającą się charakterystykę Internetu.

Powyższa praca zakładająca użycie metod eksploracji danych używała technik klasyfikacji. Metody regresyjne zostały opisane w innej publikacji [26]. Porównano w niej wybrane techniki dostępne w profesjonalnych systemach implementujących mechanizmy Data Mining: Microsoft SQL Server i IBM Intelligent Miner. Najlepsze wyniki predykcji wartości przepustowości TCP uzyskano dla opatentowanego algorytmu Transform Regression, należącego do pakietu IBM. Błąd średniokwadratowy predykcji dla tego przypadku był równy

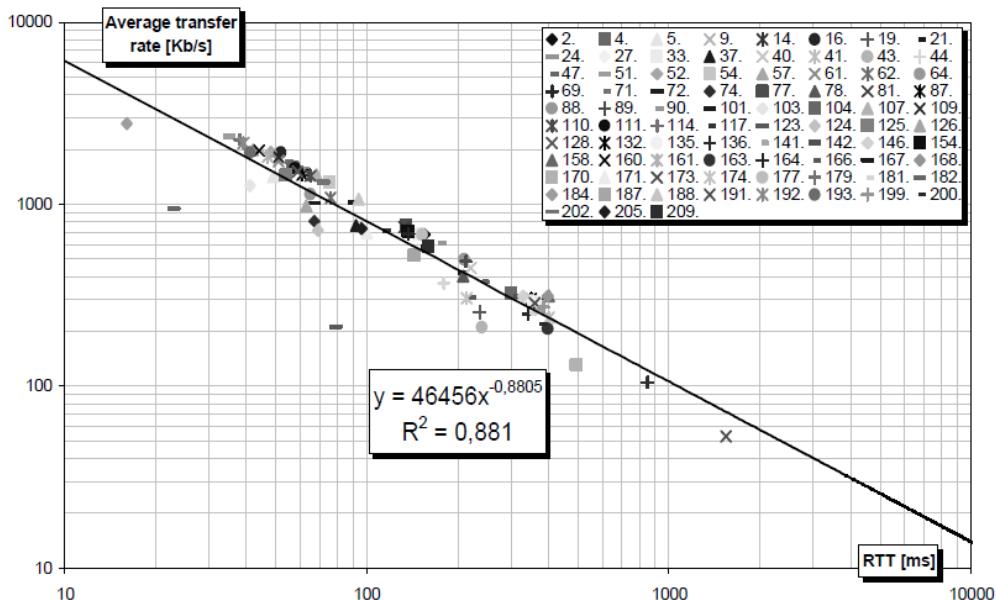
0. Wykres predykcji przepustowości TCP dla algorytmu IBM Transform Regression przedstawiono na Rysunku 2.7.



Rysunek 2.7 Predykcja przepustowości dla algorytmu Transform Regression dostępnym w pakiecie Intelligent Miner (źródło [26])

W ostatniej publikacji poruszającej temat badań wydajności sieci w obrębie Politechniki Wrocławskiej, dla każdego z wybranych serwerów końcowych wyliczono mediany otrzymanej w pomiarach przepustowości TCP oraz czasu odpowiedzi serwera w nawiązywanych połączeniach (RTT). Wartości te zostały zestawione na wykresie zaprezentowanym w skali logarytmicznej (Rysunek 2.8).

Dla rozkładu wartości przepustowości i RTT wykonano aproksymację funkcji potęgowej (będącej prostą w podziale logarytmicznym przedstawionym na wykresie). Zaobserwowane wyniki w postaci wysokiego dopasowania  $R^2$  sugerowały, że można przyjąć słuszność prawa potęgowego dla tych dwóch wartości. Takie stwierdzenie wskazuje, że znajomość wartości odpowiedzi serwera (łatwiej do otrzymania na podstawie narzędzi takich jak *ping*) może być wystarczająca do określenia jego przepustowości.



Rysunek 2.8 Rozkład wartości mediany średniej szybkości transferu a RTT (źródło [27])

### 3. Technika Data Mining

Eksploracja danych (*Data Mining*) to proces przetwarzania ogromnych zbiorów danych w celu odkrycia występujących w nich wzorców i trendów, których dostrzeżenie wykracza ponad prostą analizę. Eksploracja danych używa wymyślnych algorytmów matematycznych w celu dzielenia tych zbiorów i ocenienia prawdopodobieństwa przyszłych zdarzeń. Jest znana również jako odkrywanie wiedzy z baz danych (*Knowledge Discovery in Data – KDD*) [28].

Kluczowymi założeniami techniki eksploracji danych są:

- *Automatyczne odkrywanie wzorców*

Eksploracja danych wykonywana jest za pomocą budowanych modeli. Modele te używają algorytmów, działających automatycznie na zbiorze danych, bez ingerencji użytkownika.

- *Predykcja prawdopodobnych wyników*

Wiele form eksploracji danych ma charakter predykcyjny. Zbudowany model może na przykład przewidywać dochody, biorąc pod uwagę różne czynniki demograficzne, takie jak dostęp do edukacji.

Niektóre z modeli używane są do tworzenia reguł, będących warunkami, implikującymi określony wynik. Nawiązując do podjętego tematu pracy magisterskiej, może być to na przykład zasada, że serwery o krótkich czasach odpowiedzi (RTT) będą cechować się największą szybkością transferu.

- *Tworzenie trafnych informacji*

Eksploracja danych może przynieść dające się wykorzystać w praktyczny sposób informacje. Przykładowo, bank może udzielić pożyczki na podstawie modelu identyfikującego klientów, który sprawdza, czym cechują się osoby pewne w spłacaniu swoich zobowiązań.

- *Skupienie na dużych zbiorach danych*

Eksploracja danych skupia się przede wszystkim na dużych zbiorach danych, mogą być to na przykład zebrane obserwacje z zakresu medycyny lub fizyki. Oczywiście technika ta znajduje również powszechnie zastosowanie w przypadkach biznesowych.

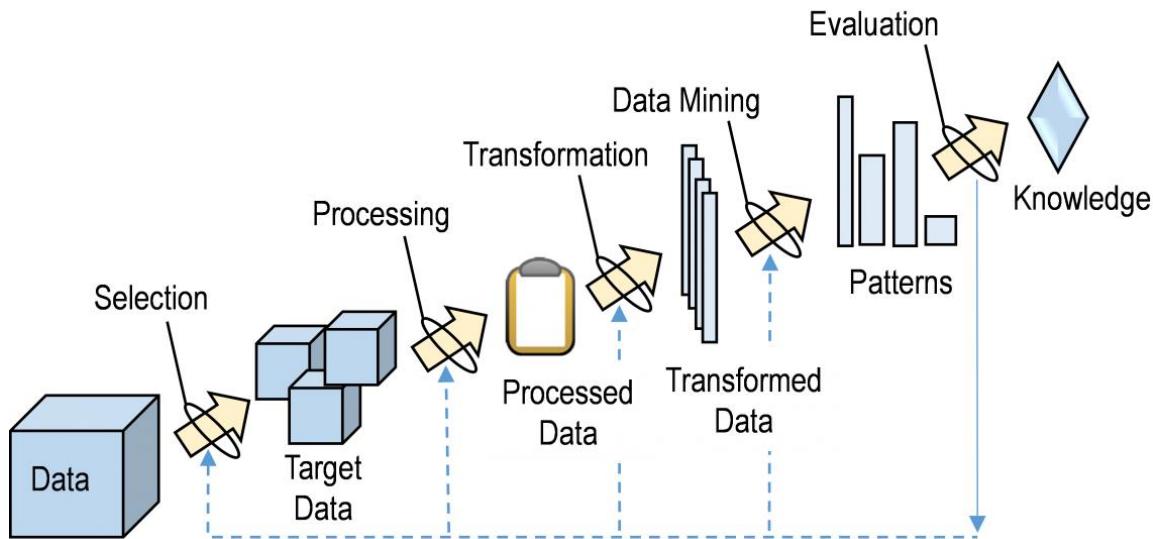
Technika Data Mining, pomimo pewnych podobieństw, różni się od tradycyjnych technik statystycznych. Metody statystyczne zazwyczaj wymagają dużej interakcji z użytkownikiem, w celu sprawdzenia prawidłowości tworzonego modelu. W rezultacie proces ten może być trudny do zautomatyzowania. Dodatkowo, metody statystyczne nie skalują się zbyt dobrze dla bardzo dużych zbiorów danych. Zazwyczaj używane są do sprawdzania hipotez lub znajdywania korelacji w oparciu o mniejsze, reprezentatywne próbki obserwacji.

Proces eksploracji danych składa się z zasadniczych, następujących po sobie etapów [29]:

- *Integracja danych.* Zbieranie wszystkich danych i integrowanie ich z różnych źródeł.
- *Selekcja danych.* Wybór danych, które są użyteczne z perspektywy eksploracji danych.
- *Czyszczenie danych.* Wybrane wcześniej dane mogą posiadać brakujące wartości, mocne odchylenia lub błędy. Na tym etapie używa się odpowiednich technik pozwalających wyczyścić podobne nieprawidłowości.

- *Transformacja danych.* Dane po czyszczeniu mogą nie nadawać się do budowania wzorców, w tym celu należy je zmienić na odpowiedniejsze formy za pomocą technik takich jak wygładzanie, agregacja czy normalizacja.
- *Eksploracja danych.* Właściwy etap, w którym wykorzystuje się różne techniki pozwalające na odkrywanie interesujących wzorców.
- *Ocena wzorców i prezentacja wiedzy.* Wizualizacja, transformacja i usuwanie zbędnych wzorców.
- *Użycie odkrytej wiedzy.* Na podstawie uzyskanej wiedzy użytkownik może wykonywać lepsze decyzje. Wzrasta świadomość badanego zjawiska.

Proces ten został przedstawiony graficznie na Rysunku 3.1, znajdującym się poniżej.



Rysunek 3.1 Etapy Data Mining  
(źródło: [30])

### 3.1 Metodyka Web Performance Mining

*Web mining* to zastosowanie technik eksploracji danych do odkrycia wiedzy z płaszczyzny *World Wide Web*.

Zazwyczaj w Web mining, analizowane są źródła danych takie jak [2]:

- *Zawartość dokumentów Webowych* – obiekty wbudowane w stronę internetową, występujące w formie m.in. tekstu lub obrazków,
- *Logi z serwerów WWW* – pliki zawierające adresy IP klientów, wraz z datą i czasem dostępu do strony Web,
- *Dane opisujące strukturę strony Web* – na przykład znaczniki HTML lub XML,
- *Dane profilu Web użytkownika*.

Analiza wymienionych źródeł danych przekłada się na odpowiednie działy Web mining:

- *Web content mining* – odkrywa tematykę strony internetowej i wyciąga z nich nową wiedzę,
- *Web usage mining* – identyfikuje wzorce poruszania się użytkownika przez strony internetowe. Używane w celu personalizacji, ulepszaniu systemów, charakteryzacji ich użycia,

- *Web structure mining* – sprawdza jak zbudowane są dokumenty Web i wyciąga wiedzę na temat modelu znajdującego się u podstaw struktur linków w sieci WWW,
- *Web user profile mining* – odkrywa profile użytkownika bazując na ich zachowaniu w sieci, na przykład dla potrzeb systemów rekomendacji e-Commerce.

Autor publikacji [2] proponuje nowe podejście Web mining – *Web performance mining*. Ma ono za zadanie odkrywać wiedzę na temat wydajności sieci Web na podstawie analizy danych, możliwych do zebrania poprzez pomiary wykonywane w sieci. Celem tego podejścia jest przede wszystkim charakteryzacja wydajności sieci z perspektywy użytkownika końcowego (np. przeglądarek internetowych), czyli odpowiedź na pytanie, jak będzie zmieniał się czas załadowania strony w różnych warunkach sieci.

### 3.2 Wybrane algorytmy Data Mining

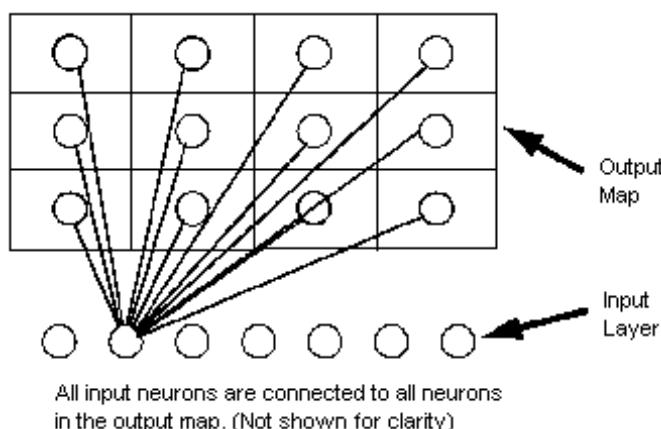
Poniżej przedstawione zostaną wykorzystane w pracy magisterskiej algorytmy Data Mining, udostępnione przez aplikację IBM SPSS Modeler 18. W przypadku technik klasyfikacji i regresji, będą to jedynie wybrane algorytmy, pojawiające się najczęściej w uzyskanych wynikach.

#### 3.2.1 Algorytmy klasteryzacji

Algorytmy klasteryzacji mają na celu grupowanie danych w charakteryzujące się podobnymi wartościami klasy. Działanie to pozwala na uzyskanie jednorodnych przedmiotów badań. W aplikacji zastosowanie znajdują trzy metody klasteryzacji:

- *Sieć Kohonena*

Sieć Kohonena jest typem sieci neuronowej, znanej również jako sieć samoorganizująca się. Podstawą działania tego algorytmu są neurony, ułożone w dwie warstwy – wejściową i wyjściową. Wszystkie neurony wejściowe połączone są z każdym neuronem wyjściowym. Połączenia te mają nadane odpowiednie wagi [31]. Schemat poglądowy został przedstawiony na Rysunku 3.2.



Rysunek 3.2 Sieć Kohonena  
(źródło: [31])

Neurony wejściowe przyjmują wartości danych używanych do klasteryzacji. Dane te są odpowiednio przeliczane do neuronów wyjściowych, będących poszczególnymi klasami wyjściowymi. Neuron wyjściowy z największą wyliczoną wartością określany jest jako klasa badanego przypadku.

Początkowo, wszystkie wagi są ustalane losowo. Z czasem działania algorytmu, ustalane są wartości wag w sposób, który pozwala na lepsze dopasowanie predykcji. Proces ten wykonywany jest, dopóki zmiany te nie będą minimalne.

- *Metoda K-średnich*

Algorytm ten znajduje szerokie zastosowanie w klasteryzacji. Jego głównym założeniem jest reprezentacja każdego z klastrów przez wektor wartości średnich numerycznych atrybutów wejściowych oraz wektor modalnych w przypadku danych nominalnych. Taka reprezentacja nazywana jest środkiem klastra [32].

K w nazwie metody odpowiada liczbie generowanych klastrów. Może być ona wyliczona automatycznie lub narzucona z góry przez użytkownika. W przypadku metody K-średnich można użyć różnych technik wyznaczających odległość między ich centrami, pasujących do badanego zjawiska .

- *Metody hierarchiczne*

W klasteryzacji hierarchicznej nie jest ustalana odgórna liczba klastrów. Na początku działania algorytmu, każda z obserwacji jest oddzielnym klastrem. W każdym kroku algorytmu łączone są dwa klastry, aż do momentu połączenia wszystkich danych w jedną grupę. Wybór ostatecznej liczby klastrów opierany jest na wykresie separowalności, obliczanej dla każdego kroku łącznia [33].

W pakiecie IBM SPSS Modeler, metody hierarchiczne używane są w algorytmie dwuetapowym. Na początku wszystkie obserwacje dzielone są na dużą liczbę małych klastrów (za pomocą bardziej wydajnych metod). Następnie wykonywana jest faktyczna klasteryzacja, w której używanymi algorytmami są techniki hierarchiczne.

### 3.2.2 Algorytmy klasyfikacji i regresji

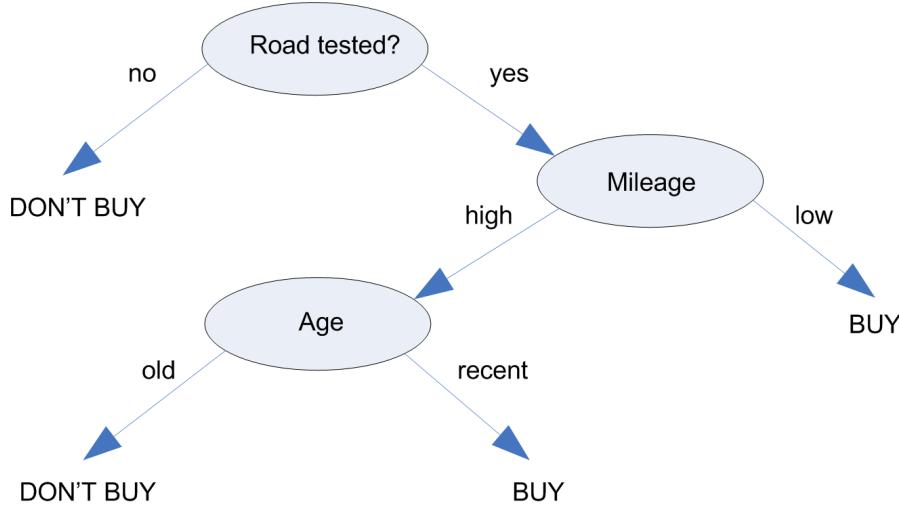
Zadaniem algorytmów klasyfikacji jest predykcja wybranej obserwacji do jednej z wcześniej zdefiniowanych klas, które są umownym przedstawieniem pewnego zjawiska. Mogą być to na przykład przedziały wartości jakiejś cechy lub nominalne wartości typu TAK/NIE. Istotą algorytmów regresji jest natomiast predykcja dokładnej wartości zmiennej wyjściowej.

Podstawowymi algorytmami stosowanymi w klasyfikacji i regresji są drzewa decyzyjne. W pakiecie SPSS dostępne są cztery algorytmy wykonujące taką analizę. Wszystkie z nich robią właściwie to samo – badają dane w celu znalezienia najlepszej klasyfikacji poprzez podział obserwacji na podgrupy. Proces ten jest wykonywany rekursywnie i dzieli grupy na coraz mniejsze, aż budowa drzewa jest zakończona (jak zdefiniowany jest koniec, określają kryteria zatrzymania). Zależnie od prognozowanej wartości, mamy do czynienia z drzewami klasyfikacyjnymi (wartość nominalna) lub drzewami regresji (wartość ciągła) [34].

W pakiecie SPSS używa się następujących drzew decyzyjnych:

- *Classification and Regression Tree (C&RT)*. Metoda ta wykorzystuje rekursywne techniki dzielenia rekordów poprzez minimalizację zakłóceń na każdym kroku. Węzeł drzewa uznany jest za czysty, jeżeli wszystkie przypadki w tym węźle mieszkają się w określonej kategorii argumentu wyjściowego. Wszystkie wykonywane podziały są binarne. Metodę tę używa się zarówno do klasyfikacji jak również regresji.
- *CHAID*. Algorytm ten używa statystyk chi-kwadrat do identyfikacji optymalnych podziałów. Tworzone węzły nie są binarne, może być ich więcej niż dwie generowane gałęzie. Wartość docelowa może być nominalna i numeryczna.

- *QUEST*. Wykonuje binarną klasyfikację, stworzoną do skrócenia czasu przetwarzania potrzebnego przez analizy algorytmu C&RT jak również zmniejszenia tendencji do wykorzystania argumentów wejściowych, które mogą być podzielone na więcej grup. Pomimo że wejścia mogą być wartościami numerycznymi, wartość prognozowana musi być nominalna.
- *C5.0*. Metoda ta tworzy drzewo decyzyjne lub zbiór zasad. Model ten działa poprzez podzielenie próbki w oparciu o argument, który zapewnia maksimum zysku informacji na każdym poziomie. Wartość prognozowana musi być w tym przypadku nominalna.



Rysunek 3.3 Przykład drzewa decyzyjnego  
(źródło: [34])

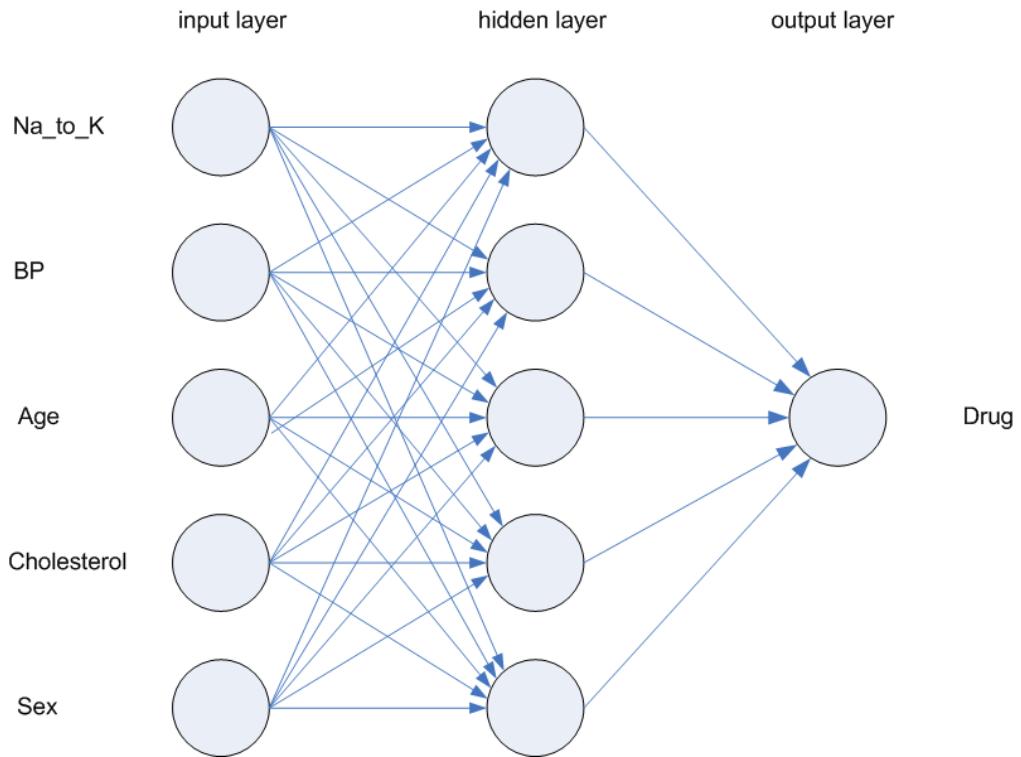
Dodatkowo, w analizach używana będzie również metoda *analizy najbliższych sąsiadów*. Bazuje ona na podobieństwie zebranych obserwacji. W uczeniu maszynowym, algorytm ten został stworzony jako sposób na rozpoznanie wzorców bez wymaganego dokładnego dopasowania do zebranych obserwacji.

Metoda ta tworzy przestrzeń, w której podobne obserwacje (sąsiedzi) są rozmieszczone blisko siebie, a różniące się – znajdują się w pewnej odległości. Dystans między tymi przypadkami jest miarą ich niepodobieństwa.

W algorytmie *K-najbliższych sąsiadów*, wartość K odpowiada za liczbę sąsiadów do sprawdzenia. Jeżeli większość wybranego sąsiedztwa należy do tej samej kategorii (szukanej), badana obserwacja również zostanie do niej zaliczona. Analiza ta może być również wykorzystana w celu obliczenia przewidywanej wartości numerycznej. W tej sytuacji, obliczana jest średnia lub mediana wartości cechujących wybranych najbliższych sąsiadów.

W wynikach również można zauważać modele oparte o działanie *sieci neuronowych*. Przykładem takiej sieci jest przytoczona wcześniej Sieć Kohonena. Jednostki przetwarzające dane są zorganizowane w odpowiednie warstwy. Zazwyczaj sieć dzieli się na trzy części: warstwę wejściową (reprezentującą parametry wykorzystywane do budowy modelu), jedną lub więcej warstw ukrytych i warstwę wyjściową [35]. Przykład widać na Rysunku 3.4.

Jednostki te powiązane są ze sobą połączeniami o odpowiednich wagach. Za pomocą danych wejściowych w warstwie pierwszej, do kolejnych neuronów propagowane są kolejne przetworzone wartości, które ostatecznie przedstawiane są w warstwie wyjściowej, której węzły odpowiadają za możliwości predykcji. Największa wartość w tej warstwie wyjściowej jest uznana za zwycięzcę – na przykład klasę danej obserwacji.



Rysunek 3.4 Przykładowa sieć neuronowa  
(źródło [35])

Poza wymienionymi wyżej algorytmami, których wyniki zazwyczaj pojawiały się w przeprowadzonych badaniach, pakiet IBM SPSS Modeler korzystał z implementacji wielu innych metod, m.in.:

- Dla problemu klasyfikacji:
  - *Lista decyzyjna*
  - *Sieć Bayesa*
  - *Analiza dyskryminacyjna*
  - *Drzewa losowe*
  - *Drzewo-AS*
  - *SVM*
  - *Regresja logistyczna*
- Dla problemu regresji:
  - *Regresja liniowa*
  - *Uogólniony model liniowy*
  - *LSVM*
  - *Drzewa losowe*
  - *SVM*
  - *Drzewo-AS*
  - *Modele liniowe*

## **4. Opracowanie stanowiska badawczego**

W rozdziale tym opisane zostaną poszczególne etapy opracowywania stanowiska badawczego. Na początku przedstawione zostaną wymagania dotyczące przeprowadzanych badań. W następnych podrozdziałach dokładnie opisana zostanie selekcja serwerów wraz z określeniem ich własności. Po wybraniu serwerów zdalnych, należało określić wielkość pliku bądź plików, które posłużą jako przedmiot pomiarów. W celu wykonania dłuższych pomiarów (badania w analizie literatury wykazały, że okres tygodnia jest niewystarczający do uzyskania dobrych wartości predykcji [2]), należało upewnić się, że wybrane serwery cechują się stabilnością działania.

W ostatnich podrozdziałach opisany zostanie system MWING, użyty na potrzeby wykonywania niezbędnych pomiarów. Przedstawione zostaną rozważane możliwości i ostateczna lokalizacja stanowisk (agentów), w których system ten zostanie zainstalowany.

### **4.1 Wymagania dotyczące badań**

Wymagania dotyczące badań zostały wylistowane poniżej:

- Wybór rozmieszczonego na całym świecie serwerów, udostępniających taką samą zawartość,
- Zawartość ta musi być binarnie identyczna,
- Liczba wybranych serwerów powinna być wysoka, należy upewnić się, że są one stabilne,
- Wybór mierzonych zawartości powinien być nieprzypadkowy,
- Użycie środowiska pomiarowego, pozwalającego na zapis charakterystyk wykonywanych pomiarów Web,
- Agenci powinni znajdować się w różnych punktach geograficznych,
- Czas wykonywanych pomiarów dla każdego z agentów to minimum 2 tygodnie,
- Użycie pakietów statystycznych, udostępniających możliwość użycia technik eksploracji danych,
- Stworzenie skryptów wykonujących predykcję wydajności Web.

### **4.2 Selekcja serwerów i określenie ich własności**

#### **4.2.1 Selekcja serwerów**

Pierwszym działaniem w celu opracowania stanowiska badawczego, była odpowiednia selekcja serwerów, udostępniających taką samą zawartość. Ponieważ jednym z założeń było, by ilość badanych próbek była jak największa, poszukiwania skupiono na serwisach udostępniających tzw. mirroring plików.

Istnieją również rozwiązania, które pozwalają na mirroring stron Web (serwer lustrzany). Jednak ze względów niewielkiej ilości takich stron i trudności w zapewnieniu ich identyczności w czasie, skupiono się na serwerach plików.

W badaniach zdecydowano się na użycie mirroringu, zapewnianego w ramach działania systemu operacyjnego *Debian*, dostępnego na stronie internetowej [36]. Serwery te rozmieszczone są na całym świecie i umożliwiają użytkownikom systemu operacyjnego na pobieranie zawartości z punktów im najbliższych. Łączna ilość oficjalnych serwerów (stan liczony 8 czerwca 2017r.) to 457.

Mirrory Debian dzielone są na dwa typy:

- *Pierwotne* – wykazujące się lepszymi parametrami łącza, dostępne całą dobę, powiązane z określonym państwem,
- *Wtórne* – mogącym posiadać ograniczenia (nie oznacza to jednak, że jest on wolniejszy niż serwer pierwotny).

Z racji takiego podziału, zdecydowano się na wykorzystanie wszystkich serwerów pierwotnych. W państwach, które nie posiadały takiego serwera, decydowano się na użycie serwerów wtórnego.

Każdy z wybranych serwerów musiał posiadać wspólną architekturę wybranej dystrybucji Debian, w celu możliwości późniejszego wyboru mirrorowanych plików. Z racji, że istniała ona na każdym z wybranych hostów, zdecydowano się na architekturę *amd64*.

Łączna ilość wybranych serwerów to 87. Liczba ta zakłada zmiany, które pojawią się w podrozdziale 4.4, dotyczącym wstępnych badań działania wybranych hostów.

Do każdego z serwerów przydzielony jest adres URL z dystrybucją systemu Debian (serwery te mogą być używane również pobierania plików niezwiązanych z tym systemem operacyjnym).

#### 4.2.2 Położenie geograficzne

Położenie geograficzne serwerów miało pomóc zarówno w wizualnej prezentacji wybranych punktów na mapie świata, a także zostać wykorzystane w przyszłości do wyliczenia odległości geograficznej między agentami a serwerem.

Ponieważ istnieje dużo narzędzi udostępniających możliwość sprawdzenia lokalizacji geograficznej wybranego adresu IP, wykorzystano serwis <https://www.iplocation.net/> wykonujący to zapytanie do najczęściej używanych API, m.in.: *IP2Location*, *ipinfo.io*, *EurekAPI*, *DB-IP*. Takie rozwiązanie miało zapobiec błędny wartościom, które mogły znaleźć się w bazie pojedynczego serwera.

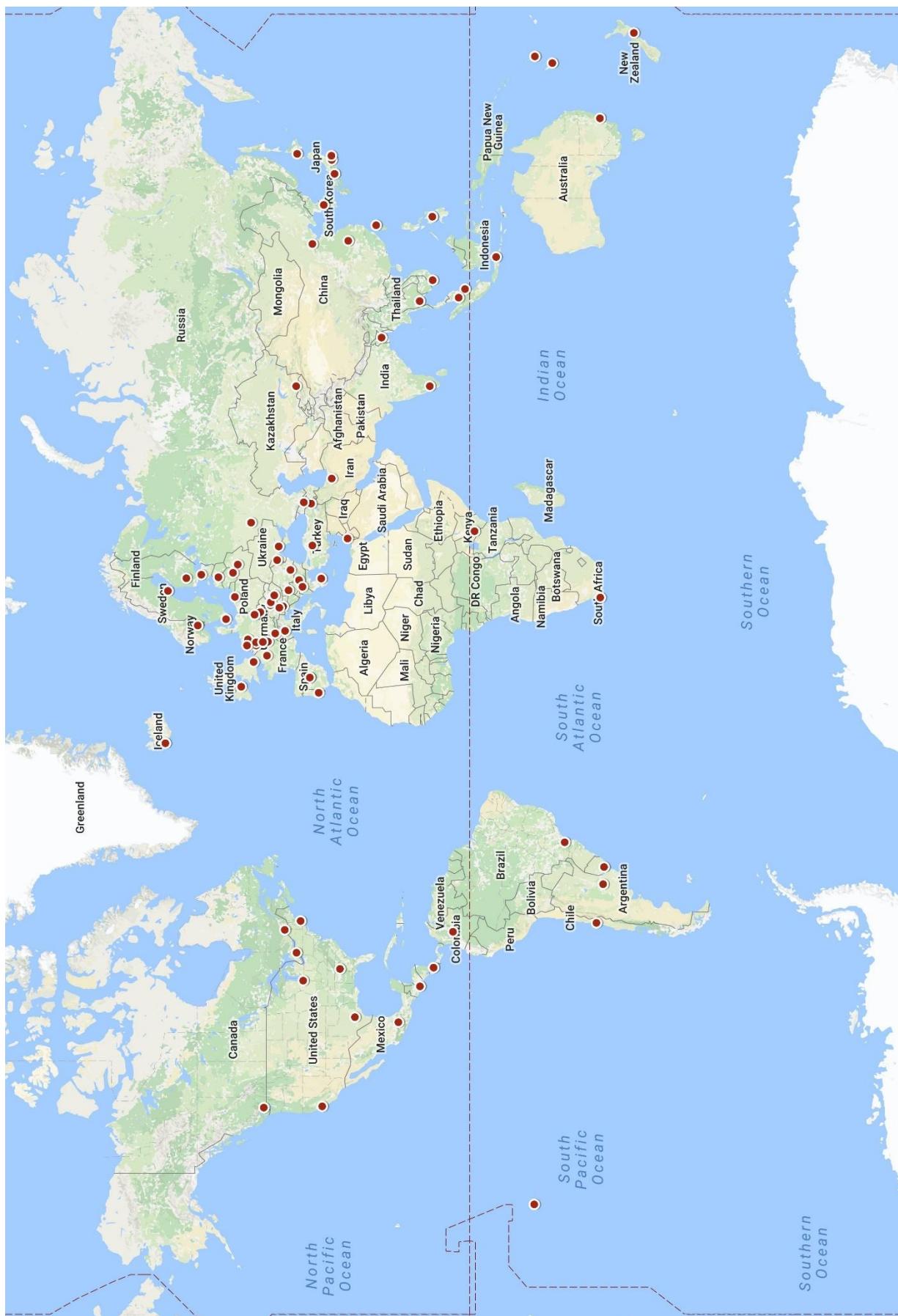
Dzięki informacjom udostępnianym przez serwis, można było wykonać walidację państwa, w którym znajdował się serwer. Oprócz tego, w odpowiedzi otrzymywano również następujące wartości:

- *Długość geograficzna*,
- *Szerokość geograficzna*,
- *Nazwa ISP (dostawcy usług internetowych)*,
- *Nazwa organizacji*.

Z racji, że nie we wszystkich bazach danych znajdowały się rekordy na temat wybranych serwerów lub były one ze sobą niezgodne, na podstawie dodatkowych poszukiwań, wybierano wyniki najbardziej prawdopodobne.

Przykładowa odpowiedź dla polskiego serwera pierwotnego, znajdującego się w Gdańsku, została przedstawiona na Rysunku 4.2.

Na następnej stronie pokazano Rysunek 4.1, który przedstawia mapę wybranych serwerów. Największe skupisko hostów znajduje się w Europie. Wynika to z dużej liczby państw (tym samym dużej liczby serwerów pierwotnych), a także rozwiniętej topologii sieci w tym obszarze. W pozostałych przypadkach, ilość serwerów zależy od możliwości internetowych danego kontynentu. Stany Zjednoczone posiadały dużą liczbę serwerów Debian, zdecydowano się więc na taki wybór hostów, który wystarczająco pokrywał obszar Ameryki Północnej.



Rysunek 4.1 Mapa położenia wybranych serwerów

Geolocation data from IP2Location (Product: DB6, updated on 2017-6-1)

Domain Name	Country	Region	City
ftp.task.gda.pl	Poland 	Pomorskie	Gdansk
ISP	Organization	Latitude	Longitude
Technical University of Gdansk Academic Computer Center Task	Not Available	54.3521	18.6464
Geolocation data from ipinfo.io (Product: API, real-time)			
Domain Name	Country	Region	City
ftp.task.gda.pl	Poland 	Pomerania	Gdańsk
ISP	Organization	Latitude	Longitude
Technical University of Gdansk, Academic Computer Center TASK	Technical University of Gdansk, Academic Computer Center TASK	54.3608	18.6583

Geolocation data from EurekAPI (Product: API, real-time)

Domain Name	Country	Region	City
ftp.task.gda.pl	Poland 	Pomorskie	Gdansk
ISP	Organization	Latitude	Longitude
Technical University of Gdansk, Academic Computer Center	Technical University of Gdansk, Academic Computer Center	54.3608	18.6583
Rysunek 4.2 Przykład odpowiedzi z serwisu <i>iplocation.net</i>			

#### 4.2.3 Internet Service Provider (ISP)

Kolejną własnością cechującą poszczególne serwery był dostawca usług internetowych. W podpunkcie wcześniejszym dla każdego z wybranych punktów wyciągnięto nazwę ISP, która nie wnosi żadnej informacji użytecznej w kontekście badania danych. Pierwotnym założeniem było wyekstrahowanie informacji, czy dostawca należy do rozwiązań przemysłowych (np. Orange Polska SA), czy sieci budowanych na potrzeby akademickie.

Okręslenie tej wartości dało wykonać się jedynie przez analizę wyników zdobytych na podstawie wyszukiwania nazwy ISP w przeglądarce. 40 serwerów należało do zaklasyfikowane o rozwiązań akademickich, zostawiając 47 serwerów należących do przemysłu.

Niestety przez fakt, że sprawdzenie typu ISP wiązało się w niektórych przypadkach z subiektywnym wyborem na podstawie opisów znajdujących się na różnych stronach WWW, uznano, że wartość tę należy odrzucić w dalszych analizach. Zmienne używane w procesach eksploracji danych powinny cechować się możliwie jak największą automatyzacją, również na poziomie ich zbierania.

#### 4.2.4 Mechanizm CDN

W analizie literatury przedstawiony został mechanizm CDN. Jest to usługa często wykupywana przez serwery Web, mająca na celu zapis zawartości strony na serwerach zdalnych, znajdujących się bliżej użytkownika końcowego. Użytkownik, wykonujący żądanie zawartości z serwera bazowego, jest przekierowywany do hosta należącego do CDN i to z niego pobierane zostają wymagane dane.

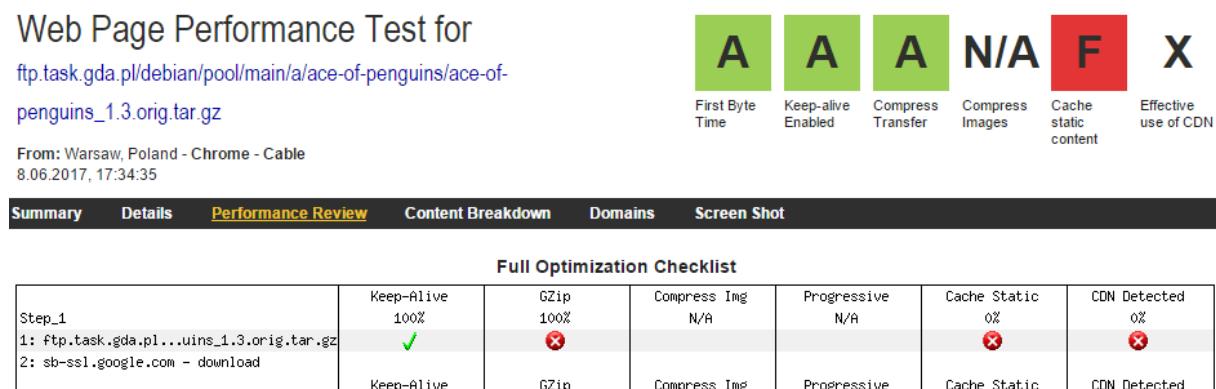
Jeżeli któryś z wybranych serwerów posiadałby taką usługę, należałoby usunąć go z listy badanych. Wartość czasu załadowania pliku byłaby obliczana dla serwera CDN, swoimi parametrami niezwiązanego z serwerem bazowym, co byłoby czynnikiem zniekształcającym wyniki.

Do sprawdzenia, czy serwer posiada mechanizmy CDN, użyto narzędzi dostępnych w Internecie. Pierwszym z nich jest *CDN Finder Tool*, udostępniane przez firmę *CDNPlanet* [37]. Narzędzie to pozwalało sprawdzić nazwy wszystkich używanych CDN na podstawie strony bazowej serwera. Użycie go wykazało, że większość z wybranych serwerów nie korzysta mechanizmu CDN.

Serwery, na których użycie CDN zostało wykryte (w liczbie 14), zostały poddane dalszej analizie. W każdym z tych przypadków okazało się, że pliki statyczne udostępniane przez CDN to *favicon*, czyli ikona znajdująca się przy adresie strony internetowej w przeglądarce. Pozostałe udostępniane elementy były również pomniejsze i niezwiązane z założonymi badaniami.

W celu pełnego upewnienia się w temacie mechanizmów CDN, wykorzystano z drugiego narzędzia, używanego w badaniu wydajności tworzonych stron Web – *WebPagetest* rozwijany przez firmę *Google* [38].

Na stronie tej możliwe było wykonanie m.in. sprawdzenia działania mechanizmu CDN dla wybranych plików (więcej informacji na ich temat zostanie przytoczona w kolejnym podrozdziale). Przykładowe wykonanie dla pliku należącego serwera w Gdańsku znajduje się na Rysunku 4.3.



Rysunek 4.3 Przykładowe sprawdzenie wydajności strony w serwisie *WebPagetest*

Dla wszystkich wybranych plików, serwis wskazywał użycie CDN (*CDN Detected*) na poziomie 0%. Wynik ten wydaje się logiczny, zważając na fakt, że cały sens istnienia mirrorów Debian ma za zadanie udostępnienie możliwości wyboru serwera najbliższego użytkownikowi. Użycie mechanizmów podobnych do CDN następuje prawdopodobnie, gdy użytkownik systemu operacyjnego Debian postanowi wykonać aktualizacje. Jej zawartość powinna być pobierana za pomocą znajdującego się najbliżej, autoryzowanego serwera.

Strona *WebPageTest* wskazuje, że żeby zaklasyfikować stronę jako używającą CDN, wartość użycia CDN powinna wskazywać przynajmniej 80%.

W związku z wykonaną analizą, żaden z serwerów nie został zaklasyfikowany jako używający CDN. Tym samym w kroku tym nie odrzucono żadnego z hostów.

#### 4.2.5 Protokół HTTP

W analizie literatury przytoczony został opis protokołu HTTP. Wskazano tam, że na większości stron, używana jest wersja HTTP/1.1. W chwili obecnej, jedynie 14.6% stron internetowych używa nowszej wersji HTTP/2 [10]. W dalszej analizie serwerów należało sprawdzić, która z wersji protokołów jest przez nie używana.

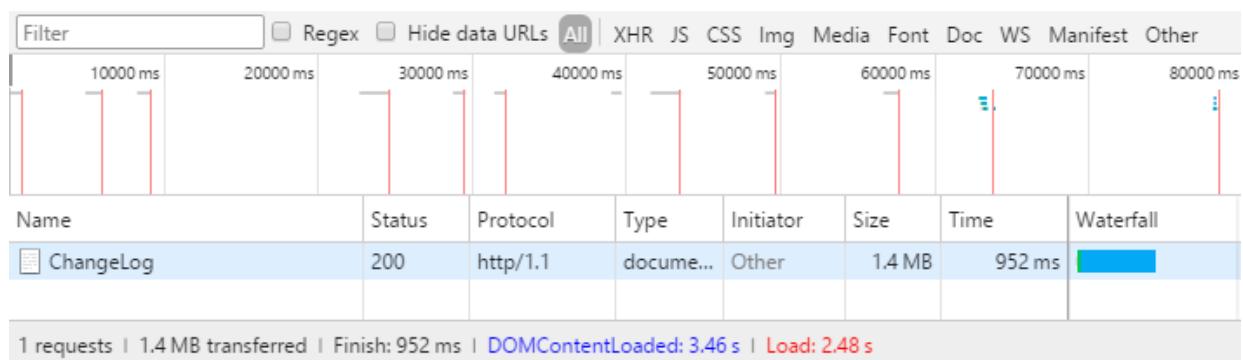
Użycie protokołu HTTP/2, z założenia szybszego, mogłoby dawać lepsze wyniki wydajności dla wykorzystujących go serwerów. Z tego powodu, taka charakterystyka powinna zostać odnotowana.

Na chwilę obecną, istnieje wiele sposobów określenia, czy serwer zdalny wykorzystuje protokół HTTP/2 [39].

Na początku, zdecydowano się na wykorzystanie serwisu online *HTTP/2 Test* udostępnianego przez firmę *KeyCDN* [40]. Niestety, okazało się, że narzędzie to nie było w stanie uzyskać prawidłowych wyników działania dla podanych adresów serwerów. Należało wykorzystać inną możliwość.

Kolejnym sposobem było użycie rozszerzenia do narzędzia *curl* występującego w dystrybucjach Linuxa. Niestety pomimo licznych prób jego konfiguracji, również nie udało się skorzystać z tej możliwości.

Określenie wersji protokołu HTTP uzyskano ostatecznie za pomocą narzędzi dla programistów dostępnych w przeglądarce Chrome (*Chrome DevTools*). Na podstawie znajdujących się tam logów, można było określić za pomocą jakich protokołów zostały pobrane kolejne zawartości strony. Przykładowe okno zakładki zostało pokazane na Rysunku 4.4. Można zauważyć, że znajdujący się tam plik *ChangeLog* został pobrany ze strony Web za pomocą protokołu HTTP/1.1.



Rysunek 4.4 Okno wtyczki *DevTools* w przeglądarce *Chrome*

Analiza wszystkich serwerów wykazała, że żaden z nich nie używa nowszej wersji protokołu HTTP/2. Z tego powodu, czynnik ten nie zostanie uwzględniony w dalszej części badań.

W celach odtworzenia badań, fakt użycia nowszego (lub innego) protokołu przez któryś z serwerów, powinien zostać odnotowany.

#### 4.2.6 Dystans geograficzny

Kolejną wyliczaną wartością był dystans geograficzny. Na podstawie danych o położeniu geograficznym serwera zdalnego i agenta pomiarowego, odległość ta wyliczana została za pomocą funkcji Excel, dostępnej pod adresem [41]. Prawidłowość działania tego algorytmu została sprawdzona na podstawie serwisu zewnętrznego *Distance Calculator* [42].

Dystans geograficzny należało policzyć osobno dla każdego z punktów wykonujących pomiary. Odległość od serwerów dla agenta we Wrocławiu wahała się od 234 km (serwer w Czechach) do 17977 km (Nowa Zelandia).

#### 4.2.7 Długość ścieżki IP/AS

Na podstawie wyników narzędzia *traceroute*, dla każdego z serwerów zdalnych została wyliczona również odległość rozumiana jako liczba przeskoków IP (pakietu na routerach), a także liczba przeskoków AS (po kolejnych systemach autonomicznych). Szczegółowe informacje na temat narzędzia *traceroute* i dalsze badania tych ścieżek zostaną poruszone w Rozdziale 5.

#### 4.2.8 Utworzenie tabeli z danymi

Tabela serwerów wraz z ich własnościami została zapisana w postaci skoroszytu programu Excel. Wycinek z niej dostępny jest poniżej, w Tabeli 4.1.

BAZOWY URL	KRAJ	MIASTO	ISP	LATITUDE	LONGITUDE	DISTANCE FROM WROCŁAW
mirrors.asnet.am	Armenia	Yerevan	ACADEMIC	40,1811	44,5136	2416,641237
debian.sil.at	Austria	Vienna	INDUSTRIAL	48,2085	16,3721	328,8734426
mirror.as35701.net	Belgium	Sint-Truiden	INDUSTRIAL	50,8168	5,1865	850,3773205
ftp.by.debian.org	Belarus	Minsk	INDUSTRIAL	53,9000	27,5667	758,5504162
debian.c3sl.ufpr.br	Brazil	Curitiba	ACADEMIC	-25,4278	-49,2731	10703,23915
debian.spnet.net	Bulgaria	Sofia	INDUSTRIAL	42,6975	23,3242	1038,453982
mirrors.tuna.tsinghua.edu.cn	China	Beijing	ACADEMIC	39,9075	116,3972	7222,194205
mirrors.ustc.edu.cn	China	Hefei	ACADEMIC	31,8639	117,2808	7941,01811
debian.carnet.hr	Croatia	Zagreb	ACADEMIC	45,8144	15,9780	596,1400139
ftp.debian.cz	Czech Republic	Prague	INDUSTRIAL	50,0880	14,4208	234,344689
mirrors.dotsrc.org	Denmark	Kongens Lyngby	ACADEMIC	55,7704	12,5038	609,6231736
ftp.aso.ee	Estonia	Tallinn	INDUSTRIAL	59,4370	24,7535	1037,943887
trumpetti.atm.tut.fi	Finland	Tampere	ACADEMIC	61,4991	23,7871	1221,533594
debian.proxad.net	France	Paris	INDUSTRIAL	48,8534	2,3488	1098,166913
debian.noc.ntua.gr	Greece	Athens	ACADEMIC	37,9795	23,7162	1542,688993
ulises.hostalia.com	Spain	Madrid	INDUSTRIAL	40,4165	-3,7026	2006,71779
ftp.snt.utwente.nl	Netherlands	Enschede	ACADEMIC	52,2183	6,8958	729,9531964
debian.heanet.ie	Ireland	Dublin	ACADEMIC	53,3440	-6,2672	1619,217895
debian.simnet.is	Iceland	Reykjavik	INDUSTRIAL	64,1355	-21,8954	2685,149678

Tabela 4.1 Wybrane serwery (wycinek)

#### 4.3 Selekcja plików wykorzystywanych w pomiarach

Po wybraniu serwerów udostępniających identyczną zawartość, należało dokonać również selekcji plików, które będą pobierane przez agentów w trakcie trwania badań. Rozdział ten opisze krótką analizę Web, której wynikiem będzie przemyślana selekcja tych plików.

##### 4.3.1 Wstępna analiza Web

Rozszerzenie plików nie ma znaczenia z perspektywy wykonywanych pomiarów, jedyną własnością, którą należało się kierować, był ich rozmiar. Wybór badanych wielkości pliku (bądź plików) mógł być kierowany jedną z dwóch potrzeb biznesowych:

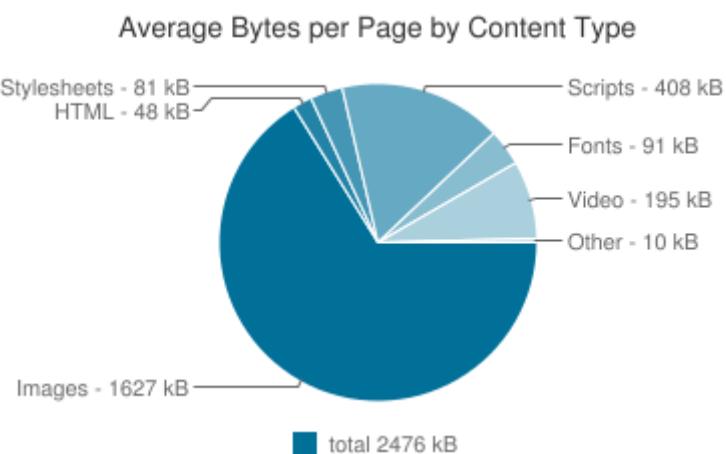
- Pliki duże, których pobieranie jest kosztowne czasowo. Wydajność czasu ładowania tych plików jest znacznie bardziej odczuwalna przez użytkownika końcowego.
- Pliki, których wielkość przypomina rozmiar strony internetowej. Badanie tego rodzaju wydajności rzutowałoby wyniki bezpośrednio na predykcję czasu załadowania stron Web w Internecie.

Ze względu na ograniczenie, jakim był dokonany już wybór serwerów mirroringu, opcja pierwsza była zbyt trudna do praktycznego wykonania. Dystrybucje systemu operacyjnego Debian cechują się dużą liczbą plików o niewielkich rozmiarach. Dodatkowo, zadanie to mogło być bardziej podatne na nieprawidłowe działanie szeroko rozumianej sieci, przyczyniając się do większej ilości błędnych pomiarów. Badanie plików o dużych rozmiarach przekładałoby się na niewielką ilość pomiarów jednego dnia, co w przypadku krótkich badań dałoby niewystarczającą liczbę próbek do prawidłowego użycia technik eksploracji danych.

Między innymi z powyższego powodu, zdecydowano się na wybór drugiego przypadku biznesowego. W tym celu należało określić, jaka wielkość z plików będzie adekwatna do średniej wielkości stron, znajdujących się w Web.

Przeanalizowano dane znajdujące się na serwisie *HTTP Archive* [43]. Bada on wybraną próbke stron internetowych (około pół miliona) w celu określenia statystyk związanych z funkcjonowaniem Web, m.in. średnią wielkość strony, czy użycie mechanizmów CDN.

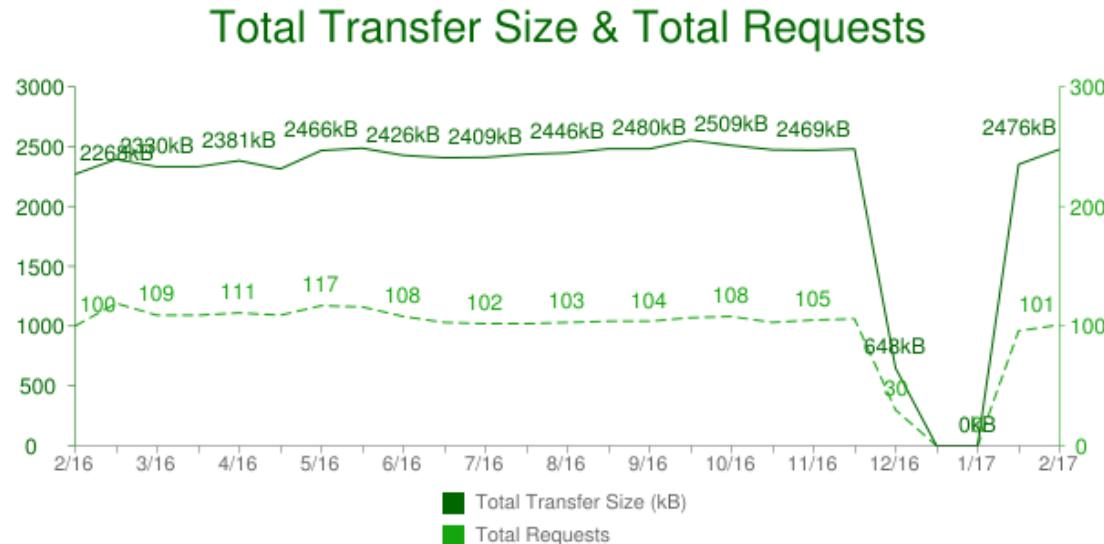
Na Rysunku 4.5 przedstawiony został wykres kołowy, pokazujący średnią wielkość stron internetowych, wraz z podziałem na poszczególne elementy.



Rysunek 4.5 Budowa strony internetowej [Stan na 3.03.2017r.]  
(źródło: [43])

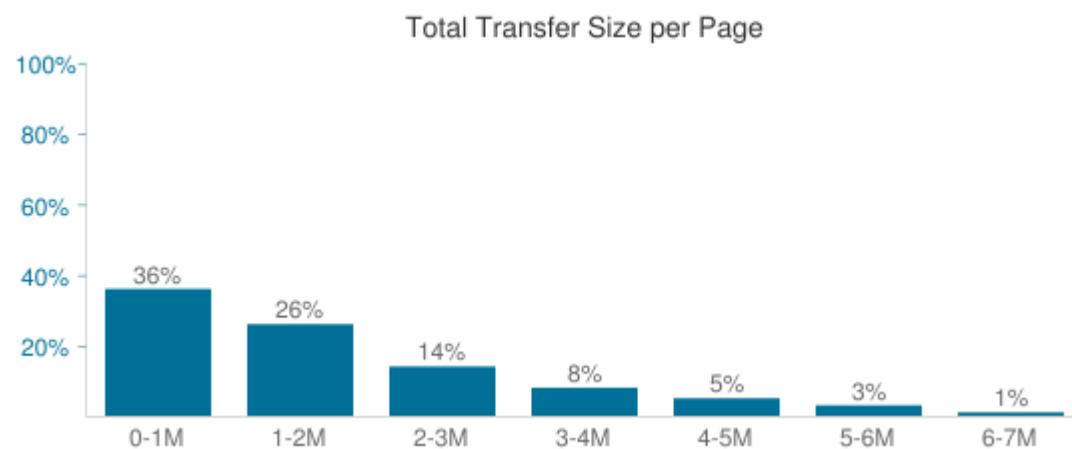
Jeżeli należałyby się zdecydować na wybór pojedynczego pliku, zawartość o rozmiarze około 2.5 MB byłaby dobrym przybliżeniem strony internetowej. Tym samym wykonane badania bezpośrednio odnosiłyby się do wydajności, rozumianej jako czas załadowania strony Web.

Warto zauważyć, że rozmiar ten zmienił się nieznacznie przez okres roku przed rozpoczęciem badań. Przedstawia to Rysunek 4.6 (spadek w okresie grudnia i stycznia spowodowany jest błędami działania serwisu).



Rysunek 4.6 Zmiana średniego rozmiaru strony w czasie [Stan na 3.03.2017r.]  
(źródło: [43])

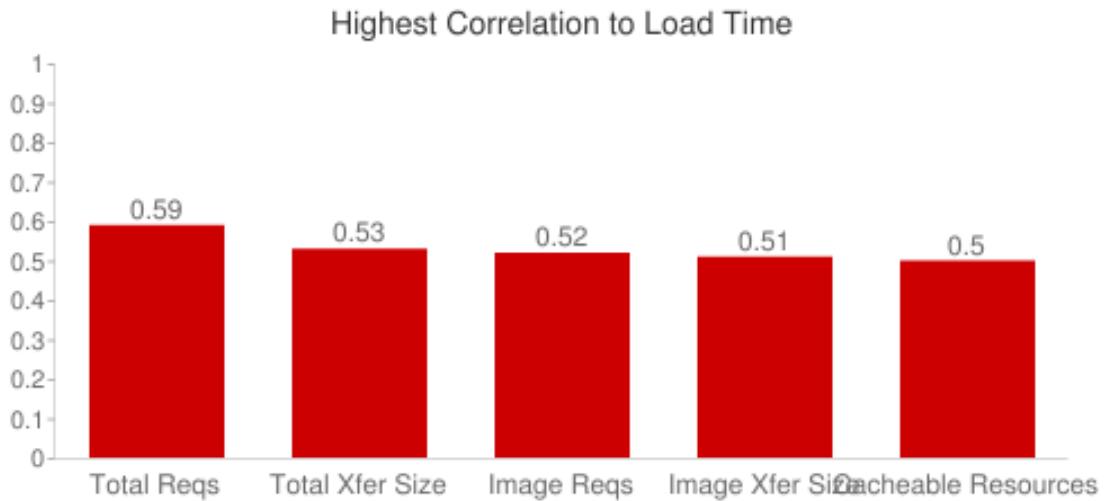
Sama wartość średnia nie powinna jednak być wskaźnikiem ostatecznego wyboru. Dlatego w analizie zapoznano się również z przedziałami całkowitego rozmiaru stron. Histogram, przedstawiający ten podział, został przedstawiony na Rysunku 4.7.



Rysunek 4.7 Przedziały rozmiaru stron [Stan na 3.03.2017r.]  
(źródło: [43])

Z racji otrzymanych wyników sugerujących, że znaczna większość stron znajduje się poniżej średniej (przynajmniej 62%), zdecydowano się na wybór kilku plików w celach badawczych. Ich wielkości powinny ważyć się od przynajmniej 0.5 MB (wartości mniejsze są zbyt podatne na zakłócenia) do 3 MB.

Dodatkowo, w serwisie znaleziono wykres korelacji do załadowania strony. Znajduje się on na Rysunku 4.8. Można zauważyć, że całkowita wielkość strony jest drugim parametrem, który najmocniej wpływa na wydajność rozumianą przez użytkownika końcowego. Potwierdza to sens wykonywanych badań.



Rysunek 4.8 Korelacja do całkowitego załadowania strony [Stan na 3.03.2017r.]  
(źródło: [43])

#### 4.3.2 Dokonanie wyboru plików

Powyższa analiza była podstawą do wykonania selekcji plików. Na serwerze należało znaleźć pliki, których wielkość będzie odpowiadała określonym przedziałom, a także zmniejszyć prawdopodobieństwo, że wykorzystywane pliki nie ulegną zmianie w trakcie zbierania pomiarów.

Zdecydowano się na wybór sześciu plików, rozpoczynając od wielkości 0.5 MB, idąc krokiem o takiej samej wartości aż do 3 MB. Selekcja dokładnych założonych rozmiarów była niemożliwa, dlatego wybrane pliki miały je jak najlepiej przybliżać.

Ostateczna selekcja przedstawiona została poniżej, wraz z rzeczywistym rozmiarem pliku:

- ace-of-penguins\_1.3.orig.tar.gz (517 KB)
- actionaz\_3.4.2.orig.tar.gz (1002 KB)
- actionaz\_3.8.0-1\_amd64.deb (1538 KB)
- actionaz\_3.4.2-1\_amd64.deb (2055 KB)
- afnix\_2.2.0-2\_amd64.deb (2606 KB)
- libacegi-security-java-doc\_1.0.7-3\_all.deb (3122 KB)

Według wcześniejszej analizy, plik 2606 KB powinien być przypadkiem, który jest adekwatny do średniej wielkości strony Web.

Niestety, ponieważ za serwery odpowiada firma zewnętrzna, nie istniała możliwość zapewnienia, by pliki nie zostały zmienione w trakcie zbierania pomiarów. Prawdopodobieństwo to zostało jedynie zmniejszone za sprawą selekcji plików, których czas ostatniej modyfikacji przekraczał przynajmniej cztery lata.

#### **4.4 Wstępna analiza zachowania wybranych serwerów**

Nim rozpoczął się etap faktycznego prowadzenia pomiarów, dopełniono starań, by żaden z wyborów (serwera lub plików) nie miał negatywnego wpływu na ich wykonanie. Na poziomie obecnej wiedzy można było wykonać sprawdzenie dostępności wybranych plików w czasie, a także upewnić się, że każdy pobrany plik ma taki sam rozmiar.

##### **4.4.1 Dostępność plików**

Badanie dostępności plików zostało wykonane przy pomocy napisanego skryptu C#. Istotą jego działania było wysyłanie zapytania HTTP HEAD do adresów, na których znajdowała się zawartość wybranych plików.

Metoda HEAD często używana jest w celu testowania linków pod względem ich poprawności i dostępności [44]. Prawidłowa odpowiedź serwera następowała w przypadkach, gdy zawartość znajdowała się pod danym adresem. Odpowiedzi błędu mogły wskazywać, że:

- Plik pod wybranym adresem nie istnieje,
- Serwer zablokował żądanie,
- Odpowiedź serwera była zbyt wolna, przekroczeno maksymalny czas odpowiedzi.

Na podstawie uzyskanych wyników odfiltrowano jeden z serwerów (pierwotnie było ich 88), znajdujący się w mieście Perth, w Australii. Powodem było powolne jego działanie.

##### **4.4.2 Porównanie zawartości pobieranych plików**

Drugą możliwą do wykonania analizą było sprawdzenie zawartości pobieranych plików. Powinna być ona taka sama dla danego rozmiaru pliku. Wartości różne zakłócałyby wyniki całkowitego czasu transferu. W celu wykonania tej analizy, ponownie napisano skrypt C#.

Jego działanie opierało się na pobraniu pliku bazowego z wybranego serwera, następnie zawartość ta pobierana była z innych serwerów i porównywana. Pomimo, że można było sprawdzić jedynie rozmiar plików, wykonano zbadanie ich identyczności w kontekście binarnym. By to wykonać, należało użyć algorytmu wykonującego poniższe czynności:

- Sprawdzić referencje do plików,
- Sprawdzić rozmiar plików,
- Na podstawie działania bufora, odczytywać kolejne bajty i porównywać je ze sobą, aż do końca plików – porównanie binarne.

Przeprowadzona analiza wykazała, że wszystkie pliki były binarnie identyczne. Nie było więc potrzeby określania przedziałów dopuszczalnych wielkości plików i odfiltrowania serwerów, które by za nie wykraczały.

We wstępnej analizie zakładano również zbadanie, czy pobieranie zawartości z serwerów wykonywane jest w sposób jednowątkowy lub wielowątkowy. W tym celu starano się wykorzystać oprogramowanie *VisualEther Protocol Analyzer* dostępne na stronie [45]. Niestety jego konfiguracja okazała się zbyt czasochłonna na potrzeby tak prostego badania, którego wyniki miałyby mały wpływ na wyciągane wnioski.

#### **4.5 Stanowisko pomiarowe – System MWING**

W celu zautomatyzowania pomiarów transakcji wybranych plików należało skonfigurować stanowisko pomiarowe, które zainstalowane zostało na każdym z wybranych agentów. Zdecydowano się na użycie istniejącego rozwiązania *MWING*, stworzonego na potrzeby podobnych badań, wykonywanych wcześniej na terenie Politechniki Wrocławskiej [2] [20] [19]. W kolejnych podrozdziałach przedstawiona zostanie specyfikacja tego systemu oraz czynności związane z jego adaptacją dla własnych badań.

##### **4.5.1 Opis systemu**

System *MWING* to właściwie wieloagentowa wersja pierwotnego rozwiązania *WING*. Jego opis przytoczony zostanie na podstawie publikacji [46].

Serwis *WING* (Web ping) został opracowany na potrzeby badania Web, a także analizy i wizualizacji wydajności z perspektywy użytkownika końcowego. Narzędzie to używa bazy danych, w której przechowywane są wartości otrzymywane na podstawie wizualizacji działań prawdziwych przeglądarek internetowych.

Serwis używa wszystkie cechy HTTP/1.1 i HTTP/2. Potrafi przetwarzać skrypty w celu automatyzacji jego użycia. Pozwala to na wykonywanie zaawansowanych pomiarów Internetu.

*WING* używa lokalnego klienta Web, który wysyła żądanie GET do docelowego adresu URL, jak również takie same zapytania do wszystkich obiektów, wbudowanych w stronę znajdującej się pod tym adresem. *WING* monitoruje i odnotowuje czasy wszystkich czynności wykonywanych przez przeglądarkę, takie jak koniec pobierania zawartości Web. Wartości te zapisywane są w bazie danych, w formacie pozwalającym na ich przyszłą analizę.

*WING* wspiera działanie protokołów IP, TCP, UDP, DNS i HTTP. Zapisuje parametry transakcji HTTP i połączeń TCP, ułatwiając tym samym bardziej szczegółową analizę pomiarów. Dzięki narzędziu można wyliczyć, m.in. wartość przepustowości HTTP, a także czas RTT dla połączenia TCP. Przykładowa transakcja mierzona przez system została przedstawiona w analizie literatury, na Rysunku 2.3.

##### **4.5.2 Adaptacja systemu do celów badawczych**

Podstawą rozpoczęcia badań w systemie *MWING* było uruchomienie narzędzia *Cron*, znajdującego się na maszynach Linux. Jest to demon systemowy, za pomocą którego użytkownik może zalecić systemowi cykliczne uruchamianie programów.

Określono, że seria pomiarowa (6 plików \* 87 serwerów, czyli łącznie 522 pomiary) ma być rozpoczęta o pełnych godzinach parzystych każdego dnia, czyli godzinach 0, 2, 4 ... Tym samym ilość wykonanych pomiarów na jeden dzień wynosiła 12.

Po uruchomieniu badań na każdym z agentów sprawdzono, czy pojedyncza seria pomiarowa nie przekracza założonych 2 godzin. Jeżeli średnia trwania serii pomiarowej byłaby bliska lub większa od tej wartości, pomiary powinny być wykonywane z mniejszą częstotliwością. W przeciwnym wypadku, system rozpoczynał nowe pomiary w trakcie trwania starych, zwiększałcąc faktycznie wyniki wydajności połączenia.

Adresy URL, pod którymi dostępne były pliki na każdym z badanych serwerów, należało dostarczyć narzędziu *MWING* w pliku XML. Jego generację wykonano za pomocą skryptu C#, który do każdego z adresów przypisywał odpowiadający mu numer identyfikacyjny, wyliczany na podstawie wzoru: [100 \* numerPliku + numerSerwera].

## 4.6 Hosting agentów

Przedstawiony powyżej system MWING należało umieścić w punktach pomiarowych, zwanych dalej agentami. W tym podrozdziale przedstawiona zostanie analiza możliwości, jak również miejsca, w których ostatecznie zainstalowano agentów. Na końcu pokazane zostaną wyniki prostych badań, mających na celu sprawdzenie działania wybranych punktów pomiarowych.

### 4.6.1 Analiza możliwości

Dotychczas używane środowisko, posiadające zainstalowany system pomiarowy MWING, znajduje się na terenie Politechniki Wrocławskiej, w gmachu budynku C-3. Ponieważ konfiguracja narzędzia na czystej wersji systemu operacyjnego byłaby zbyt złożona, zdecydowano się na stworzenie jej wirtualnej kopii, którą można uruchomić narzędziami takimi jak *VMware Workstation* czy *Oracle VirtualBox*.

Pierwotnym założeniem było rozłożenie serwerów pomiarowych w różnych punktach na mapie świata. Ponieważ na chwilę obecną nie istniały możliwości nawiązania kontaktu z uczelniami z innych krajów, w tym celu trzeba było wykorzystać rozwiązania biznesowe.

Niestety, większość serwisów internetowych, udostępniających możliwość zdalnego wynajmowania maszyn wirtualnych, nie posiada możliwości wgrania istniejącej maszyny. Po wykonaniu przeglądu w Internecie, znaleziono trzy wspierane rozwiązania:

- *Microsoft Azure* [47]
- *Amazon Web Services* [48]
- *DigitalOcean* [49]

Niestety problemem okazały się trudności na poziomie konfiguracji tych systemów. Ponadto, rzetelność badań wykonywanych na tych serwerach mogła być podważana przez działanie mechanizmów w technologiach chmury, która mogła posiadać infrastrukturę poprawiającą wydajność połączenia z hostami znajdującymi się na całym świecie. Z tych powodów zdecydowano się na inną opcję.

Obszar badań ograniczono do terenu Polski, a serwery pomiarowe znajdowały się w sieci PIONIER, opisanej dalej.

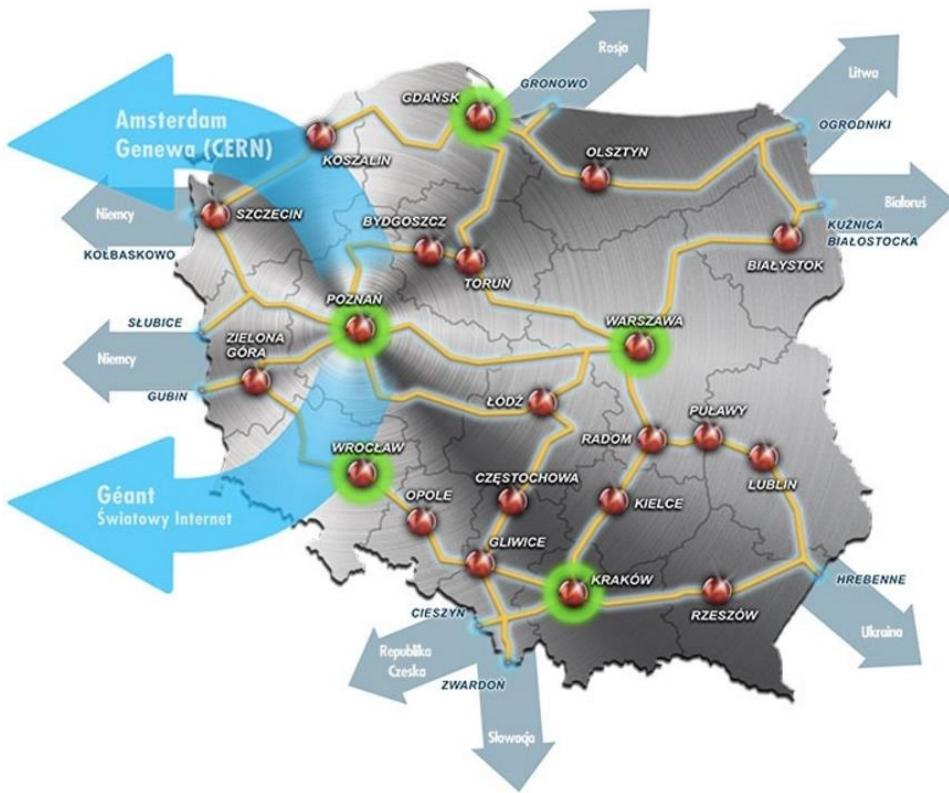
### 4.6.2 Sieć PIONIER

Sieć PIONIER to szerokopasmowa sieć optyczna, łącząca duże ośrodki naukowe na terenie całej Polski. Mapa połączeń przedstawiona została na Rysunku 4.9.

Platforma ta udostępnia możliwości wykonywania badań naukowych, z naciskiem na dziedziny informatyki i telekomunikacji [50]. Użytkownicy sieci mają możliwość wykorzystania usług takich jak:

- Zdalna praca z aplikacjami inżynierskimi (np. Matlab, AutoCad),
- Dostęp do bazy danych MySQL w chmurze,
- Dostęp do narzędzi Microsoft Office 365,
- Wynajmowanie maszyn wirtualnych (systemy MS Windows i Linux).

Ostatnia z przytoczonych usług była wykorzystywana w celu uruchamiania maszyn wirtualnych systemu *Debian 7*, będących kopią obrazu oryginalnej maszyny pomiarowej z narzędziem MWING.



Rysunek 4.9 Mapa połączeń PIONIER  
 (źródło: [50])

W ramach pracy magisterskiej udało się nawiązać współpracę z ośrodkami we Wrocławiu i w Poznaniu. Skontaktowano się również z punktami w Rzeszowie i Gdańsku. W pierwszym z nich nie było możliwości na uruchomienie wymaganej maszyny wirtualnej, zaś w drugim przeszkodziły tymczasowe problemy działania sieci wewnętrznej.

Instalacja gotowej maszyny wirtualnej na serwerach PIONIER była również zadaniem trudnym do skonfigurowania. Udało się ją wykonać jedynie dzięki sprawnej pomocy osób zarządzających poszczególnymi ośrodkami.

#### 4.6.3 Wybór agentów

Do wykonywania pomiarów użyto 3 agentów:

- **Koral.** Jest to serwer na terenie Politechniki Wrocławskiej, w budynku C-3. Pierwotny agent pomiarowy, który jest oryginalnym systemem zainstalowanym na maszynie fizycznej.
- **WCSS Platon.** Jest to pierwszy z serwerów w sieci PIONIER. Znajduje się również na terenie Politechniki Wrocławskiej, w budynku D-2. WCSS to skrót od Wrocławskiego Centrum Sieciowo-Superkomputerowego.
- **PCSS Platon.** Jest to poznański odpowiednik serwera w sieci PIONIER.

Decyzję o wybraniu dwóch serwerów we Wrocławiu podjęto dla celów analizy różnic wydajności dla każdego z nich. Serwery w sieci PIONIER znajdują się bliżej kręgosłupa sieci, dlatego ich działanie powinno być potencjalnie lepsze, niż na serwerze Koral. Pierwotny serwer pomiarowy jest bardziej zależny od wewnętrznego ruchu sieci na kampusie Politechniki Wrocławskiej.

#### 4.6.4 Analiza wybranych agentów

Zarówno przed, jak i w trakcie trwania pomiarów, wykonywano analizę działania wybranych agentów. Nim rozpoczęto wyjaśnianie wyników pomiarów, należało sprawdzić, czy istnieją różnice w konfiguracji serwerów pomiarowych. Zbadano wybrane własności systemów operacyjnych, jak również sprawdzono działanie sieci, w której znajdował się każdy z agentów.

Pierwszym badaniem było sprawdzenie początkowej wartości okna przeciążenia TCP. Jest to parametr, który odpowiada za ilość segmentów przesyłanych podczas pierwszego transferu danych z serwera. Większa jego wartość mogłaby przyczynić się do szybszego transferu (więcej przesyłanych pakietów, zmniejszenie problemu „wolnego startu”) bądź jego spowolnienia (więcej zagubionych pakietów i konieczna ich retransmisja).

Dane o tej wartości dostępne są w odpowiednich plikach systemowych. Na każdym z serwerów, początkowa wartość okna przeciążenia TCP wynosiła 10. Czynnik został wyeliminowany jako przyczyna różnicy w działaniu serwerów.

Kolejną czynnością było sprawdzenie stosu TCP używanego przez serwery. Na podstawie czynności podanych na stronie [51] otrzymano informację, że każdy z agentów korzysta ze stosu Cubic, będącego standardowym rozwiązaniem dla systemów Linux.

Ostatnim elementem sprawdzanym na poziomie systemu operacyjnego było wykorzystanie Network Time Protocol (NTP). Sprawdzenie działania tego protokołu miało za zadanie upewnić się, czy wykonywane pomiary są zsynchronizowane do zegara atomowego. Niestety ze względu na zainstalowane zapory sieciowe w serwerach sieci PIONIER, informacja taka okazała się niemożliwa do zdobycia.

W okresie od 15 maja do 28 maja 2017 roku przeprowadzone zostały również pomiary połączenia między trzema serwerami, zarówno pod względem czasu odpowiedzi (RTT), jak i ilości przeskoków na routerach, potrzebnych pojedynczemu pakietowi na osiągnięcia drugiego serwera. Wykorzystano narzędzia *ping* oraz *traceroute*.

Badania te nie wykazały żadnych odchyleń w wartościach na tych samych trasach (np. Koral-Poznań, Poznań-Koral), zarówno dla czasu odpowiedzi i ilości przeskoków pakietów. Badanie to nie przyniosło żadnych dodatkowych wniosków.

W analizach planowano również sprawdzenie szybkości transferu między serwerami. Niestety ze względów bezpieczeństwa, utworzenie serwera plików Apache2 okazało się czynnością wymagającą zbyt dużych nakładów formalnych dla podejmowania dalszych przedsięwzięć.

#### 4.7 Narzędzia wykorzystane w badaniach

Poniżej wylistowano pakiety, które wykorzystane zostały na potrzeby przeprowadzanych badań w dalszych rozdziałach pracy magisterskiej:

- *IBM SPSS Statistics 24,*
- *IBM SPSS Modeler 18,*
- *MathWorks MATLAB R2016a,*
- *Microsoft Visual Studio Community 2015,*
- *Microsoft Office 365 ProPlus.*

Każdy z programów wykorzystywany był w wersji próbnej, bądź na zasadach licencji studenckiej.

## 5. Szczegółowe badania dotyczące wybranych serwerów

Czas wykonywania pomiarów wykorzystano na dokładną analizę serwerów zdalnych, udostępniających system Debian. Istniało wiele możliwości dodatkowych badań. Zdecydowano, że wykorzystany czas poświęcony będzie szczegółowej analizie Systemów Autonomicznych.

System Autonomiczny (w skrócie AS) to zbiór adresów IP pod znajdujących się pod wspólną kontrolą administracyjną. Oznacza to, że pakiet poruszający się w Internecie może poruszać się przez kilka routerów, znajdujących się w tym samym AS.

Systemy te mogą znajdować się na różnych poziomach dostawców Internetu. Pomimo, że nie ma żadnego formalnego podziału, przyjęto poniższą strukturę poziomów [1]:

- Poziom 1 (Tier 1) – Dostawcy znajdujący się w nim mają dostęp do każdego miejsca Internetu za sprawą darmowych umów peeringowych z innymi dostawcami Internetu poziomie 1.
- Poziom 2 – Dostawcy łączący się z innymi darmowymi umowami peeringowymi, lecz również wykonujący opłaty tranzytowe w celu dostępu do dużej części Internetu.
- Poziom 3 – Dostawcy ci zawsze wykonują opłaty tranzytowe w celu dostępu do dowolnej części Internetu.

### 5.1 Analiza Systemów Autonomicznych (AS)

Poniżej przedstawione zostaną badania, mające na celu uzupełnienie własności serwera o szczegóły, dotyczące systemu autonomicznego, do którego należał zdalny host. Na początku opisana zostanie metodologia wykonanych poszukiwań. W kolejnych krokach stworzony zostanie graf połączeń AS, jak również prezentacja połączeń na mapie geograficznej.

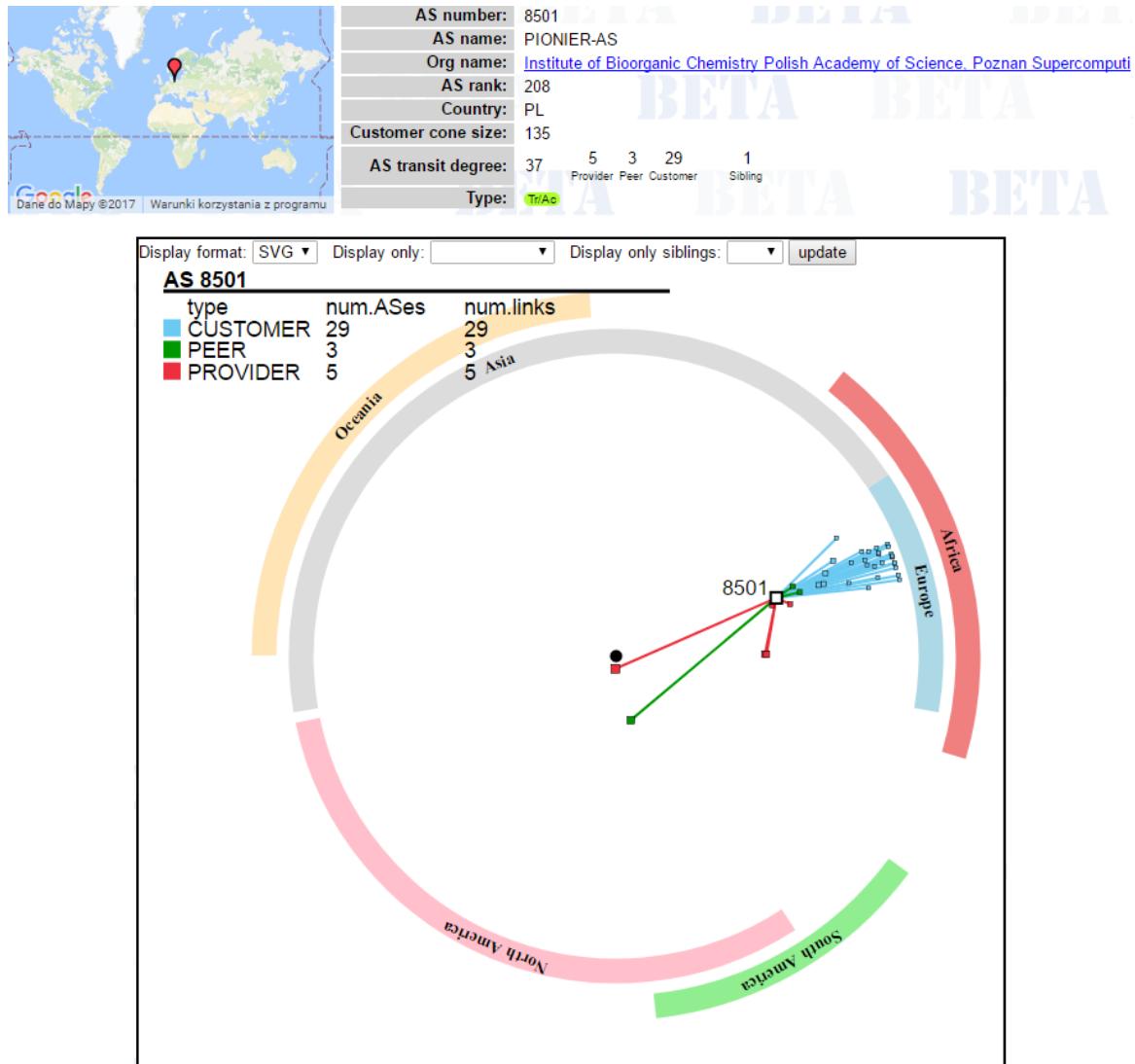
#### 5.1.1 Wyciągnięcie charakterystyk AS

Charakterystyki AS otrzymano za pomocą serwisu Web Caida, jednego z największych ośrodków badających Internet [52]. W swoich pracach ocenia on systemy autonomiczne, wyprowadzając takie statystyki jak [53]:

- Typ AS, m.in.:
  - Tranzytowy – używany jako punkt przejściowy między innymi ISP,
  - Dostępu – pozwalający użytkownikom końcowym na dostęp do Internetu,
  - Zawartości – używany do hostingu i systemów dystrybucji,
  - Przemysłowy – używany przez organizacje i przemysł, posiadający przed wszystkim użytkowników końcowych.
- Ranga AS – bazująca na ilości klientów (w postaci innych AS),
- Wielkość tranzytu – liczba innych AS, które w swoich ścieżkach routingu korzystały z danego AS,
- Liczba klientów i dostawców.

Nim można było zająć się ich sprawdzaniem, należało wyciągnąć informacje na temat numerów systemów autonomicznych wybranych punktów końcowych. Działanie to wykonano opierając się przede wszystkim na podstawie narzędzia [54]. Wyszukiwało ono numer AS na podstawie podanego adresu IP, wykonując zapytania do serwisu WHOIS, będącym bazą informacji o domenach. W przypadkach niewyjaśnionych przez serwis, korzystano z baz dostępnych na innych stronach Web.

Dla każdego z systemów końcowych wykonywano zapytanie w serwisie CAIDA. Przykładowa odpowiedź dla wyszukiwania systemu autonomicznego AS8501 (PIONIER) przedstawiona została na Rysunku 5.1. Można wyczytać z niego wartości wylistowane wcześniej. Dodatkowo w odpowiedzi udostępniana jest mapa połączeń z innymi AS. Pokazuje ona, jakie umowy łączą sieć PIONIER z innymi częściami świata.



Rysunek 5.1 Przykładowa odpowiedź serwisu CAIDA  
 (źródło: <http://as-rank.caida.org/?mode0=as-info&mode1=as-core&as=8501&data-selected-id=39>)

Wartości otrzymane z zapytań zostały dodane do tabeli własności hostów. Na podstawie liczby klientów i dostawców danego AS określono poziom sieci dostawcy usług internetowych. W przypadku systemów posiadających wyłącznie dostawców, był to poziom 3. Systemy posiadające zarówno dostawców, jak również klientów oceniono na poziom 2.

Przypadki, posiadające jedynie klientów, były sprawdzane na podstawie ogólnodostępnych w Internecie informacji o sieciach poziomu 1-wszego. Jeżeli nazwy dostawców się pokrywały, były zapisywane jako ten właśnie poziom.

Informacje o systemach autonomicznych z brakującymi wartościami na portalu CAIDA, uzupełniane były w oparciu o użycie baz danych znajdujących się w Internecie, m.in. na stronie *CIDR REPORT* [55].

### 5.1.2 Tworzenie grafu ścieżek AS

Z racji sprawnie przeprowadzonych badań punktów końcowych, zdecydowano się na głębszą analizę systemów autonomicznych. Tym razem dokonano starań, by stworzyć ścieżki, którymi wędrują pakiety wysyłane z punktu pomiarowego (dokładniej z WCSS) do poszczególnych serwerów końcowych.

W tym celu skorzystano z narzędzia *traceroute*, znajdującego się na dystrybucjach systemu Linux. Narzędzie to pozwala na zbadanie ścieżki wędrówki wysyłanego pakietu. Jego działanie można wywoływać przy pomocy różnych argumentów. Jednym z nich jest argument ‘-A’, który służy do pokazywania numeru systemu autonomicznego, do którego należy każdy z routerów znajdujących się w ścieżce.

Przykładowe wywołanie traceroute z argumentem ‘-A’ do serwera w Czechach, zostało przedstawione na Rysunku 5.2. Można zauważyć, że pakiet ten przechodzi przez systemy autonomiczne o numerach AS8970, AS196844, AS1200 i dociera do końcowego AS2852, w którym znajduje się czeski serwer.

```
traceroute to ftp.debian.cz (195.113.161.73), 30 hops max, 60 byte packets
1 * * *
2 smielec-u3.wask.wroc.pl (156.17.252.237) [AS8970] 0.722 ms 0.719 ms 0.853 ms
3 centrumrtr-do-smielec.wask.wroc.pl (156.17.252.1) [AS8970] 0.603 ms 0.600 ms 0.598 ms
4 karkonosz-centrum-rtr.wask.wroc.pl (156.17.254.111) [AS8970] 0.449 ms 0.588 ms 0.589 ms
5 sniezka-karkonosz.wask.wroc.pl (156.17.250.222) [AS8970] 0.586 ms 0.583 ms 0.581 ms
6 z-wroclaw-com.poznan-gw2-amsix.rtr.pionier.gov.pl (212.191.237.121) [AS196844] 4.832 ms 4.831 ms 4.927 ms
7 ams-ix-1.cesnet.cz (80.249.209.106) [AS1200] 29.030 ms 29.023 ms 29.138 ms
8 r105-r40.cesnet.cz (195.113.156.106) [AS2852] 29.456 ms 29.683 ms 29.722 ms
9 www.debian.cz (195.113.161.73) [AS2852] 28.125 ms 28.091 ms 28.202 ms
```

Rysunek 5.2 Przykładowe wywołanie traceroute

Dla każdego z wybranych serwerów wylistowano ścieżkę przeskoków po systemach autonomicznych. Z racji mechanizmów działania Internetu, należało pamiętać, że pojedynczy pomiar trasy ścieżek jest jedynie pewnym przybliżeniem, z następujących powodów:

- *Internet jest asymetryczny.*

Dane otrzymane na podstawie pomiaru traceroute wskazują, w jaki sposób pakiety poruszały się do serwera docelowego. Ponieważ podczas transferu plików, pakiety idą w kierunku przeciwnym, należałoby użyć traceroute po stronie serwera. W tym celu starano się wykorzystać narzędzie reverse-traceroute dostępne w pakiecie *VisualRoute*, niestety jest ono dostępne wyłącznie dla osób posiadających wykupioną pełną usługę programu.

- *Ścieżki routingu ciągle się zmieniają.*

Ponieważ Internet dostosowuje się do wszelkich zmian w sieci (np. zmiany routingu lub nagłe błędy po stronie serwerów), ścieżka wędrówki pakietu uzyskana przez pojedynczy pomiar może zmienić się podczas kolejnego wykorzystania narzędzia.

W celu uproszczenia dalszych badań na temat tras systemów autonomicznych, powyższych czynników nie brano pod uwagę. Przytoczenie ich miało na celu wskazanie niedoskonałości wyprowadzanych założeń.

Wykstrahowaną listę skoków użyto w tworzeniu grafu ścieżek AS z ośrodka WCSS Platon, w oparciu o poniższe założenia:

- Ścieżka każdego skoku rozpoczyna się w AS8970 (system autonomiczny, do którego należy Politechnika Wrocławskiego),

- Kolejne skoki na routerach znajdujących się w tym samym systemie autonomicznym nie były uwzględniane,
- Powrót do tego samego AS, również nie był uwzględniany,
- Routery, których sprawdzenie numeru AS nie udało się automatycznie przy użyciu narzędzia traceroute, zostały zweryfikowane na podstawie serwisów wykorzystywanych wcześniej [54],
- Jeżeli nie udało się znaleźć takich informacji, skok był pomijany.

Przygotowano kolejny skrypt C#, który z otrzymanych danych generował tzw. macierz sąsiedztwa, wykorzystywaną do określania połączeń w grafie [56]. Wielkość wag w tym przypadku, odpowiadała za ilość przejść pakietów przez określony odcinek między dwoma systemami autonomicznymi.

Wyniki programu były następnie przetwarzane przez napisany skrypt w środowisku Matlab. Poprzez wymuszenie jak najszerzego rozmieszczenia punktów na grafie, udało uzyskać się obraz znajdujący się na Rysunku 5.3.

Węzłami grafu są poszczególne systemy autonomiczne. Numer systemu autonomicznego (ASN) znajduje się obok punktu. W zależności od poziomu ISP, różni się on kolorem. Jest to kolejno – czerwony dla poziomu 1, niebieski dla poziomu 2, czarny dla poziomu 3.

Krawędziami grafu są połączenia między systemami autonomicznymi. Im częściej pakiet poruszał się określonymi ścieżkami, tym grubsza jest krawędź grafu. Na otrzymanych wynikach można zauważać, że najczęstszą trasą wędrówki pakietów była ścieżka AS8970 – AS8501, czyli z sieci Politechniki Wrocławskiej do sieci PIONIER.

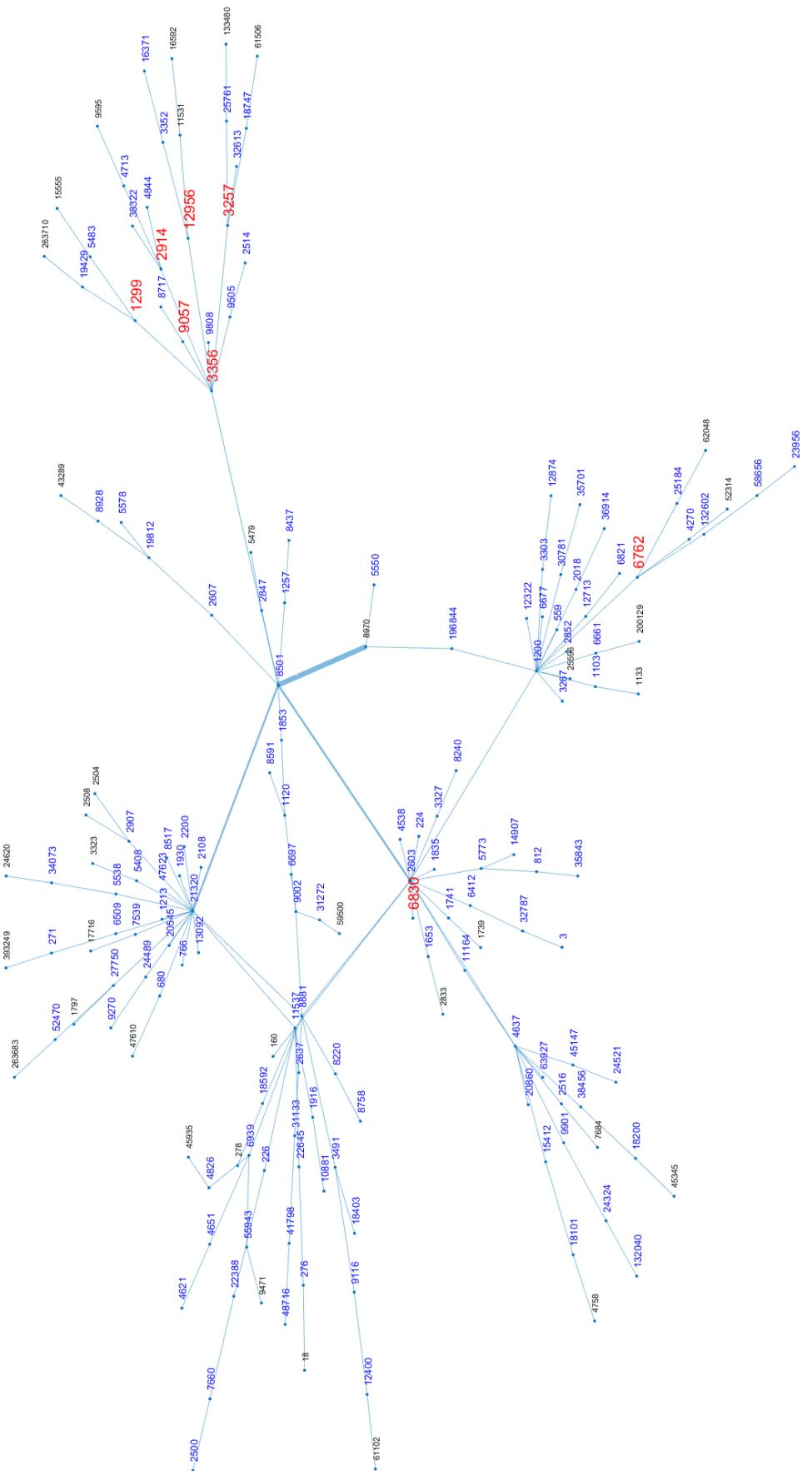
Na podstawie tak wygenerowanego grafu można porównać wydajność systemów autonomicznych znajdujących się zaraz po rozwidleniu ścieżki. Jeżeli będą wykazywały się one podobieństwem w działaniu, ich wydajność zależna jest prawdopodobnie od punktów znajdujących się w środku trasy poprowadzonej od agenta pomiarowego. Znając taką zależność istnieje szansa, że wykonując pomiar dla jednego z tych serwerów, można byłoby wykonać predykcję działania drugiego.

### 5.1.3 Tworzenie geograficznej mapy ścieżek AS

Wyniki otrzymane we wcześniejszym badaniu pokazywały jedynie zauważone połączenia między systemami autonomicznymi, wyciągnięte na podstawie tras wędrówki pakietów do każdego ze zdalnych hostów. Brakowało w nich informacji na temat geograficznego położenia kolejnych punktów znajdujących się w ścieżce. Następne wykonane badania miały na celu stworzenie grafu połączeń AS, który zostanie nałożony na mapie świata.

Wcześniej należało jednak wyciągnąć informacje na temat położenia geograficznego każdego z punktów pośrednich. Lokalizacja serwerów końcowych była już znana po wykonaniu wcześniejszych analiz. W przypadkach pośrednich, skala zadania była zbyt duża, by sprawdzanie położenia geograficznego było przeprowadzane w sposób manualny. Badanie to należało zautomatyzować.

W pomiarach zdecydowano się na ponowne użycie narzędzia traceroute w celu odpowiedniego formatowania danych wejściowych. Tym razem była to ulepszona wersja tego narzędzia (według zapewnienia autorów) – *paris-traceroute*, której opis dostępny jest na stronie [57]. Wyniki jego działania wykorzystano w stworzonym skrypcie.



Rysunek 5.3 Utworzony graf ścieżek AS

W celu wyciągnięcia współrzędnych geograficznych serwerów pośrednich, znajdujących się w ścieżkach połączeń, ponownie napisano skrypt C#, którego działanie zostało opisane poniżej.

Po wczytaniu ścieżki przeskoków do każdego serwera zdalnego (powstałej w wyniku działania narzędzia paris-traceroute), skrypt wykonywał:

- Wyekstrahowanie wartości adresów IP routerów znajdujących się w ścieżce (za pomocą narzędzia Regex). Wartości zaszyfrowane były omijane.
- Sprawdzenie systemu autonomicznego każdego z adresów IP za pomocą zapytania wykonywanego do serwisu *Cymru* [54]. W przypadkach, w których nie znaleziono odpowiedniego ASN, należało wykonać validację manualną.
- Filtrację powtarzających się numerów AS. Zostawiano wyłącznie pierwszy skok (wejście) do systemu autonomicznego. Zbyt duża ilość skoków na mapie mogła być niekorzystna dla wyraźności prezentowanych wyników.
- Jako punkt końcowy każdej z przefiltrowanych ścieżek, dodawany zostawał adres IP hosta docelowego.

Kolejnym etapem działania skryptu było wykonywanie zapytań do API *ipinfo.io* na podstawie udostępnionej przez serwis dokumentacji. Żądania HTTP zwracały informacje, z których parsowana była lokalizacja w postaci długości i szerokości geograficznej dla każdego IP, znajdującego się w przefiltrowanej ścieżce.

Wartości zwracane w odpowiedzi z serwera to IP, nazwa domenowa, długość i szerokość geograficzna, organizacja, miasto, region oraz państwo. Podczas jednego dnia można było wykonać 1000 takich zapytań, co wystarczyło dla badanych przypadków.

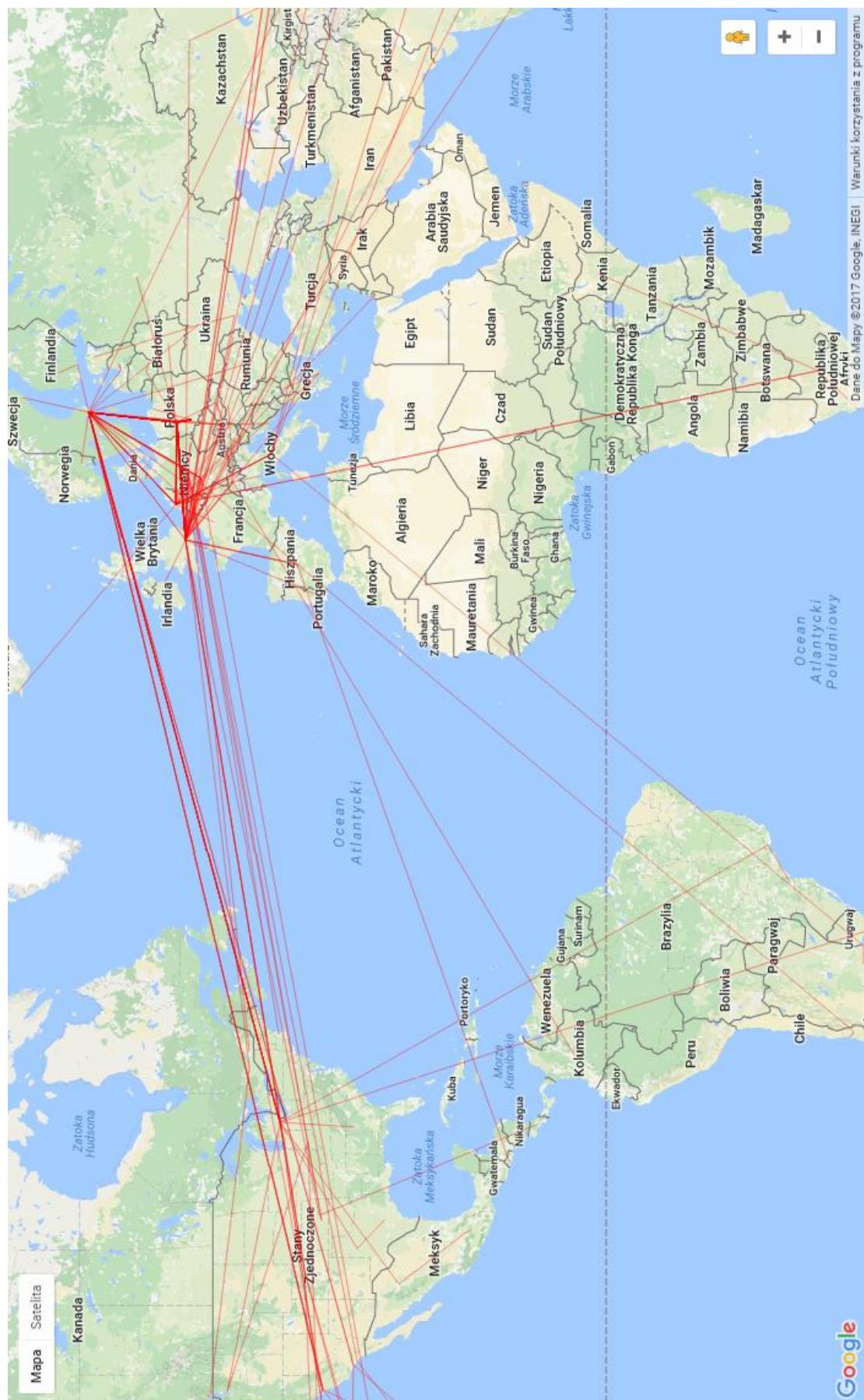
Z otrzymanych wyników należało stworzyć mapę świata. Pierwotnym założeniem było ponowne użycie środowiska Matlab. Wykonanie tego zadania było możliwe, przez manualne rozmieszczenie punktów grafu na mapie świata w odwzorowaniu Merkatora [58]. Jednakże opcja ta wydawała się przynosić wymierne efekty w porównaniu do pracy, jaką należało włożyć w jej wykonanie.

Kolejną możliwością było wykorzystanie Google Static Maps API. Jest to serwis, który na podstawie wpisanego URL, pozwala na generację obrazów map z nałożonymi wartościami. W możliwości działania wpisuje się między innymi łącznie punktów na mapie. Po stworzeniu generatora URL na podstawie posiadanych danych okazało się jednak, że przekracza on maksymalną długość założoną dla zapytania.

Ostatecznie, w celu stworzenia geograficznej prezentacji grafu systemów autonomicznych, użyto Framework ASP.NET MVC. Stworzono dzięki niemu stronę internetową, która używała Google Maps API w celu wyświetlania połączeń między poszczególnymi punktami ścieżek.

Strona internetowa została opublikowana w sieci pod adresem <http://mgrmap.azurewebsites.net/>. Użycie technologii Google zapewnia możliwość przesuwania się po mapie świata, jej oddalanie i zbliżanie w celu dokładniejszego przyjrzenia się ścieżkom. Przykładowy widok mapy został przedstawiony na Rysunku 5.4.

Podczas poszukiwań znaleziono również mapę przedstawiającą podwodne połączenia sieciowe, która może przydać się w celu dopełnienia wykonywanej analizy [59].



Rysunek 5.4 Mapa połączeń między systemami autonomicznymi

## 5.2 Analiza najdłuższych ścieżek AS

Pomimo wcześniejszych założeń o niezmienności ścieżki AS, w badaniach postanowiono wykonać dodatkowe pomiary mające sprawdzić, jak faktycznie ścieżki te zmieniają się w czasie.

Na początek wybrane zostały wybrane cztery serwery, o maksymalnej odnotowanej długości ścieżki AS (7 skoków):

- Hasaki, Japonia (*ftp.nara.wide.ad.jp*)
- Numea, Nowa Kaledonia (*debian.nautile.nc*)
- Dhaka, Bangladesz (*mirror.dhakacom.com*)
- Holon, Izrael (*debian.co.il*)

Przez kilka dni (dokładnie od 7 do 9 kwietnia 2017 roku), do powyższych serwerów wykonywano zapytania narzędzia *traceroute*. Na podstawie analizy tych wstępnych badań zauważono, że jedynie serwer znajdujący się w Bangladeszu wykazywał się zmianami w trakcie ich trwania.

Z tego powodu postanowiono wykonać dalszą analizę tego serwera, jako przypadku o największej możliwości zmian w dłuższym czasie trwania pomiarów.

Początek kolejnej serii pomiarowej, wyłącznie dla serwera w Bangladeszu, miał miejsce 21 kwietnia 2017 roku i trwał aż do 16 maja tegoż roku. Pomiar narzędziem *traceroute* był wykonywany co 15 minut.

Otrzymane dane przetworzono skryptem napisanym w C#. Jego wyniki pokazywały osobliwe ścieżki, którymi poruszały się pakiety oraz wyliczał ilość takich samych ścieżek w trakcie trwania badań. Wyniki zapisane zostały do tabeli w programie Excel, przedstawionej poniżej jako Tabela 5.1.

1871				
1864	3	3	1	412
[AS8970]	[AS8970]	[AS8970]	[AS8970]	[AS8970]
[AS8501]	[AS8501]	[AS8501]	[AS8501]	[AS8501]
[AS2603]	[AS2603]	[AS21320]	[AS21320]	[AS2603]
[AS1200]	[AS6762]	[AS21320/AS20965]	[AS21320/AS20965]	[AS51531/AS6695]
[AS6762]	[AS132602]	[AS1200]	[AS1200]	[AS3491]
[AS132602]	[AS58656]	[AS6762]	[AS6762]	[AS9498]
[AS58656]	[AS23956]	[AS132602]	[AS17494]	[AS58656]
[AS23956]		[AS58656]	[AS58656]	[AS23956]
		[AS23956]	[AS23956]	

10 zł

12 maj 12:15+

Tabela 5.1 Zmiany ścieżki AS dla serwera w Bangladeszu

Wyniki pomiarów pokazały, że pomimo znacznej odległości do serwera docelowego, długość ścieżki AS była stała. Zielone wyniki tabeli są do siebie bardzo podobne, a ilość innych ścieżek niż główna w tym okresie wynosi zaledwie 7, co sugeruje chwilowe przeciążenie na routerach pośrednich. Od 12 maja (czyli po 3 tygodniach) ścieżka uległa stałej zmianie (tabela niebieska), lecz jej długość pozostała taka sama. Na podstawie tych wyników uznano, że założenie niezmienności ścieżek AS nie powinno być rzutujące na wyniki ewentualnych analiz.

## 6. Przeprowadzenie badań pomiarowych w rzeczywistych warunkach Internetu w systemie MWING

W rozdziale tym opisany zostanie proces zbierania pomiarów przez agentów MWING zainstalowanych we Wrocławiu i Poznaniu. Sekcje te opiszą czas wykonywania badań oraz zachowanie agentów zaobserwowane podczas tych pomiarów.

### 6.1 Czas wykonywania badań

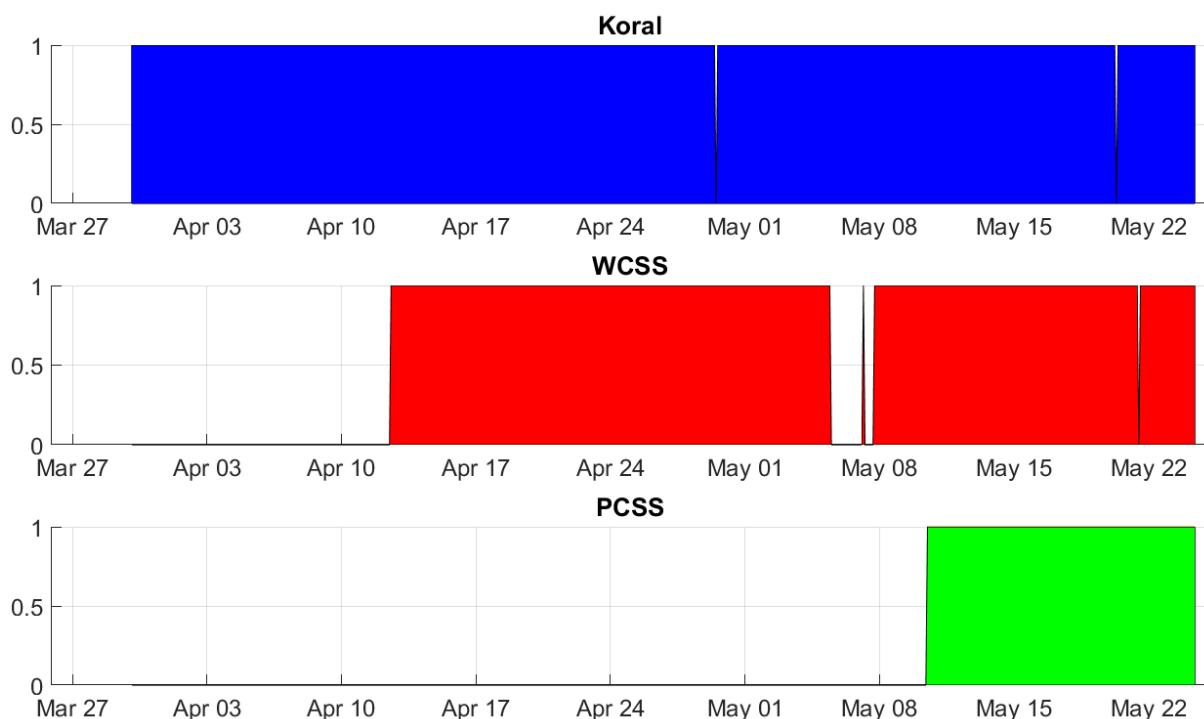
Terminy początku wykonywania pomiarów dla każdego z serwerów były różne. Wynikało to przede wszystkim z pierwotnej koncepcji używania hostów rozmieszczonych w różnych punktach świata, jak również nakładu pracy włożonego na dopełnienie wszelkich formalności w infrastrukturze PLATON oraz konfigurację uruchomionych serwerów.

Czasy rozpoczęcia pomiarów zostały przedstawione poniżej, godzina została podana w czasie lokalnym dla Polski w okresie letnim (GMT+2).

- **Koral** – czwartek, 30 marca 2017 2:00,
- **WCSS Platon** – środa, 12 kwietnia 2017 14:00,
- **PCSS Platon** – środa 10 maja 2017 12:00.

Jak widać, różnica między czasem rozpoczęcia pomiaru na serwerze oryginalnym Koral, a serwerem PCSS wynosi około 40 dni. Ponieważ w założeniach badań ustalono, że czas trwania pomiarów powinien wynosić przynajmniej dwa tygodnie, obserwacje z każdego z tych serwerów zebrano w dniu **24 maja 2017, o godzinie 12:00**.

Rysunek 6.1 przedstawia czas działania tych serwerów, razem z uwzględnieniem okresów nieprawidłowego ich działania.



Rysunek 6.1 Wykresy stabilności działania serwerów

Na serwerze Koral odnotowano dwa momenty, w których serwer nie zebrał danych pomiarowych. Prezentowane są one przez dwa pojedyncze skoki na wykresie. Ponieważ dotyczą one wyłącznie pojedynczych serii pomiarowych, nie stanowi to większego problemu.

Stabilność działania wyglądała inaczej w przypadku agenta WCSS, w którym można zaobserwować dłuższy okres trwania nieprawidłowości. Problem rozpoczął się 5 maja. Związały był z nagłym, nieoczekiwany wyrejestrowaniem maszyny wirtualnej z serwisu. Istnieją różne możliwości, które mogły być przyczyną takiego działania. Najbardziej prawdopodobną były niewystarczające zasoby na serwerze i zwalnianie stanowisk o niższym priorytecie (w tym agenta MWING). Niestety sytuacji tej nie można było przeciwodzielić, a ponowne uruchomienie maszyny na dłuższy okres stało się możliwe dopiero 8 maja. W późniejszym okresie pomiarowym można zauważać również błąd pojedynczej serii pomiarowej, związany z niewydłużeniem czasu trwania maszyny na serwerze (błąd autora).

Serwer w Poznaniu wykazywał się pełną stabilnością działania w okresie trwania pomiarów.

Łącznie, liczba ominiętych pomiarów dla serwera Koral wynosi 2 (0,3%), a dla WCSS 27 (5,3%).

Dłuższe okresy nieprawidłowego działania serwera są znacznym problemem w przypadkach używania metod analizy szeregow czasowych, gdzie kluczowym założeniem jest ciągłość wykonywanych pomiarów. Temat ten zostanie poruszony szerzej przy okazji przedstawiania przebiegów mierzonych wartości w czasie.

## 6.2 Stabilność serii pomiarowych

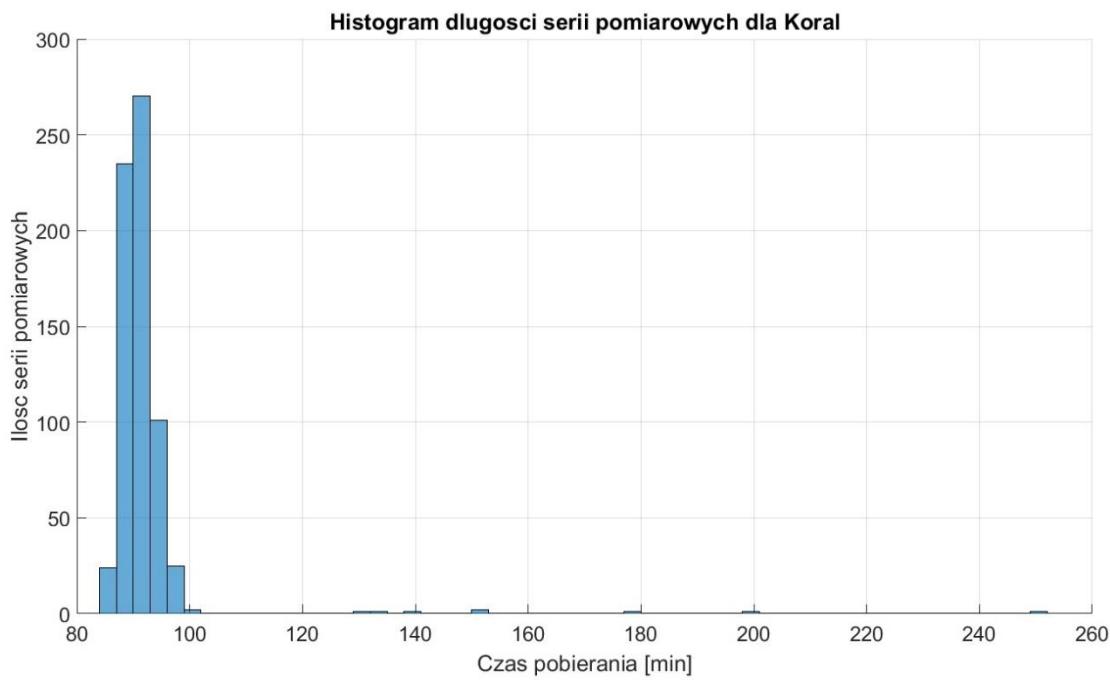
Stabilność działania serwera nie musi oznaczać stabilności zbierania pomiarów. Na etapie tworzenia stanowiska badawczego ustalone, że pomiary wykonywane będą co dwie godziny. Prawidłowość założenia została empirycznie potwierdzona dla każdego z serwerów.

Jednakże, w przypadku przeciążenia sieci, w której znajdował się agent pomiarowy, mogło dojść do przypadków, w których wykonanie pojedynczej serii pomiarowej trwało dłużej niż założony czas dwóch godzin. Takie sytuacje należało odnotować, a w przypadku częstego ich występowania, zwiększyć czas między wykonywanymi seriami.

Zarówno w tym rozdziale, jak również w kolejnych, na wykresach przedstawiane będą wyłącznie pojedyncze przypadki, dające ogólny pogląd opisywanych danych. Pokazywanie zachowań wszystkich serwerów wiąże się ze zbyt dużą ilością danych graficznych, co mogłoby przyczynić się do braku przejrzystości pracy magisterskiej. Opisywane przypadki, które nie wystąpią w głównym tekście pracy, znajdują się na końcu, w sekcji **Załączniki**.

Na potrzeby badania długości serii pomiarowych stworzono skrypt C#, który na podstawie pliku zwierającego wszystkie obserwacje wyliczał całkowity czas trwania serii pomiarowych. Czasy trwania były wyliczane również osobno dla każdej wielkości plików (mechanizm w agentach zakładał pobieranie plików o jednym rozmiarze ze wszystkich serwerów, a następnie rozmiarze kolejnym).

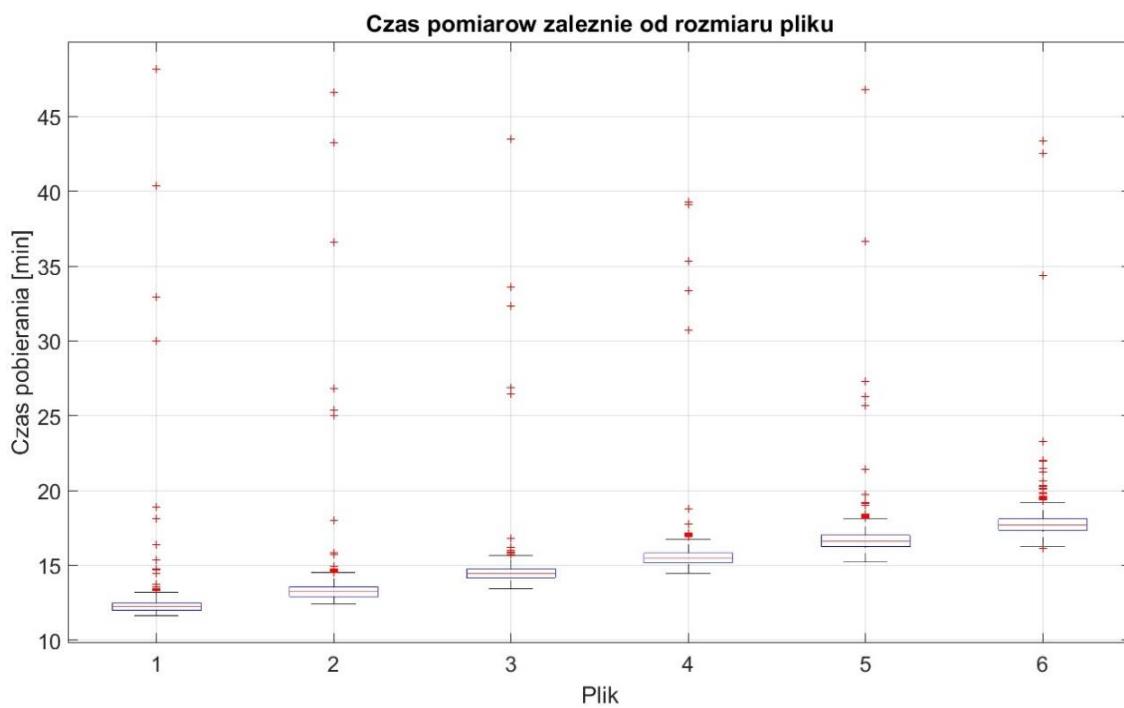
Wyniki działania powyższego skryptu zostały użyte w środowisku Matlab, za pomocą którego otrzymano histogram całkowitego czasu trwania pomiarów, a także wykresy pudełkowe tegoż czasu dla każdej z wielkości plików. Rysunki 6.2 i 6.3 przedstawiają przykładowe wyniki dla serwera Koral.



Rysunek 6.2 Histogram całkowitego czasu serii pomiarowych [Koral]

Na powyższym histogramie można dostrzec, że większość wykonanych serii pomiarowych znajduje się w założonym czasie (poniżej 120 minut). Pojedyncze przypadki związane były z przeciążeniami wewnętrznej sieci Politechniki Wrocławskiej. Większość z nich (m.in. długości serii bliskie 4 godzinom) zostało odnotowane w okresie, w którym sieć wewnętrzna Politechniki miała problem z brakiem odpowiedzi serwera DNS.

Dla dwóch pozostałych serwerów serie pomiarowe nie przekroczyły progu 120 minut. Znajdująca się blisko kręgosłupa sieć PIONIER wykazywała się więc dużą stabilnością.



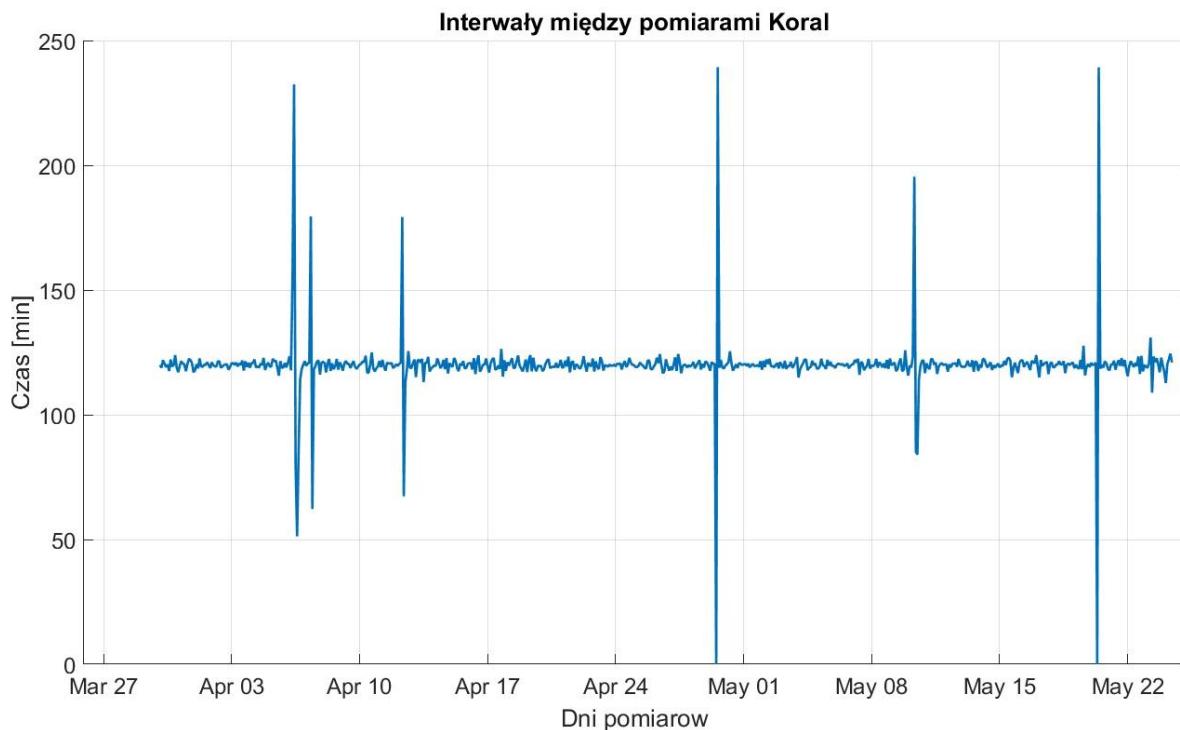
Rysunek 6.3 Wykres pudełkowy czasów pobierania dla każdego pliku [Koral]

Wykres pudełkowy pokazuje czas pobrania dla każdego z plików. Widać na nim, że transfer zawartości dla serwera Koral zawiera się w przedziale 13 minut dla plików o rozmiarze 0.5 MB do 19 minut dla 3 MB. Stosunek ten nie rośnie liniowo, co spowodowane jest mechanizmem *Slow Start* w połączeniach TCP, które umówiono wcześniej.

Na powyższym wykresie pudełkowym widać również, że zbierane pomiary cechują się niską wariancją. Wartości odstające wynikają przede wszystkim z istnienia okresów przeciążenia sieci.

Agenci w sieci PIONIER posiadają mniej wartości odstających, a czasy ich transferów są krótsze, przede wszystkim dla większych plików. Stanowisko w Poznaniu działa szybciej niż to we Wrocławiu (około minuty krócej dla każdej wielkości pliku).

Innym sposobem przedstawienia stabilności pomiarów jest wykres, wykorzystany we wcześniejszych badaniach w sieci Politechniki Wrocławskiej w publikacji [27]. Przedstawia on interwały między kolejnymi pomiarami tego samego pliku z tego samego serwera. W przypadku zebranych pomiarów był to plik o rozmiarze 3MB dla serwera 47 (wykorzystywanego w późniejszych analizach).



Rysunek 6.4 Interwały między pobraniemiami pliku 3MB na serwerze #47 [Koral]

Wykres ten przedstawia wyniki potwierdzające wcześniejszy histogram stabilności serii pomiarowych. W przypadku tym można zauważać jednak okresy, w których pomiary cechowały się największymi wartościami odchyлеń od założonych 120 minut. Z wykresu agenta pomiarowego Koral należy wyłączyć dwa skoki (koniec kwietnia i 18 maja), wynikające z błędów systemu.

Podobne wykresy zostały stworzone dla pozostałych agentów pomiarowych. Oprócz okresu, w którym maszyna w WCSS Platon była niesprawna, wykresy dla obu przypadków cechują się stabilnością przebiegu na poziomie 120 minut.

Kolejny rozdział przedstawi dokładniejszą analizę zebranych pomiarów.

## 7. Analiza zebranych pomiarów

W rozdziale tym przedstawione zostaną wyniki badań na przeprowadzonych pomiarach, niezwiązane bezpośrednio z metodą eksploracji danych. Ich analiza jest dopełnieniem poruszанego problemu predykcji wydajności. W kolejnych sekcjach omówione zostaną opracowane skrypty, których zadaniem było odkrywanie charakterystyk i zależności w otrzymanych obserwacjach.

Przytoczony zostanie problem brakujących danych pomiarowych, który pojawił się podczas zbierania obserwacji przez agentów. Wynikał on z czynników przedstawionych w rozdziale wcześniejszym – problemów po stronie serwera pomiarowego, obecnych przede wszystkim dla agenta WCSS.

W części dalszej, przedstawiony zostanie związek zaobserwowanej wartości przepustowości połączenia TCP oraz czasu odpowiedzi serwera (RTT). Wcześniej badania potwierdziły, że zmienne te są od siebie zależne [27]. Postawione zostanie pytanie, czy czas odpowiedzi wystarczy, by otrzymać przybliżoną wartość transferu z danego serwera.

Przytoczone zostaną inne badania, wykonywane na terenie Politechniki Wrocławskiej. Uzyskane w nich wyniki zostaną porównane z tymi, otrzymanymi z wykonywanych pomiarów.

W ostatniej sekcji przedstawiony zostanie przypadek, który zaobserwowano w trakcie wykonywania pomiarów. Fakt istnienia przytoczonego zjawiska stanie się przyczyną pytań, dotyczących wykonywania badań wydajności sieci na maszynach wirtualnych.

### 7.1 Problem brakujących obserwacji

We wcześniejszym rozdziale pracy magisterskiej wspomniano, że w badaniach wykorzystujących analizę szeregów czasowych, jednym z kluczowych założeń jest ciągłość pomiarów. Rozumiana jest ona jako występujące w równych odstępach czasu obserwacje, które nie posiadają wartości brakujących.

Podczas wykonywania pomiarów przez agentów pomiarowych dochodziło do sytuacji, gdy obserwacje te należało odrzucić. Możliwymi powodami ich wystąpienia były m.in.:

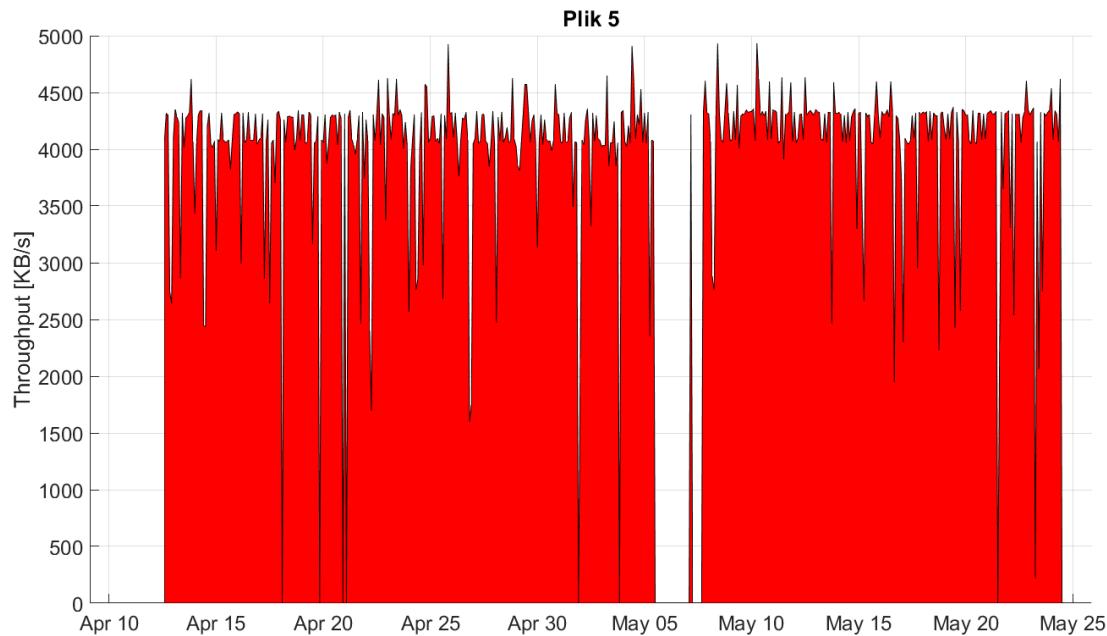
- Błędne działanie agenta, kończące się niewykonaniem danej serii pomiarowej, która powinna wystąpić co 2 godziny.
- Błędy w zebranej próbce – wynikające z nieprawidłowego działania połączenia TCP, problemów po stronie serwera zdalnego lub innego błędu, odnotowanego przez narzędzie pomiarowe MWING.

Na potrzeby przedstawienia problemu brakujących pomiarów, wykorzystano skrypt opracowany w środowisku Matlab.

Główym założeniem tego skryptu było przedstawienie zmian przepustowości połączenia TCP oraz czasu odpowiedzi w czasie dla wybranego serwera. Działanie skryptu zakładało możliwość zbadania przebiegów tych wartości w różnych odcinkach czasu – możliwe było ustalenie okna trwania przebiegu. Obserwacje błędne były odrzucane i liczone jako brak pomiaru na potrzeby rysowanego wykresu zmian wartości w czasie.

Przykładowe jego wywołanie, wykonane na potrzebę wizualizacji brakujących pomiarów, zostało przedstawione na Rysunku 7.1. Jest to przebieg przepustowości jednego z serwerów (dokładnie #9) dla agenta WCSS, czyli agenta cechującego się największą ilością niewykonanych pomiarów. Widać na nim zarówno pojedyncze odrzucone obserwacje

(wartości schodzące nagle do 0), jak również okres na początku maja, w którym agent był wyłączony.



Rysunek 7.1 Przykładowy przebieg przepustowości [WCSS]

W przypadku, gdy badania uzależnione są od ciągłości pomiarów, dla powyższego przebiegu można wykonać różne działania, które pozwolą na uzyskanie ciągłości.

Najprostszym z używanych sposobów jest wyliczenie średniej dla całego badanego zjawiska i uzupełnienie brakujących miejsc tą właśnie wartością. Bardziej wysublimowaną metodą jest wyliczenie wartości średniej z ostatniego i kolejnego poprawnie wykonanego pomiaru. Możliwe jest również wykorzystanie okna, które uśredni wartości znajdujące się w określonym przedziale czasowym.

Znając postać przebiegu badanej funkcji, można również wykorzystać wiedzę ekspercką (świadomość badanego zjawiska) dla wygenerowania brakujących wartości.

Większość z wykorzystywanych metod eksploracji danych jest niepodatna na brakujące wartości. Dlatego w badaniach prowadzonych na potrzeby pracy magisterskiej problem braku pomiarów nie powinien wpływać na wyniki analiz. Metodologia ta uzależniona jest przede wszystkim od ilości badanych próbek. Dla przytoczonego agenta WCSS posiadana jest wystarczająco duża liczba obserwacji (476 przeprowadzonych serii pomiarowych).

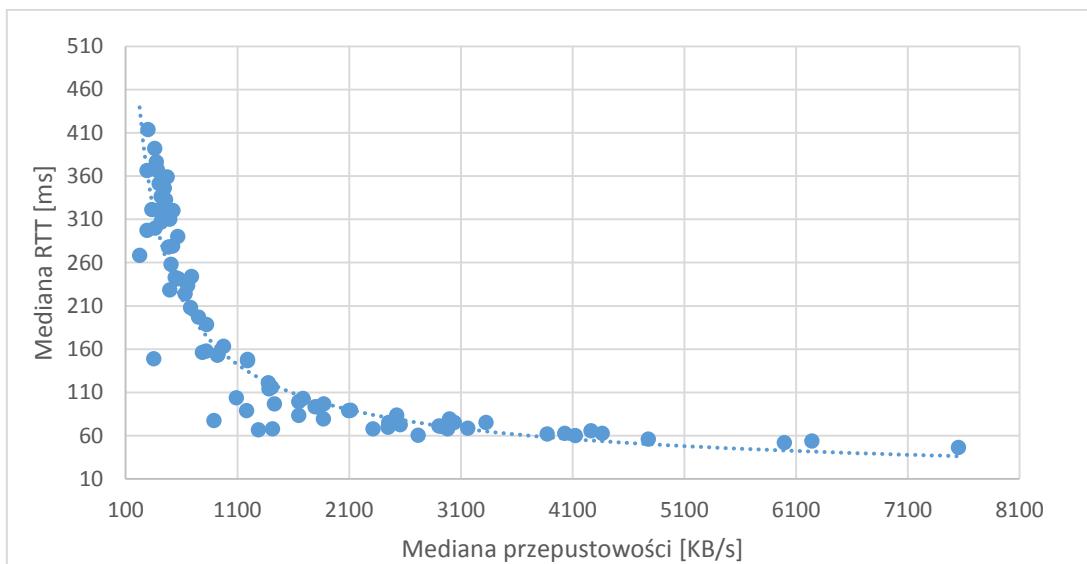
## 7.2 Zależność RTT i przepustowości TCP

W badaniach przeprowadzonych w latach wcześniejszych na terenie Politechniki Wrocławskiej, pokazano wykres zależności średniej wartości przepustowości od czasu odpowiedzi (RTT) dla badanych serwerów [27]. Starano się sprawdzić własność, że połączenia z niższą wartością RTT mają skłonność do przesyłania większej ilości danych w warstwie HTTP, w jednakowym okresie czasu.

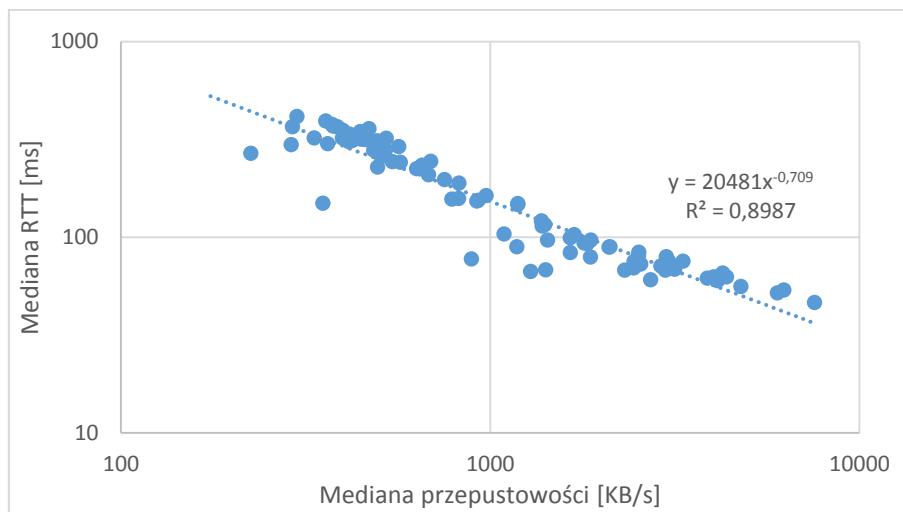
Celem przeprowadzonej analizy było ocenienie, czy znajomość wartości RTT (która można uzyskać dzięki prostym narzędziom, takim jak *ping*) może być wystarczająca do wyprowadzenia wartości przepustowości dla nowego serwera, który nie pojawił się w przeprowadzonych pomiarach.

Ta zależność została zbadana również na potrzeby pracy magisterskiej, przy wykorzystaniu wykonanych pomiarów. Opracowano skrypt, którego celem było wyliczenie median zaobserwowanych wartości RTT dla każdego z serwerów, a także wyliczenie mediany wartości przepustowości dla każdej z wielkości plików, pobieranych z określonego serwera.

Na podstawie uzyskanych wyników tego skryptu, udało się wyrowadzić wykres, na którym naniesiona została zależność mediany przepustowości i RTT dla każdego z serwerów (punkty na wykresie). Przykładowy wykres znajduje się na Rysunku 7.2. Można zauważyć, że rozłożenie tych wartości przypomina funkcję potęgową. Z tego powodu wykonano aproksymację, która jest znacznie lepiej widoczna przy zmienieniu podziału na osiach na postać logarytmiczną (Rysunek 7.3).



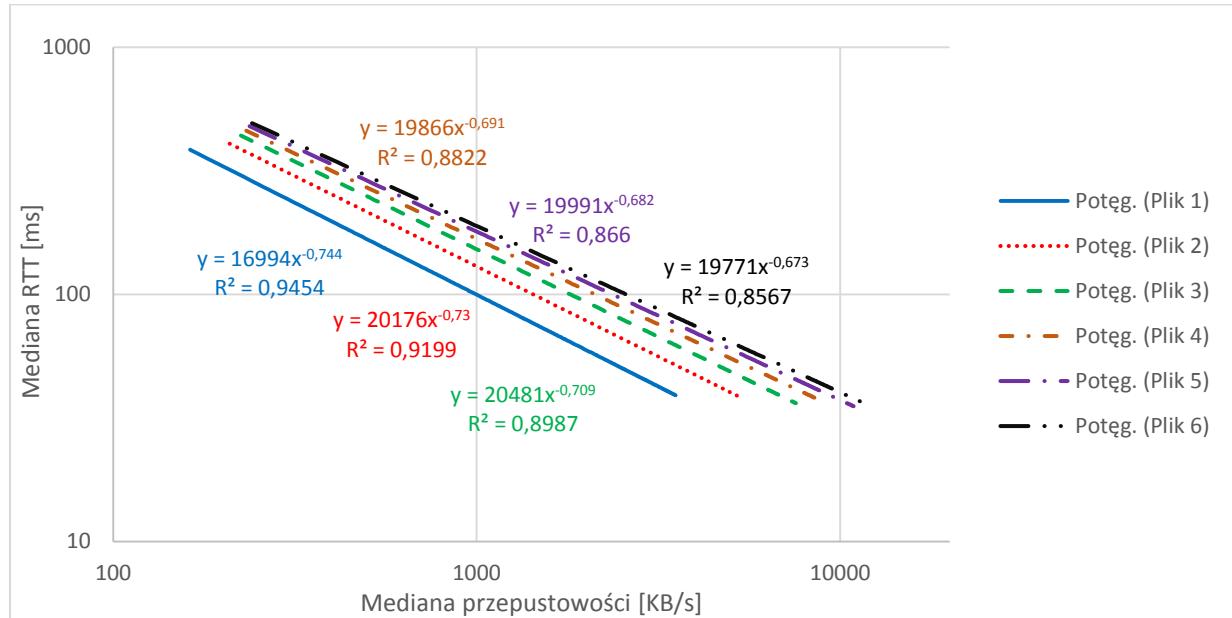
Rysunek 7.2 Zależność median RTT od przepustowości dla pliku 1.5MB [WCSS]



Rysunek 7.3 Zależność median RTT od przepustowości dla pliku 1.5MB (logarytmicznie) [WCSS]

Znajdujący się na wykresie współczynnik  $R^2$  sugeruje, że dopasowanie poprowadzonej linii regresji jest bardzo dobre. Już wartości tego współczynnika znajdująca się powyżej 0.7 powinny sugerować, że przyjęcie liniowej zależności (na wykresie logarytmicznym) między dwoma badanymi zmiennymi jest możliwe.

Dla każdego z agentów pomiarowych wyliczono linie regresji dla median RTT i przepustowości. Zapisano je w postaci pojedynczych wykresów, których przykład znajduje się na Rysunku 7.3. Wyniki aproksymacji dla każdej wielkości pliku naniesiono następnie na pojedynczy wykres, osobno dla każdego serwera pomiarowego. Wykres dla agenta WCSS przedstawiono na Rysunku 7.4.



Rysunek 7.4 Poprowadzenie linii regresji mediany RTT i przepustowości dla każdej wielkości pliku [WCSS]

Na zamieszczonym wykresie widać, że wartość współczynników dopasowania  $R^2$  znajduje się na wysokim poziomie. Można również zauważyć tendencję, że im większy plik, tym mniejsza jest wartość tego dopasowania. Wynik taki wydaje się logiczny, zważając na fakt, że istotą pomiaru czasu odpowiedzi serwera jest wysyłanie bardzo małego pakietu danych. Im bliżej jego wielkości znajduje się badany plik, tym bardziej wyniki RTT i przepustowości powinny być do siebie podobne.

Dopasowanie funkcji dla każdego z agentów zostało przedstawione w Tabeli 7.1. Uzyskane wyniki pokazują, że najlepsze dopasowanie uzyskano dla serwera WCSS. Zarówno Koral, jak i PCSS cechują bardzo podobne wyniki. Wartość dopasowania dla pierwszego z nich spada poniżej 0,8 dla największego badanego pliku, niemniej nadal sugeruje ona korelację między badanymi zmiennymi (jest większa niż 0,7). W przyszłości, wartości te warto byłoby sprawdzić dla plików o znacznie większym rozmiarze.

Wielkość pliku [MB]	Koral	WCSS	PCSS
0,5	0,9355	0,9454	0,9334
1	0,9077	0,9199	0,9038
1,5	0,8786	0,8987	0,8614
2	0,853	0,8822	0,8542
2,5	0,8291	0,866	0,827
3	0,7995	0,8567	0,8218

Tabela 7.1 Dopasowanie linii regresji ( $R^2$ ) median RTT i przepustowości dla agentów pomiarowych

Na podstawie funkcji tangens, wyliczone zostało nachylenie funkcji regresji liniowej na otrzymanych wykresach z podziałem logarytmicznym. Wyniki kątów nachylenia zaprezentowano w Tabeli 7.2. Okazało się, że serwery Koral i PCSS (których wyniki dopasowania były podobne), mają nachylenie na poziomie około 45 stopni, podczas gdy w WCSS waha się ono od 34 do 36 stopni. Wyniki sugerują, że wartości mniejsze niż te, utrzymujące się na poziomie 45 stopni pozwalają na lepsze dopasowanie czasu odpowiedzi do przepustowości. Zaobserwowany dla WCSS przebieg funkcji sugeruje odrzucenie twierdzenia o rozkładzie Poissona, stosowanego w wielu modelach analitycznych ruchu w Internecie.

Wielkość pliku [MB]	Koral	WCSS	PCSS
0,5	45,82	36,58	45,82
1	45,79	35,94	45,2
1,5	45,65	35,11	44,42
2	45,54	34,45	43,89
2,5	45,43	34,14	43,38
3	45,09	33,78	42,95

Tabela 7.2 Kąt nachylenia [ $^{\circ}$ ] funkcji liniowej na wykresie logarytmicznym

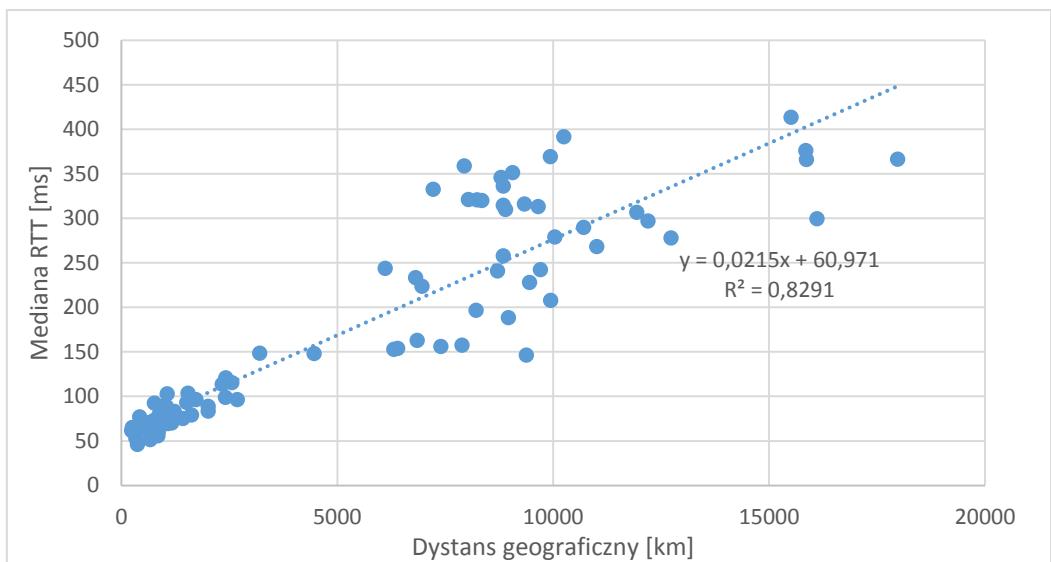
W dalszych analizach zmierzono również zależność czasu odpowiedzi serwerów (zachowanych w postaci mediany wszystkich pomiarów dla danego kierunku) od wartości, uznawanych za odległość serwerów w Internecie: odległości geograficznej, długości ścieżki IP oraz długości ścieżki AS. Badania te wykonano wyłącznie dla danych dostępnych dla agenta WCSS, który wykazywał się największym dopasowaniem poprowadzonej linii regresji. Jeżeli między miarami odległości również będzie silna zależność, powinny być one znaczące w przewidywaniu przepustowości (mocno powiązanej z RTT).

Na Rysunku 7.5 przedstawiono zależność liniową między RTT a dystansem geograficznym. Poprowadzona regresja liniowa cechuje się wysokim współczynnikiem dopasowania, o wartości około 0,83. Uzyskany wynik wydaje się dostateczny do stwierdzenia, że użycie dystansu do predykcji przepustowości powinno dawać zblżone do faktycznych wartości predykcje.

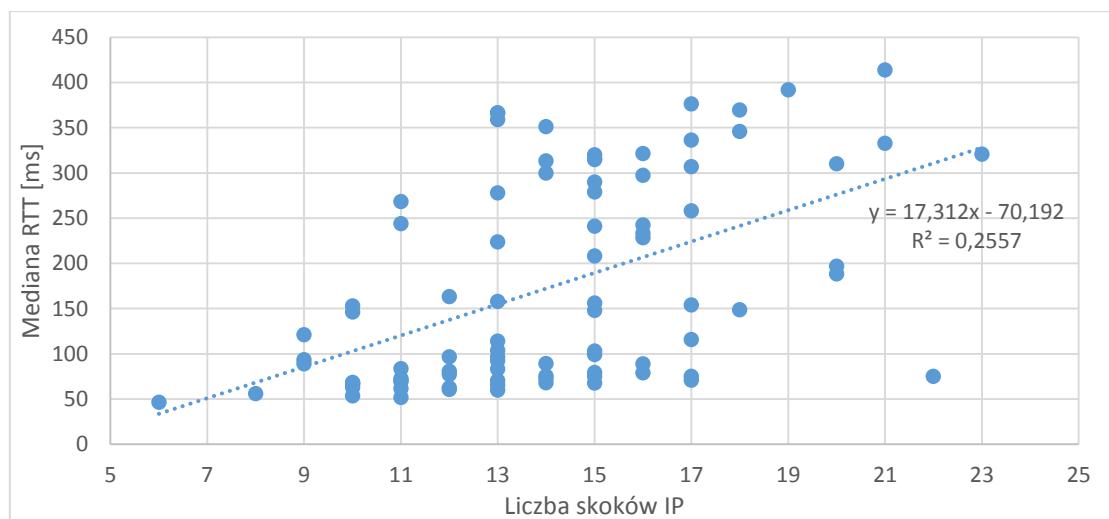
Na Rysunkach 7.6 i 7.7 pokazano zależność RTT między dwoma pozostałymi miarami odległości, uzależnionymi od przeskoków pakietów danych w kierunku serwera. Niestety, całkowite wartości, jakimi cechują się te pomiary, bardzo słabo dopasowują się do poprowadzonej regresji liniowej. Dla badanych danych współczynniki te byłyby niewystarczające do wykonania przewidywań.

Zmierzone dane mogą mieć inną postać niż funkcja liniowa. Być może dopasowanie to byłoby lepsze dla aproksymacji innego przebiegu funkcji.

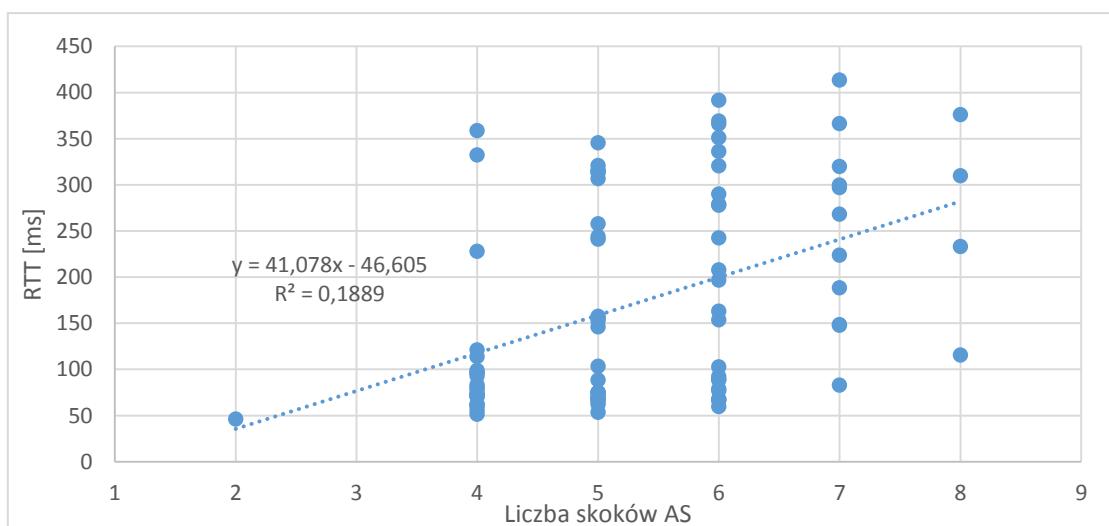
W wynikach uzyskanych za pomocą narzędzia traceroute należy również pamiętać, że we wcześniejszych rozdziałach pracy magisterskiej przyjęto założenie niezmienności ilości przeskoków w czasie i przyjęcie tej wartości na podstawie pojedynczego pomiaru dla każdego z serwerów zdalnych. Założenie to może być mniej rzutujące na wyniki dotyczące przeskoków AS (wniosek z przeprowadzonej analizy najdłuższych ścieżek). Jednak ilość routerów, przez które przechodzi pakiet (skoki IP), jest wartością cechującą się większym zróżnicowaniem w czasie. Dlatego też, słabe dopasowanie w tym przypadku wynikać może z niedostatecznej liczby pomiarów, z których powinno się wyciągnąć średnią ilość przeskoków do każdego z serwerów.



Rysunek 7.5 Wykres zależności median RTT od dystansu geograficznego [WCSS]



Rysunek 7.6 Wykres zależności median RTT od ilości skoków IP [WCSS]



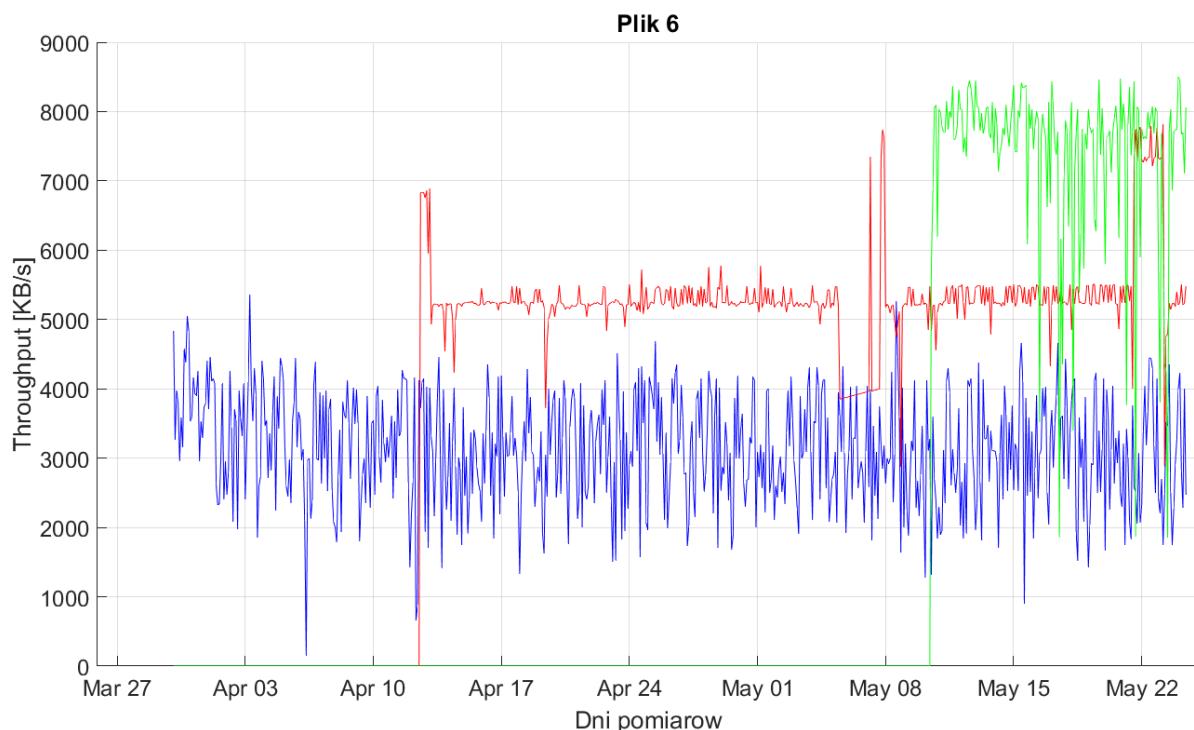
Rysunek 7.7 Wykres zależności median RTT od ilości skoków AS [WCSS]

### 7.3 Problem pomiarów na maszynach wirtualnych

Podeczas wykonywania badań postanowiono sprawdzić działanie każdego z agentów pomiarowych dla serwerów znajdujących się najbliżej i najdalej od Wrocławia w rozumieniu dystansu geograficznego. Były to serwery w Czechach (w Pradze) oraz w Nowej Zelandii.

Uzyskane przebiegi przepustowości TCP dla serwera czeskiego (pliku 3MB) zostały przedstawione na Rysunku 7.8. Podczas ich analizy okazało się, że charakterystyki na serwerze WCSS wykazywały się bardzo dużą stabilnością działania, w porównaniu zarówno do znajdującego się obok (na kampusie Politechniki Wrocławskiej) serwera Koral, jak również drugiego serwera znajdującego się w sieci PIONIER (Poznań). Potwierdzić to miały dodatkowo wyprowadzone wykresy pudełkowe (Rysunek 7.9).

W przypadku pomiarów brakujących (zarówno z powodów niedziałającego serwera, jak i błędu pomiaru), wartość przepustowości na wykresach była uśredniania na podstawie całych danych. Niebieski przebieg należy do agenta **Koral**, czerwony – **WCSS**, zielony – **PCSS**.

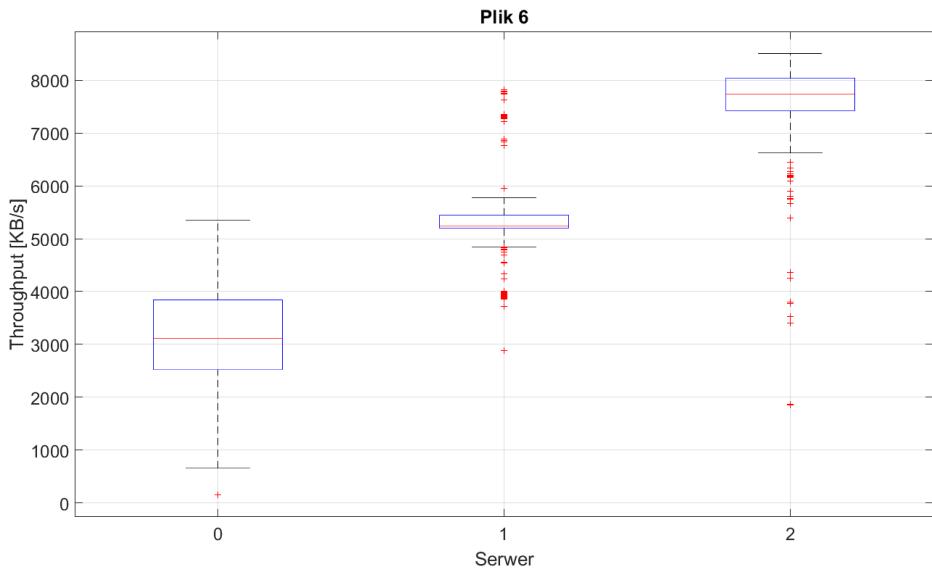


Rysunek 7.8 Przebieg przepustowości dla serwera w Czechach [Plik 3MB]

Zauważone działanie serwera WCSS stało się przyczyną pytań, czy docierający do wrocławskiej sieci PIONIER ruch nie jest w jakiś sposób sterowany. Nie tylko na serwerze czeskim, lecz również w przypadku innych serwerów zdalnych można było zwrócić uwagę, że ruch ten jest uporządkowany, a jego zmiany przebiegają skokowo.

Szukano przyczyny takiego zachowania. Z tego powodu wykonane zostały badania działania agentów, przytoczone wcześniej w podrozdziale 4.6.4. Niestety żadne z nich nie przyniosły odpowiedzi, dlaczego na serwerze WCSS można zaobserwować takie przebiegi. Sprawa ta pozostała niewyjaśniona, na podstawie zebranych obserwacji należy zwrócić się w kierunku administratorów sieci.

Takie działanie serwera jest niewskazane dla badania wydajności Web. Niska zmienność danych sprawia, że ciężko podzielić wartość przepustowości na przedziały.

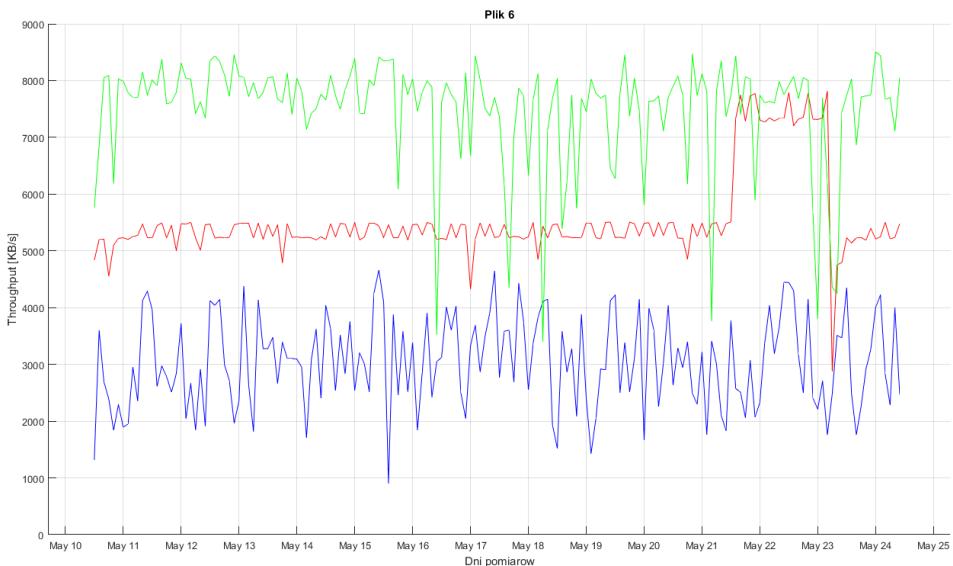


Rysunek 7.9 Wykres pudełkowy przepustowości dla serwera českiego [Plik 3MB]

Bardzo możliwe, że problem stoi po stronie mechanizmów wirtualizacji agenta pomiarowego. W przypadku, gdy na jednym systemie wirtualizacji zarejestrowanych jest dużo maszyn, bardzo prawdopodobnym jest, że ruch takiej sieci jest sterowany w celu optymalizacji działania hostów. Niestety taki mechanizm znacznie zakłoca wyniki pomiarów, które stają się uzależnione przede wszystkim od lokalnego stanu sieci. Tym samym wyniki otrzymane w sieci WCSS nie mogą być rzutujące zarówno na sieci znajdujące się we Wrocławiu, jak również działanie ogólnopolskiej sieci PIONIER.

Pierwotnie działanie to zauważono porównując hosty znajdujące się we Wrocławiu. Przypuszczano, że w Poznaniu będzie można zaobserwować podobne zachowania. Jednak po zebraniu pomiarów z PCSS okazało się, że sterowanie ruchem istnieje tylko we wrocławskim ośrodku sieci PIONIER.

W celu potwierdzenia przypuszczeń, zbadano koreacje przebiegów przepustowości dla ostatnich dwóch tygodni pomiarowych. Ich wygląd przedstawiono na Rysunku 7.10.



Rysunek 7.10 Przebiegi przepustowości TCP dla serwera českiego [6 plik, ostatnie 2 tygodnie]

Sprawdzenie korelacji między przebiegami przepustowości na różnych serwerach wykonano za pomocą skryptu w środowisku Matlab. Współczynniki korelacji Pearsona dla powyższych przebiegów zostały zaprezentowane w Tabeli 7.3. Niskie wartości między każdym z tych serwerów sugerują, że przebiegi nie są uzależnione od działania zdalnego hosta w Czechach (w takim przypadku, serwery powinny zachowywać się podobnie). Otrzymane przebiegi wynikają albo z działania sieci wewnętrznych, w której znajdują się agenci lub przeciążeń występujących po drodze.

Warto zaznaczyć, że w okresie badanego czasu żaden z serwerów nie wykazywał się większymi problemami stabilności działania.

	Koral	WCSS	PCSS
Koral	1	0,0023	0,0843
WCSS	0,0023	1	0,0538
PCSS	0,0843	0,0538	1

Tabela 7.3 Współczynniki korelacji Pearsona dla serwera czeskiego [6 plik, ostatnie 2 tygodnie]

W celu dopełnienia badań serwera w Czechach, sprawdzono przeskoki pakietów do tego hosta za pomocą narzędzia *traceroute*. Działanie to wykonano dla każdego z agentów pomiarowych. Wyniki pokazały, że dla serwerów znajdujących się we Wrocławiu, ścieżka ta przechodzi przez Poznań, a następnie przez Holandię, która dopiero kieruje pakiety do serwera w Czechach. Pomimo zbliżonych ścieżek, przebiegi wrocławskich agentów są od siebie różne, co sugeruje wcześniejszy wniosek o kontroli ruchu we wrocławskiej sieci PIONIER.

Serwer w Poznaniu łączy się bezpośrednio przez punkt sieci PIONIER, znajdujący się po czeskiej stronie granicy. Stamtąd wędruje bezpośrednio do systemu autonomicznego, w którym znajduje się serwer zdalny. Tłumaczy to wysokie wyniki przepustowości otrzymane dla tego agenta.

#### 7.4 Porównanie poprawy wydajności sieci na Politechnice Wrocławskiej

W tym podrozdziale zostanie wykonane porównanie wyników pomiarów z serwera Koral, wykorzystywanego wcześniej w badaniach wydajności HTTP na terenie Politechniki Wrocławskiej. Wykonywane były one kolejno w latach 2003 i 2008, pod uwagę brana była średnia arytmetyczna poszczególnych etapów połączenia, to znaczy:

- DNS – czas potrzebny na uzyskanie adresu IP hosta z serwisów DNS,
- CONN – czas potrzebny na ustalenie połączenia z hostem, równowartość RTT w połączeniach TCP,
- FIRST\_BYT – czas potrzebny na otrzymanie pierwszego pakietu pobieranych danych,
- LEFT\_BYTES – czas transferu pliku.

Prosta transakcja webowa, zawierająca w sobie powyższe etapy, została przedstawiona na Rysunku 2.3 w wykonanej analizie literatury.

Wyliczone zostały wartości średnie poszczególnych etapów połączeń do każdego z badanych kierunków na serwerze Koral. Ponieważ oryginalne badania korzystały z pliku

tekstowego o wielkości około 130 kB, dane zostały ograniczone do najmniejszego pobieranego pliku na potrzeby pracy magisterskiej (0.5 MB).

Wyniki, wraz z wyliczonymi współczynnikami przyspieszenia, przedstawione zostały w Tabeli 7.4.

Czas trwania fazy	Średnia arytmetyczna [ms]			Wsp. przyspieszenia (2003-2008)	Wsp. przyspieszenia (2008-2017)
	2003 r.	2008 r.	2017 r.		
DNS	866	315	191	<b>2,75</b>	<b>1,65</b>
CONN	269	232	146	<b>1,16</b>	<b>1,59</b>
FIRST_BYTE	498	219	220	<b>2,27</b>	<b>0,995</b>
LEFT_BYTES	2672	3143	1494	<b>0,85</b>	<b>2,1</b>
TOTAL	4568	3936	2180	<b>1,16</b>	<b>1,81</b>

Tabela 7.4 Wydajność sieci w Politechnice Wrocławskiej na przełomie kolejnych lat

Otrzymane wyniki pokazują, że wydajność w sieci Politechniki poprawiła się prawie na każdym etapie wykonywania połączenia Web. Jedyny czas, który nie uległ zmianie to czas pobrania pierwszego pakietu danych.

Poprawienie czasów DNS związane może być z większą dostępnością do tych serwisów. Bardzo możliwe, że Politechnika Wrocławska działa w oparciu o wysoką jakość serwery DNS, używane do wyszukiwania IP podanych nazw domenowych.

Czasy CONN, czyli właściwie RTT, również uległy znacznej poprawie. W kolejnym rozdziale pokazane zostanie, że wartość ta dla każdego z serwerów zachowuje się bardzo stabilnie. Obserwacje te sugerują, że przyspieszenie wynikać może z ogólnej poprawy czasu nawiązania połączenia i wysyłania małych pakietów przez sieć.

Zmianom natomiast nie uległ średni czas wysyłania pierwszego pakietu danych. Być może obciążenie serwerów (kolejkowanie wykonywania transferów) pozostało na podobnym poziomie.

Bardzo dużemu przyspieszeniu uległ czas samego transferu pliku. Warto zaznaczyć, że dane dla lat wcześniejszych dotyczą pliku mniejszego (130 kB), podczas gdy wyniki z 2017 roku bazują na pliku około 0.5 MB. Należy pamiętać, że z racji mechanizmu *Slow Start*, czas transferu nie rośnie liniowo, więc otrzymanego współczynnika przyspieszenia nie można wymnożyć przez 4, by uzyskać wyniki, jakie zostały otrzymane dla pliku, na którym bazowały wcześniejsze pomiary.

Należy zaznaczyć, że wyniki zostały otrzymane dla serwerów innych niż te, które zostały wykorzystane w badaniach wcześniejszych. W celu faktycznego dopasowania wyników, należałoby powtórzyć je zarówno dla takiej samej wielkości plików, jak również tych samych serwerów. Przytoczona tabela ma przede wszystkim charakter poglądowy i przybliżający poprawę wydajności obecnego stanu wewnętrznej sieci Politechniki Wrocławskiej.

## **8. Przygotowanie do badań Data Mining**

Rozdział ten skupia się już bezpośrednio na metodyce Data Mining i założeniach związanych z jej wykorzystaniem. Na samym początku przedstawiony zostanie problem, dotyczący różnych możliwości badania dużej ilości danych, otrzymanych na podstawie wykonanych pomiarów. W sekcji pierwszej zostaną wybrane przypadki, których działanie będzie najbardziej interesujące z punktu widzenia pisanej pracy. Kolejny rozdział, przedstawiający wyniki użycia technik eksploracji danych, będzie starał się odpowiedzieć na wybrane możliwości badawcze.

Podstawą jednego z wybranych zadań predykcji będzie określenie serwerów działających w sposób najmniej i najbardziej przewidywalny. Kolejna sekcja skupi się na tym właśnie aspekcie. Przedstawiony w niej będzie współczynnik Hursta, na którego podstawie zostaną wyłonione oba serwery.

Przedostatni podrozdział będzie opisywał poszczególne etapy Data Mining, które zostały wyszczególnione w Rozdziale 3. Teoretyczne założenia, które były w nim zawarte, zostaną wykorzystane w sposób praktyczny.

Ostatnim z punktów tego rozdziału będzie opracowywanie programu, używającego wybrane techniki eksploracji danych. Zostanie przytoczony proces tworzenia skryptu w pakiecie SPSS Modeler 18 pod kątem zamierzonych badań.

### **8.1 Definicja zadania predykcji**

Na podstawie wykonywanych pomiarów, otrzymano bardzo dużą liczbę obserwacji. Głównym zadaniem technik eksploracji danych jest odkrycie wzorców, które nie są łatwo dostrzegalne. Niestety zależności te można odkryć jedynie wtedy, gdy wykorzysta się do tego odpowiednie wartości pomiarowe. Wybór obserwacji poddawanych analizie powinien być nieprzypadkowy i wynikać z istniejącego problemu biznesowego.

Poniżej przedstawione zostaną statystyki na temat przeprowadzonych pomiarów, które potwierdzą konieczność definicji zadania predykcji:

- Trzech agentów pomiarowych: Koral, WCSS, PCSS,
- Łączna ilość obserwacji: 683 tys. w tym:
  - 347096 na serwerze Koral
  - 248468 w WCSS
  - 87696 w PCSS
- 87 badanych serwerów zdalnych,
- 6 badanych wielkości plików.

Definicja zadania predykcji bazowała na problemach użytkownika końcowego, przedstawionych w rozdziałach wcześniejszych:

- *Predykcja wydajności nowego serwera*, bazowana na analizie charakterystyk wszystkich serwerów uczestniczących w pomiarach. Problem ten pojawia się na przykład w przypadku, gdy użytkownik musi dokonać wyboru na serwisie udostępniającym mirroring plików.
- *Predykcja wydajności jednego z badanych serwerów*, bazująca na wcześniejszych obserwacjach jego charakterystyk. Narzędzie to może przydać się w przypadku, gdy użytkownik zobligowany jest do użycia określonego serwisu, lecz swoje działanie chce wykonać w warunkach, kiedy usługa ta będzie działała w sposób wydajny.

Po zdefiniowaniu dwóch podstawowych zadań predykcji, należało przyjrzeć się możliwościami wartości prognozowanych:

- *Wyliczenie wartości przepustowości TCP*, czyli predykcja z użyciem technik regresji. Przepustowość jest parametrem, który bezpośrednio przekłada się na rozumienie wydajności usługi Web przez użytkownika końcowego.
- *Predykcja do jednego z ustalonych przedziałów wartości przepustowości*, powstały przez grupowanie tego parametru (*data binning*). Mogą być to przepustowości uważane za wysokie lub niskie. W prognozach tych wykorzystywane będą algorytmy klasyfikacji.
- *Predykcja do jednego z zauważonych stanów wydajności sieci*. Stany te zbudowane zostaną na podstawie algorytmów klasteryzacji. W prognozach takich przypadków ponownie zostaną użyte techniki klasyfikacyjne. Jest to dwustopniowa technika, która została wykorzystana we wcześniej prowadzonych badaniach [2].

Pewnym było, że w przeprowadzonych badanach wybrane zostaną oba przytoczone zadania predykcji. Pierwsze z nich bazowało będzie na wszystkich pomiarach wykonanych przez poszczególnych agentów. Użyte zostanie do sprawdzenia, w którym ze środowisk pomiarowych można wykonać najlepszą predykcję nowego serwera.

W przypadku drugiego zadania predykcji, na podstawie współczynnika Hursta, wybrane zostaną dwa serwery poddane dalszym analizom. Będą to hosty o potencjalnie najtrudniejszym i najłatwiejszym w predykcji zachowaniu charakterystyki przepustowości.

Badania będą skupiać się na plikach o największym rozmiarze (3 MB). Pliki te, według obecnych informacji na stronie *HttpArchive* [43], powinny być przybliżeniem przeciętnej strony Web (2929 kB). Jest to wielkość większa od tej, która przedstawiona została w podrozdziale 4.2, przed wykonywaniem pomiarów MWING. Sprawdzona zostanie również predykcja dla najmniejszych badanych plików.

Wartości prognozowane przez budowane predyktory będą należały do jednej z trzech grup, przytoczonych wyżej. Przede wszystkim skupiono się na predykcjach z użyciem technik klasyfikacji, w których wyniki predykcji będą w postaci nominalnej.

## 8.2 Estymacja współczynnika Hursta przy wyborze serwerów

W przypadku badań wydajności dla pojedynczego serwera, zdecydowano się na wybór dwóch kierunków, w których jakość predykcji powinna być teoretycznie największa i najmniejsza. Określenie takich serwerów można wykonać na podstawie badania samopodobieństwa otrzymanych w wyniku pomiarów przebiegów przepustowości TCP i czasów odpowiedzi (RTT).

Samopodobieństwo to jeden z terminów charakteryzujących ruch w sieci. Na podstawie przeprowadzonych badań w literaturze stwierdzono, że ruch ten ma podobne charakterystyki, niezależnie od liczby jednoczesnych połączeń w danym łączu fizycznym. Samopodobieństwo jest zjawiskiem, które zachowuje własności statystyczne modelu mimo zmiany skali mierzonego czasu [60]. Oznacza to, że bez względu na przyjęty okres przebiegów ruchu w sieci, powinien on wyglądać identycznie.

Najczęściej stosowaną miarą samopodobieństwa i wybuchowości (będącej również cechą ruchu w Internecie) jest wartość współczynnika Hursta  $H$ , które przyjmuje wartości  $0 < H < 1$ . Wykładnik  $H$  służy do analiz w szeregach czasowych i wykrywania zjawiska tzw. długiej

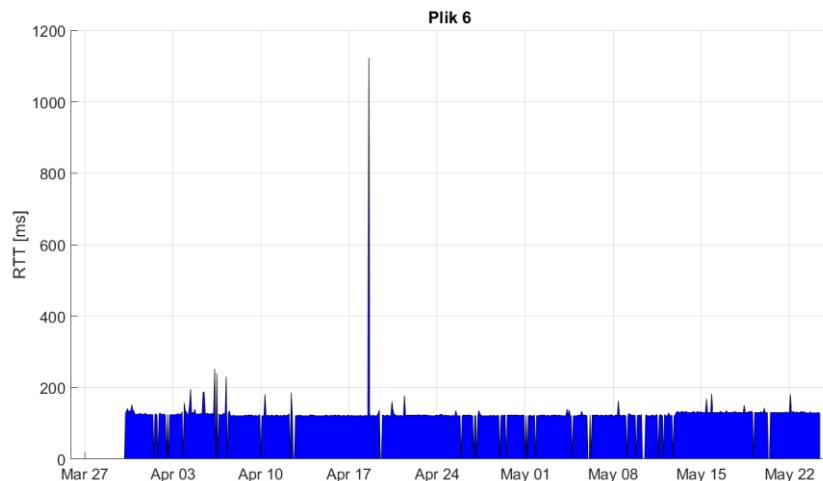
pamięci. Zjawisko to mówi, że w przebiegach w pewnym okresie czasu można zaobserwować powtarzający się trend.

Wartość  $H = 0,5$  oznacza, że proces reprezentowany przez badany szereg wartości zachowuje się jak błędzenie losowe, czyli nie można zaobserwować w nim żadnych zależności. Z punktu widzenia przeprowadzanych badań, serwer najbliższy tej wartości powinien cechować się najgorszą jakością w wykonywanych predykcjach jego działania.

Jeżeli wykładownik Hursta przyjmuje wartość większą niż powyższa, w danym szeregu można zaobserwować zjawisko długiej pamięci. Im większa wartość współczynnika (bliższa liczbie 1), tym efekt ten jest bardziej wyraźny. Są to tak zwane serie trwałe. Serwer o największym współczynniku powinien być najlepszy z punktu widzenia jakości wykonywanych predykcji. Dodatkowo, przy założeniach, że ruch w sieci jest faktycznie samopodobny, wyniki predykcji wydajności tego serwera powinny dać lepsze pojęcie o zastosowaniu technik eksploracji danych w kontekście całego Internetu.

Współczynnik Hursta został wykorzystany we wcześniejszych badaniach na Politechnice Wrocławskiej [2]. Wykładownik ten został oszacowany na podstawie przebiegów przepustowości i RTT. Ostatecznie wybrano serwer, na którym obie te wartości były na poziomie  $H = 0,63$ .

Podczas wykonywanych szacowań zauważono, że wartość RTT jest słabo zmienna w czasie w obrębie jednego serwera zdalnego. Przykładowy przebieg pomiarów RTT dla wybranego później kierunku, zaprezentowano na Rysunku 8.1. Podobne charakterystyki (niewielka zmienność) zaobserwowano zarówno na innych agentach pomiarowych, jak również pozostałych badanych serwerach zdalnych.



Rysunek 8.1 Przebiegi RTT dla serwera #47 [Koral]

Na podstawie otrzymanych wyników można wyciągnąć wniosek, że charakterystyka czasu odpowiedzi serwerów ustabilizowała się, w porównaniu do badań wykonywanych wcześniej na serwerze Koral. Być może przyczyną takich przebiegów jest wybór oficjalnych, wydajnych serwerów dystrybucji systemu Debian.

W przypadku otrzymanych danych, charakterystyki przebiegów RTT nie powinny być więc brane pod uwagę przy określaniu współczynnika Hursta. Ich mała zmienność może wprowadzić jedynie zniekształcenie wartości szacowanej przez używane później algorytmy.

Na potrzeby wyliczenia wykładownika Hursta stworzono skrypt w środowisku Matlab. Istotą jego działania była estymacja tego parametru na podstawie przebiegów przepustowości dla

każdego z plików, znajdujących się na wybranych serwerach zdalnych. Następnie wyciągana była średnia wartość dla każdego z hostów, która była podstawą podejmowanego wyboru dwóch serwerów.

Nim opisany zostanie sam etap selekcji, warto zaznaczyć, że w badaniach korzystano z różnych metod estymacji parametru Hursta. Znajdowały się one w serwisie *MathWorks File Exchange*, udostępnione przez autorów do użytku publicznego:

- *Analiza R/S* [61]
- *Ogólny wykładek Hursta* [62]

Obie te metody napisane zostały na podstawie istniejących algorytmów szacowania współczynnika Hursta. Niestety, ani jedna, ani druga funkcja, nie działały prawidłowo dla wykorzystywanych danych pomiarowych. Pierwsza z metod bardzo często wyprowadzała wartości powyżej 1, co jest niezgodne z teorią. W drugiej zaś, oszacowane współczynniki były w większości bliskie wartości 0. Dopiero serwery, na których zauważono zjawisko długiej pamięci, posiadały wartość większą od 0,5. Takie działanie programu wskazywało jednak, że należy znaleźć inną metodę wyliczeń.

Na powyższym serwisie znaleziono również funkcję, która została użyta do ostatecznych estymacji parametru Hursta [63]. Wyniki działania tej metody będą przedstawiane niżej. Nim zastosowano ją na faktycznych danych pomiarowych, przetestowano jej działanie na przebiegach teoretycznych, m.in. na ciągu losowych wartości – wyniki działania były zgodne z założeniami teoretycznymi i znajdowały się na poziomie  $H = 0,5$ .

Należy pamiętać, że parametr  $H$  jest estymowany. Prawdopodobnie należałoby posiadać własną procedurę szacowania, w której można byłoby dobrać najlepsze parametry dla badanych pomiarów.

Na podstawie wykonanych badań dla każdego z agentów otrzymano wyniki dotyczące wyboru serwerów. Serwerem zdalnym, cechującym się największą wartością współczynnika Hursta był serwer #47 (Chicago, USA). Z drugiej strony, serwerem o najbardziej losowym działaniu okazał się serwer #9 (Zagrzeb, Chorwacja).

Jednakże, wyników tych nie przyjęto na podstawie pojedynczego szacowania parametru Hursta. Ponieważ powinien on przedstawiać samopodobieństwo sieci, estymacje tego współczynnika powinny znajdować się na podobnym poziomie, niezależnie od przyjętego okresu pomiarowego. W kolejnych tabelach (Tabela 8.1 i 8.2) zostały przedstawione zmiany tego parametru dla wybranych serwerów, z zastosowaniem rozszerzającego się okna uwzględnianych pomiarów.

Uzyskane wyniki wskazują, że parametr ten jest utrzymywany na podobnej wartości, niezależnie od przyjętego okna przebiegu. Najmniej stabilną wartość współczynnika uzyskiwał agent pomiarowy w Poznaniu. Może być to spowodowane faktem skoku wartości na początku pomiarów, który będzie można zauważać w badanych przebiegach dla tego agenta.

W przypadku agenta we Wrocławskim punkcie sieci PIONIER, można zauważać bardzo duży współczynnik tej wartości dla okresu jednego tygodnia. Ze względu na zauważone wcześniej (prawdopodobne) sterowanie ruchem, przebieg przepustowości w przypadku WCSS zmienia się ostro, wręcz schodkowo. Zauważono, że takie zmiany mają znaczny wpływ na wyniki uzyskiwane przy pomocy funkcji estymujących parametr Hursta.

Wartość współczynnika Hursta dla pełnego przebiegu agentów pomiarowych znajduje się w ostatniej kolumnie.

		Wielkość okna [tyg.]			
Serwer 47		1	2	6	8
Koral		0.8340	0.8149	0.9481	0.9396
WCSS		0.9990	0.9427	0.9086	-
PCSS		0.9044	0.7987	-	-

Tabela 8.1 Zmiana współczynnika Hursta dla okna rozszerzającego się [Serwer #47]

		Wielkość okna [tyg.]			
Serwer 9		1	2	6	8
Koral		0.5597	0.5527	0.5650	0.5565
WCSS		0.4869	0.4883	0.5398	-
PCSS		0.3730	0.4567	-	-

Tabela 8.2 Zmiana współczynnika Hursta dla okna rozszerzającego się [Serwer #9]

Na podstawie wykonanych badań nie było więc podstaw do odrzucenia dwóch wybranych serwerów. Sprawdzono, czy ich przebiegi faktycznie przypominają przebiegi samopodobne. Poniżej przedstawione zostaną przebiegi dla pliku o wielkości 3 MB dla każdego z agentów, w kontekście dwóch wybranych serwerów zdalnych.

Bardzo dobry przykład ruchu samopodobnego można zauważać na serwerze Koral, przybliżenie go w dowolnym z miejsc powinno dać ruch o takiej samej charakterystyce. Charakterystyczny histogram przedstawiający długogonowość pomiarów, znajduje się w sekcji Załączniki.

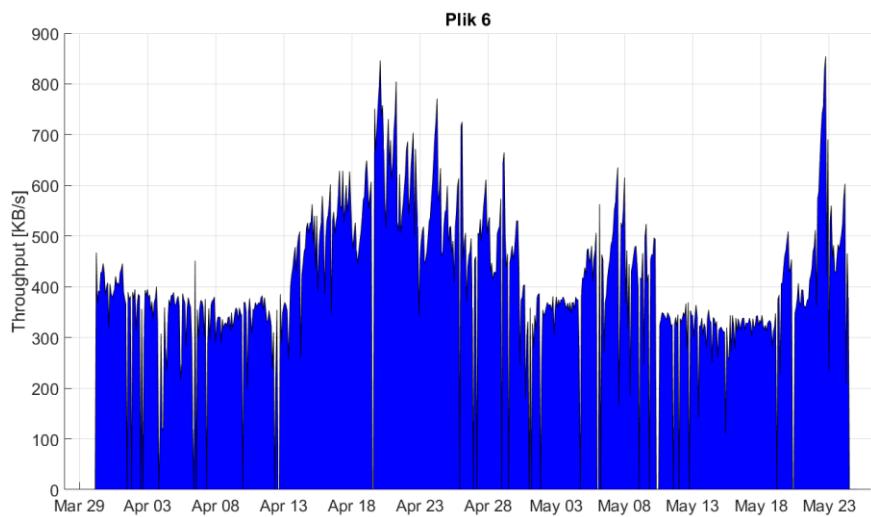
Niestety, podobnego zachowania nie można potwierdzić na podstawie przebiegów dla agenta pomiarowego w WCSS. Zauważone zostały na nim nagłe skoki przepustowości, które mogą mieć powiązanie ze sterowanym ruchem na maszynach wirtualnych. Wysokie wyniki parametru Hursta dla tego agenta sugerowały, że mogą być one podatne na schodkową postać badanych pomiarów.

W tym przypadku należało znaleźć jeszcze inny sposób na szacowanie współczynnika. Zauważono, że m.in. w okresie od 15 do 21 kwietnia przebiegi są bardzo stabilne. Okres ten byłby interesujący dla estymacji wartości parametru Hursta i sprawdzenia, czy jest ona na równie wysokim poziomie, co przy uwzględnianiu funkcji schodkowej.

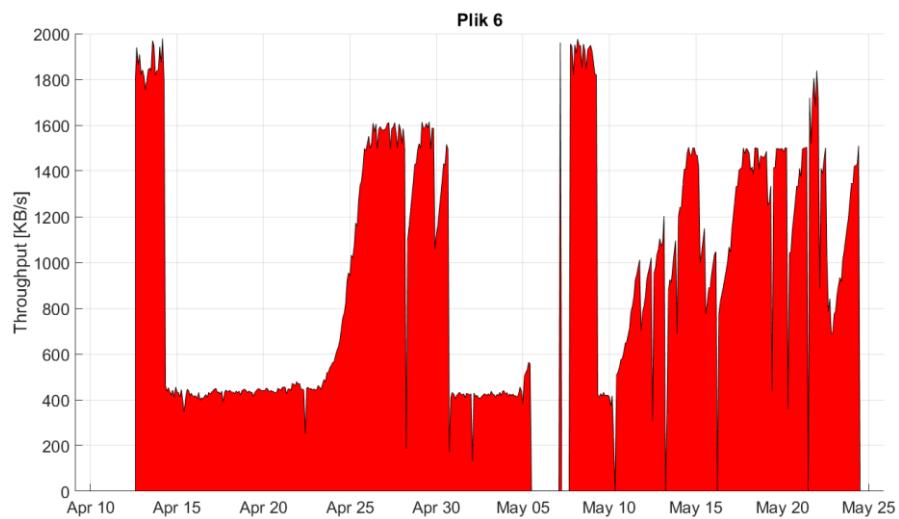
Dla sprawdzenia tej własności postanowiono napisać skrypt, który będzie wykonywał szacowanie parametru Hursta dla przesuwającego się okna pomiarowego. Przyjmując okno wielkości 5 dni (60 pomiarów) i przesuwając je w czasie, otrzymano wykresy zmian parametru Hursta w kolejnych jego położeniach. Zostały one przedstawione na Rysunku 8.8. Można zauważać, że w przypadku WCSS, współczynnik ten ma mniejszą wartość, gdy przesuwające się okno posiada pomiary bez skoków. Jednakże, nie wiąże się to z mocnym spadkiem wartości współczynnika Hursta (mocny spadek widoczny w środku przebiegu wynika z brakujących pomiarów).

Następne wykresy (Rysunki 8.9 i 8.10) przedstawiają wyniki dla większych rozmiarów okna, kolejno 10 i 20 dni. Można zauważać, że im więcej uwzględnianych dni pomiarowych, tym bardziej płaska postać otrzymanej funkcji zmiany współczynnika Hursta.

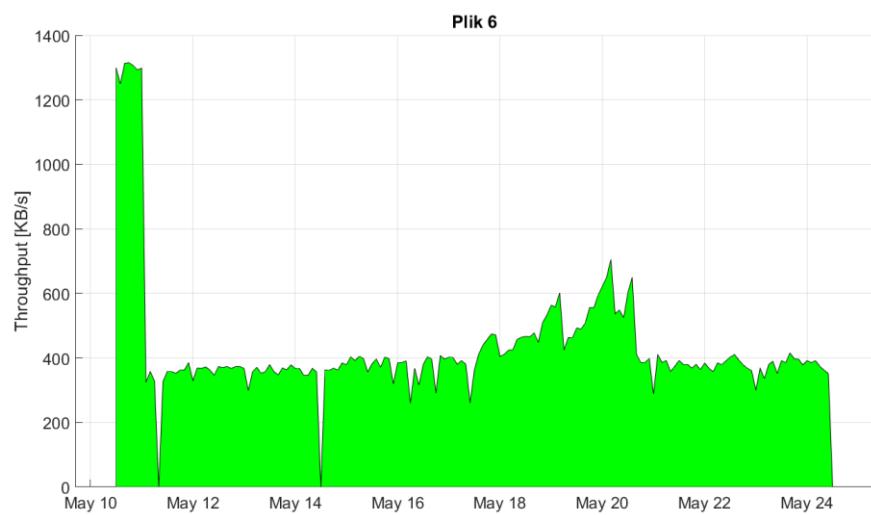
Przebiegi dla serwera #9 przypominają ciąg losowy. Potwierdzają to zarówno histogramy dla serwera, jak również utrzymująca się na poziomie  $H = 0,5$  wartość współczynnika Hursta dla okna przesuwnego (dostępna w Załącznikach).



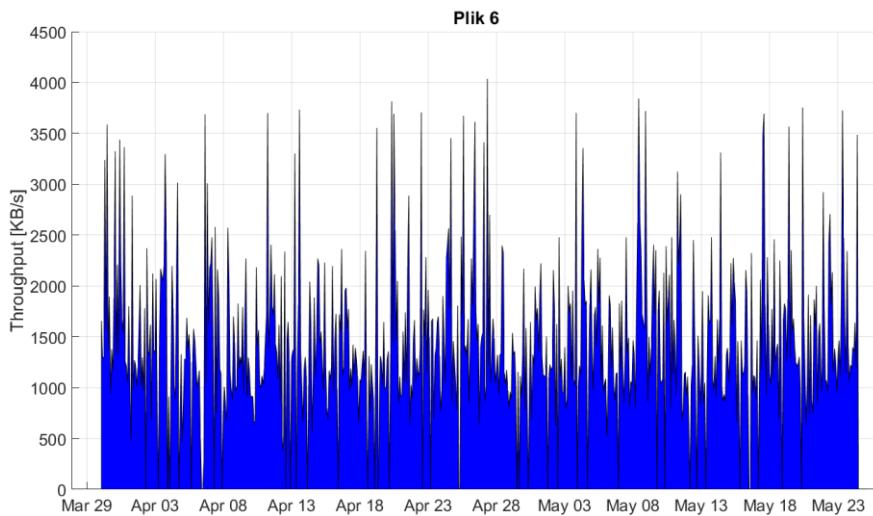
Rysunek 8.2 Przebiegi przepustowości dla pliku 6MB [Koral, serwer #47]



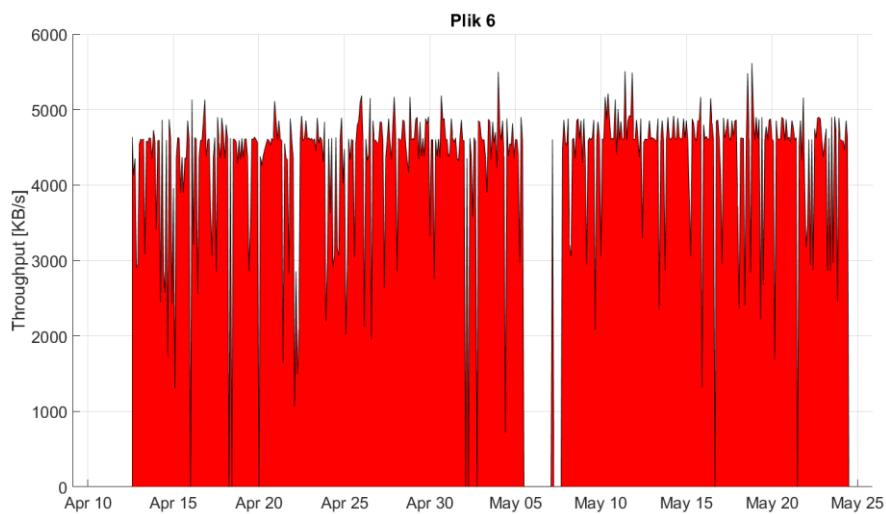
Rysunek 8.3 Przebiegi przepustowości dla pliku 6MB [WCSS, serwer #47]



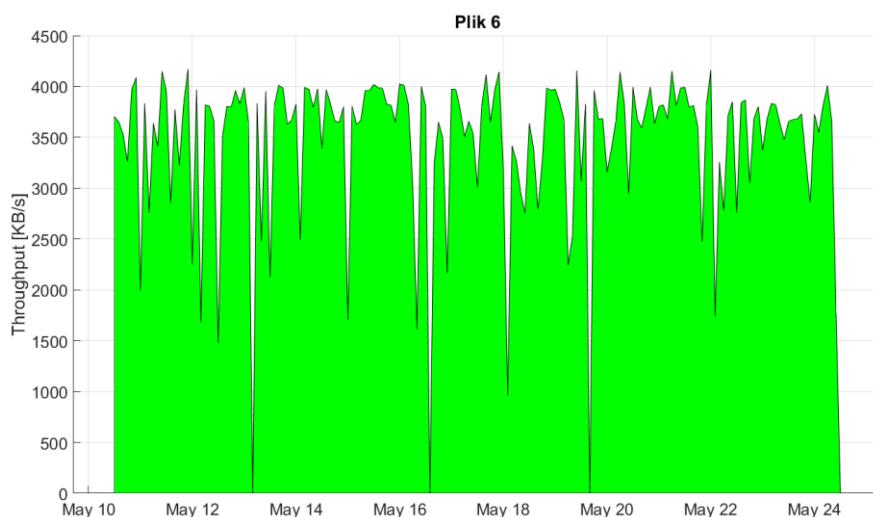
Rysunek 8.4 Przebiegi przepustowości dla pliku 6MB [PCSS, serwer #47]



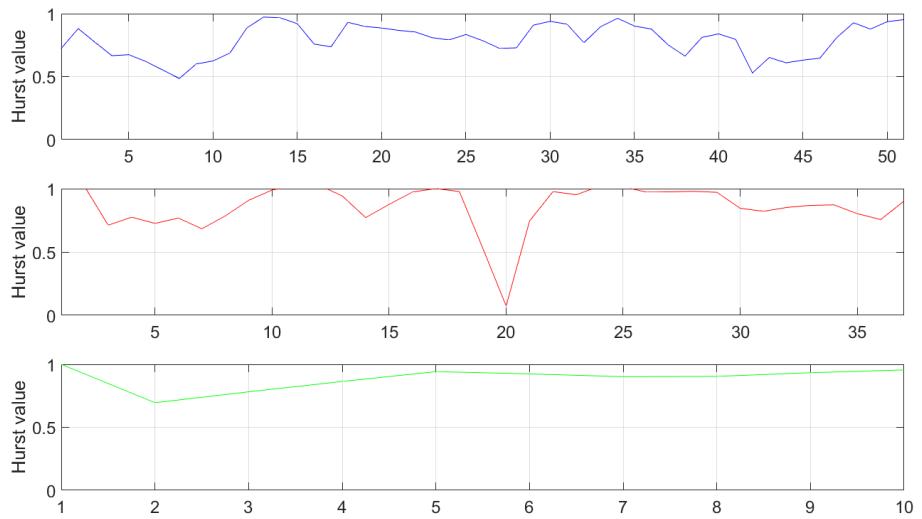
Rysunek 8.5 Przebiegi przepustowości dla pliku 6MB [Koral, serwer #9]



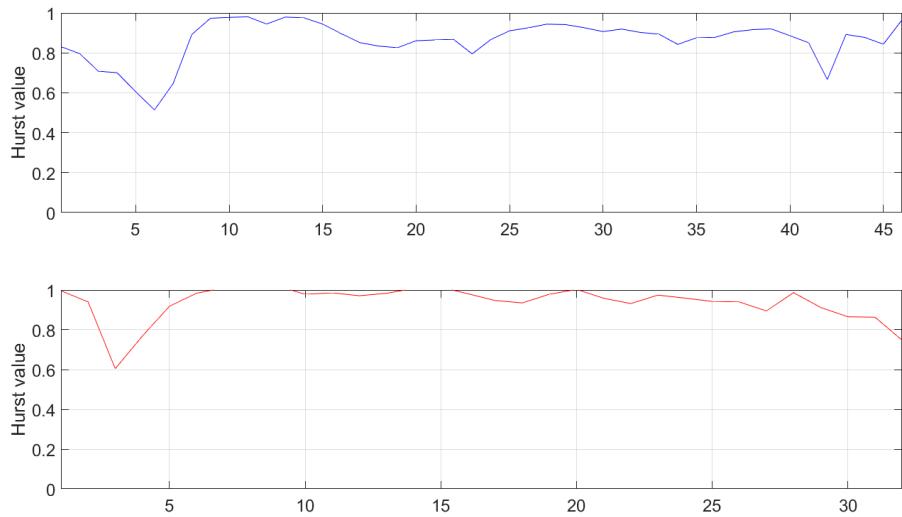
Rysunek 8.6 Przebiegi przepustowości dla pliku 6MB [WCSS, serwer #9]



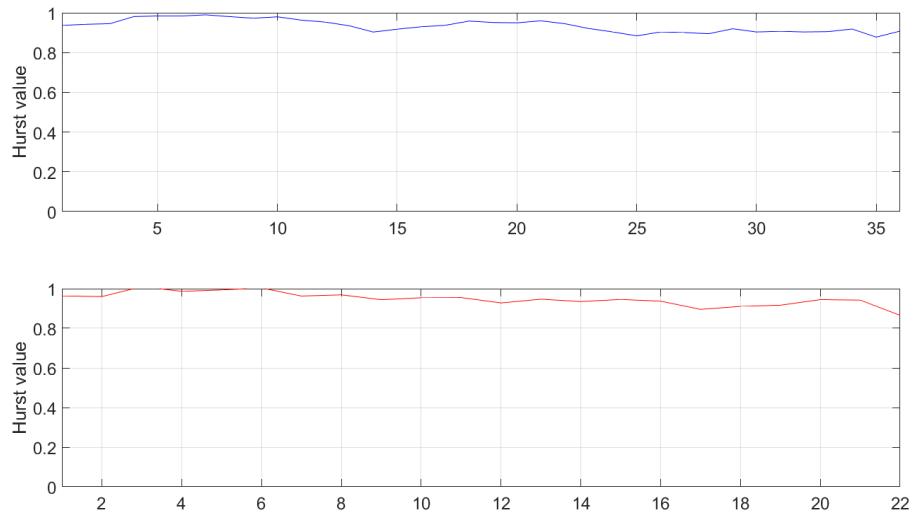
Rysunek 8.7 Przebiegi przepustowości dla pliku 6MB [PCSS, serwer #9]



Rysunek 8.8 Wartość współczynnika Hursta dla okna przesuwnego (5) [serwer #47]



Rysunek 8.9 Wartość współczynnika Hursta dla okna przesuwnego (10) [serwer #47]



Rysunek 8.10 Wartość współczynnika Hursta dla okna przesuwnego (20) [serwer #47]

### **8.3 Wykonanie kolejnych etapów Data Mining**

W tym podrozdziale przedstawiono czynności przyporządkowane do kolejnych etapów Data Mining, wykonywane dla przeprowadzonych pomiarów. Opisane zostaną tutaj poszczególne kroki, które należało wykonać w celu przygotowania danych do ich przyszłej eksploracji.

#### **8.3.1 Selekция danych**

Pierwszą czynnością była odpowiednia selekcja zebranych danych. Powinny być to parametry, które będą przydatne z punktu widzenia wykonywanych pomiarów. Poniżej przytoczone zostaną wartości, zebrane na podstawie zarówno systemu pomiarowego MWING, jak również w wyniku przeprowadzonych wcześniej badań:

##### Wartości systemu MWING:

- Dane porządkowe:
  - **mid** – identyfikator pomiaru,
  - **agentId** – identyfikator agenta pomiarowego,
  - **slotId** – indywidualny id pobieranej zawartości,
  - **mStart** – czas rozpoczęcia pomiaru,
  - **sStart** – czas rozpoczęcia serii pomiarowej.
- Dane pobierania pliku:
  - **MOR** – łączny czas wykonanej transakcji, zmienna sprawdzająca poprawność uzyskanych wyników,
  - **DNS** – czas na uzyskiwanie adresu IP hosta z serwerów DNS,
  - **D2S** – czas między fazą DNS a rozpoczęciem transakcji TCP,
  - **CONN** – czas ustalania połączenia z hostem (RTT),
  - **A2G** – czas między ustaleniem połączenia a wysłaniem żądania pobrania zawartości,
  - **FIRST** – czas potrzebny na pobranie pierwszego pakietu danych,
  - **LEFT** – czas faktycznego transferu pliku,
  - **SIZE** – wielkość pobranego pliku,
  - **URL** – adres, z którego pobrany został plik.

##### Dane serwerów zebrane dodatkowo na potrzeby pracy:

- **DISTANCE** - Dystans geograficzny od Wrocławia/Poznania,
- **ASRANK** - Ranga AS na podstawie serwisu CAIDA (zależna od ilości klientów w AS),
- **ASTRANSIT** - Tranzyt AS,
- **ASTYPE** - Typ AS (Transit, Content, Enterprise),
- **TIER** - Poziom sieci ISP (1, 2, 3).

Na etapie selekcji danych zdecydowano o odrzuceniu identyfikatora pomiaru (mid), który ma znaczenie wyłącznie dla potrzeb odróżniania obserwacji w systemie pomiarowym. Z poziomu wykonywanych analiz niepotrzebne były również zmienne D2S oraz A2G, ponieważ ich czasy związane były bezpośrednio z systemem pomiarowym, a nie działaniem na poziomie sieci. Z punktu widzenia badań nie przydał się adres URL.

Warto zaznaczyć, że na podstawie przeprowadzonych badań można było uwzględnić więcej wartości, m.in. typ ISP (akademicki lub przemysłowy), jednak parametry te zostały odrzucone już na etapach zbierania danych, a działanie to było odpowiednio argumentowane.

### 8.3.2 Czyszczenie danych

Kolejnym etapem, kluczowym dla wykonywanych badań, było oczyszczenie danych z błędnych wartości, jak również wszelkich odchyleń. Działanie to musiało być wy tłumaczone z perspektywy wiedzy eksperckiej – potwierdzone odpowiednią argumentacją. Każdy z przytoczonych kroków filtracji będzie przedstawiony w formie wypunktowanych warunków.

Pierwszymi obserwacjami usuniętymi z dalszej analizy były te, które posiadały nieprawidłowe wartości dotyczące nawiązanego połączenia TCP. Były to wartości, które były mniejsze lub równe zero.

- MOR > 0
- DNS > 0
- CONN > 0
- FIRST > 0
- LEFT > 0

W kroku kolejnym należało sprawdzić, czy parametr MOR, którego założeniem było zsumowanie wartości połączenia TCP, został wyliczony poprawnie.

- MOR = DNS + D2S + CONN + A2G + FIRST + LEFT

Następnie odfiltrowano zbyt wysokie wartości RTT. Uznaje się, że wartości, których czas odpowiedzi serwera wynosi ponad 3000 ms, są wartościami błędnymi. Ponieważ dane były zbierane w  $\mu$ s, należało odpowiednio je przemnożyć.

- CONN < 3 000 000

Kolejnym argumentem podlegającym analizie z perspektywy czyszczenia danych, była zmierzona wielkość pobranego pliku. We wcześniejszej części pracy wykonane zostały pomiary, mające na celu sprawdzenie binarnej tożsamości tych samych plików, znajdujących się na różnych serwerach. Pomimo pozytywnych wyników tych badań, okazało się, że pliki pobrane przez narzędzie pomiarowe MWING mają różnice się od siebie wartości. W obrębie jednego serwera wartości te były stałe. Należało więc znaleźć granicę wielkości pliku, w której stwierdzano, że jest on niepełny. Dla każdego z plików została określona wartość tej granicy, w odpowiedniej ilości bajtów.

- PLIK 1 – SIZE > 500 000
- PLIK 2 – SIZE > 1 000 000
- PLIK 3 – SIZE > 1 500 000
- PLIK 4 – SIZE > 2 100 000
- PLIK 5 – SIZE > 2 600 000
- PLIK 6 – SIZE > 3 100 000

Po wykonaniu przedstawionych powyżej filtracji, dla każdego z agentów pomiarowych pozostał poniższy procent danych:

- Koral – 93,97%
- WCSS – 98,48%
- PCSS – 97,91%

Można zauważyć, że serwer Koral cechował się największą ilością niepoprawnie wykonanych pomiarów. Fakt, że każdy z agentów badał serwery w jednakowym czasie sugeruje, że winą błędnych pomiarów stoi po stronie wewnętrznej sieci Politechniki

Wrocławskiej. Bardzo prawdopodobne, że takie wyniki pomiarów związane były z obciążeniem sieci, które można było zaobserwować w badaniach stabilności pomiarów. Serwer ten jako jedyny posiadał serie pomiarowe, które czasem na siebie nachodziły.

Ostatnim etapem czyszczenia danych była eliminacja serwerów z wysokim niepowodzeniem transakcji. Działanie to miało na celu odfiltrować próbki, które mogły mieć negatywny wpływ na tworzenie modeli w problemie predykcji wydajności nowego serwera.

Dla każdego z agentów pomiarowych ustalono progi, po których przekroczeniu serwer powinien zostać odfiltrowany z badań eksploracji danych. Wartości te były zależne od ogólnej liczby odfiltrowanych pomiarów.

- Koral – przynajmniej 10% nieudanych transakcji
- WCSS i PCSS – przynajmniej 5% nieudanych transakcji

Przy takich założeniach odfiltrowane zostały serwery w Japonii, Włochach, Macedonii, Australii, Vanuatu, Hiszpanii, Holandii i Tajlandii. Do usunięcia z listy badanych serwerów wystarczyło przekroczenie założonego progu na jednym z agentów pomiarowych.

Ciekawym był również fakt, że dla agenta WCSS próg ten przekroczył również serwer w Gdańsku, znajdujący się w tej samej sieci PIONIER. Dla dwóch pozostałych punktów pomiarowych nie zauważono znacznej liczby błędnych transakcji dla tego kierunku. Nie chcąc pozbywać się serwera cechującego się największymi wartościami przepustowości, postanowiono zostawić go w badaniach mimo przekroczonego progu.

Etap czyszczenia danych wykonano za pomocą pakietu *IBM SPSS Statistics 24*, w którym przytoczone wyżej filtracje były pierwotnie wykonywane manualnie. Każda użyta opcja generowała odpowiednią składnię, która została wykorzystana do opracowania skryptu, wykonującego w przyszłości filtrowanie w sposób zautomatyzowany, niewymagający działania użytkownika.

Zastanawiano się nad usunięciem wartości odstających i ekstremalnych dla danych charakteryzujących połączenie TCP. Być może działanie to byłoby wskazane, gdyby przypuszczano, że rozkład parametrów ma postać gaussowską. Badania w publikacjach na temat Internetu pokazały jednak, że rozkłady badanego ruchu mają postać tak zwanego długiego ogona. Przyjęcie postaci rozkładu normalnego dla tych danych byłoby założeniem błędny. Przy odcinaniu wartości odstających, bardzo możliwa jest utrata prawidłowych charakterystyk połączenia.

W wykonywanym etapie czyszczenia każda z próbek, która nie wpisywała się w określone reguły, była usuwana. Działanie to jest wytlumaczalne, ponieważ procent błędnych obserwacji jest niski, a operacja usunięcia pozostawia nadal bardzo dużą ilość próbek do analizy. W przypadku, gdyby pomiarów było znacznie mniej, należałoby się zastanowić m.in. nad uśrednianiem wartości brakujących argumentów i usuwać jedynie te pomiary, których poprawa w kontekście prawidłowości byłaby niemożliwa.

### 8.3.3 Transformacja danych

Transformacja danych polegała na zamienieniu wybranych parametrów na użyteczne z perspektywy wykonywanych badań. Poniżej przedstawione zostaną wszystkie zmiany wraz z wyjaśnieniem ich podjęcia. Transformacja została wykonana przy użyciu obu pakietów IBM – SPSS Modeler 18 i SPSS Statistics 24.

Pierwsza transformacja dotyczyła daty wykonywanych pomiarów. Zmienna mStart zapisana była w formacie czasu uniksowego. Jest to notacja, która mierzy liczbę sekund od początku roku 1970. Wiedząc, że 1 stycznia 1970 roku to środa, w łatwy sposób można wyliczyć dzień tygodnia. Przy użyciu funkcji modulo, wyciągnięto również godzinę wykonywanego pomiaru. Można było również wykonać taką samą transformację na minuty, jednak dla prostoty pomiarów, nie zdecydowano się na taki ruch.

- DAY – dzień tygodnia (wartości nominalne 1-7)
- HOUR – godzina rozpoczęcia pomiaru (0-23)

Nastecną transformacją z poziomu skryptu statystycznego, było wyliczenie przepustowości TCP. W obliczeniach wykorzystano zarówno wielkość pobranego pliku, jak również czas transferu. Wzór na przepustowość widoczny jest poniżej. Można zauważyć, że wykorzystywane są w nim kilobajty na sekundę, przy czym kilobajt rozumiany jest w sposób informatyczny ( $1 \text{ KB} = 2^{10} \text{ B} = 1024 \text{ B}$ ).

$$\text{THROUGHPUT} = \frac{\text{SIZE} * 10^6}{\text{LEFT} * 1024} [\text{KB/s}]$$

Ponieważ RTT jest standardowo mierzone w ms, a system pomiarowy mierzył tę wartość w  $\mu\text{s}$ , wykonano transformacje dla przedstawienia jej w dłuższej jednostce czasu.

$$RTT = \frac{CONN}{1000}$$

Na podstawie wartości slotID (indywidualnego identyfikatora połączenia plik-serwer), określony został zarówno serwer pomiarowy, jak również pobierany plik. Działanie to wykonano dla potrzeb łatwiejszego porządkowania badań. Dzięki przyjętej wcześniej notacji (podrozdział 4.5.2), były to zmienne bardzo proste do wyekstrahowania.

Kolejnymi etapami transformacji danych było grupowanie (*binning*) zarówno wartości RTT, jak również przepustowości TCP w odpowiednie przedziały. Grupowania używano wyłącznie, jeżeli predykcja była wykonywana przy pomocy technik klasyfikacji. Podstawowe dwie możliwości grupowania to:

- Dzielenie na przedziały wartości ustalone przez użytkownika,
- Dzielenie programowe na równej wielkości przedziały wartości.

Dla innego typu danych, możliwe może być również ustalanie przedziałów na podstawie wartości odchylenia standardowego od średniej. Jednak, jak zostało to wspomniane wcześniej, ruch w Internecie nie posiada rozkładu normalnego, więc taki podział nie powinien być wykorzystanych dla zebranych obserwacji.

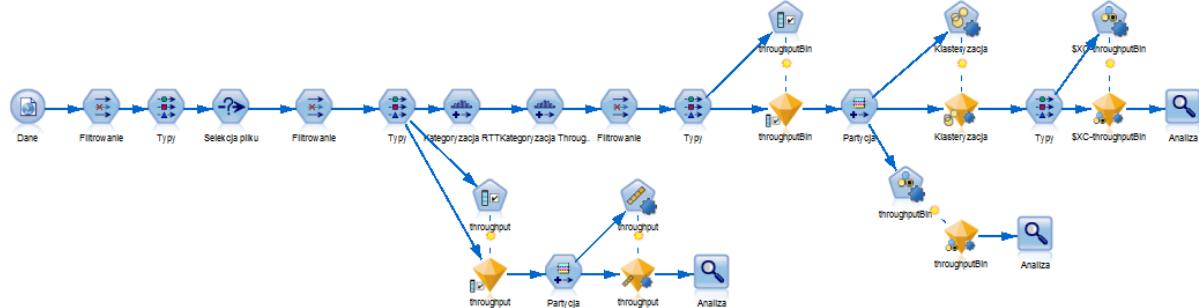
Dokładnie przyjęte podziały zostaną przedstawione w kolejnym podrozdziale, w którym krótko zostanie opisany każdy z węzłów w użytym pakiecie do wykonania eksploracji danych – IBM SPSS Modeler 18.

#### 8.4 Opracowanie skryptu IBM SPSS Modeler

Pierwotnie zakładano, że eksploracja danych zostanie wykonana w środowisku IBM SPSS Statistics. Szybko jednak okazało się, że pomimo takiej możliwości, pakiet ten nie wspiera automatyzacji wykonywanych badań. Rozwiążanie, które ułatwia użytkownikowi wykorzystanie metodologii eksploracji danych, dostępne jest w innym produkcie firmy IBM – SPSS Modeler. Narzędzie to nadaje się do wykorzystania przez użytkownika końcowego, który nie ma obowiązku posiadać wysoko rozwiniętej wiedzy na temat Data Mining.

Oprogramowanie to pozwala na stworzenie strumienia, który wyznacza kolejne działania na danych. Wykonywane są w nim m.in. niektóre etapy Data Mining, które zostały przedstawione w poprzedniej sekcji. Kluczowym założeniem tego strumienia jest predykcja przygotowanych danych i prezentacja w postaci odpowiednich technik graficznych.

Strumień wykonany na potrzeby sprawdzenia jakości predykcji wydajności dla nowego serwera został zaprezentowany na Rysunku 8.11, jako przykład poglądowy. Większa wersja obrazu dostępna jest w sekcji Załączniki.



Rysunek 8.11 Strumień dla pierwszego zadania predykcji [SPSS Modeler]

Strumień ten zawiera kolejne węzły, których działanie wykorzystuje wprowadzone na wejście dane. Poniżej postarano się opisać działanie powyższego skryptu, skupiając się na funkcjonalności poszczególnych węzłów.

Na początku wczytywane są dane otrzymane w wyniku działania jednego z serwerów. Są to obserwacje pozostawione w wyniku wcześniejszego etapu filtracji. Po usunięciu wartości niepotrzebnych z punktu widzenia predykcji, należy określić ich typy i rolę w wykonywanych pomiarach. Okno wykonujące tę funkcjonalność zostało zaprezentowane na Rysunku 8.12.

Zmienna	Poziom pomiaru	Wartości	Braki	Sprawdź	Rola
# mid	Ilościowy	[1.235068...]	Brak		ID rekordu
# dns	Ilościowy	[221.0,356...]	Brak		Dane wej...
# first	Ilościowy	[120544.0,...]	Brak		Dane wej...
# size	Ilościowy	[529216.0,...]	Brak		Dane wej...
# day	Nominalny	1.0,2.0,3.0...	Brak		Dane wej...
# hour	Ilościowy	[0.0,23.0]	Brak		Dane wej...
# rtt	Ilościowy	[153.809,1...]	Brak		Dane wej...
# file	Nominalny	1.0,2.0,3.0...	Brak		Brak
# serverId	Nominalny	1.0,2.0,3.0...	Brak		Dane wej...
# throughput	Ilościowy	[122.0886...]	Brak		Przewidy...
# distance	Ilościowy	[7397.614...]	Brak		Dane wej...
# asRank	Ilościowy	[9449.0,94...]	Brak		Dane wej...
# asTransit	Ilościowy	[0.0,0.0]	Brak		Dane wej...
# asType	Nominalny	1.0	Brak		Dane wej...
# networkTier	Nominalny	3.0	Brak		Dane wej...

Rysunek 8.12 Okno określania typów i roli argumentów [SPSS Modeler]

Jeżeli wykorzystana metodologia zakładała predykcję wartości nominalnej (techniki klasifikacji), zanim przystąpiono do użycia metod eksploracji danych, należało określić odpowiednie przedziały dla wybranych argumentów. W badaniach na grupy zazwyczaj dzielono przepustowość i RTT. Działanie to miało sens w przypadku badania wszystkich serwerów, gdzie obie te wartości cechowały się dużym rozrzutem. W obrębie jednego serwera jednak, przypadki podziału należało rozpatrzyć indywidualnie, ze względu na możliwy mały

rozrzut wartości. Dokładne granice zakładanych przedziałów będą wskazywane przy wynikach eksploracji danych w kolejnym rozdziale.

Przykładowe okna, dzielące wartości przepustowości i RTT dla wszystkich serwerów, znajdują się na Rysunkach 8.13 i 8.14. Dla ułatwienia ich zrozumienia, podziałom można nadać odpowiednie nazwy. W przypadku czterech wyliczonych kategorii przepustowości mogą być to kolejno wartości LOW, MEDIUM, HIGH, VERY HIGH.

Kategoria	Dolna	Góra
1	$\geq 101,20707535$	$< 452,91657715$
2	$\geq 452,91657715$	$< 821,93247861$
3	$\geq 821,93247861$	$< 1775,06990428$
4	$\geq 1775,06990428$	$\leq 11088,81217322$

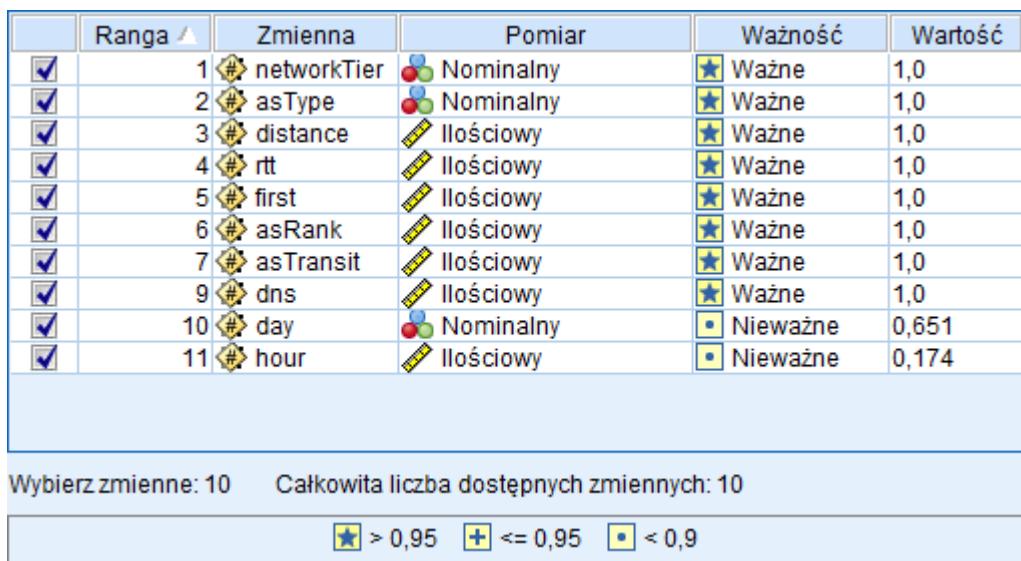
Rysunek 8.13 Grupowanie (*binning*) wartości przepustowości [SPSS Modeler]

Kategoria	Dolna	Góra
1	$\geq 12,834$	$< 33,739$
2	$\geq 33,739$	$< 41,75$
3	$\geq 41,75$	$< 56,791$
4	$\geq 56,791$	$< 100,913$
5	$\geq 100,913$	$< 163,674$
6	$\geq 163,674$	$< 244,485$
7	$\geq 244,485$	$< 296,866$
8	$\geq 296,866$	$< 2372,194$

Rysunek 8.14 Grupowanie (*binning*) wartości RTT [SPSS Modeler]

Nim użyto technik eksploracji danych, wykonywano sprawdzenie mające na celu wylistowanie argumentów, które są najbardziej użyteczne z punktu widzenia wykonywanej analizy. Użycie tego węzła miało charakter poglądowy, w dalszych krokach często wykorzystywano również dane nieważne według zastosowanych algorytmów. Dodatkowym zadaniem tego węzła było wskazanie, które ze zmiennych nie nadają się jako dane wejściowe (na przykład z racji znakomej zmienności).

Węzeł ten wykorzystuje różne techniki do określenia ważności danych wejściowych. Przykładowe wyniki jego działania zaprezentowano na Rysunku 8.15.

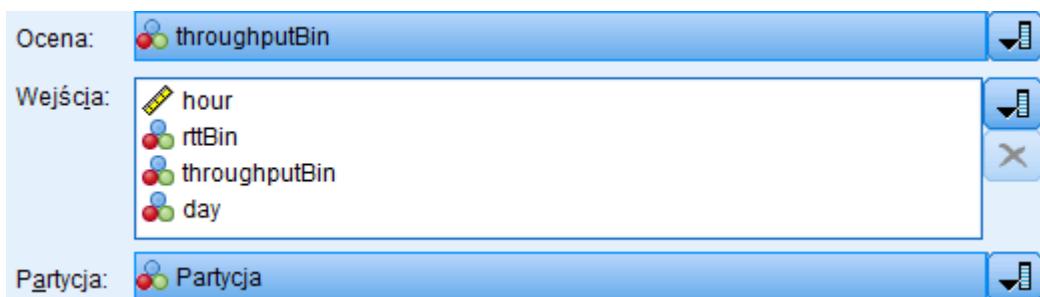


Rysunek 8.15 Określenie użytecznych danych [SPSS Modeler]

Pierwotnie w celu wykrycia użytecznych danych, zakładano użycie metody Principal Component Analysis (PCA) [64], jednak z racji przytoczonej wyżej funkcjonalności w pakiecie IBM SPSS Modeler, zaniechano dalszych analiz w tym kierunku.

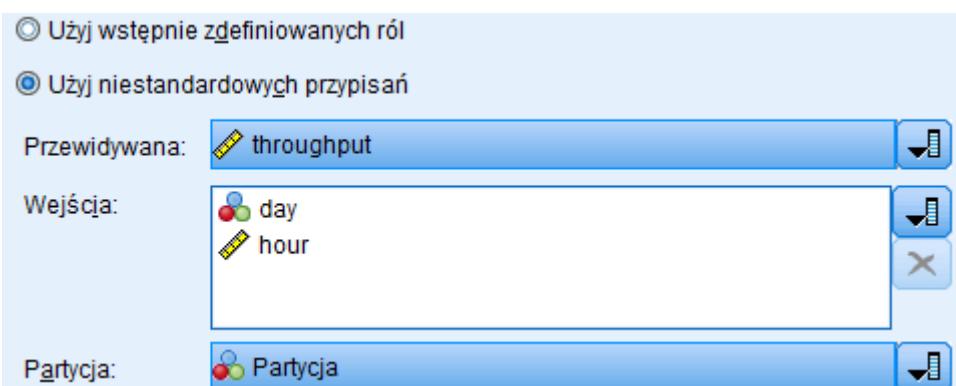
W przypadku wykorzystywania metod klasteryzacji, należało określić, z których zmiennych klastry te będą budowane. Następnie jakość ich budowy była poddawana ocenie na bazie jednego z argumentów (zazwyczaj przepustowości), którego rozrzut w przestrzeni był miarą dopasowania wyników tej techniki.

Przykładowe okno wyboru argumentów do budowy klastrów pokazane jest na Rysunku 8.16. Widać w nim, że klastry tworzone będą z czterech zmiennych: godziny i dnia pomiaru, przedziałów RTT i przepustowości.



Rysunek 8.16 Wybór argumentów do budowy klastrów [SPSS Modeler]

Wyniki predykcji użytych technik klasyfikacji i regresji są uzależnione od używanych przez algorytmy argumentów. Możliwe było użycie wstępnie zdefiniowanych ról (wszystkich argumentów wejściowych), jak również odpowiedni wybór zmiennych. Przykład wyboru argumentów znajduje się na Rysunku 8.17, gdzie dla wybranej techniki używa się wyłącznie dnia tygodnia i godziny w przewidywaniu przepustowości.



Rysunek 8.17 Wybór argumentów w metodach klasyfikacji lub regresji [SPSS Modeler]

Techniki wykorzystywane w klasteryzacji, a także klasyfikacji i regresji przez pakiet IBM SPSS Modeler 18, zostały przytoczone w sekcji dotyczącej pojęcia eksploracji danych (Rozdział 3).

Wyniki działania tych technik zostaną przedstawione w kolejnym rozdziale, w którym wykonana zostanie predykcja dla założonych zadań pomiarowych. Pokazane zostaną tam graficzne reprezentacje z użytego pakietu, które staną się podstawą do wyciągania wniosków na temat przeprowadzonych pomiarów.

## 9. Analiza wybranych wyników Data Mining

Poniżej prezentowane będą wyniki otrzymane na podstawie wyliczeń pakietu IBM SPSS Modeler 18. Szczegółowe rezultaty przedstawione zostaną dla serwera Koral, którego przypadek powinien być najbardziej interesujący z perspektywy użytkowników Politechniki Wrocławskiej. W oparciu o analizę wyników, podejmowane będą kolejne kroki, sprawdzające zmiany predykcji przy różnych podejmowanych założeniach.

Na podstawie działań wykonanych na danych z serwera Koral bazować będą badania dla pozostałych agentów pomiarowych. W tym przypadku, rysunki obrazujące kolejne kroki eksploracji danych, zostaną przedstawione jedynie w ciekawszych obserwacjach. Część obrazów, na których bazowane będą analizy znajdujące się w tych podrozdziałach, znajduje się w sekcji Załączniki.

Każdy z algorytmów eksploracji danych korzysta z dwóch zbiorów danych. Pierwszy z nich, zwany zbiorem uczącym, służy do budowania modeli klasyfikacji i regresji. Drugi zbiór (testowy), służy do sprawdzania jakości zbudowanych modeli. To właśnie na tej podstawie oceniana jest jakość predykcji. Dla wszystkich wykonanych badań przyjęto stosunek zbiorów uczących i testowych 70:30.

W badaniach skupiono się na największej założonej wielkości pliku (3 MB). Decyzja ta podyktowana była faktem, że jest to rozmiar zbliżony do obecnej średniej wielkości strony Web [43]. Sprawdzenie wszystkich wielkości wiążałoby się ze zbyt szerokim obszarem badań.

### 9.1 Agent Koral

Agent ten był wykorzystywany we wcześniejszych badaniach z użyciem metod eksploracji danych, wykonywanych wewnętrznej sieci Politechniki Wrocławskiej. W porównaniu do dwóch pozostałych punktów pomiarowych, jego działanie było najbardziej podatne na zakłócenia. Był to także punkt, w którym występowało przeciążenie sieci w kilku odcinkach czasu.

Pierwszym zadaniem predykcji była ocena wydajności wielu serwerów pomiarowych, na podstawie której użytkownik może wybrać ten, który cechuje się największymi możliwościami w zakresie przepustowości. We wcześniejszym rozdziale zadanie to określono jako *predykcję wydajności nowego serwera*.

Na początku sprawdzona została jakość budowanych modeli, używających technik regresji. Nimi jednak użyto związanych z nimi algorytmów, sprawdzono, które z argumentów mogą mieć największy wpływ na wartość przepustowości TCP (na podstawie korelacji Pearsona).

Ranga	Zmienna	Pomiar	Ważność	Wartość
1	networkTier	Nominalny	Ważne	1,0
2	asType	Nominalny	Ważne	1,0
3	distance	Ilościowy	Ważne	1,0
4	rtt	Ilościowy	Ważne	1,0
5	asRank	Ilościowy	Ważne	1,0
6	first	Ilościowy	Ważne	1,0
7	asTransit	Ilościowy	Ważne	1,0
9	dns	Ilościowy	Ważne	1,0
10	day	Nominalny	Nieważne	0,655
11	hour	Ilościowy	Nieważne	0,366

Rysunek 9.1 Ważność algorytmów bez grupowania [Koral, wszystkie serwery]

Na Rysunku 9.1 zauważać można, że zarówno dzień tygodnia, jak również pora dnia, określone zostały jako mało ważne w predykcji przepustowości. Wydaje się to logiczne, ponieważ każdy z hostów zdalnych działa w innych strefach czasowych.

Po użyciu algorytmów regresji, wybrane zostały trzy najlepsze uzyskane modele. Dwa z nich bazowały na implementacji drzew decyzyjnych (CHAID i C&RT), podczas gdy jeden używał rozwiązań w postaci sieci neuronowej. Wyniki zostały przedstawione na Rysunku 9.2. Korelacja danych znajdowała się na wysokim poziomie, około 0,85. Błąd względny wynosił 25% dla najlepszego modelu. IBM sugeruje, że dopiero wartość tego błędu na poziomie 1 wskazuje, że stworzone predyktory są bezużyteczne.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		CHAID 1	< 1	0,865	8	0,252
<input checked="" type="checkbox"/>		C&RT 1	< 1	0,864	8	0,254
<input checked="" type="checkbox"/>		Sieci neuronowe 1	< 1	0,841	8	0,294

Rysunek 9.2 Wyniki technik regresji dla wszystkich argumentów [Koral, wszystkie serwery]

Pomimo zadowalających wyników predykcji uznano, że dwa argumenty wejściowe – wyszukiwane DNS i czas odebrania pierwszego pakietu (FIRST) są bezpośrednio związane z wykonaniem połączenia TCP. Wartości te są możliwe, lecz trudne do szybkiego ich otrzymania, co może być problematyczne z punktu widzenia biznesowego. Dlatego ponownie przetworzono dane, bez przytoczonych parametrów. Na Rysunku 9.3 widać, że korelacja dla modeli nie uległa zmianie, a działanie to zmniejszyło nawet błąd względny.

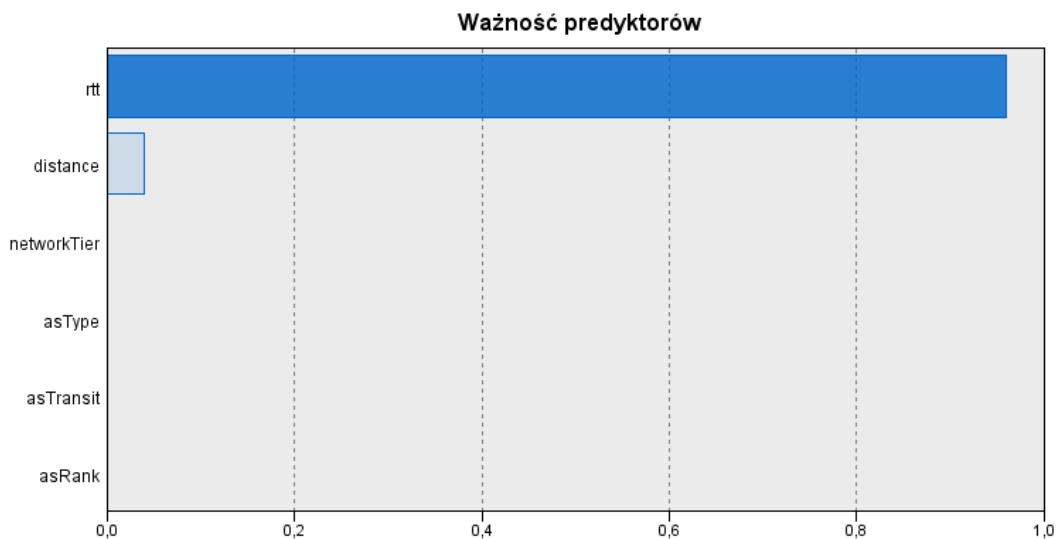
Zdecydowano, że użycie wartości DNS i FIRST będzie wykonywane wyłącznie, gdy wykonywane predykcje będą cechować się niską jakością. Taka sytuacja pojawi się przy badaniu serwerów szczególnych. Dla badań wszystkich serwerów, wartości te nie zostaną ponownie użyte.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		CHAID 1	< 1	0,874	6	0,237
<input checked="" type="checkbox"/>		C&RT 1	< 1	0,859	6	0,263
<input checked="" type="checkbox"/>		Sieci neuronowe 1	< 1	0,845	6	0,286

Rysunek 9.3 Wyniki technik regresji dla wybranych argumentów [Koral, wszystkie serwery]

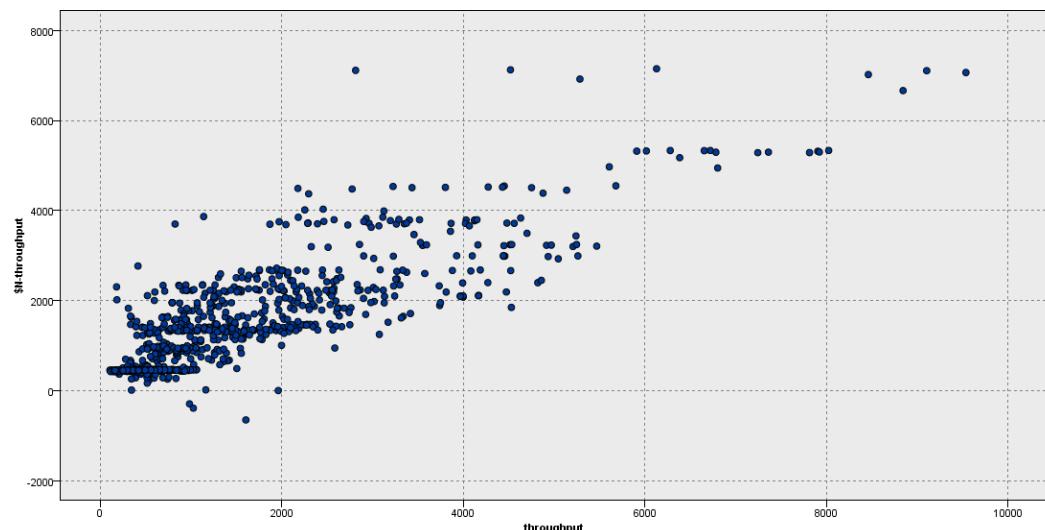
Dla stworzonych trzech modeli, sprawdzono ważność predyktorów (argumentów wejściowych). Ważność ta, przedstawiona na Rysunku 9.4 pokazuje, że najsilniejszym z nich jest czas odpowiedzi serwera (RTT). Potwierdza to, że można wykorzystać wykonaną wcześniej aproksymację dla wartości median RTT i przepustowości badanych serwerów.

Wartości inne niż RTT i dystans geograficzny, również uczestniczą w tworzeniu modeli (bez względu na wyniki pokazywane przez wykres), ich znaczenie jest jednak na tyle małe, że przy sprawdzaniu wykresu ważności dla trzech modeli, wygląda ono na zerowe.



Rysunek 9.4 Ważność predyktorów dla otrzymanych modeli regresji [Koral, wszystkie serwery]

Przykładowy wykres badania jakości predykcji zbioru testowego, otrzymany za pomocą sieci neuronowych, został przedstawiony poniżej na Rysunku 9.5. Znajdują się na nim wartości właściwe przepustowości (*throughput*) i prognozowane (*\$N-throughput*).



Rysunek 9.5 Wyniki regresji dla sieci neuronowej [Koral, wszystkie serwery]

Techniki regresyjne posiadają mniej założeń niż metody klasyfikacyjne, które zostaną przedstawione dalej. W przypadku klasyfikacji, należało przekształcić numeryczną wartość przepustowości, poprzez określenie jej odpowiednich przedziałów. Czynność przekształcenia została zautomatyzowana przez jeden z węzłów pakietu SPSS Modeler. Założono, że przepustowość należy podzielić na 4 wielkości, które można będzie opisać w sposób następujący: NISKA, ŚREDNIA, WYSOKA, BARDZO WYSOKA.

Kategoria	Dolna	Góra
1	$\geq 101,20707535$	$< 452,91657715$
2	$\geq 452,91657715$	$< 821,93247861$
3	$\geq 821,93247861$	$< 1775,06990428$
4	$\geq 1775,06990428$	$\leq 11088,81217322$

Rysunek 9.6 Grupowanie przepustowości TCP [Koral, wszystkie serwery]

Określone w ten sposób grupy miały być równej liczności. Dzięki temu założeniu, znacznie łatwiejsze miało być określenie jakości wykonywanych predykcji. W przypadku czterech równej wielkości grup, losowy wybór serwera (czyli taki, jaki użytkownik wykonuje bez żadnej wcześniejszej wiedzy) to predykcja na poziomie 25%. Prognoza powyżej tej wartości jest uznawana za lepszą, lecz bardzo często niewystarczającą. Każdy z wyników w postaci zbudowanych modeli powinien być indywidualnie oceniany pod względem otrzymanej jakości.

Grupowanie (*bining*) wykonano również dla wartości RTT. Uznano, że stworzone zostanie 8 równych przedziałów. Wyniki tego podziału wydają się logiczne i zostały przedstawione na Rysunku 9.7.

Kategoria	Dolna	Górna
1	$\geq 12,834$	$< 33,739$
2	$\geq 33,739$	$< 41,75$
3	$\geq 41,75$	$< 56,791$
4	$\geq 56,791$	$< 100,913$
5	$\geq 100,913$	$< 163,674$
6	$\geq 163,674$	$< 244,485$
7	$\geq 244,485$	$< 296,866$
8	$\geq 296,866$	$< 2372,194$

Rysunek 9.7 Grupowanie RTT [Koral, wszystkie serwery]

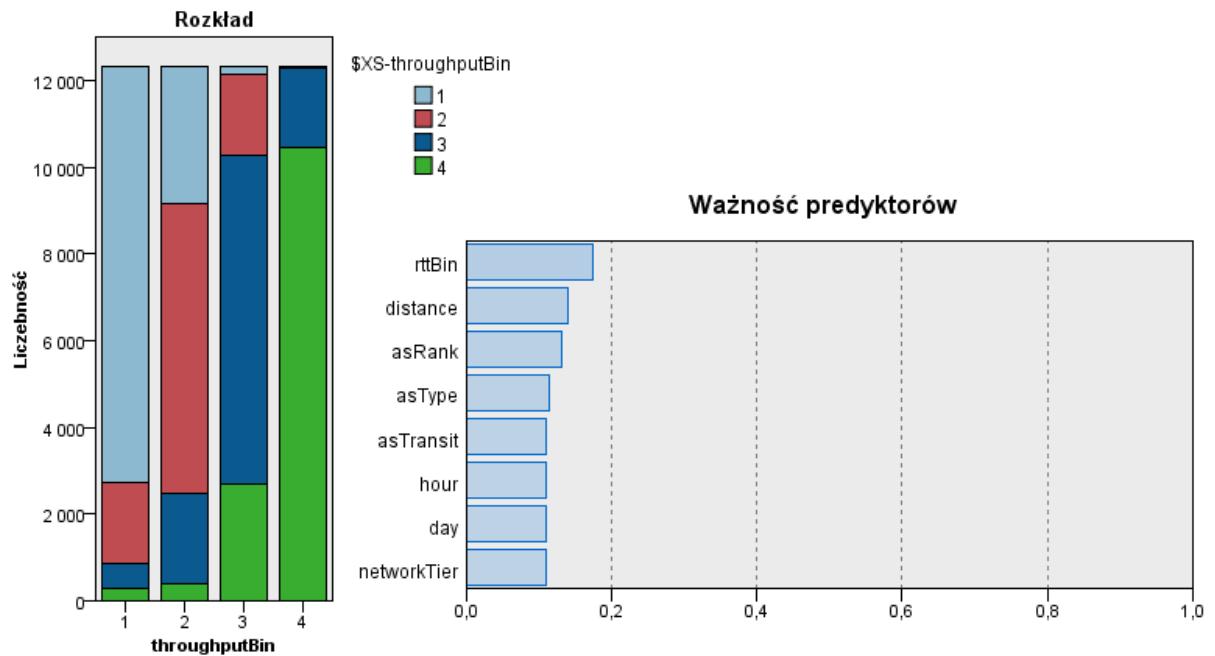
Dla tak pogrupowanych wartości, ponownie oceniono ważność poszczególnych predyktorów. W tym przypadku program zalecał również przyjęcie dnia tygodnia i godziny pomiaru (wartości równe bądź bliskie 1), czyli inaczej niż przy użyciu wartości numerycznych. Z racji wcześniej przyjętych założeń, odrzucających porę i dzień tygodnia, nie zdecydowano się jednak na użycie tych predyktorów. Wyniki trzech najlepiej prognozujących technik klasyfikacji zostały zaprezentowane na Rysunku 9.8.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5 1	53	67,786	8
<input checked="" type="checkbox"/>		CHAID 1	53	67,631	6
<input checked="" type="checkbox"/>		Sieci neuronowe 1	53	66,171	8

Rysunek 9.8 Wyniki predykcji przedziałów przepustowości [Koral, Wszystkie serwery]

Wyniki rzędu 67% poprawnych predykcji są wynikiem dobrym. W porównaniu z wyborem losowym, jest to wynik prawie trzykrotnie lepszy. Najlepszym dopasowaniem cechuje się algorytm C5, czyli kolejne drzewo decyzyjne używane w badaniach. Bardzo zbliżoną wartością poprawnych predykcji charakteryzuje się ponownie CHAID i sieci neuronowe. Ogromnym problemem związanym z użyciem wszystkich predyktorów (wyłączając FIRST i DNS) jest czas budowy modelu. Jak widać na załączonym rysunku, czas konstruowania tych modeli wynosił łącznie 53 minuty (należy w to wliczyć czas tworzenia 7-miu innych modeli, których wyniki predykcji były gorsze od przedstawionych).

Przestrzeń pomiarowa była zbyt duża. W przypadku, gdy predykcje powinny być wykonywane Online, tak długi czas przetwarzania jest niemożliwy do przyjęcia. Z tego powodu sprawdzono, które z argumentów mają największe znaczenie w budowie stworzonych modeli. Wyniki przedstawiono na Rysunku 9.9.



Rysunek 9.9 Ważność predyktorów dla klasyfikacji przedziału przepustowości [Koral, wszystkie serwery]

Na podstawie powyższego wykresu wyciągnięto dwa najważniejsze parametry, które według programu najmocniej wpływają na trzy przedstawione modele. Ostatecznie, dane wejściowe w postaci czasu odpowiedzi serwera (RTT) i dystansu geograficznego, zostały użyte do stworzenia kolejnych modeli, zaprezentowanych na Rysunku 9.10. Zaletą przyjęcia tych parametrów był fakt, że nie wykorzystywały one serwisu CAIDA, którego działanie było stosunkowo wolne. Zebranie informacji ad hoc mogło być więc problematyczne.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5 1	8	63,351	2
<input checked="" type="checkbox"/>		C&RT 1	8	62,045	2
<input checked="" type="checkbox"/>		CHAID 1	8	60,638	2

Rysunek 9.10 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [Koral, Wszystkie serwery]

Wyniki z wykorzystaniem tylko tych dwóch parametrów pogorszyły się o około 4% poprawności predykcji. Można zauważyć, że trzy najlepsze algorytmy dla takich założeń to różnego typu drzewa decyzyjne. Znacznej poprawie uległ jednak czas tworzenia modeli, który spadł o 45 minut.

Predykcje na poziomie 62% dokładności są zadowalającym wynikiem, nadal znacznie lepszym niż przypadek losowego wyboru serwera przez użytkownika.

Kolejna technika predykcji zakłada dwa następujące po sobie etapy. W pierwszym z nich tworzone są klastry, które powinny odpowiadać różnym warunkom działania sieci. Dopiero

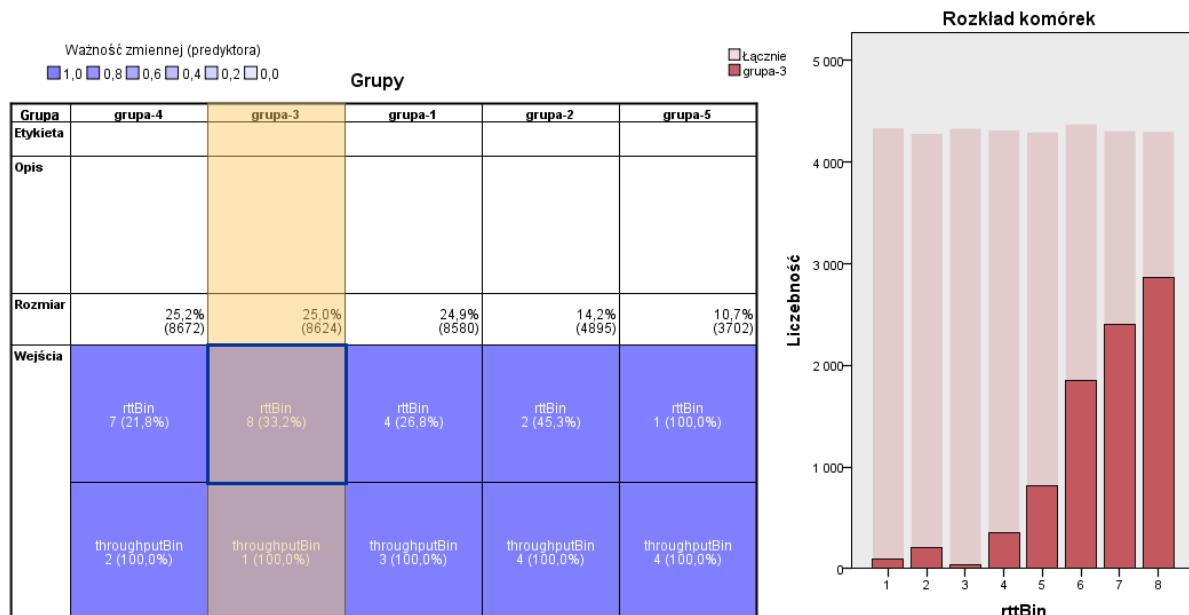
kolejny etap wykonuje właściwą predykcję, klasyfikując badany przypadek do jednego z otrzymanych klastrów.

Klastry stworzono na podstawie dwóch wartości, które dotyczyły były podstawą określania działania serwerów – RTT i przepustowość TCP. Stworzone za pomocą różnych technik klastry zostały pokazane poniżej (Rysunek 9.11).

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Silhouette	Liczba grup	Najmniejsza grupa (N)	Najmniejsza grupa (%)	Największa grupa (N)	Największa grupa (%)	Najmniejsza grupa (N)	Najmniejsza grupa (%)	Największa grupa (N)	Największa grupa (%)	Ważność
<input checked="" type="checkbox"/>		K-średnich 1	< 1	0,616	5	1577	10	3753	25	0,42	0,0			
<input type="checkbox"/>		Sieć Kohonena 1	< 1	0,509	8	91	0	3618	24	0,025	0,0			
<input type="checkbox"/>		Dwustopniowa 1	< 1	0,263	2	7168	48	7689	51	0,932	0,0			

Rysunek 9.11 Stworzone klastry wydajności [Koral, wszystkie serwery]

Najlepszym dopasowaniem (Silhouette) cechuje się klasteryzacja K-średnich. Każdy ze stworzonych klastrów został sprawdzony pod względem jego logiki. Z pięciu wygenerowanych zbiorów można było wyróżnić zauważone wcześniej przypadki zależności przepustowości i RTT. Na przykład serwery o wysokich wartościach RTT, znajdują się w najmniejszym przedziale przepustowości (Rysunek 9.12).



Rysunek 9.12 Przykład grupy w klastrze [Koral, wszystkie serwery]

Stworzone klastry były zmiennymi przewidywanymi w wykonywanych predykcjach. Wyniki zastosowanych technik klasyfikacji widoczne są na Rysunku 9.13. Zaletą wykorzystanej techniki (dwuetapowej) jest bardzo krótki czas budowania modeli, zarówno na poziomie klasteryzacji, jak również klasyfikacji wynosił on parę sekund.

Wyniki z użyciem wszystkich argumentów są na poziomie 68%, co jest wynikiem dobrym. W znajdujących się na obrazku wykresach można zauważyć, że predykcja rozróżnia poszczególne klastry, w najgorszym przypadku poprawnie przypisując połowę wartości klastra czwartego. W badaniach sprawdzono również klasyfikację wykonywaną jedynie przy użyciu

parametrów RTT i dystansu geograficznego. Jakość predykcji, podobnie jak w przypadku przedziału przepustowości, spadła o około 5%.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5.1	< 1	68,325	8
<input checked="" type="checkbox"/>		CHAID 1	< 1	67,631	6
<input checked="" type="checkbox"/>		Sieci neuronowe 1	< 1	67,093	8
<input checked="" type="checkbox"/>		C&RT 1	< 1	63,115	6
<input checked="" type="checkbox"/>		Sieć Bayesa 1	< 1	62,381	8
<input checked="" type="checkbox"/>		Regresja logistyczna 1	< 1	58,585	8
<input checked="" type="checkbox"/>		Quest 1	< 1	57,939	8
<input checked="" type="checkbox"/>		Analiza dyskryminacyjna 1	< 1	41,455	4

Rysunek 9.13 Wyniki predykcji stworzonych klastrów [Koral, Wszystkie serwery]

Kolejne badania sprawdzały predykcję wykonywaną dla wybranych we wcześniejszej analizie hostów. Pierwszym z nich był serwer znajdujący się w Chicago (#47), charakteryzujący się największą wartością oszacowanego współczynnika Hursta. Według teoretycznych założeń, powinien być to serwer, którego zachowanie jest potencjalnie najłatwiejsze do przewidzenia.

W jego przypadku (jak również we wszystkich kolejnych) zastosowano podobne postępowanie, co dla badań wszystkich serwerów, przytoczone powyżej.

Ponieważ wykonywano badania pojedynczego serwera, należało odfiltrować wszystkie wartości, które pozostawały niezmienne w jego obrębie (na przykład dystans geograficzny).

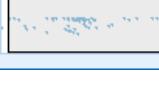
Wartości numeryczne używane w technikach regresji w tym przypadku posiadały małą ważność według wykorzystywanego przez program algorytmu. Jedynymi argumentami, które zostały zakwalifikowane był czas otrzymania pierwszego pakietu, jak również czas zapytania DNS. Dlatego wartości te nie zostały odrzucone w dalszej części badań dla tego serwera.

	Ranga	Zmienna	Pomiar	Ważność	Wartość
<input checked="" type="checkbox"/>	2	# first	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	3	# dns	Ilościowy	Ważne	0,954
<input checked="" type="checkbox"/>	4	# day	Nominalny	Nieważne	0,845
<input checked="" type="checkbox"/>	5	# rtt	Ilościowy	Nieważne	0,788
<input checked="" type="checkbox"/>	6	# hour	Ilościowy	Nieważne	0,307

Rysunek 9.14 Ważność algorytmów bez grupowania [Koral, serwer #47]

Techniki regresyjne wykonane dla predykcji wartości przepustowości w przypadku tego serwera cechują się niską wartością korelacji (około 0,25), jak również bardzo dużym błędem względnym, zbliżonym do 1. Modele te są bezużyteczne z punktu widzenia predykcji.

Niestety, pomimo poczynionych kroków, nie udało się ulepszyć jakości predykcji dla tego przypadku. Znaczenia nie miała tutaj również wielkość plików, ponieważ dla każdego badanego rozmiaru, wartości korelacji i błędu były na podobnym poziomie. Wynik taki sugerował, że predykcja działania tego serwera może okazać się niemożliwa.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		 CHAID 1 < 1	0,256	1	0,961	
<input checked="" type="checkbox"/>		 Sieci neuronowe 1 < 1	0,245	2	0,960	
<input checked="" type="checkbox"/>		 Modele liściaste 1 < 1	0,230	1	0,971	

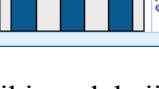
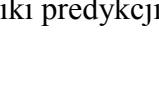
Rysunek 9.15 Wyniki technik regresji dla wszystkich argumentów [Koral, serwer #47]

Dla wybranego serwera wykonano również technikę grupowania. Działanie to dotyczyło wyłącznie przepustowości TCP, która podzielona została na 3 równej wielkości grupy (Rysunek 9.16). RTT wykazuje się zbyt małą zmiennością, by można było je podzielić.

Kategoria	Dolna	Góra
1	>= 107,6461632	< 359,22920704
2	>= 359,22920704	< 463,71100931
3	>= 463,71100931	< 854,37823332

Rysunek 9.16 Grupowanie przepustowości TCP [Koral, serwer #47]

Algorytm badania ważności parametrów ponownie wskazuje zmienne FIRST i DNS. Dodatkowo, jako predyktor brzegowy uznany został dzień tygodnia. Uzyskane na podstawie wszystkich zmiennych wyniki predykcji tych przedziałów znajdują się na poziomie 50% poprawności (Rysunek 9.17). Zakładając, że 33% to wybór wykonywany losowo, wartość ta jest od niego lepsza. Niestety, uzyskany poziom predykcji jest nadal niezadowalający.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		 C&RT 1 < 1	50,521	5	
<input checked="" type="checkbox"/>		 CHAID 1 < 1	48,958	1	
<input checked="" type="checkbox"/>		 Regresja logistyczna 1 < 1	45,833	5	
<input checked="" type="checkbox"/>		 C5 1 < 1	44,792	5	
<input checked="" type="checkbox"/>		 Sieci neuronowe 1 < 1	44,792	5	
<input checked="" type="checkbox"/>		 Analiza dyskryminacyjna 1 < 1	35,938	4	
<input checked="" type="checkbox"/>		 Sieć Bayesa 1 < 1	35,417	5	
<input checked="" type="checkbox"/>		 SVM 1 < 1	32,812	5	
<input checked="" type="checkbox"/>		 Quest 1 < 1	31,771	5	

Rysunek 9.17 Wyniki predykcji przedziałów przepustowości [Koral, serwer #47]

We wcześniejszych badaniach wykonanych na terenie Politechniki Wrocławskiej, wykonujących predykcję pojedynczego serwera, posługiwano się jedynie wyliczonymi wartościami DAY i HOUR. W tym przypadku, wyłącznie ich użycie daje wyniki na poziomie losowym. Wyniki predykcji utrzymują swój poziom niezależnie od wielkości badanego pliku.

Ostatnią techniką do wykonania dla jednego serwera było ponownie użycie technik klasteryzacji, a następnie klasyfikacja stworzonych klastrów. Klastry dla pojedynczego serwera postanowiono utworzyć na podstawie przepustowości TCP, dnia tygodnia i godziny pomiaru. Powiązanie tych wartości miało za zadanie próbę odkrycia typowych dla serwera zachowań.

Wyniki tworzenia klastrów są znacznie gorsze niż w przypadku badania wszystkich serwerów. Parametr Silhouette dla najlepszego przypadku jest na poziomie znajdującym się na pograniczu poprawnie stworzonego modelu (Rysunek 9.18).

Wykorzystanie	Wykres	Model	Czas tworzenia	Silhouette	Liczba grup	Najmniejsza grupa (N)	Najmniejsza grupa (%)	Największa grupa (N)	Największa grupa (%)	Najmniejszy/Największy	Ważność
<input checked="" type="checkbox"/>		K-śr... < 1		0,296	5	26	13	61	31	0,426	0,0
<input type="checkbox"/>		Sie... < 1		0,174	8	5	2	61	31	0,082	0,0

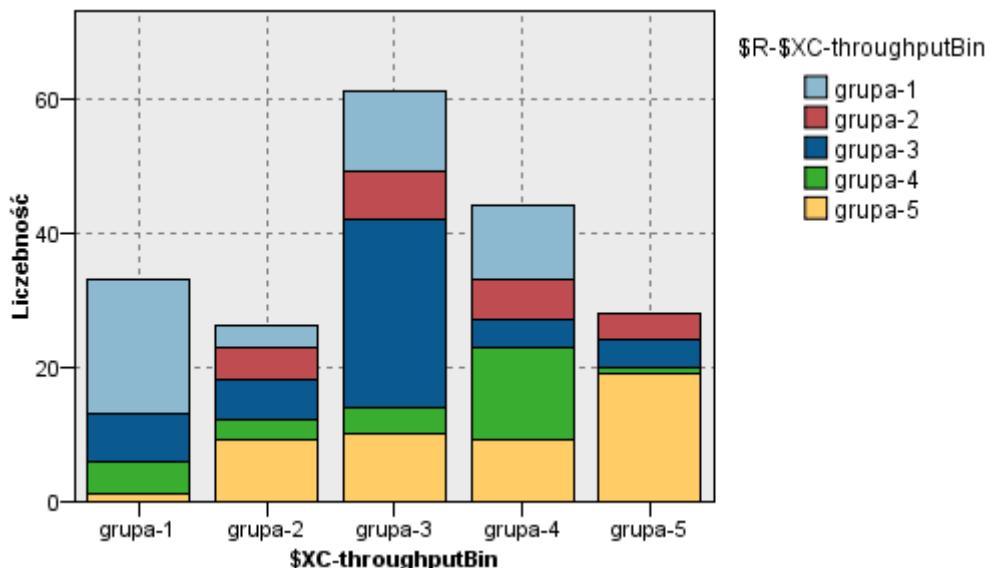
Rysunek 9.18 Stworzone klastry wydajności [Koral, serwer #47]

Badanie poszczególnych grup w zbudowanym modelu techniką K-średnich nie jest tak łatwe, jak w przypadku wcześniejszym. Pora i dzień tygodnia mogą być w jakiś sposób powiązane, jednak ciężko znaleźć logiczne wyjaśnienie takiego działania serwera. Serwer cechujący się samopodobieństwem, powinien charakteryzować się pewną powtarzalnością, której w tym przypadku nie można wyciągnąć. Być może jest ona uzależniona od innych czynników niż te, wykorzystywane w kontekście badań.

Wyniki najlepszego algorytmu (ponownie drzewo C&RT) w klasyfikacji do stworzonych klastrów sięgają 44% poprawnych predykcji. Wartość ta jest lepsza niż wybór losowy, nie wystarczy jednak do wydajnego używania badanego serwera. Wyniki dla algorytmu C&RT wskazują, że przynajmniej połowa obserwacji jest przydzielana do właściwego klastra (oprócz klastra drugiego). Zaobserwować to można na powiększeniu wykresu (Rysunek 9.20).

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C&RT 1	< 1	44,792	5
<input checked="" type="checkbox"/>		Regresja logistyczna	< 1	40,104	5
<input checked="" type="checkbox"/>		C5.0	< 1	38,542	5
<input checked="" type="checkbox"/>		CHAID 1	< 1	37,5	3
<input checked="" type="checkbox"/>		SVM 1	< 1	36,979	5
<input checked="" type="checkbox"/>		Quest 1	< 1	36,458	5
<input checked="" type="checkbox"/>		Sieci neuronowe	< 1	35,417	5
<input checked="" type="checkbox"/>		Sieć Bayesa 1	< 1	33,333	5
<input checked="" type="checkbox"/>		Analiza dyskretowa	< 1	30,729	4

Rysunek 9.19 Wyniki predykcji stworzonych klastrów [Koral, serwer #47]



Rysunek 9.20 Wyniki działania metody C&RT [Koral, serwer #47]

Ostatecznie nie znaleziono sposobu wykonania lepszej prognozy dla wybranego serwera, który rzekomo miał cechować się najlepszymi możliwościami predykcyjnymi, wynikającymi z jego samopodobieństwa. Przyczyna takich wyników może leżeć zarówno po stronie słabo zależnych od siebie parametrów wykorzystywanych w badaniach, jak również błędami w użyciu metodologii Data Mining. W przeszłości, należało sprawdzić inne serwery cechujące się wysoką wartością współczynnika Hursta. Być może serwer w Chicago był w tym przypadku źle dobrana próbka, wykazującą się indywidualnym zachowaniem.

Według przyjętych wcześniej założeń, predykcja ta powinna być jednak lepsza od serwera, którego działanie jest bardzo losowe. Na podstawie wcześniejszych badań ustalono, że najbliższy takiemu zachowaniu jest serwer #9 znajdujący się w Zagrzebie, w Chorwacji.

Techniki regresji wykonane dla tego serwera zakończyły się nawet gorszymi wynikami od przypadku serwera wcześniejszego. Korelacja w tych przypadkach jest bliska zera, podczas gdy błąd względny jest większy niż 1, co bezpośrednio przekłada się na wniosek, że modele te są bezużyteczne z punktu widzenia predykcji.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		Sieci ne...	< 1	0,166	5	1,090
<input checked="" type="checkbox"/>		Modele li...	< 1	0,084	1	1,001
<input checked="" type="checkbox"/>		CHAID 1	< 1	0,048	4	1,168

Rysunek 9.21 Wyniki technik regresji dla wszystkich argumentów [Koral, serwer #9]

Pomimo losowych przebiegów przepustowości, wykonanie grupowania odbyło się bez większych problemów. Podział tej wartości na równe przedziały zakończył się nie budzącym zastrzeżeń wynikiem (Rysunek 9.22).

Kategoria	Dolna	Góra
1	$\geq 272,49870986$	$< 1152,74957331$
2	$\geq 1152,74957331$	$< 1660,24739594$
3	$\geq 1660,24739594$	$< 4035,60402724$

Rysunek 9.22 Grupowanie przepustowości TCP [Koral, serwer #9]

Użycie technik klasyfikacji do predykcji tych przedziałów kończy się na poziomie równym 40% dokładności, czyli o 10% gorszym, niż w przypadku serwera w Chicago przy zastosowaniu tych samych argumentów wejściowych. Wartość predykcji tak bliska wyborowi losowemu sugeruje, że przebiegi przepustowości mogą być faktycznie niezależne w swoim działaniu. Wyniki klasyfikacji przedziałów zaprezentowano na Rysunku 9.23.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5 1	< 1	39,583	5
<input checked="" type="checkbox"/>		CHAID 1	< 1	38,542	2
<input checked="" type="checkbox"/>		Sieci neuron...	< 1	38,542	5
<input checked="" type="checkbox"/>		Sieć Bayesa 1	< 1	38,021	5
<input checked="" type="checkbox"/>		Regresja liniowa	< 1	36,979	5
<input checked="" type="checkbox"/>		SVM 1	< 1	36,979	5
<input checked="" type="checkbox"/>		Quest 1	< 1	36,979	5
<input checked="" type="checkbox"/>		Analiza dyskretowa	< 1	35,938	4
<input checked="" type="checkbox"/>		C&RT 1	< 1	31,771	5

Rysunek 9.23 Wyniki predykcji przedziałów przepustowości [Koral, serwer #9]

Dla tego serwera również stworzono klastry, zawierające te same argumenty – przepustowość, dzień tygodnia i godzinę pomiaru. Model zbudowany za pomocą algorytmu K-średnich wykazywał się dobrym dopasowaniem do danych. Niestety badania poszczególnych grup nie potrafiły wskazać logicznego wyjaśnienia opracowanych klastrów.

Wykorzystanie	Wykres	Model	Czas tworzenia	Silhouette	Liczba grup	Najmniejsza grupa (N)	Najmniejsza grupa (%)	Największa grupa (N)	Największa grupa (%)	Najmniejszy/Największy	Ważność
<input checked="" type="checkbox"/>		K-średnich	< 1	0,433	5	28	14	61	31	0,459	0,0
<input type="checkbox"/>		Sieci neuron...	< 1	0,169	8	7	3	57	29	0,123	0,0

Rysunek 9.24 Stworzone klastry wydajności [Koral, serwer #9]

Uzyskana predykcja za pomocą technik klasyfikacji utrzymuje się na poziomie 42%, niezależnie od tego, czy użyte zostały wszystkie dane wejściowe, czy tylko dzień tygodnia i pora dnia. Klasyfikacja ta jest na podobnym poziomie, jak w przypadku badań wykonanych dla serwera #47. Kluczowym pytaniem jest, czy znajomość trudnych do opisania klastrów przekłada się na predykcję wydajności serwera.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		Analiza dysk... < 1		42,188	4
<input checked="" type="checkbox"/>		C5 1 < 1		41,667	5
<input checked="" type="checkbox"/>		Regresja lo... < 1		41,667	5
<input checked="" type="checkbox"/>		SVM 1 < 1		40,625	5
<input checked="" type="checkbox"/>		Sieć Bayesa 1 < 1		38,542	5
<input checked="" type="checkbox"/>		CHAID 1 < 1		38,542	1
<input checked="" type="checkbox"/>		Sieci neuron... < 1		38,542	5
<input checked="" type="checkbox"/>		Quest 1 < 1		31,771	5
<input checked="" type="checkbox"/>		C&RT 1 < 1		31,771	5

Rysunek 9.25 Wyniki predykcji stworzonych klastrów [Koral, serwer #9]

## 9.2 Agent WCSS

Przypadek wrocławskiego agenta znajdującego się w sieci PIONIER został we wcześniejszym rozdziale przedstawiony jako problematyczny. Wynikało to z analizy zaobserwowanych przebiegów przepustowości, które sugerowały, że ruch na tym serwerze może być sterowany. Z tego względu wykonane predykcje mogą być zniekształcone i w ograniczonym stopniu zależne od działania serwera zdalnego.

Ten agent, jak również serwer pomiarowy w PCSS zostanie opisany znacznie krócej niż Koral. Pod spodem przedstawione zostaną najważniejsze wnioski wyciągnięte z badań dotyczących wszystkich serwerów zdalnych, jak również hostów #47 i #9.

W badaniach wszystkich serwerów ponownie odrzucono argumenty wejściowe w postaci godziny pomiaru i dnia tygodnia. Ze sprawdzanych korelacji, jedynie dzień tygodnia był mało powiązany z wartością przepustowości. Otrzymane wyniki regresji są znacznie lepsze od serwera Koral. Ich korelacja jest na poziomie 0,92, a błąd względny zaledwie 14%. Zaobserwowaną jakość predykcji przedstawiono na Rysunku 9.26.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		CHAID 1 < 1		0,929	7	0,136
<input checked="" type="checkbox"/>		C&RT 1 < 1		0,928	7	0,140
<input checked="" type="checkbox"/>		Sieci ne... < 1		0,923	7	0,148

Rysunek 9.26 Wyniki technik regresji dla wszystkich argumentów [WCSS, wszystkie serwery]

Lepsze wyniki pokazują, że pomimo rzekomo sterowanych przebiegów, rozróżnienie na poziomie wszystkich serwerów jest wysokiej jakości. Być może różnica w wynikach związana jest z małym wpływem zakłóceń na wykonane pomiary przepustowości. Ponownie, to RTT jest najsilniejszym predyktorem – potwierdza to bardzo dobre dopasowanie RTT-przepustowość, otrzymane we wcześniejszych badaniach. Dla serwera WCSS było ono największe.

W przypadku predykcji przedziałów przepustowości, ponownie podzielono wartości numeryczne na odpowiednie grupy. Podczas wykonywania tej operacji dla RTT zaobserwowano, że każdy z utworzonych przedziałów ma około 30 ms większą wartość niż w przypadku serwera Koral. Była to kolejna sugestia, że ruch musi być w jakiś sposób sterowany bądź analizowany. Nielogicznym wydawał się fakt, by znajdujący się bliżej kręgosłupa sieci serwer we Wrocławiu otrzymywał odpowiedź wolniej, niż agent znajdujący się wewnątrz sieci Politechniki Wrocławskiej.

Podczas wykorzystania wszystkich argumentów do algorytmów klasyfikacji, osiągnięto wyniki na poziomie 79%. Najlepsze okazały się techniki sieci neuronowych, SVM i drzew decyzyjnych C5. Niestety budowa tych modeli była dłużna (około 15 minut). Na podstawie badań ważności predyktorów, wyciągnięto dwa najważniejsze argumenty wejściowe – były to ponownie RTT i dystans geograficzny. Użyto je w kolejnej klasyfikacji przedziałów przepustowości (Rysunek 9.27). Okazało się, że przy takich założeniach jeszcze lepszą predykcję wykazywał się algorytm KNN (K-najbliższych sąsiadów). Była ona na poziomie 80%, a jej skuteczność sugeruje dobre rozmieszczenie problemu predykcji w przestrzeni. Wynik ten jest około 20% lepszy niż w przypadku serwera Koral. Dodatkowo, budowa modeli wynosiła już tylko 3 minuty.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		Algorytm KN...	3	80,397	2
<input checked="" type="checkbox"/>		C5 1	3	76,6	2
<input checked="" type="checkbox"/>		C&RT 1	3	74,502	2

Rysunek 9.27 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [WCSS, wszystkie serwery]

Użycie technik klasteryzacji dla danych zebranych w WCSS, cechuje się wysoką wartością dopasowania. W odróżnieniu od przypadków wcześniejszych, dobrym dopasowaniem charakteryzuje się również sieć Kohonena. Jest ona jednak gorsza od algorytmu K-średnich, której klastry ponownie układają się w zauważone wcześniej relacje RTT-przepustowość. Wyniki przedstawiono na Rysunku 9.28.

Wykorzystanie	Wykres	Model	Czas tworzenia	Silhouette	Liczba grup	Najmniejsza grupa (N)	Najmniejsza grupa (%)	Największa grupa (N)	Największa grupa (%)	Najmniejszy/Największy	Ważność
<input checked="" type="checkbox"/>		K-śr... < 1	0,645		5	1187	10	2849	25	0,417	0,0
<input type="checkbox"/>		Sie... < 1	0,480		9	114	1	2655	23	0,043	0,0
<input type="checkbox"/>		Dw... < 1	0,342		2	5580	49	5668	50	0,984	0,0

Rysunek 9.28 Stworzone klastry wydajności [WCSS, wszystkie serwery]

Klasyfikacja utworzonych klastrów przebiegła pomyślnie i znajdowała się na poziomie 82%, z kilkuprocentową przewagą sieci neuronowych. Po sprawdzeniu ważności predyktorów dla wszystkich użytych metod okazało się, że oprócz używanego RTT, wysoki współczynnik posiadała również rangę AS. Wykorzystanie tych dwóch argumentów zostało przedstawione na Rysunku 9.29. Osiągnięto bardzo dobre wyniki na poziomie 83% dokładności dla drzewa decyzyjnego C5.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5 1	< 1	83,046	2
<input checked="" type="checkbox"/>		CHAID 1	< 1	76,725	2
<input checked="" type="checkbox"/>		C&RT 1	< 1	76,174	2
<input checked="" type="checkbox"/>		Sieci neuron...	< 1	75,622	2
<input checked="" type="checkbox"/>		Sieć Bayesa	< 1	73,862	2
<input checked="" type="checkbox"/>		Quest 1	< 1	70,466	2
<input checked="" type="checkbox"/>		Regresja lo...	< 1	70,404	2
<input checked="" type="checkbox"/>		Analiza dysk...	< 1	23,302	1

Rysunek 9.29 Wyniki predykcji stworzonych klastrów dla RTT i dystansu [WCSS, wszystkie serwery]

Zbadano przypadek serwera #47. Ważność argumentów dla tego kierunku była bardzo niewielka. Jedyny parametr, który miał znaczenie (brzegowe) to dzień tygodnia. Wartość RTT została automatycznie odrzucona ze względu na małą zmienność.

Wyniki technik regresji cechują się bardzo wysokim błędem względnym, o wartości powyżej 0,91. W tym przypadku również nie można było w poprawny sposób prognozować wartości przepustowości na podstawie badanych danych.

Pomimo akceptowalnego podziału wartości przepustowości na odpowiednie grupy, techniki klasyfikacyjne wykorzystane w predykcji cechowały się maksymalnie 51% dokładności dla danych testowych. Podobne wyniki uzyskano dla tego serwera na podstawie danych z agenta Koral, również przy użyciu drzew decyzyjnych. Z racji braku korelacji między tymi dwoma serwerami, uzyskanie podobnych wartości może być przypadkowe (tym bardziej przy założeniu sterowania ruchem przez agenta WCSS).

Podczas prób klasteryzacji nie udało się utworzyć klastrów o dobrym dopasowaniu do danych. Będące na skraju poprawności modele posiadały bardzo trudne do rozróżnienia grupy. Dlatego też podjęto decyzję o braku kontynuacji badań w tym zakresie.

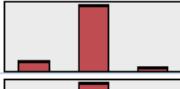
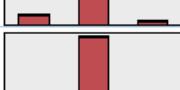
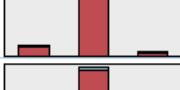
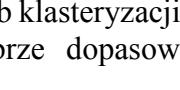
Techniki regresji użyte dla predykcji zachowania serwera #9 zakończyły się podobnie. Tak jak w przypadku wcześniejszym (dla agenta Koral), miały one regresję bliską zeru, a błąd względny był w każdym modelu większy od 1, co jest wartością gorszą nawet od serwera w Chicago. Jakości predykcji nie pomogł nawet fakt, że ustalona przez program ważność predyktorów wskazywała jedynie na odrzucenie wartości DNS i RTT.

Po obserwacji podziału przepustowości na równe grupy zauważono, że taka metoda dzielenia jest w tym przypadku niewskazana. Ponieważ znaczna część danych znajdowała się w środku rozkładu, zdecydowano się na podział zjawiska z użyciem odchylenia standardowego. Dane zostały podzielone na grupy, które były stworzone na zasadzie odległości od średniej. Grupą zerową były wartości z przedziału [średnia  $\pm$  odchylenie standardowe], a grupy -1 i 1 to odpowiednio wartości mniejsze i większe od grupy zerowej (Rysunek 9.30).

Kategoria	Dolna	Góra
-1		< 3603,75000213
0	$\geq 3603,75000213$	$\leq 5133,46927787$
1	$> 5133,46927787$	

Rysunek 9.30 Grupowanie przepustowości TCP [WCSS, serwer #9]

Użyte dla tego podziału metody klasyfikacji nie potrafiły wykonać rozpoznania innej grupy niż grupa zerowa. Grupa ta zajmowała 80% obserwacji, dlatego taką dokładnością cechowała się jakość predykcji. Niestety liczba ta jest złudna ze świadomością, że każdy z badanych przypadków został zaliczony do grupy zerowej (pomimo, że należał do grup -1 i 1). Wynik zobaczyć można na Rysunku 9.31. Pokazuje on, że tego typu badania nie powinny pokazywać wyłącznie procentu predykcji jako potwierdzenia skuteczności zbudowanych modeli.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		 C5.1	< 1	82,877	4
<input checked="" type="checkbox"/>		 Sieć Bayesa 1	< 1	82,877	4
<input checked="" type="checkbox"/>		 CHAID 1	< 1	82,877	1
<input checked="" type="checkbox"/>		 Quest 1	< 1	82,877	4
<input checked="" type="checkbox"/>		 C&RT 1	< 1	82,877	4
<input checked="" type="checkbox"/>		 Sieci neuronowe 1	< 1	82,877	4
<input checked="" type="checkbox"/>		 Regresja logistyczna 1	< 1	81,507	4
<input checked="" type="checkbox"/>		 SVM 1	< 1	81,507	4
<input checked="" type="checkbox"/>		 Analiza dyskryminacyjna 1	< 1	50,0	3

Rysunek 9.31 Wyniki predykcji przedziałów przepustowości [WCSS, serwer #9]

Przypadek prób klasteryzacji zakończył się podobnie, jak dla serwera w Chicago. Nie udało się stworzyć dobrze dopasowanych klastrów, których użycie miałoby jakieś logiczne wyjaśnienie.

Uzyskane wyniki dla agenta WCSS są znacznie lepsze dla badań wszystkich serwerów, jednak zarówno w jednym, jak również drugim przypadku szczególnym, nie udało się uzyskać przyzwoitych wyników predykcji dla wykorzystanych metod eksploracji danych.

### 9.3 Agent PCSS

Pomimo tego, że również znajduje się on w sieci PIONIER, agent PCSS nie wykazywał żadnych zachowań, które mogłyby wskazywać na sterowanie ruchem internetowym. Działanie tego punktu pomiarowego i uzyskane predykcje powinny być porównywane przede wszystkim z serwerem znajdującym się w WCSS.

Wyniki technik regresyjnych dla tego agenta są bardzo podobne do tych, uzyskanych w WCSS. Na Rysunku 9.32 przedstawione zostały modele zbudowane na podstawie dwóch wartości – RTT i dystansu geograficznego. Cechują się one nawet mniejszym błędem względnym, niż w przypadku drugiego agenta sieci PIONIER. Komputery znajdujące się w pobliżu kręgosłupa sieci znacznie lepiej potrafią rozwiązać problem wyboru jednego serwera z wielu posiadanych możliwości.

Najlepsze wyniki uzyskano ponownie dla tych samych algorytmów – C&RT i sieci neuronowych. Sugeruje to użycie tych technik do badania przypadku wszystkich serwerów w przyszłości.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		C&RT 1	< 1	0,940	2	0,117
<input checked="" type="checkbox"/>		Sieci neuronowej	< 1	0,923	2	0,148
<input checked="" type="checkbox"/>		CHAID 1	< 1	0,858	2	0,264

Rysunek 9.32 Wyniki technik regresji dla wybranych argumentów [PCSS, wszystkie serwery]

Podczas grupowania wartości RTT zauważono, że była ona najmniejsza ze wszystkich badanych serwerów. Porównując ten parametr do agenta WCSS można przypuszczać, że ruch znajdujący się w poznańskiej sieci jest poddawany mniejszej ilości analiz (o ile w ogóle).

Dla stworzonych przedziałów przepustowości (ponownie w ilości czterech), techniki klasyfikacji cechują się dokładnością na poziomie 80% i wyższym. Wyniki te otrzymane zostały jedynie na podstawie algorytmów drzew decyzyjnych. W przypadku ograniczenia argumentów wejściowych do RTT i dystansu, podobną wartość predykcji uzyskuje algorytm K-najbliższych sąsiadów. Otrzymane wyniki są bardzo podobne do tych, którymi charakteryzował się agent WCSS.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		Algorytm KN...	< 1	80,035	2
<input checked="" type="checkbox"/>		C&RT 1	< 1	77,223	2
<input checked="" type="checkbox"/>		Quest 1	< 1	74,360	2

Rysunek 9.33 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [PCSS, Wszystkie serwery]

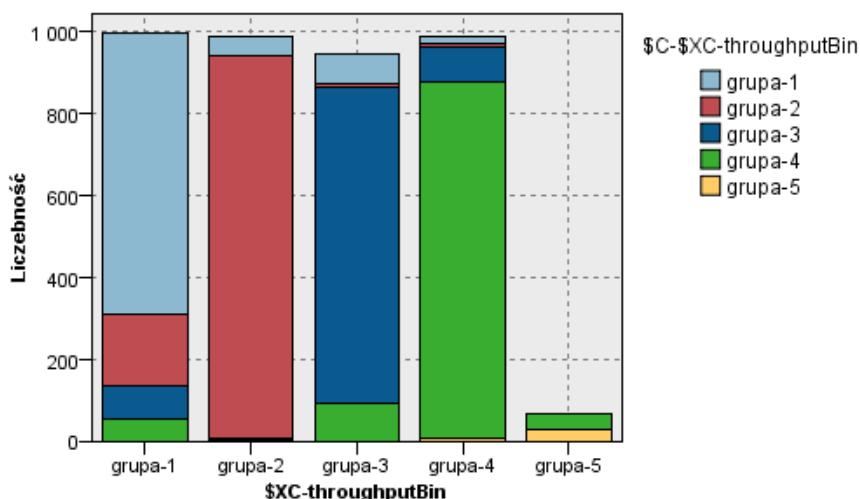
Metody klasteryzacji dla PCSS tworzą klastry o bardzo dobrym dopasowaniu. W tym przypadku określone zostały 4 duże klastry oraz jeden o wiele mniejszy, który cechował się bardzo wysokimi wartościami RTT, lecz średnią (a nie wolną) szybkością transmisji.

Otrzymane wyniki znajdują się na poziomie 80% i wyżej, podobnie jak w przypadku prognozy przedziału przepustowości (Rysunek 9.34). Predykcja jest również na bardzo podobnym poziomie do tego, który uzyskano dla agenta WCSS. Użycie RTT i dystansu geograficznego jako jedynych argumentów wejściowych, zmniejsza jakość predykcji o około 4%.

Ostatnia, najmniejsza grupa dla stworzonych modeli jest prawidłowo prognozowana w 50% przypadków. Oznacza to, że nie została ona tak mocno dominowana przez pozostałe, większe klastry (Rysunek 9.35).

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		 C5 1	< 1	82,697	7
<input checked="" type="checkbox"/>		 C&RT 1	< 1	81,115	7
<input checked="" type="checkbox"/>		 CHAID 1	< 1	79,759	7
<input checked="" type="checkbox"/>		 Sieci neuronowe 1	< 1	79,131	7
<input checked="" type="checkbox"/>		 Sieć Bayesa 1	< 1	74,611	7
<input checked="" type="checkbox"/>		 Quest 1	< 1	74,46	7
<input checked="" type="checkbox"/>		 Regresja logistyczna 1	< 1	71,899	7
<input checked="" type="checkbox"/>		 Analiza dyskryminacyjna 1	< 1	50,954	4

Rysunek 9.34 Wyniki predykcji stworzonych klastrów [PCSS, wszystkie serwery]



Rysunek 9.35 Rozpoznane przypadki dla algorytmu C5 [PCSS, wszystkie serwery]

W przypadku serwera #47 techniki regresji wykazywały się lepszym działaniem niż w przypadku dwóch wcześniejszych agentów pomiarowych. Najlepszy z opracowanych modeli (drzewo C&RT) posiada wysoki wskaźnik korelacji, znajdujący się na poziomie 0,7. Niestety błąd względny jest nadal wysoki – około 88%. Wartość ta sugeruje, że predykcja wykonywana przez opracowane modele jest bardzo niskiej jakości.

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Korelacja	Liczba użytych zmiennych	Błąd względny
<input checked="" type="checkbox"/>		C&RT 1	< 1	0,708	4	0,88
<input checked="" type="checkbox"/>		CHAID 1	< 1	0,565	3	0,802
<input checked="" type="checkbox"/>		Modele li... < 1		0,356	2	1,054

Rysunek 9.36 Wyniki technik regresji dla wszystkich argumentów [PCSS, serwer #47]

Podział wartości przepustowości na trzy grupy okazał się dobrze rozdzielać używane dane. Wykorzystanie technik klasyfikacji zakończyło się zbudowaniem modeli o dokładności około 47%. Jest to wartość o 3% mniejsza niż dla wcześniejszych agentów. W tym przypadku jednak, jakością predykcji bliskiej 50% cechował się nie pojedynczy algorytm, lecz kilka – głównie drzew decyzyjnych.

W przypadku klastryzacji po raz pierwszy zaobserwowano lepsze dopasowanie z użyciem sieci Kohonena. Niestety jakość tego dopasowania graniczyła z poprawną, a stworzona ilość klastrów (9) i ich wstępna analiza podważała użyteczność modelu.

Podczas badań serwera #9 zauważono, że jakość modeli zbudowanych przez techniki regresyjne dla hosta w Zagrzebie posiadają błąd względny o wartości powyżej 1, co ponownie potwierdza, że uzyskane modele są bezużyteczne.

Wyniki otrzymane przez badania wszystkich agentów pokazują, że metody regresyjne mają gorsze działanie dla serwera, którego przebiegi nie cechują się samopodobieństwem. Niestety z racji bardzo słabej hosta #47, w którym współczynnik Hursta był największy, różnica ta nie ma większego znaczenia (nie można powiedzieć, że jest on lepiej prognozowany).

W podziale przepustowości serwera #9 również zastanawiano się nad użyciem odchylenia standardowego, jednak ostatecznie zdecydowano się na grupy o równej ilości obserwacji. Żadne z wykorzystanych technik klasyfikacji nie okazały się wystarczająco dobre, by poprawić jakość losowego wyboru predykcji – dokładność wszystkich modeli znajdowała się na poziomie bliskim 33%.

Stworzenie klastrów zakończyło się poprawnie, chociaż również w tym przypadku ze zbudowanych modeli ciężko było wywnioskować specyficzne zachowania dla serwera w Zagrzebie. Wyniki klasyfikacji tych klastrów były na poziomie 38%, co oczywiście nie jest żadną poprawą w stosunku do wyboru losowego.

Predykcja serwera #9 za pomocą każdą z użytych technik okazuje się być na poziomie wyboru losowego. Zachowanie to jest zgodne z przewidywaniami.

## **10. Podsumowanie**

W pracy magisterskiej wykonano dokładną analizę zebranych danych dotyczących wydajności webowej. Pomiary, na których podstawie wyprowadzano kolejne wnioski, zbierane były przez system pomiarowy MWING, wykorzystywany wcześniej w badaniach na terenie Politechniki Wrocławskiej. Instancje tego środowiska zostały zainstalowane w trzech punktach pomiarowych, znajdujących się w Polsce. Pierwszy z nich znajdował się wewnętrznej sieci Politechniki Wrocławskiej, natomiast dwa kolejne punkty były węzłami sieci ogólnopolskiej infrastruktury sieci PIONIER, znajdującymi się we Wrocławiu i Poznaniu.

Kluczowym aspektem pracy było przeprowadzenie badań z wykorzystaniem metodologii eksploracji danych. Po wykonaniu kolejnych etapów zakładanych przez tę technikę, udało się otrzymać obserwacje, które następnie były poddawane odpowiednim metodom grupowania, klasyfikacji i regresji.

Problem badawczy podzielono na dwie części. Pierwsze z przewidzianych zadań polegało na predykcji wydajności wielu serwerów pomiarowych. Znajomość jakości ich działania jest kluczowa w przypadkach, gdy celem użytkownika jest wybór serwera działającego najszybciej. Użycie technik eksploracji danych dla tego przypadku przyniosło bardzo dobre wyniki rzędu 83% poprawnych klasyfikacji.

Zauważono, że serwery znajdujące się w SIECI PIONIER (bliżej tzw. „kręgosłupa sieci”) cechują się znacznie lepszą jakością uzyskanych predykcji. Szczegółowe analizy pokazały, że przebiegi zbierane przez obydwa punkty pomiarowe nie są ze sobą skorelowane. Mimo braku takiej charakterystyki, dokładność wykonanych predykcji w kontekście działania wszystkich serwerów jest na bardzo zbliżonym poziomie. Wynik ten sugeruje, że zastosowanie algorytmów eksploracji danych może być dobrym sposobem predykcji prognozowania wydajności dla klientów znajdujących się bliżej najwyższego poziomu sieci.

Inaczej prezentują się wartości predykcji dla serwera Koral, znajdującego się wewnętrz sieci Politechniki Wrocławskiej. Wykonane prognozy osiągają poprawność około 68%. Jest to wynik dobry, znacznie lepszy niż przypadek najgorszy, czyli wybór losowy, oscylujący w wartościach około 25% (zależnie od przyjętych założeń). W porównaniu do dwóch pozostałych serwerów, agent ten cechował się dużymi zakłóceniami w pomiarach, jak również największą wrażliwością na przeciążenia sieci – może być to przyczyną gorszej predykcji.

W porównaniu do wyników otrzymanych we wcześniejszych badaniach w środowisku Politechniki Wrocławskiej, jest to spadek o około 12% poprawności wykonanych predykcji. Należy jednak pamiętać, że wykonane pomiary przeprowadzono na różnych kierunkach, przy różnym stanie dynamicznie zmieniającej się topologii sieci. Bezpośrednie ich porównywanie ma wyłącznie charakter poglądowy.

Pomimo, że dwa z agentów pomiarowych są położone obok siebie w kontekście geograficznym, topologia sieci w której się znajdują jest różna. Po sprawdzeniu korelacji przebiegów przepustowości na wybranych kierunkach stwierdzono, że nie można wykorzystać pomiarów z jednego agenta do predykcji wydajności dla drugiego. Zauważono również, że we wrocławskim systemie znajdującym się w infrastrukturze PIONIER, ruch w sieci może być sterowany. Działanie takie mogło znacznie zniekształcić wyniki przeprowadzonych analiz.

W pracy porównano również przyspieszenie poszczególnych etapów połączenia TCP względem pomiarów wykonywanych w 2008 roku. Okazało się, że przyspieszeniu uległ każdy z kroków, z wyjątkiem czasu pobranie pierwszego pakietu danych z serwera. W ogólnym

rozrachunku, współczynnik przyspieszenia osiągnął wartość 181%. Biorąc pod uwagę fakt, że w pomiarach użyty został inny rozmiar plików, współczynnik ten może być większy.

Druga część badawcza z wykorzystaniem metod eksploracji danych polegała na predykcji działania wybranego serwera. Ponieważ posiadano dużą próbke (87) serwerów pomiarowych, zdecydowano się na wybór dwóch, cechujących się potencjalnie najwyższą i najmniejszą predykcją, o czym powiedzieć miała wartość ich samopodobieństwa.

Informację o samopodobieństwie serwera otrzymano za pomocą estymacji współczynnika Hursta. Na jego podstawie określono dwa serwery – w Chicago, której wartość tego parametru była największa i w Zagrzebie, gdzie nie zauważono powtarzalności w działaniu.

Podczas tych badań okazało się również, że zmienność czasu odpowiedzi (RTT) dla wybranych serwerów wykazuje się małą zmiennością. Parametr ten przestaje być ważny w obrębie pojedynczego serwera i nie jest użyteczny w predykcjach, tak jak miało to miejsce we wcześniej przeprowadzonych badaniach.

Wyniki predykcji wykorzystanych technik eksploracji danych dla obu wybranych serwerów są bardzo niskiej radości. Jest to zaledwie 50% w przypadku serwera cechującego się największym samopodobieństwem (33% to wybór losowy). Wynik ten sugeruje, że wartość przepustowości TCP może być zależna od czynników innych niż te, założone w badaniach – dzień tygodnia, godzina pomiaru i inne parametry połączenia TCP.

Serwer, którego działanie zostało ocenione jako losowe, cechuje się predykcją na poziomie 33%, czyli wyboru losowego. Podobne wyniki zaobserwowano dla wszystkich zebranych danych. Założenia te zgadzają się z teoretycznymi.

Wyniki metod eksploracji danych są zależne od każdego podjętego kroku – zarówno wyboru danych, jak również ich filtracji, tworzonych podziałów, przyjętych klastrów itd. Dla zebranych danych na pewno istnieją sposoby na uzyskanie lepiej dopasowanych modeli.

Najlepiej działającymi algorytmami eksploracji danych dla badanych obserwacji okazały się drzewa decyzyjne i ich poszczególne implementacje – CHAID, C&RT, QUEST, C5. Bardzo podobnymi (a czasem lepszymi) wynikami cechowały się sieci neuronowe. Dla niewielkiej ilości argumentów wejściowych, bardzo dobre wyniki można było uzyskać za pomocą algorytmu K-najbliższych sąsiadów.

W pracy prowadzono również inne analizy, które były dopełnieniem dla tematu eksploracji danych. Wykonano sprawdzenie zależności median RTT i przepustowości dla każdego z badanych serwerów zdalnych. Dopasowując uzyskane przebiegi do postaci funkcji potęgowej otrzymano dopasowanie  $R^2 = 0,94$  dla najmniejszej wielkości pliku (0,5 MB). Tak wysoka wartość sugeruje, że łatwiejsze do pomiaru RTT może być użyte w predykcji przepustowości. Dopasowanie to zmniejsza się wraz ze wzrostem wielkości pliku – dla 3 MB współczynnik ten wynosi już 0,85. Takie zmiany wskazują, że korzystanie z RTT może być więc przydatne w predykcji działania stron internetowych, lecz niekoniecznie przy pobieraniu dużych plików. Potwierdzają to wykonane badania eksploracji danych dla wszystkich serwerów, gdzie RTT było zawsze postrzegane jako najważniejszy z argumentów wejściowych.

Wykonano dokładną analizę systemów autonomicznych dla wybranych kierunków. Otrzymane wyniki wykorzystano w metodach eksploracji danych. Metodologię pozwalającą na otrzymanie mapy grafów można wykorzystać w badaniach, które będą sprawdzały wydajność zbliżonych do siebie serwerów zdalnych. Warto zastanowić się nad rozwiązaniem, które zautomatyzuje sprawdzanie charakterystyk wybranego serwera. Na potrzeby pracy stworzono skrypty, które wykonywały to w sposób wymagający działania użytkownika.

Stanowisko opracowane na potrzeby pracy magisterskiej może zostać wykorzystane do przyszłych badań w zakresie wydajności Web. Mogą być to m.in. pomiary kontrolne, wykonywane na tych samych kierunkach, mające na celu sprawdzenie poprawy wyników w czasie. Można również stworzyć nowe stanowiska pomiarowe w innych krajach, by badania te miały wydłużek światowy, a nie tylko lokalny.

Technika eksploracji danych jest zaimplementowana w wielu pakietach dostępnych dla użytku publicznego. Wykorzystany program IBM SPSS Modeler to jedna z licznych możliwości. W badaniach można użyć innych narzędzi – chociażby dodatku Data Mining dla pakietu Microsoft Excel lub oprogramowania Weka. Znajdujące się tam metody mogą dać znacznie lepsze wyniki predykcji dla zebranych obserwacji.

Dodatkowe badania można, a nawet należy wykonać dla danych zebranych na potrzeby napisanej pracy magisterskiej. Przestrzeń badawcza dla wykonanych pomiarów jest bardzo duża. W badaniach przytoczono zaledwie dwa wybrane problemy badawcze.

Jeden z problemów badawczych zakładał estymację współczynnika Hursta. Jest to parametr najczęściej używany w ekonomii, przez co jego implementacje mogły być niedopasowane do badanego zjawiska. W celu sprawdzenia poprawności jego wyliczenia należy stworzyć własny algorytm, którego parametryzacja byłaby lepiej dopasowana do problemów sieci.

W przyszłych, podobnych badaniach warto skupić się również na plikach większych, ponieważ to wydajność ich pobierania w znaczny sposób przyczynia się do długości oczekiwania użytkownika końcowego.

## Bibliografia

- [1] Miller L., *Internet Performance For Dummies*, Hoboken: John Wiley & Sons, 2015.
- [2] Borzemski L., „The use of Data Mining to predict Web performance,” *Cybernetics and Systems*, tom 37, nr 6, pp. 587-608, 2006.
- [3] Nah F., „A study on tolerable waiting time - how long are Web users willing to wait,” *Behaviour & Information Technology*, tom 23, nr 3, pp. 153-163, 2004.
- [4] Krishnamurthy B. i Wills C., „Analyzing factors that influence end-to-end Web performance,” *Computer Networks*, nr 33, pp. 17-32, 2000.
- [5] Bobcares, „Server Load – The Basics,” 3 Styczeń 2006. [Online]. Available: <https://bobcares.com/blog/server-load-the-basics-2/>. [Data uzyskania dostępu: 4 Czerwiec 2017].
- [6] Park D., Seong B. i Shin S., „Improving World-Wide-Web performance using domain-top approach to prefetching,” w *Proceedings Fourth International Conference / Exhibition on High Performance Computing in the Asia-Pacific Region*, 2000.
- [7] Cito J., Dustar S., Gotowka D., Leitner P., Pelette R. i Suljoti D., „Zeitschriftenartikel S. Dustdar 2015 Identifying Web Performance,” *Journal of Web Engineering*, tom 14, nr 5 i 6, pp. 414-442, 2015.
- [8] Jacobsen O., „TCP Performance,” *The Internet Protocol Journal*, tom 3, nr 2, pp. 1-3, 2000.
- [9] Williamson C., „Internet Traffic Measurement,” Calgary, 2001.
- [10] W3Techs, „Usage of HTTP/2 for websites,” [Online]. Available: <https://w3techs.com/technologies/details/ce-http2/all/all>. [Data uzyskania dostępu: 4 Czerwiec 2017].
- [11] Oprescu I., Saxce H. i Yiping C., „Is HTTP/2 Really Faster Than HTTP/1.1?,” w *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Hong Kong, 2015.
- [12] Arsenault C., „14 Important Website Performance Metrics You Should Be Analyzing,” 1 Czerwiec 2017. [Online]. Available: <https://www.keycdn.com/blog/website-performance-metrics/>. [Data uzyskania dostępu: 4 Czerwiec 2017].
- [13] Barford P., „Measurement, Modeling Modeling and Analysis Analysis of the Internet,” Marzec 2004. [Online]. Available: [http://pages.cs.wisc.edu/~pb/ima\\_tutorial.pdf](http://pages.cs.wisc.edu/~pb/ima_tutorial.pdf). [Data uzyskania dostępu: 7 Czerwiec 2017].
- [14] Bagnulo M., Eardley P., Trammell B., Trevor B. i Winter R., „Standardizing large-scale measurement platforms,” *ACM SIGCOMM Computer Communication Review*, tom 43, nr 2, pp. 58-63, 2013.

- [15] SamKnows Limited, „Test Methodology White Paper,” 1 Lipiec 2011. [Online]. Available:  
[https://availability.samknows.com/broadband/uploads/Methodology\\_White\\_Paper\\_20110701.pdf](https://availability.samknows.com/broadband/uploads/Methodology_White_Paper_20110701.pdf). [Data uzyskania dostępu: 7 Czerwiec 2017].
- [16] Burnett S., Feamster N. i Sundaresan S., „BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks,” w *Proceedings of USENIX ATC '14: 2014 USENIX Annual Technical Conference*, Philadelphia, 2014.
- [17] Bajpai V. i Schonwalder J., „Large-Scale Network Measurements,” Październik 2013. [Online]. Available: <http://www.cnsm-conf.org/2013/documents/Key-note-3-JS.pdf>. [Data uzyskania dostępu: 7 Czerwiec 2017].
- [18] PlanetLab, „About PlanetLab,” [Online]. Available: <https://www.planet-lab.org/about>. [Data uzyskania dostępu: 7 Czerwiec 2017].
- [19] Borzemski L. i Kamińska-Chuchmała A., „2013 Distributed Web Systems Performance Forecasting Using Turning Bands Method,” *IEEE Transactions on Industrial Informatics*, tom 9, nr 1, pp. 254-261, 2013.
- [20] Borzemski L. i Starczewski G., „Application of Transfer Regression to TCP Throughput Prediction,” w *First Asian Conference on Intelligent Information and Database Systems*, Dong Hoi, 2009.
- [21] Barford P., Mirza M., Sommers J. i Zhu X., „A Machine Learning Approach to TCP Throughput Prediction,” w *SIGMETRICS '07*, San Diego, 2007.
- [22] Bhulai S., „Modeling and Predicting End-to-End Response Times in Multi-Tier Internet Applications,” w *Managing Traffic Performance in Converged Networks: 20th International Teletraffic Congress*, Berlin, Heidelberg, 2007, pp. 519-532.
- [23] Faloutsos M., Karagiannis T. i Molle M., „A Nonstationary Poisson View of Internet Traffic,” w *INFOCOM 2004*, Hong Kong, 2004.
- [24] Huang T.-i. i Subhlok J., „Fast Pattern-Based Throughput Prediction for TCP Bulk Transfers,” w *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid*, 2005.
- [25] Vazhkudai S., J. Schopf i I. Foster, „Predicting the Performance of Wide Area Data Transfers,” 2002.
- [26] Borzemski L. i Kliber M., „Using Data Mining algorithms in Web performance prediction,” *Cybernetics and Systems*, tom 40, nr 2, pp. 176-187, 2009.
- [27] Borzemski L. i Nowak Z., „An Empirical Study of Web Quality: Measuring the Web from the Wroclaw University of Technology Campus,” w *Engineering Advanced Web Applications*, Princeton, Rinton, 2004, pp. 307-320.
- [28] Oracle, „What Is Data Mining?,” [Online]. Available: [https://docs.oracle.com/cd/B28359\\_01/datamine.111/b28129/process.htm#CHDFGC1J](https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process.htm#CHDFGC1J). [Data uzyskania dostępu: 7 Czerwiec 2017].

- [29] Manik S., „Steps of Data Mining,” 31 Październik 2008. [Online]. Available: <http://dataminingwarehousing.blogspot.com/2008/10/data-mining-steps-of-data-mining.html>. [Data uzyskania dostępu: 7 Czerwiec 2017].
- [30] Hong T., „Occupant Behavior Research - Data Mining,” [Online]. Available: <https://behavior.lbl.gov/?q=node/11>. [Data uzyskania dostępu: 12 Czerwiec 2017].
- [31] IBM, „IBM Knowledge Center - Kohonen Node,” [Online]. Available: [https://www.ibm.com/support/knowledgecenter/pl/SS3RA7\\_15.0.0/com.ibm.spss.modeler.help/kohonennode\\_general.htm](https://www.ibm.com/support/knowledgecenter/pl/SS3RA7_15.0.0/com.ibm.spss.modeler.help/kohonennode_general.htm). [Data uzyskania dostępu: 7 Czerwiec 2017].
- [32] IBM, „IBM Knowledge Center - K-means clustering,” [Online]. Available: [https://www.ibm.com/support/knowledgecenter/en/SSPT3X\\_4.1.0/com.ibm.db2.luw.analytics.commsql.doc/doc/r\\_kmeans\\_clustering.html](https://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.1.0/com.ibm.db2.luw.analytics.commsql.doc/doc/r_kmeans_clustering.html). [Data uzyskania dostępu: 7 Czerwiec 2017].
- [33] Pokarowski P. i Prochenka A., „Statystyka II Wykłady,” [Online]. Available: <http://mst.mimuw.edu.pl/lecture.php?lecture=st2&part=Ch6>. [Data uzyskania dostępu: 7 Czerwiec 2017].
- [34] IBM, „Decision Tree Models,” [Online]. Available: [https://www.ibm.com/support/knowledgecenter/en/SS3RA7\\_15.0.0/com.ibm.spss.modeler.help/nodes\\_treebuilding.htm](https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/nodes_treebuilding.htm). [Data uzyskania dostępu: 19 Czerwiec 2017].
- [35] IBM, „IBM Knowledge Center - Neural Net Node,” [Online]. Available: [https://www.ibm.com/support/knowledgecenter/en/SS3RA7\\_15.0.0/com.ibm.spss.modeler.help/trainnetnode\\_general.htm](https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/trainnetnode_general.htm). [Data uzyskania dostępu: 19 Czerwiec 2017].
- [36] Debian, „Światowa sieć mirrorów Debiana,” [Online]. Available: <https://www.debian.org/mirror/list>. [Data uzyskania dostępu: 8 Czerwiec 2017].
- [37] CDNPlanet, „CDN Finder Tool,” [Online]. Available: <https://www.cdnplanet.com/tools/cdnfinder/>. [Data uzyskania dostępu: 8 Czerwiec 2017].
- [38] Google, „WepPagetest,” [Online]. Available: <https://www.webpagetest.org/>. [Data uzyskania dostępu: 8 Czerwiec 2017].
- [39] Graham-Cumming J., „Tools for debugging, testing and using HTTP/2,” CloudFlare, 4 Grudzień 2015. [Online]. Available: <https://blog.cloudflare.com/tools-for-debugging-testing-and-using-http-2/>. [Data uzyskania dostępu: 8 Czerwiec 2017].
- [40] KeyCDN, „HTTP/2 Test,” [Online]. Available: <https://tools.keycdn.com/http2-test>.
- [41] BlueMM, „Excel formula to calculate distance between 2 latitude, longitude (lat/lon) points (GPS positions),” 6 Styczeń 2007. [Online]. Available: <http://bluemm.blogspot.com/2007/01/excel-formula-to-calculate-distance.html>. [Data uzyskania dostępu: 8 Czerwiec 2017].
- [42] „Distance Calculator,” [Online]. Available: <https://www.distancecalculator.net/>. [Data uzyskania dostępu: 8 Czerwiec 2017].

- [43] „HTTP Archive,” [Online]. Available: <http://httparchive.org/index.php>. [Data uzyskania dostępu: 3 Marzec 2017].
- [44] W3C, „HTTP/1.1: Method Definitions,” [Online]. Available: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [45] EventHelix.com Inc., „VisualEther Protocol Analyzer 7.0,” [Online]. Available: <https://www.eventhelix.com/VisualEther/#.WTqrDOvyiiM>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [46] Borzemski L. i Nowak Z., „WING: A Web Probing, Visualization, and Performance Analysis Service,” w *Web Engineering. ICWE 2004. Lecture Notes in Computer Science, vol 3140*, Berlin, Springer, 2004, pp. 601-602.
- [47] Microsoft, „Azure,” [Online]. Available: <https://azure.microsoft.com/pl-pl/>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [48] Amazon, „Amazon Web Services,” [Online]. Available: <https://aws.amazon.com/>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [49] „DigitalOcean,” [Online]. Available: <https://www.digitalocean.com/>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [50] PIONIER, „Pionier Cloud - O projekcie,” [Online]. Available: <https://cloud.pionier.net.pl/project>. [Data uzyskania dostępu: 9 Czerwiec 2017].
- [51] Groś S., „Controlling which congestion control algorithm is used in Linux,” 25 Grudzień 2012. [Online]. Available: <http://sgros.blogspot.com/2012/12/controlling-which-congestion-control.html>.
- [52] CAIDA, „Center for Applied Internet Data Analysis,” [Online]. Available: <http://www.caida.org/home/>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [53] CAIDA, „AS Rank: Help,” [Online]. Available: <http://as-rank.caida.org/?mode0=help&help=ranking#as-types>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [54] Team Cymru, „IP to ASN Lookup v1.0,” [Online]. Available: <https://asn.cymru.com/>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [55] „CIDR Report,” [Online]. Available: <http://www.cidr-report.org/as2.0/>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [56] Knasiecki M., „Grafy i ich reprezentacje,” algorytm.org, [Online]. Available: <http://www.algorytm.org/klasyczne/grafy-i-ich-reprezentacje.html>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [57] „PARIS TRACEROUTE,” [Online]. Available: <https://paris-traceroute.net/>. [Data uzyskania dostępu: 10 Czerwiec 2017].

- [58] Wikipedia, „Odwzorowanie walcowe równokątne,” [Online]. Available: [https://pl.wikipedia.org/wiki/Odwzorowanie\\_walcowe\\_r%C3%B3wnok%C4%85tnie](https://pl.wikipedia.org/wiki/Odwzorowanie_walcowe_r%C3%B3wnok%C4%85tnie). [Data uzyskania dostępu: 10 Czerwiec 2017].
- [59] TeleGeography, „Submarine Cable Map,” [Online]. Available: <http://www.submarinecablemap.com/#/>. [Data uzyskania dostępu: 10 Czerwiec 2017].
- [60] Leland W., „On the Self-similar Nature of Ethernet Traffic,” *IEEE/ACM Transactions on Networking*, tom 2, nr 1, pp. 1-15, 1994.
- [61] Chu C., „Hurst parameter estimate,” 11 Marzec 2008. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/19148-hurst-parameter-estimate>.
- [62] Aste T., „Generalized Hurst exponent,” 31 Styczeń 2013. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/30076-generalized-hurst-exponent>.
- [63] Davidson B., „Hurst exponent,” 1 Luty 2006. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/9842-hurst-exponent>.
- [64] StatSoft, „Składowe główne i analiza czynnikowa,” [Online]. Available: [http://www.statsoft.pl/textbook/stathome\\_stat.html?http%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstfacan.html](http://www.statsoft.pl/textbook/stathome_stat.html?http%3A%2F%2Fwww.statsoft.pl%2Ftextbook%2Fstfacan.html). [Data uzyskania dostępu: 14 Czerwiec 2017].

## Spis rysunków

Rysunek 2.1 Taksonomia przyczyn degradacji wydajności serwera .....	11
Rysunek 2.2 Schemat modelu TCP/IP .....	12
Rysunek 2.3 Prosta transakcja Web .....	14
Rysunek 2.4 Działanie HTTP/1.1. Po lewej: Wersja podstawowa. Po prawej: Z użyciem mechanizmu pipelining .....	14
Rysunek 2.5 Przykładowa mapa czasu pobierania w dniu 1 lipca 2008 r. o godzinie 6:00.....	21
Rysunek 2.6 Charakterystyki utworzonych klastrów.....	22
Rysunek 2.7 Predykcja przepustowości dla algorytmu Transform Regression dostępnym w pakiecie Intelligent Miner .....	23
Rysunek 2.8 Rozkład wartości mediany średniej szybkości transferu a RTT .....	23
Rysunek 3.1 Etapy Data Mining .....	25
Rysunek 3.2 Sieć Kohonena.....	26
Rysunek 3.3 Przykład drzewa decyzyjnego .....	28
Rysunek 3.4 Przykładowa sieć neuronowa .....	29
Rysunek 4.1 Mapa położenia wybranych serwerów .....	32
Rysunek 4.2 Przykład odpowiedzi z serwisu <i>iplocation.net</i> .....	33
Rysunek 4.3 Przykładowe sprawdzenie wydajności strony w serwisie <i>WebPagetest</i> .....	34
Rysunek 4.4 Okno wtyczki <i>DevTools</i> w przeglądarce <i>Chrome</i> .....	35
Rysunek 4.5 Budowa strony internetowej [Stan na 3.03.2017r.] .....	37
Rysunek 4.6 Zmiana średniego rozmiaru strony w czasie [Stan na 3.03.2017r.] .....	38
Rysunek 4.7 Przedziały rozmiaru stron [Stan na 3.03.2017r.].....	38

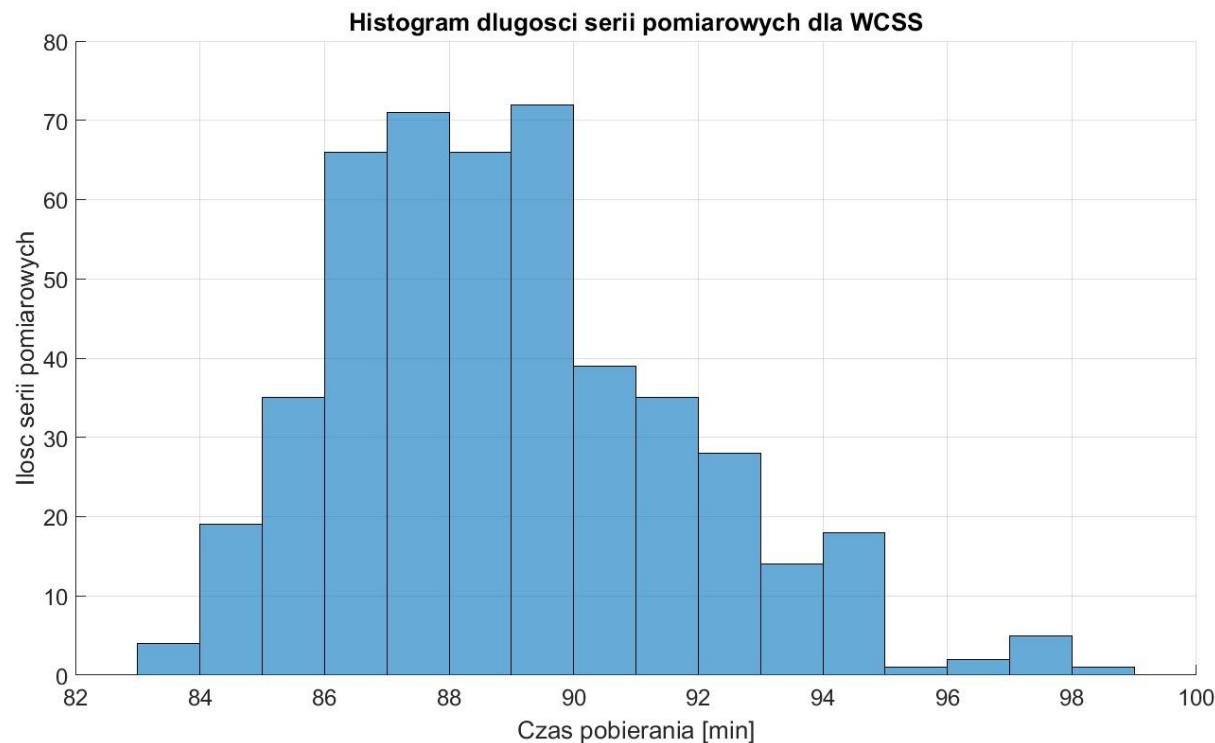
Rysunek 4.8 Korelacja do całkowitego załadowania strony [Stan na 3.03.2017r.] .....	39
Rysunek 4.9 Mapa połączeń PIONIER .....	43
Rysunek 5.1 Przykładowa odpowiedź serwisu CAIDA.....	46
Rysunek 5.2 Przykładowe wywołanie traceroute .....	47
Rysunek 5.3 Utworzony graf ścieżek AS.....	49
Rysunek 5.4 Mapa połączeń między systemami autonomicznymi.....	51
Rysunek 6.1 Wykresy stabilności działania serwerów .....	53
Rysunek 6.2 Histogram całkowitego czasu serii pomiarowych [Koral] .....	55
Rysunek 6.3 Wykres pudełkowy czasów pobierania dla każdego pliku [Koral].....	55
Rysunek 6.4 Interwały między pobraniami pliku 3MB na serwerze #47 [Koral] .....	56
Rysunek 7.1 Przykładowy przebieg przepustowości [WCSS].....	58
Rysunek 7.2 Zależność median RTT od przepustowości dla pliku 1.5MB [WCSS].....	59
Rysunek 7.3 Zależność median RTT od przepustowości dla pliku 1.5MB (logarytmicznie) [WCSS] .....	59
Rysunek 7.4 Poprowadzenie linii regresji mediany RTT i przepustowości dla każdej wielkości pliku [WCSS] .....	60
Rysunek 7.5 Wykres zależności median RTT od dystansu geograficznego [WCSS] .....	62
Rysunek 7.6 Wykres zależności median RTT od ilości skoków IP [WCSS] .....	62
Rysunek 7.7 Wykres zależności median RTT od ilości skoków AS [WCSS].....	62
Rysunek 7.8 Przebieg przepustowości dla serwera w Czechach [Plik 3MB] .....	63
Rysunek 7.9 Wykres pudełkowy przepustowości dla serwera czeskiego [Plik 3MB] .....	64
Rysunek 7.10 Przebiegi przepustowości TCP dla serwera czeskiego [6 plik, ostatnie 2 tygodnie] .....	64
Rysunek 8.1 Przebiegi RTT dla serwera #47 [Koral] .....	69
Rysunek 8.2 Przebiegi przepustowości dla pliku 6MB [Koral, serwer #47] .....	72
Rysunek 8.3 Przebiegi przepustowości dla pliku 6MB [WCSS, serwer #47] .....	72
Rysunek 8.4 Przebiegi przepustowości dla pliku 6MB [PCSS, serwer #47] .....	72
Rysunek 8.5 Przebiegi przepustowości dla pliku 6MB [Koral, serwer #9] .....	73
Rysunek 8.6 Przebiegi przepustowości dla pliku 6MB [WCSS, serwer #9] .....	73
Rysunek 8.7 Przebiegi przepustowości dla pliku 6MB [PCSS, serwer #9] .....	73
Rysunek 8.8 Wartość współczynnika Hursta dla okna przesuwnego (5) [serwer #47] .....	74
Rysunek 8.9 Wartość współczynnika Hursta dla okna przesuwnego (10) [serwer #47] .....	74
Rysunek 8.10 Wartość współczynnika Hursta dla okna przesuwnego (20) [serwer #47] .....	74
Rysunek 8.11 Strumień dla pierwszego zadania predykcji [SPSS Modeler].....	79
Rysunek 8.12 Okno określania typów i roli argumentów [SPSS Modeler] .....	79
Rysunek 8.13 Grupowanie ( <i>binning</i> ) wartości przepustowości [SPSS Modeler].....	80
Rysunek 8.14 Grupowanie ( <i>binning</i> ) wartości RTT [SPSS Modeler] .....	80
Rysunek 8.15 Określenie użytecznych danych [SPSS Modeler] .....	80
Rysunek 8.16 Wybór argumentów do budowy klastrów [SPSS Modeler] .....	81
Rysunek 8.17 Wybór argumentów w metodach klasyfikacji lub regresji [SPSS Modeler].....	81
Rysunek 9.1 Ważność algorytmów bez grupowania [Koral, wszystkie serwery] .....	82
Rysunek 9.2 Wyniki technik regresji dla wszystkich argumentów [Koral, wszystkie serwery] .....	83
Rysunek 9.3 Wyniki technik regresji dla wybranych argumentów [Koral, wszystkie serwery] .....	83
Rysunek 9.4 Ważność predyktorów dla otrzymanych modeli regresji [Koral, wszystkie serwery] .....	84
Rysunek 9.5 Wyniki regresji dla sieci neuronowej [Koral, wszystkie serwery].....	84
Rysunek 9.6 Grupowanie przepustowości TCP [Koral, wszystkie serwery] .....	84
Rysunek 9.7 Grupowanie RTT [Koral, wszystkie serwery] .....	85

Rysunek 9.8 Wyniki predykcji przedziałów przepustowości [Koral, Wszystkie serwery] .....	85
Rysunek 9.9 Ważność predyktorów dla klasyfikacji przedziału przepustowości [Koral, wszystkie serwery] .....	86
Rysunek 9.10 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [Koral, Wszystkie serwery] .....	86
Rysunek 9.11 Stworzone klastry wydajności [Koral, wszystkie serwery] .....	87
Rysunek 9.12 Przykład grupy w klastrze [Koral, wszystkie serwery] .....	87
Rysunek 9.13 Wyniki predykcji stworzonych klastrów [Koral, Wszystkie serwery] .....	88
Rysunek 9.14 Ważność algorytmów bez grupowania [Koral, serwer #47] .....	88
Rysunek 9.15 Wyniki technik regresji dla wszystkich argumentów [Koral, serwer #47] .....	89
Rysunek 9.16 Grupowanie przepustowości TCP [Koral, serwer #47] .....	89
Rysunek 9.17 Wyniki predykcji przedziałów przepustowości [Koral, serwer #47] .....	89
Rysunek 9.18 Stworzone klastry wydajności [Koral, serwer #47] .....	90
Rysunek 9.19 Wyniki predykcji stworzonych klastrów [Koral, serwer #47] .....	90
Rysunek 9.20 Wyniki działania metody C&RT [Koral, serwer #47] .....	91
Rysunek 9.21 Wyniki technik regresji dla wszystkich argumentów [Koral, serwer #9] .....	91
Rysunek 9.22 Grupowanie przepustowości TCP [Koral, serwer #9] .....	92
Rysunek 9.23 Wyniki predykcji przedziałów przepustowości [Koral, serwer #9] .....	92
Rysunek 9.24 Stworzone klastry wydajności [Koral, serwer #9] .....	92
Rysunek 9.25 Wyniki predykcji stworzonych klastrów [Koral, serwer #9] .....	93
Rysunek 9.26 Wyniki technik regresji dla wszystkich argumentów [WCSS, wszystkie serwery] .....	93
Rysunek 9.27 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [WCSS, wszystkie serwery] .....	94
Rysunek 9.28 Stworzone klastry wydajności [WCSS, wszystkie serwery] .....	94
Rysunek 9.29 Wyniki predykcji stworzonych klastrów dla RTT i dystansu [WCSS, wszystkie serwery] .....	95
Rysunek 9.30 Grupowanie przepustowości TCP [WCSS, serwer #9] .....	96
Rysunek 9.31 Wyniki predykcji przedziałów przepustowości [WCSS, serwer #9] .....	96
Rysunek 9.32 Wyniki technik regresji dla wybranych argumentów [PCSS, wszystkie serwery] .....	97
Rysunek 9.33 Wyniki predykcji przedziałów przepustowości przy użyciu RTT i dystansu geograficznego [PCSS, Wszystkie serwery] .....	97
Rysunek 9.34 Wyniki predykcji stworzonych klastrów [PCSS, wszystkie serwery] .....	98
Rysunek 9.35 Rozpoznane przypadki dla algorytmu C5 [PCSS, wszystkie serwery] .....	98
Rysunek 9.36 Wyniki technik regresji dla wszystkich argumentów [PCSS, serwer #47] .....	99

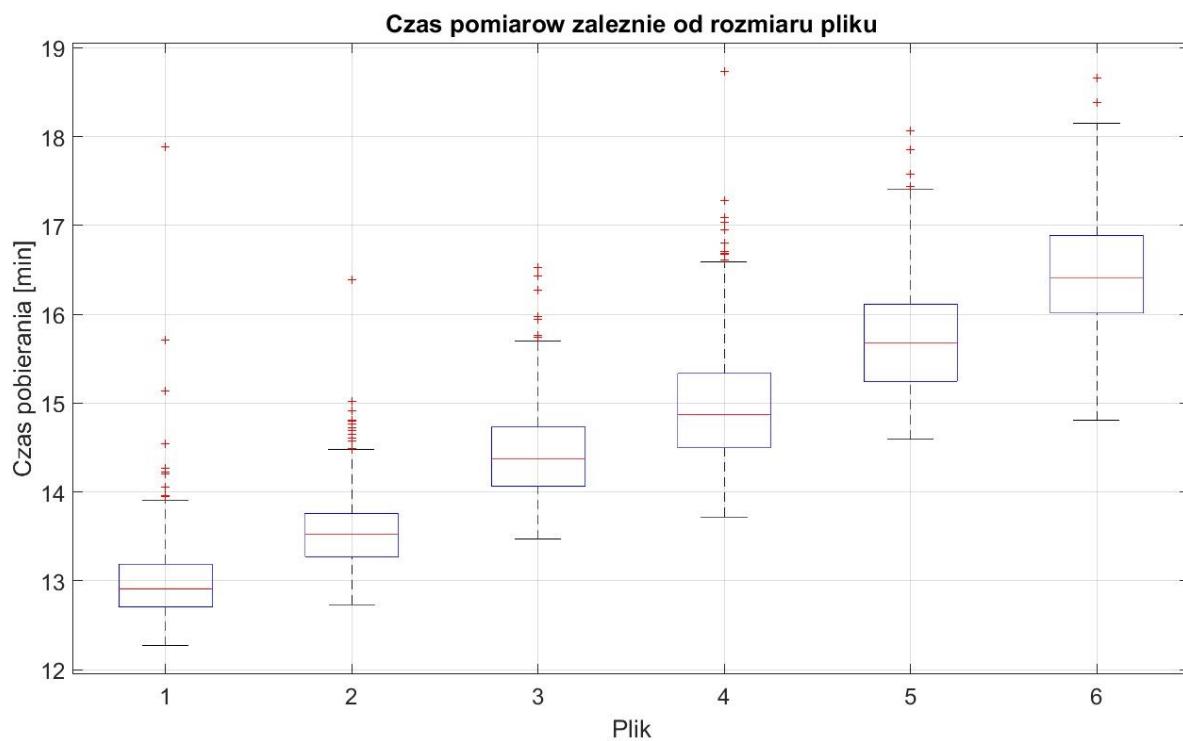
## Spis tabel

Tabela 4.1 Wybrane serwery (wycinek).....	36
Tabela 5.1 Zmiany ścieżki AS dla serwera w Bangladeszu.....	52
Tabela 7.1 Dopasowanie linii regresji ( $R^2$ ) median RTT i przepustowości dla agentów pomiarowych.....	60
Tabela 7.2 Kąt nachylenia [ $^\circ$ ] funkcji liniowej na wykresie logarytmicznym.....	61
Tabela 7.3 Współczynniki korelacji Pearsona dla serwera czeskiego [6 plik, ostatnie 2 tygodnie] .....	65
Tabela 7.4 Wydajność sieci w Politechnice Wrocławskiej na przełomie kolejnych lat .....	66
Tabela 8.1 Zmiana współczynnika Hursta dla okna rozszerzającego się [Serwer #47].....	71
Tabela 8.2 Zmiana współczynnika Hursta dla okna rozszerzającego się [Serwer #9].....	71

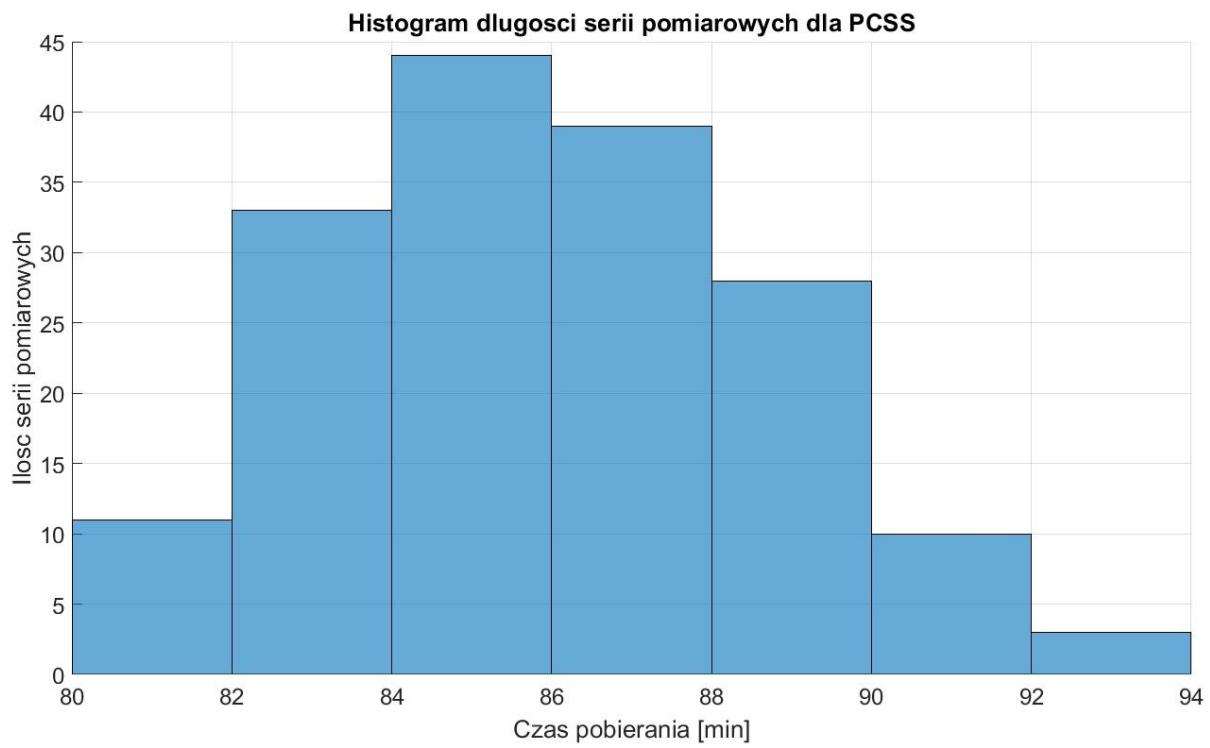
## Załączniki



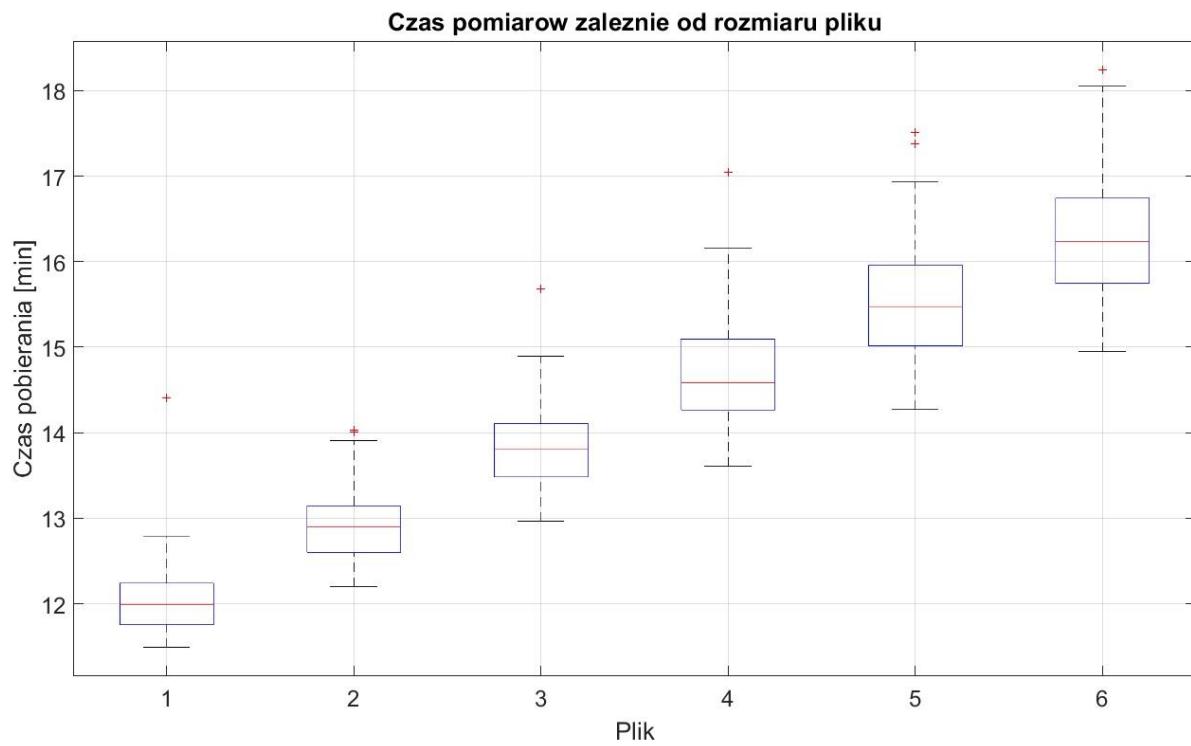
Rysunek 10.1 Histogram całkowitego czasu serii pomiarowych [WCSS]



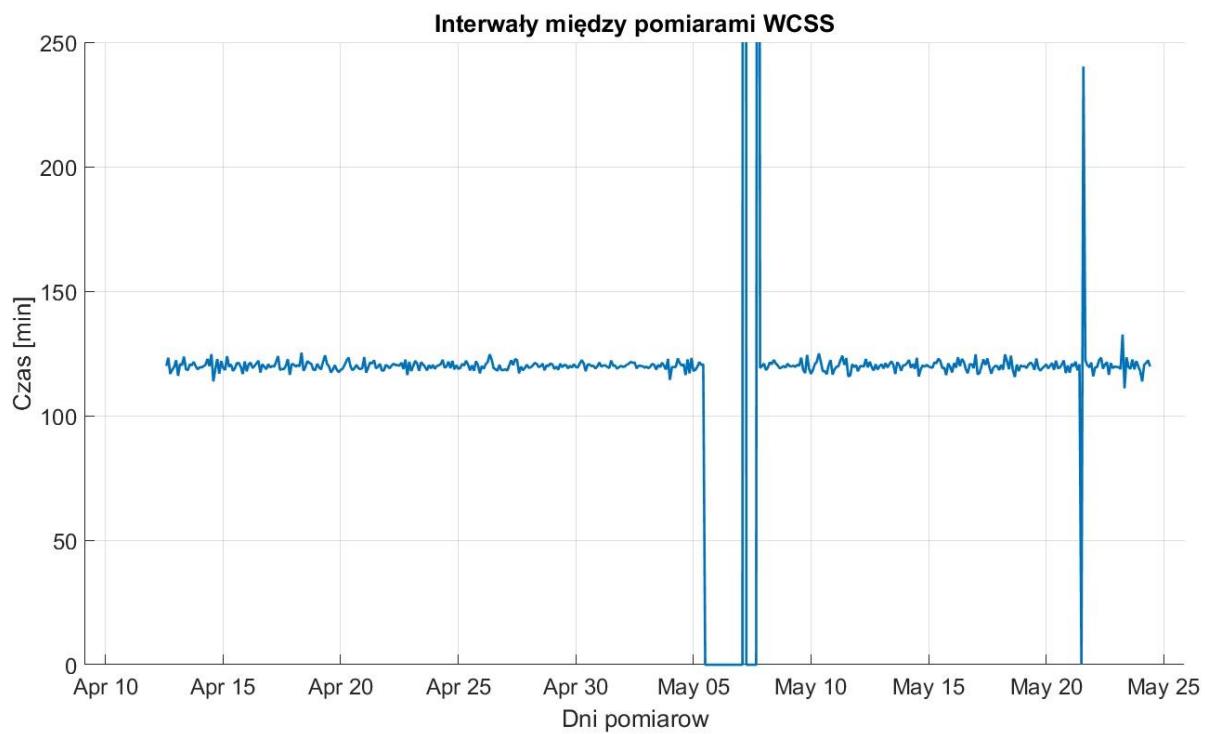
Rysunek 10.2 Wykres pudełkowy czasów pobierania dla każdego pliku [WCSS]



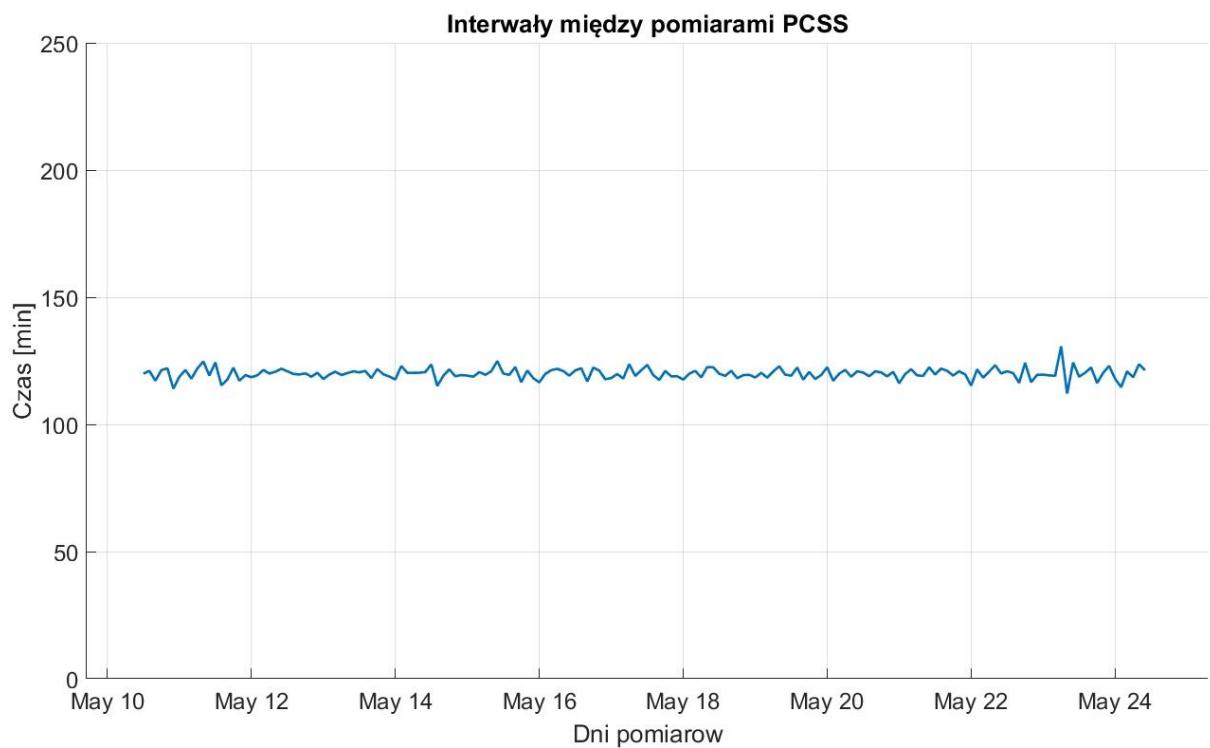
Rysunek 10.3 Histogram całkowitego czasu serii pomiarowych [PCSS]



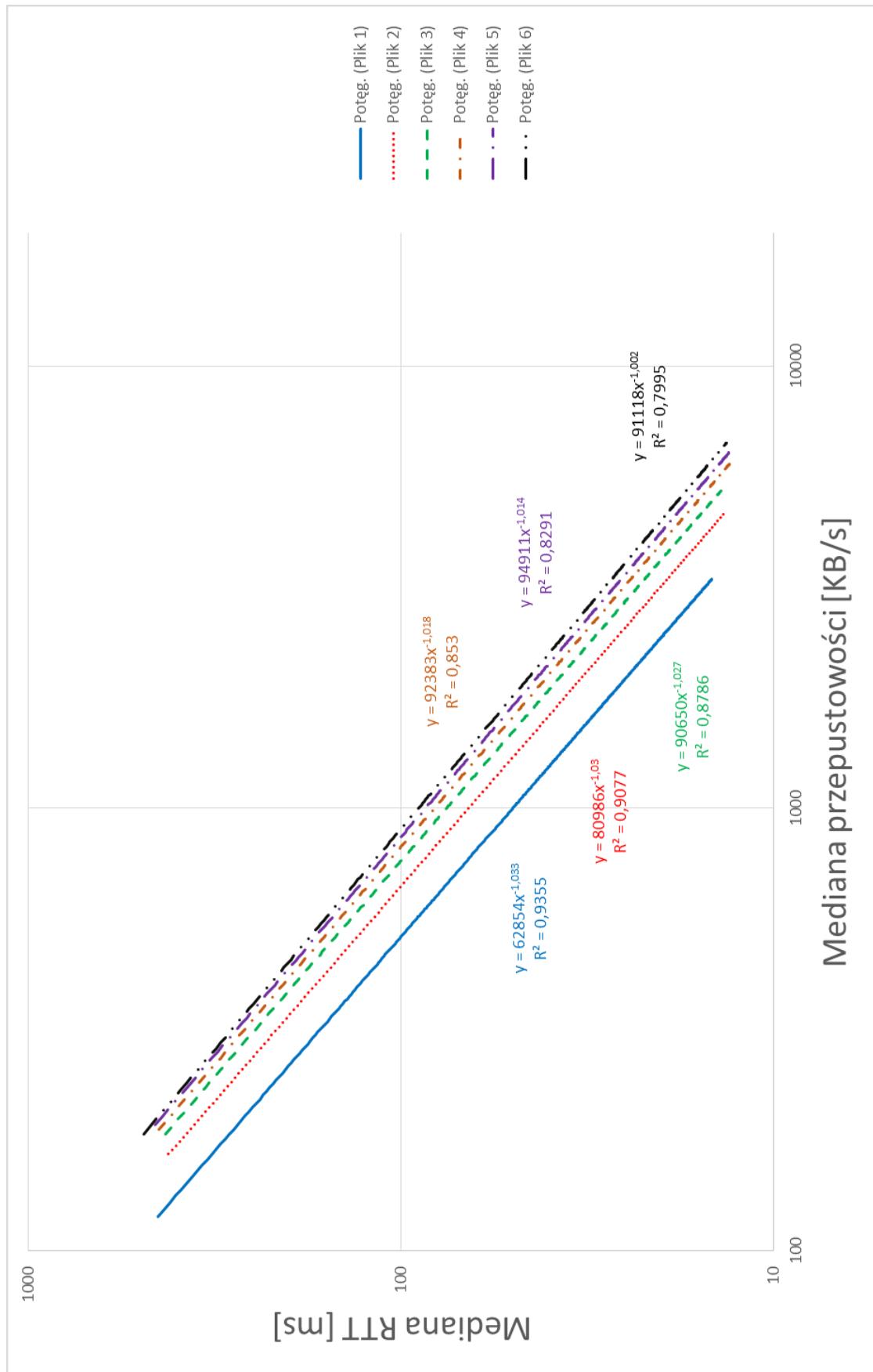
Rysunek 10.4 Wykres pudełkowy czasów pobierania dla każdego pliku [PCSS]



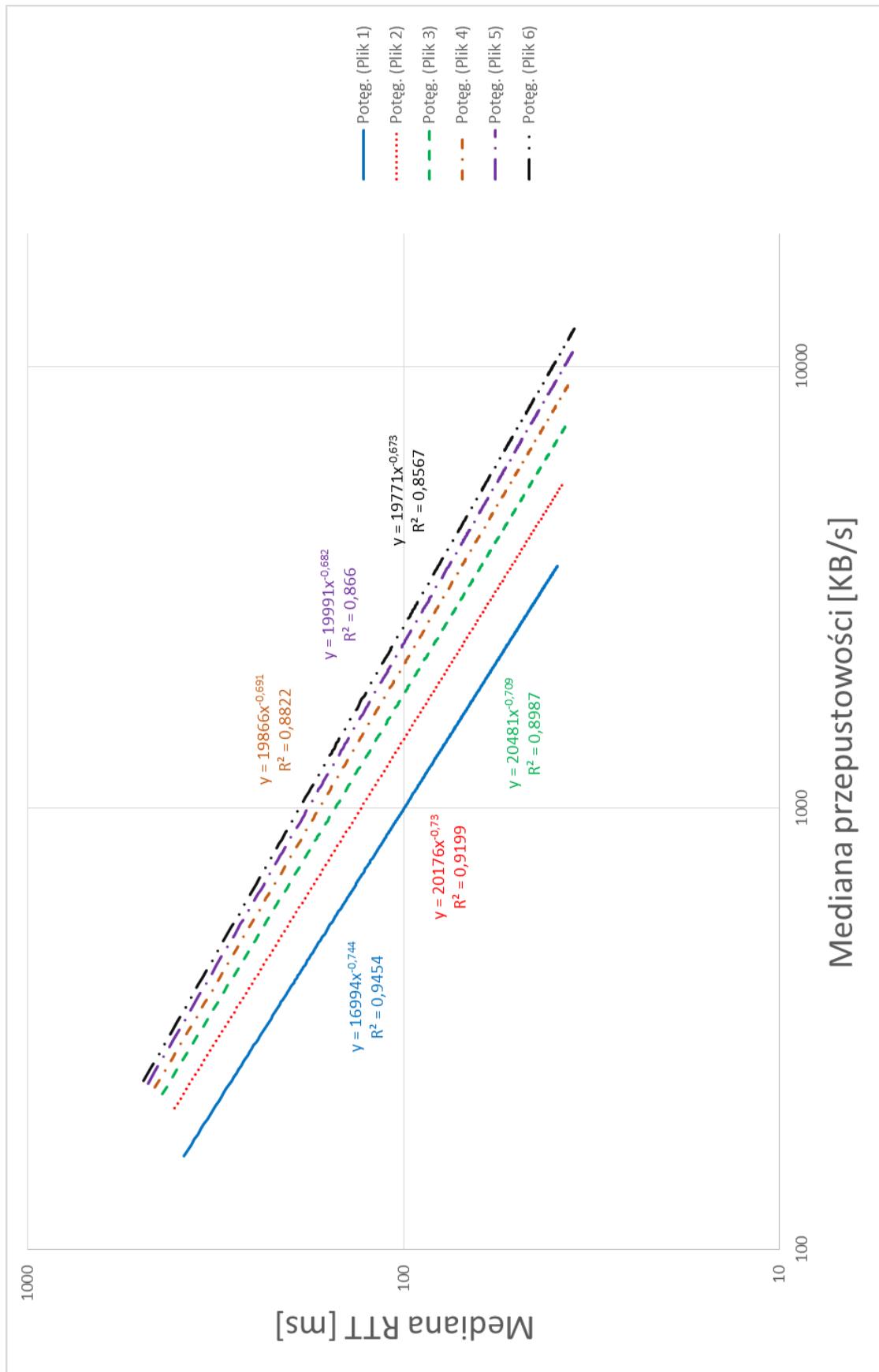
Rysunek 10.5 Interwały między pobraniami pliku 3MB na serwerze #47 [WCSS]



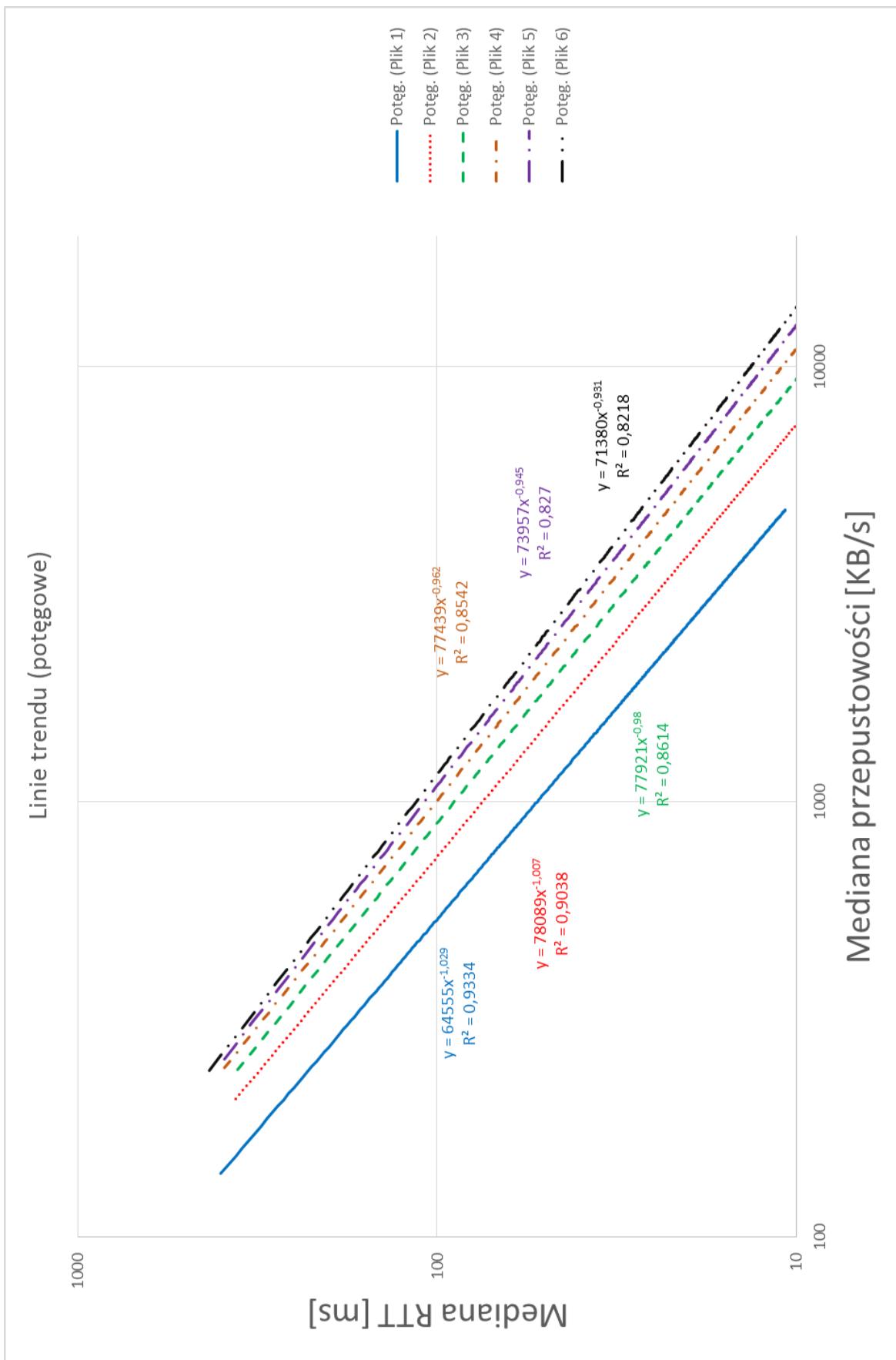
Rysunek 10.6 Interwały między pobraniami pliku 3MB na serwerze #47 [PCSS]



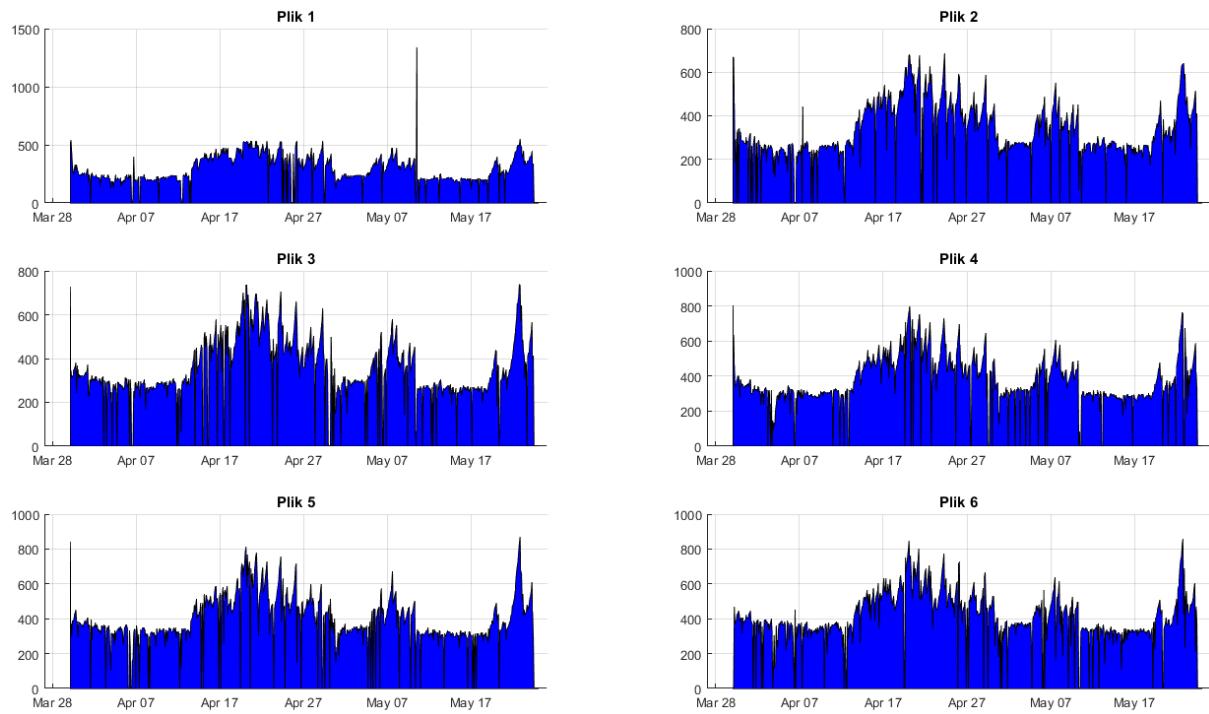
Rysunek 10.7 Poprowadzenie linii regresji mediany RTT i przepustowości dla każdej wielkości pliku [Koral]



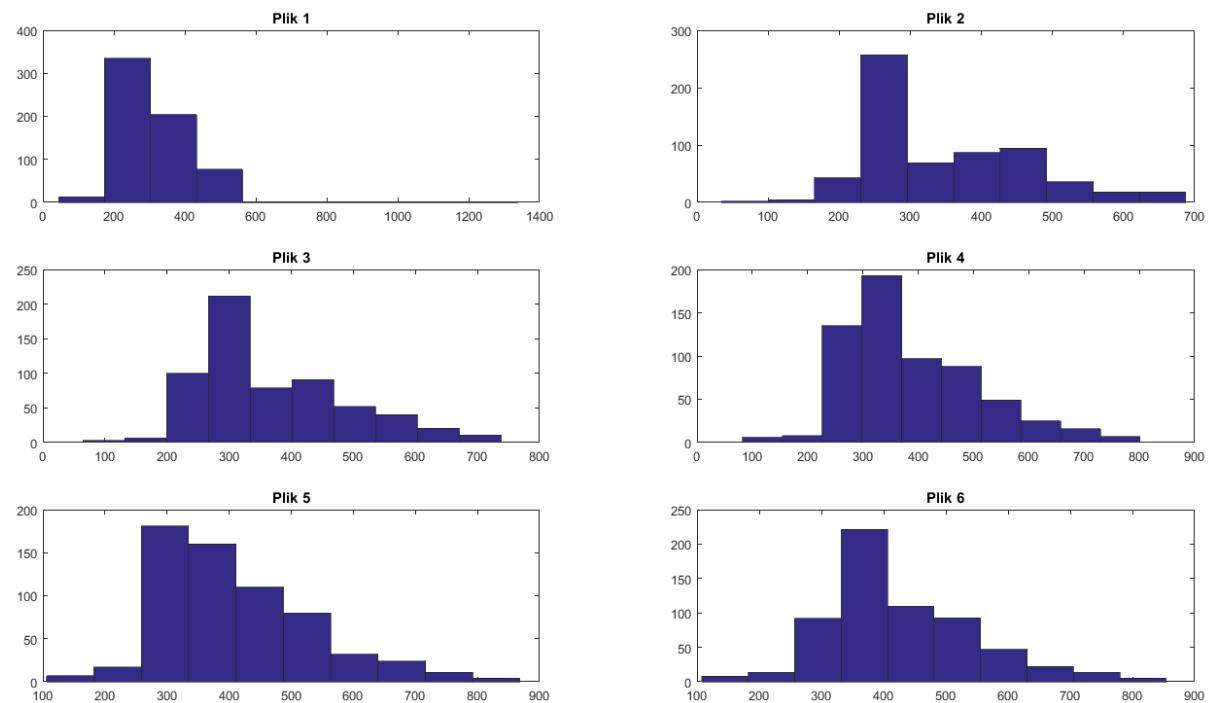
Rysunek 10.8 Poprowadzenie linii regresji mediany RTT i przepustowości dla każdej wielkości pliku [WCSS]



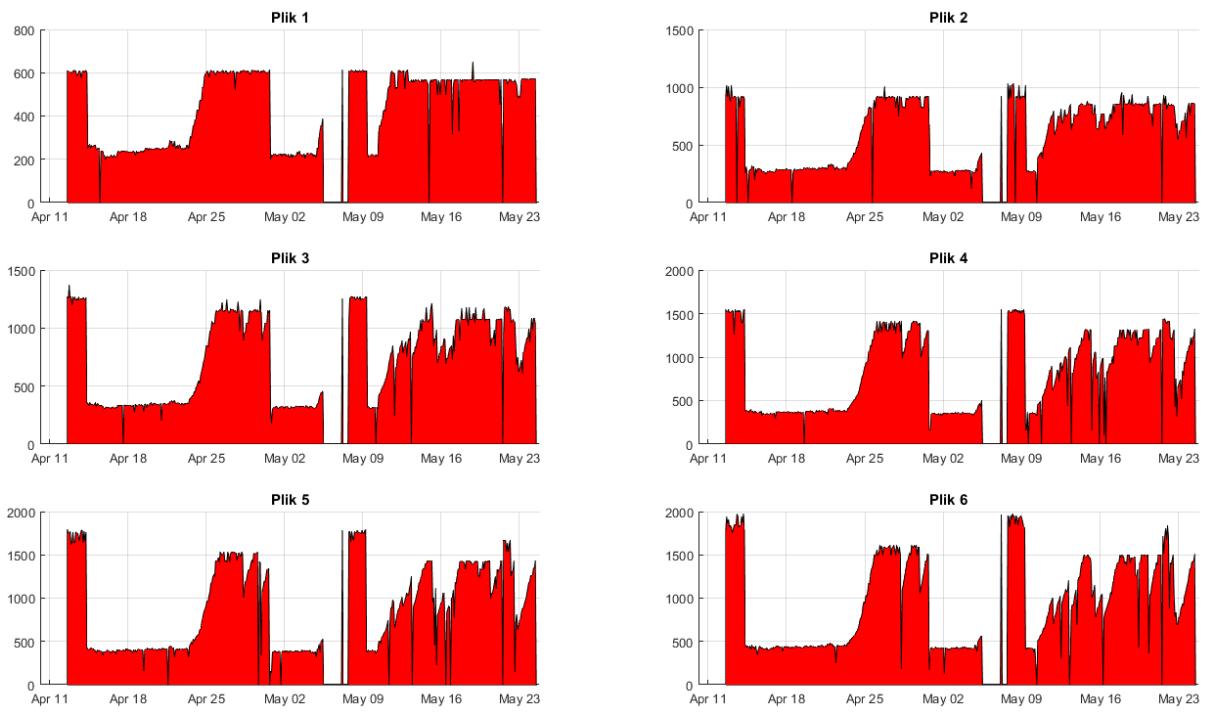
Rysunek 10.9 Poprowadzenie linii regresji mediany RTT i przepustowości dla każdej wielkości pliku [PCSS]



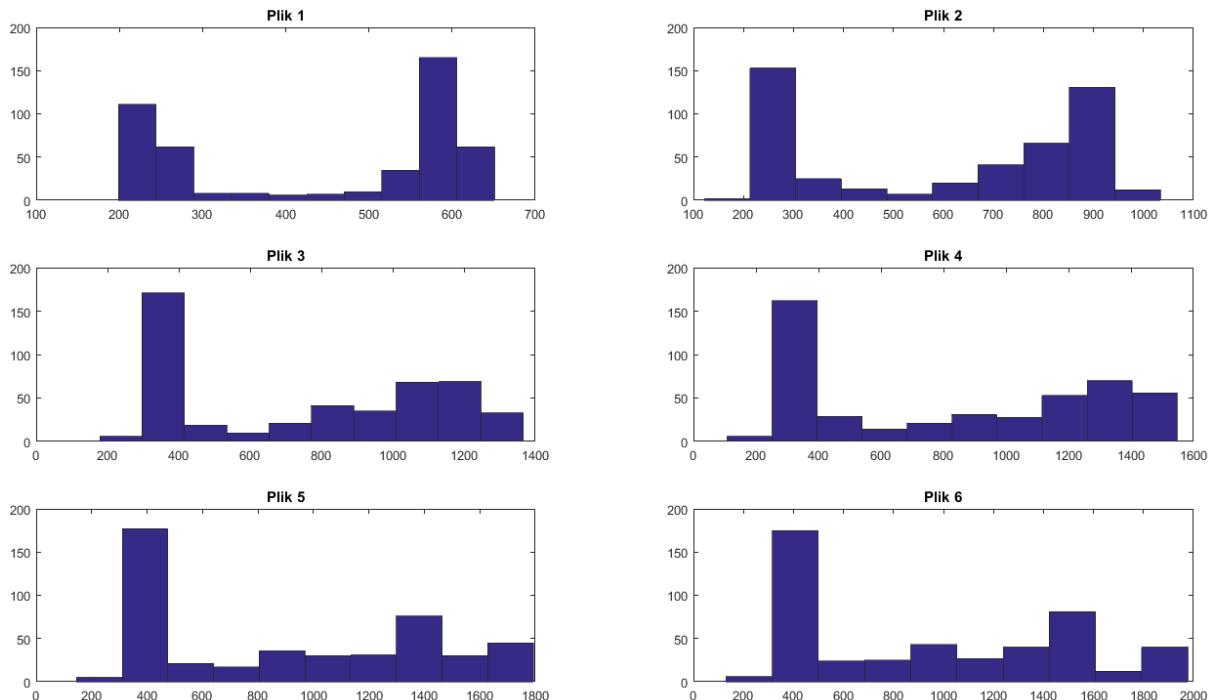
Rysunek 10.10 Przebiegi przepustowości dla serwera #47 zależnie od wielkości pliku [Koral]



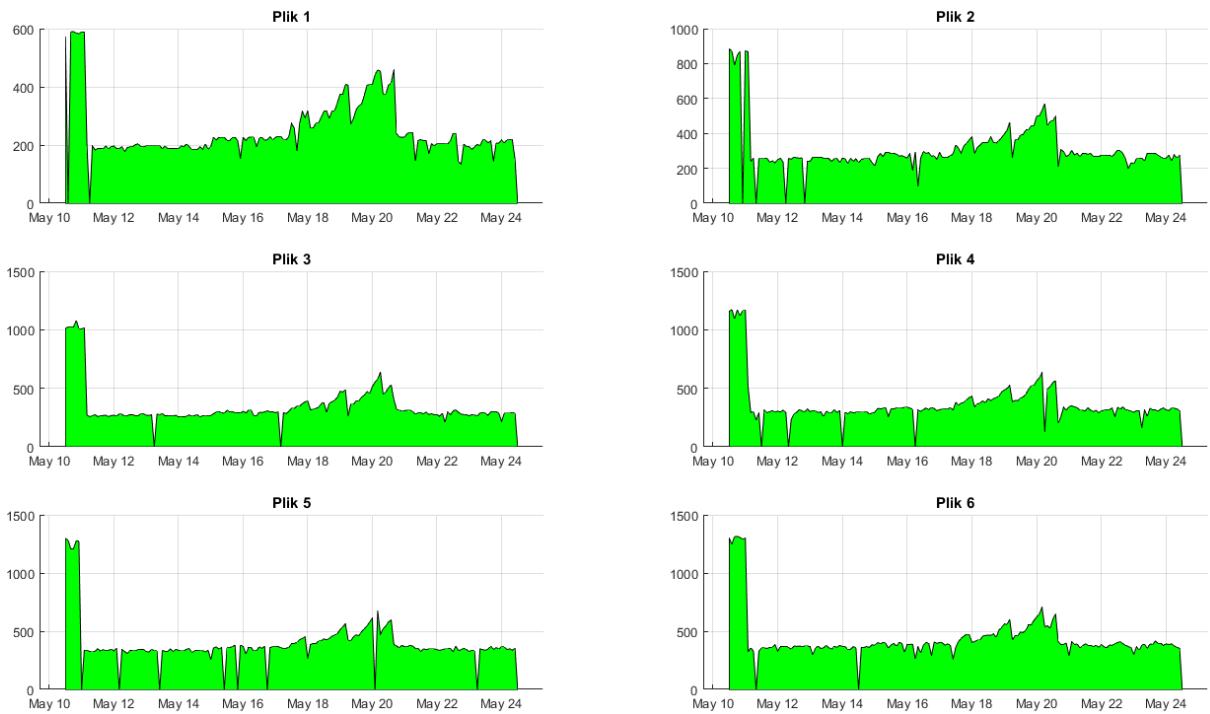
Rysunek 10.11 Histogramy przepustowości dla serwera #47 zależnie od wielkości pliku – zjawisko długogonowości [Koral]



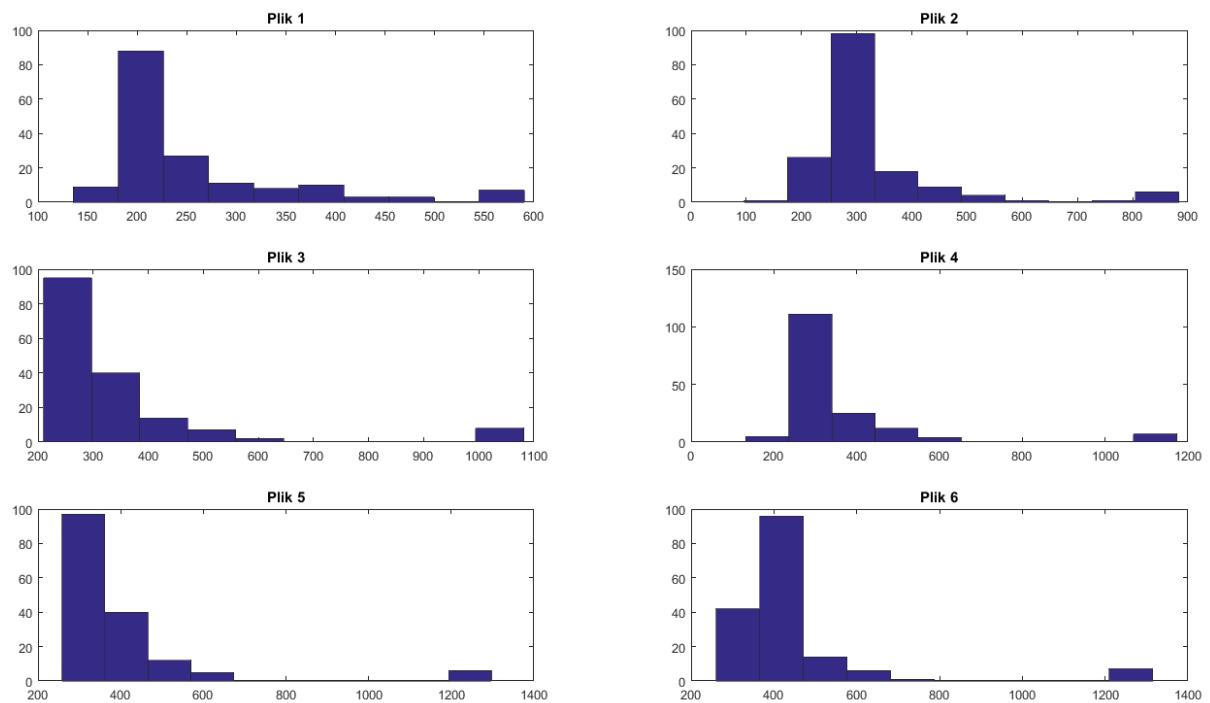
Rysunek 10.12 Przebiegi przepustowości dla serwera #47 zależnie od wielkości pliku [WCSS]



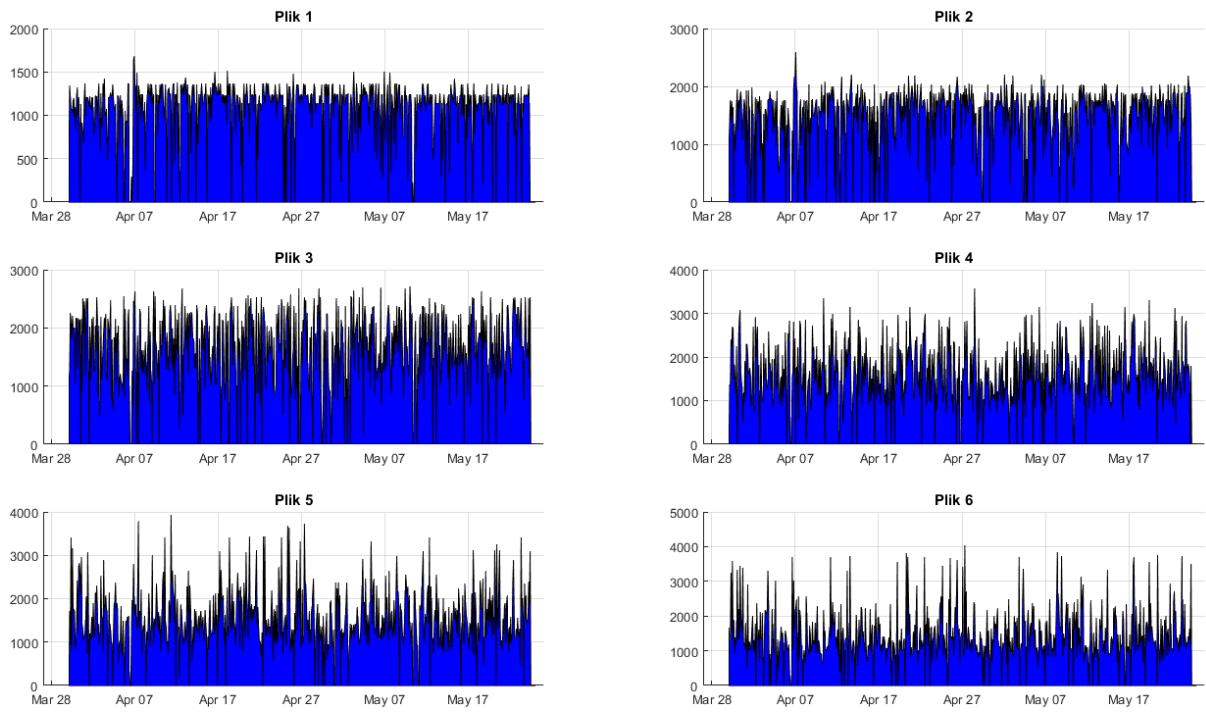
Rysunek 10.13 Histogramy przepustowości dla serwera #47 zależnie od wielkości pliku [WCSS]



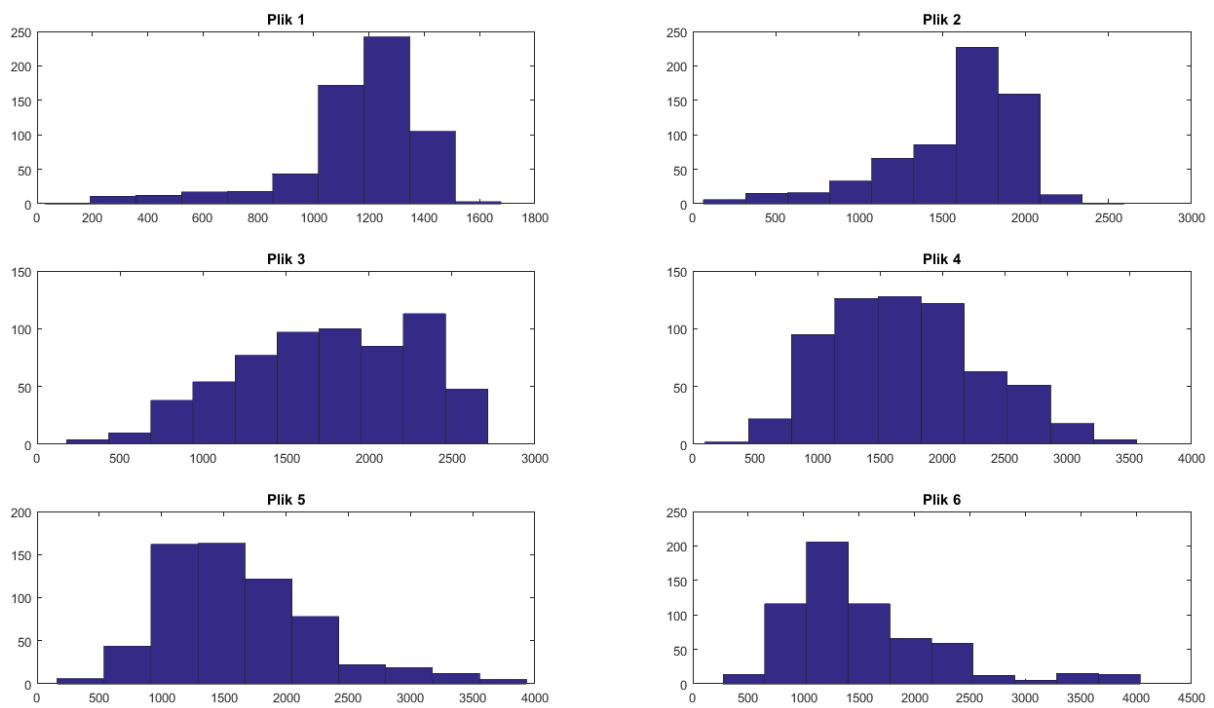
Rysunek 10.14 Przebiegi przepustowości dla serwera #47 zależnie od wielkości pliku [PCSS]



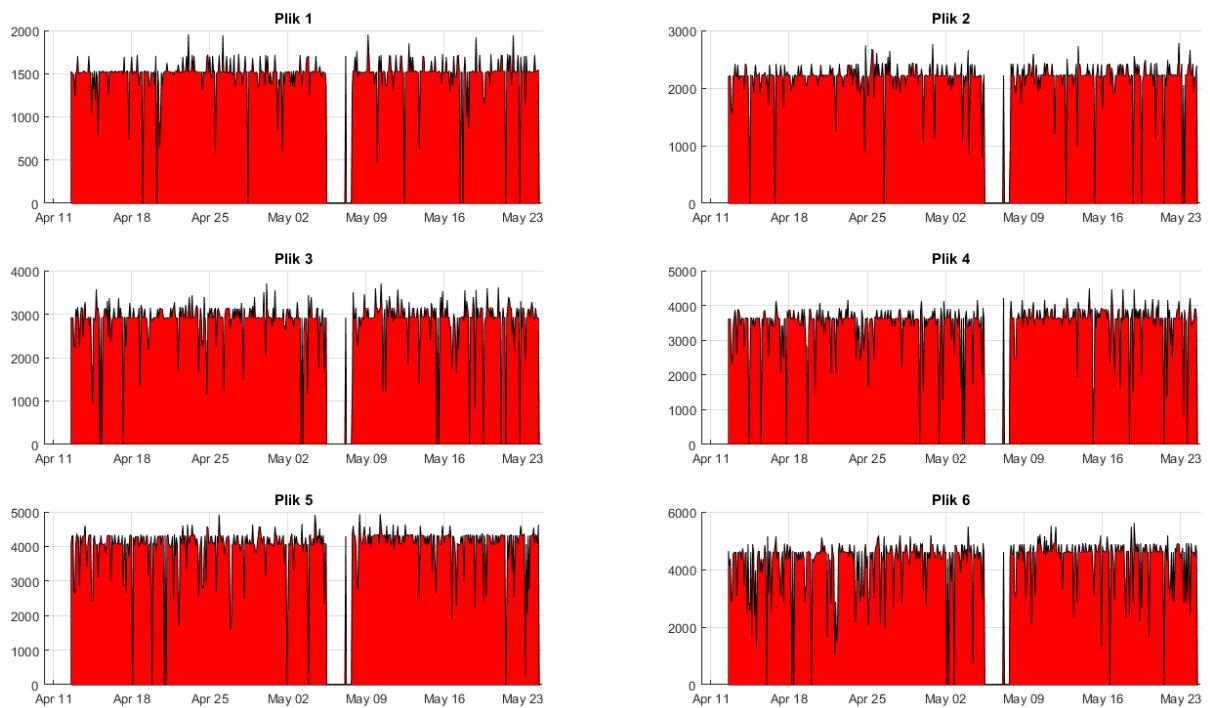
Rysunek 10.15 Histogramy przepustowości dla serwera #47 zależnie od wielkości pliku [PCSS]



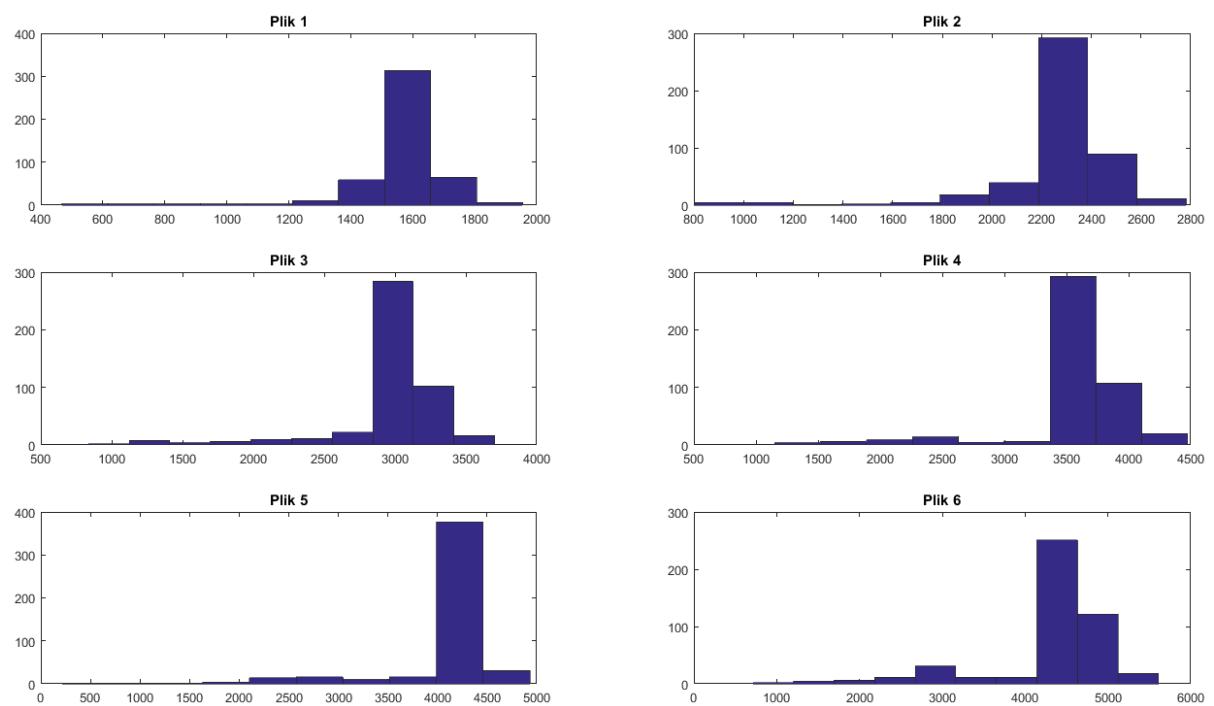
Rysunek 10.16 Przebiegi przepustowości dla serwera #9 zależnie od wielkości pliku [Koral]



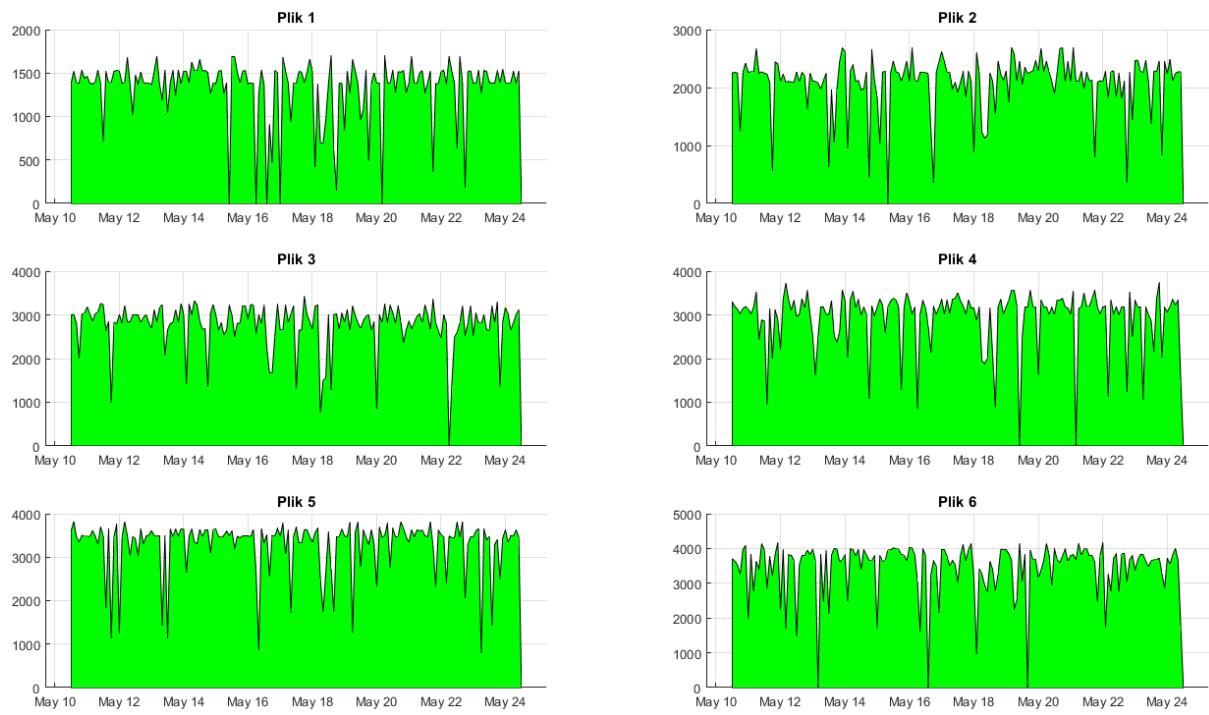
Rysunek 10.17 Histogramy przepustowości dla serwera #9 zależnie od wielkości pliku [Koral]



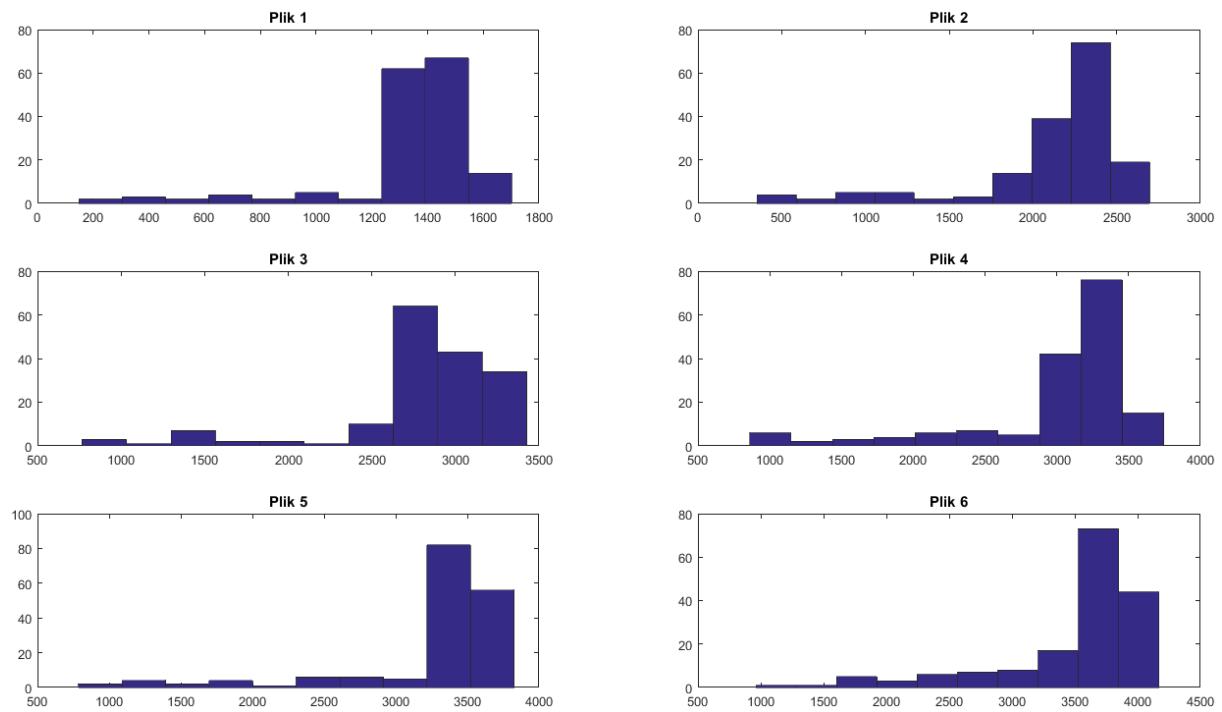
Rysunek 10.18 Przebiegi przepustowości dla serwera #9 zależnie od wielkości pliku [WCSS]



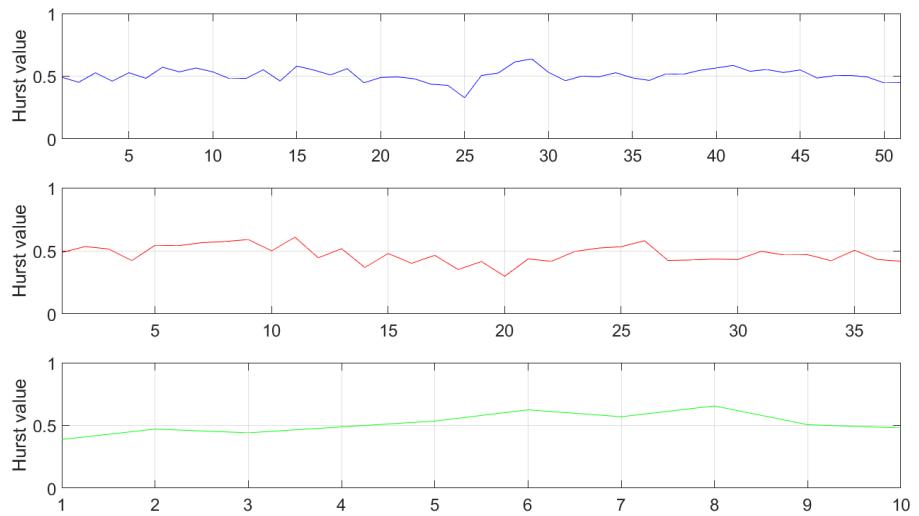
Rysunek 10.19 Histogramy przepustowości dla serwera #9 zależnie od wielkości pliku [WCSS]



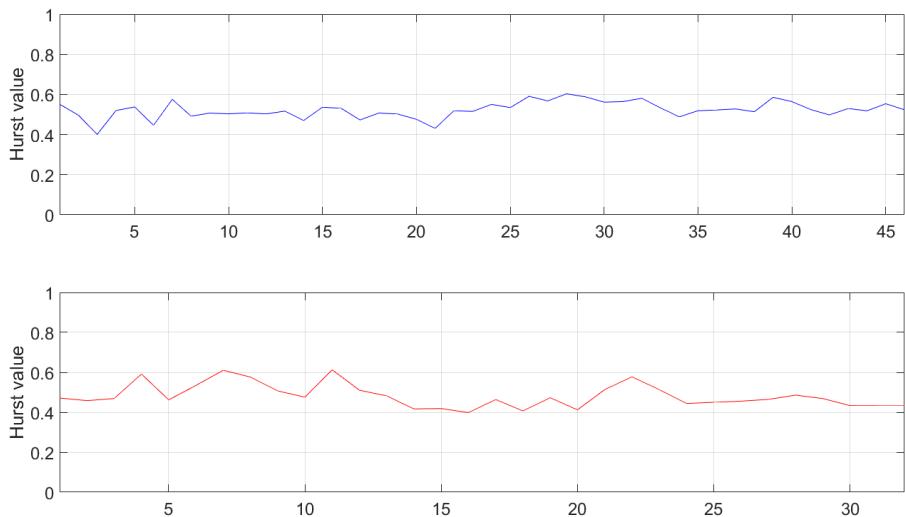
Rysunek 10.20 Przebiegi przepustowości dla serwera #9 zależnie od wielkości pliku [PCSS]



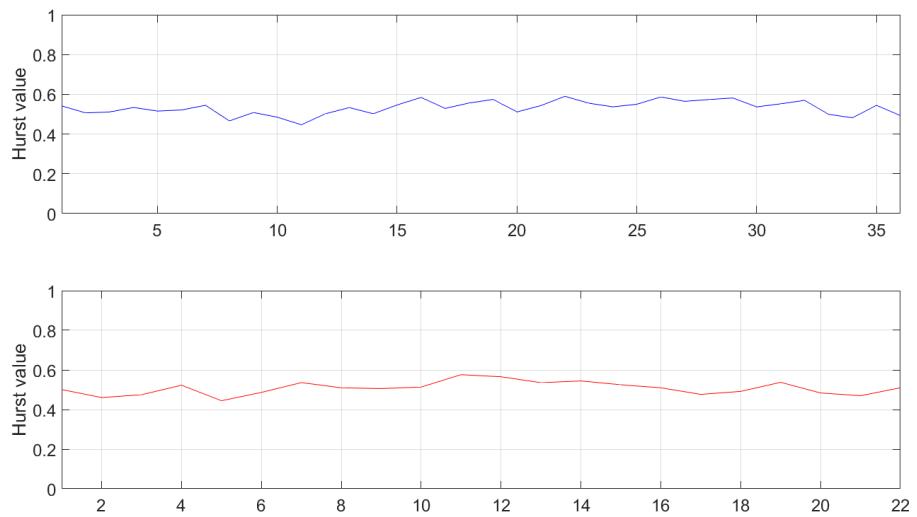
Rysunek 10.21 Histogramy przepustowości dla serwera #9 zależnie od wielkości pliku [PCSS]



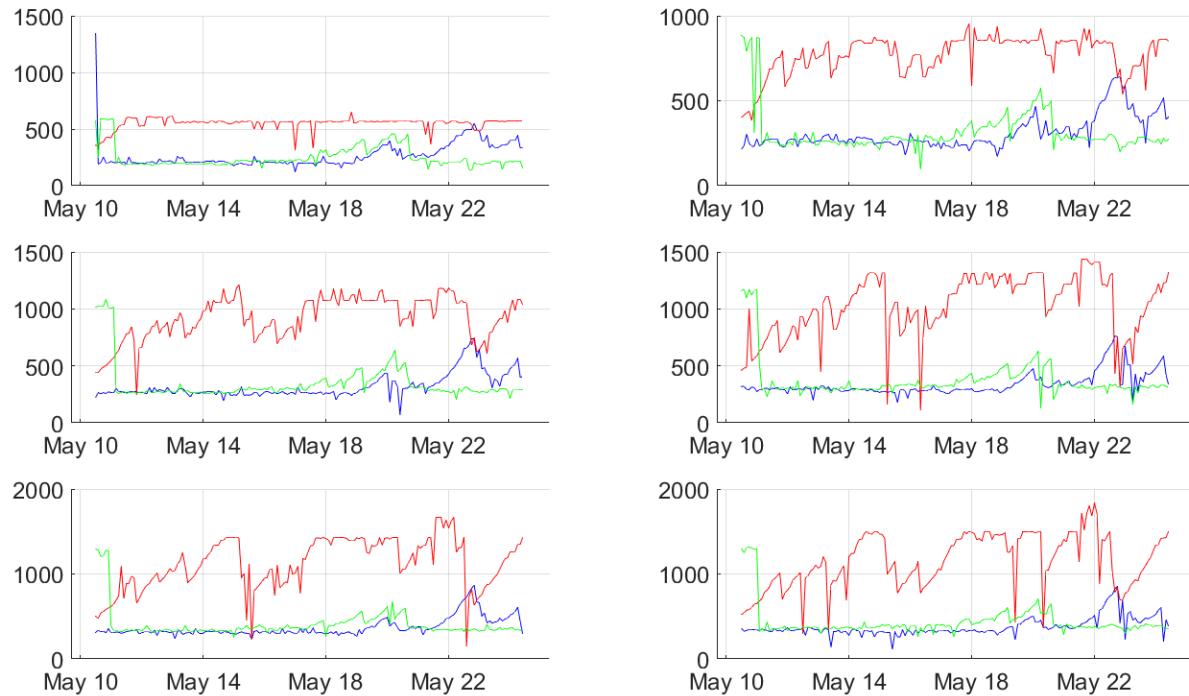
Rysunek 10.22 Wartość współczynnika Hursta dla okna przesuwnego (5) [serwer #9]



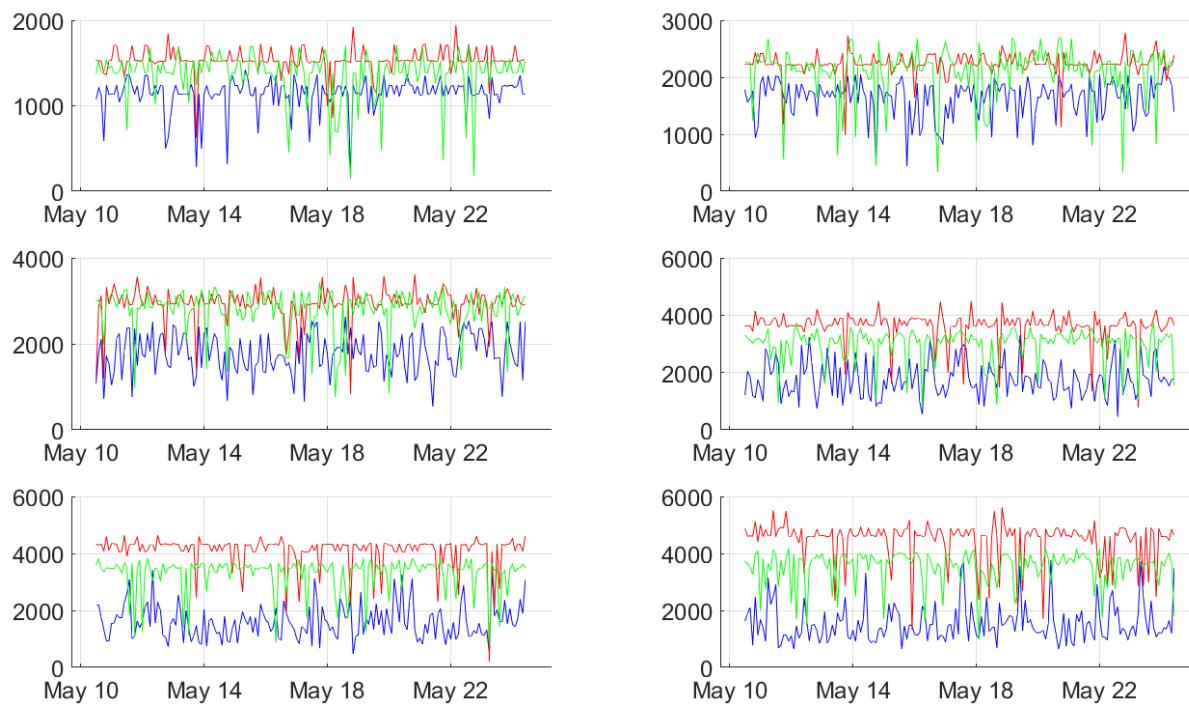
Rysunek 10.23 Wartość współczynnika Hursta dla okna przesuwnego (10) [serwer #9]



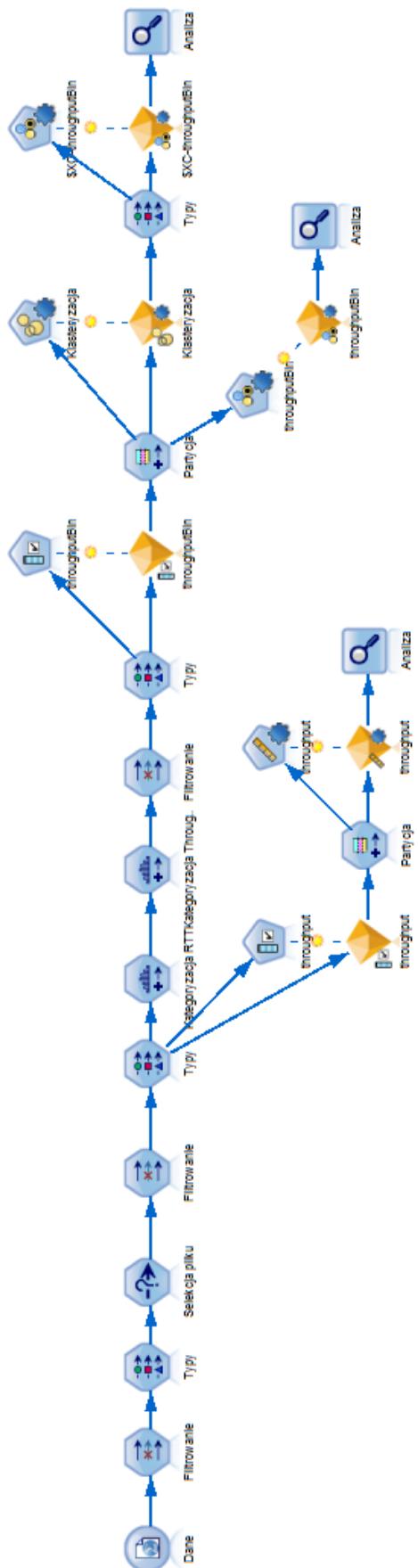
Rysunek 10.24 Wartość współczynnika Hursta dla okna przesuwnego (20) [serwer #9]



Rysunek 10.25 Przebiegi przepustowości wszystkich agentów dla serwera #47 w ostatnich dwóch tygodniach pomiarów



Rysunek 10.26 Przebiegi przepustowości wszystkich agentów dla serwera #9 w ostatnich dwóch tygodniach pomiarów



Rysunek 10.27 Strumień dla pierwszego zadania predykcji [SPSS Modeler]

	Ranga	Zmienna	Pomiar	Ważność	Wartość
<input checked="" type="checkbox"/>	1	rttBin	Nominalny	Ważne	1,0
<input checked="" type="checkbox"/>	2	distance	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	3	networkTier	Nominalny	Ważne	1,0
<input checked="" type="checkbox"/>	4	asType	Nominalny	Ważne	1,0
<input checked="" type="checkbox"/>	5	first	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	6	dns	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	7	asRank	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	8	asTransit	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	9	hour	Ilościowy	Ważne	1,0
<input checked="" type="checkbox"/>	10	day	Nominalny	Ważne	0,974

Rysunek 10.28 Ważność argumentów podzielonych na przedziały [Koral, wszystkie serwery]

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		C5.1	< 1	63,418	2
<input checked="" type="checkbox"/>		C&RT 1	< 1	62,045	2
<input checked="" type="checkbox"/>		CHAID 1	< 1	60,638	2
<input checked="" type="checkbox"/>		Quest 1	< 1	58,928	2
<input checked="" type="checkbox"/>		Sieć Bayesa 1	< 1	58,020	2
<input checked="" type="checkbox"/>		Sieci neuronowe 1	< 1	57,858	2
<input checked="" type="checkbox"/>		Regresja logistyczna	< 1	57,454	2
<input checked="" type="checkbox"/>		Analiza dyskryminacyjna	< 1	47,136	1

Rysunek 10.29 Klasyfikacja klastrów za pomocą RTT i dystansu geograficznego [Koral, Wszystkie serwery]

	Ranga	Zmienna	Pomiar	Ważność	Wartość
<input checked="" type="checkbox"/>	1	first	Ilościowy	Ważne	0,999
<input checked="" type="checkbox"/>	2	dns	Ilościowy	Ważne	0,974
<input checked="" type="checkbox"/>	3	day	Nominalny	Brzegowe	0,947
<input checked="" type="checkbox"/>	4	hour	Ilościowy	Nieważne	0,254
<input checked="" type="checkbox"/>	5	rtt	Ilościowy	Nieważne	0,24

Rysunek 10.30 Ważność dla stworzonych przedziałów [Koral, serwer #47]

Kategoria	Dolna	Góra
1	$\geq 101,51793144$	$< 650,47653858$
2	$\geq 650,47653858$	$< 1333,76681467$
3	$\geq 1333,76681467$	$< 3690,98560236$
4	$\geq 3690,98560236$	$\leq 13792,89146652$

Rysunek 10.31 Przedziały przepustowości [WCSS, wszystkie serwery]

Kategoria	Dolna	Góra
1	$\geq 45,825$	$< 65,605$
2	$\geq 65,605$	$< 73,591$
3	$\geq 73,591$	$< 89,195$
4	$\geq 89,195$	$< 122,138$
5	$\geq 122,138$	$< 196,105$
6	$\geq 196,105$	$< 278,001$
7	$\geq 278,001$	$< 328,091$
8	$\geq 328,091$	$\leq 1746,979$

Rysunek 10.32 Przedziały RTT [WCSS, wszystkie serwery]

Wykorzystanie	Wykres	Model	Czas tworzenia (min.)	Ogólna dokładność (%)	Liczba użytych zmiennych
<input checked="" type="checkbox"/>		Sieci neuronowe 1	< 1	81,899	8
<input checked="" type="checkbox"/>		C5 1	< 1	79,356	8
<input checked="" type="checkbox"/>		CHAID 1	< 1	79,205	6
<input checked="" type="checkbox"/>		C&RT 1	< 1	77,907	7
<input checked="" type="checkbox"/>		Sieć Bayesa 1	< 1	77,889	8
<input checked="" type="checkbox"/>		Regresja logistyczna 1	< 1	75,302	8
<input checked="" type="checkbox"/>		Quest 1	< 1	65,781	8
<input checked="" type="checkbox"/>		Analiza dyskryminacyjna 1	< 1	49,360	4

Rysunek 10.33 Wyniki predykcji klastrów [WCSS, wszystkie serwery]

Kategoria	Dolna	Góra
1	$\geq 101,42603924$	$< 529,67509555$
2	$\geq 529,67509555$	$< 1122,09739634$
3	$\geq 1122,09739634$	$< 3165,33928217$
4	$\geq 3165,33928217$	$\leq 15526,90710158$

Rysunek 10.34 Przedziały przepustowości [PCSS, wszystkie serwery]

Kategoria	Dolna	Góra
1	$\geq 8,32$	$< 27,457$
2	$\geq 27,457$	$< 36,822$
3	$\geq 36,822$	$< 51,987$
4	$\geq 51,987$	$< 81,547$
5	$\geq 81,547$	$< 166,725$
6	$\geq 166,725$	$< 237,23$
7	$\geq 237,23$	$< 291,614$
8	$\geq 291,614$	$\leq 1307,451$

Rysunek 10.35 Przedziały RTT [PCSS, wszystkie serwery]