



**POLITECHNIKA
GDAŃSKA**

Imię i nazwisko autora rozprawy: **Tomasz Śmiałkowski**
Dyscyplina naukowa: **Informatyka Techniczna i Telekomunikacja**

ROZPRAWA DOKTORSKA

Tytuł rozprawy w języku polskim: **Adaptacyjny system oświetlania dróg oraz inteligentnych miast**

Tytuł rozprawy w języku angielskim: **Adaptive lighting system for roads and smart cities**

Promotor	Drugi promotor
<i>podpis</i>	<i>podpis</i>
prof. dr hab. inż. Andrzej Czyżewski	
Promotor pomocniczy	Kopromotor
<i>podpis</i>	<i>podpis</i>

Gdańsk, 2023



POLITECHNIKA
GDAŃSKA

DOKTORAT WDROŻENIOWY

Program Ministerstwa Nauki i Szkolnictwa Wyższego

Gdańsk, 2023

OŚWIADCZENIE

Autor rozprawy doktorskiej: Tomasz Śmiałkowski

Ja, niżej podpisany, oświadczam, iż jestem świadomy(a), że zgodnie z przepisem art. 27 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 poz. 1062), uczelnia może korzystać z mojej rozprawy doktorskiej zatytułowanej:

Adaptacyjny system oświetlenia dróg oraz inteligentnych miast

do prowadzenia badań naukowych lub w celach dydaktycznych.¹

Świadomy odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych i konsekwencji dyscyplinarnych określonych w ustawie Prawo o szkolnictwie wyższym i nauce (Dz.U.2021.478 t.j.), a także odpowiedzialności cywilnoprawnej oświadczam, że przedkładana rozprawa doktorska została napisana przeze mnie samodzielnie.

Oświadczam, że treść rozprawy opracowana została na podstawie wyników badań prowadzonych pod kierunkiem i w ścisłej współpracy z promotorem prof. dr hab. inż. Andrzejem Czyżewskim.

Niniejsza rozprawa doktorska nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem stopnia doktora.

Wszystkie informacje umieszczone w ww. rozprawie uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami, zgodnie z przepisem art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji pracy doktorskiej z załączoną wersją elektroniczną.

Gdańsk, dnia

.....
podpis doktoranta

Ja, niżej podpisany, wyrażam zgodę na umieszczenie ww. rozprawy doktorskiej w wersji elektronicznej w otwartym, cyfrowym repozytorium instytucjonalnym Politechniki Gdańskiej.

Gdańsk, dnia

.....
podpis doktoranta

¹ Art. 27. 1. Instytucje oświatowe oraz podmioty, o których mowa w art. 7 ust. 1 pkt 1, 2 i 4–8 ustawy z dnia 20 lipca 2018 r. – Prawo o szkolnictwie wyższym i nauce, mogą na potrzeby zilustrowania treści przekazywanych w celach dydaktycznych lub w celu prowadzenia działalności naukowej korzystać z rozpowszechnionych utworów w oryginale i w tłumaczeniu oraz zwielokrotniać w tym celu rozpowszechnione drobne utwory lub fragmenty większych utworów.

2. W przypadku publicznego udostępniania utworów w taki sposób, aby każdy mógł mieć do nich dostęp w miejscu i czasie przez siebie wybranym korzystanie, o którym mowa w ust. 1, jest dozwolone wyłącznie dla ograniczonego kręgu osób uczących się, nauczających lub prowadzących badania naukowe, zidentyfikowanych przez podmioty wymienione w ust. 1.

OPIS ROZPRAWY DOKTORSKIEJ

Autor rozprawy doktorskiej: Tomasz Śmiałkowski

Tytuł rozprawy doktorskiej w języku polskim: Adaptacyjny system oświetlania dróg oraz inteligentnych miast

Tytuł rozprawy w języku angielskim: Adaptive lighting system for roads and smart cities

Język rozprawy doktorskiej: polski

Promotor rozprawy doktorskiej: prof. dr hab. inż. Andrzej Czyżewski

~~**Drugi promotor rozprawy doktorskiej*:** <imię, nazwisko>~~

~~**Promotor pomocniczy rozprawy doktorskiej*:** <imię, nazwisko>~~

~~**Kopromotor rozprawy doktorskiej*:** <imię, nazwisko>~~

Data obrony:

Słowa kluczowe rozprawy doktorskiej w języku polskim: system oświetlenia drogowego, wykrywanie anomalii, uczenie maszynowe, inteligentne miasto, inteligentne liczniki energii

Słowa kluczowe rozprawy doktorskiej w języku angielskim: road lighting system, anomaly detection, machine learning, smart city, smart meters

Streszczenie rozprawy w języku polskim:

Przedmiotem rozprawy jest zbadanie praktycznej możliwości wykrywania w czasie rzeczywistym anomalii w systemie oświetlenia drogowego w oparciu o analizę danych ze inteligentnych liczników energii. Zastosowanie inteligentnych liczników energii elektrycznej (Smart Meter) w systemach oświetlenia drogowego stwarza nowe możliwości w zakresie automatycznej diagnostyki takich niepożądanych zjawisk jak awarie lamp, odstępstwa od harmonogramu czy też kradzieże energii z sieci zasilającej. Rozwiązanie takie wpisuje się w koncepcję inteligentnych miast (Smart City) gdzie zastosowanie adaptacyjnego systemu oświetlenia stwarza nowe wyzwania dla funkcji monitorowania. Zbadano metody uczenia maszynowego oparte o modele regresyjne oraz rekurencyjne sieci neuronowe. Zaproponowano praktyczne podejście oparte na zastosowaniu algorytmów czasu rzeczywistego, nienadzorowanych oraz używających ograniczonych zasobów obliczeniowych możliwych do implementacji w urządzeniach przemysłowych. Algorytmy przetestowano na rzeczywistych danych pochodzących z instalacji systemu oświetlenia i wykazano, że obie metody umożliwiają stworzenie samouczących algorytmów detekcji anomalii, działających w czasie rzeczywistym i że jest możliwa ich implementacja na urządzeniach warstwy Edge Computing.

W rozprawie przedstawiono również architekturę uniwersalnej platformy sterowania elementami infrastruktury oświetleniowej opracowanej przy udziale autora, jako głównego konstruktora, w ramach projektu rozwojowego „INFOLIGHT - Chmurowa platforma oświetleniowa dla inteligentnych miast”.

Streszczenie rozprawy w języku angielskim:

The subject of the dissertation research is investigating the practical possibility of real-time detection of anomalies in the road lighting system based on the analysis of data from smart energy meters. Using smart electricity meters in road lighting systems creates new opportunities for automatic diagnostics of undesirable phenomena such as lamp failures, schedule deviations, or energy theft from the power grid. Such a solution fits into the Smart City concept, where an adaptive lighting system creates new challenges for monitoring functions. Machine learning methods based on regression models and recurrent neural networks were investigated. A practical approach was proposed based on applying real-time, unsupervised algorithms and using the limited computational resources possible for implementation in industrial equipment. The algorithms were tested on real data from the installation of a lighting system, and it was shown that both methods enable the creation of self-learning, real-time anomaly detection algorithms and that it is possible to implement them on Edge Computing layer devices.

The dissertation also presents the architecture of a universal control platform for lighting infrastructure elements developed with the author's participation as the leading developer within the development project "INFOLIGHT - Cloud-based lighting platform for smart cities".

Podziękowania

Dziękuję Promotorowi, Panu prof. dr. hab. inż. Andrzejowi Czyżewskiemu, za rady, sugestie, wskazówki oraz za cierpliwość i wyrozumiałość. Dziękuję opiekunowi pomocniczemu w Firmie: Radosławowi Mioduskiemu, Prezesowi Zarządu TSTRONIC Sp. z o.o. za stworzenie warunków do powstania tej pracy. Dziękuję również mojej żonie za zrozumienie i wsparcie.

Spis symboli i skrótów

W pracy zastosowano następujące skróty:

ICT	Information and Communication Technologies
IoT	Internet of Things
ISO	International Organization for Standardization
ITU	International Telecommunication Union
IEC	International Electrotechnical Commission
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2X	Vehicle-to-everything
P2I	Pedestrian to Infrastructure
GPS	Global Positioning System
MEC	Multi-access edge computing
PLC	Power Line Communication
WAN	Wide Area Network
CIE	Commission Internationale de l'Eclairage
LED	Light Emitting Diode
HPS	High Pressure Sodium
IMU	Inertial Measurement Unit
OBD	On-board diagnostics
DALI	Digital Addressable Lighting Interface
WPAN	Wireless Personal Area Network
WLAN	Wireless Local Area Network
WWAN	Wireless Wide Area Network
LPWAN	Low-Power Wide-Area Network
ECG	Electronic Control Gear
AMI	Advanced Metering Infrastructure
NILM	Non-Intrusive Load Monitoring
ML	Machine Learning
RNN	Recurrent Neuron Networks
ARIMA	Auto Regressive Integrating Moving Average
SARIMA	Seasonal Auto Regressive Integrating Moving Average
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Units
AIC	Akaike Information Criterion
STLF	Short Term Load Forecast
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
AE	Absolute Error
MaAE	Maximum Absolute Error
MSE	Mean Squared Error
TP	True Positives
FN	False Negative
TNR	True Negative Rate
LTE	Long Term Evolution

Spis treści

Spis symboli i skrótów	1
1 Wprowadzenie.....	3
1.1 Wstęp	3
1.2 Cel badań	4
1.3 Zakres	5
1.4 Konspekt pracy	5
1.5 Tezy rozprawy	6
2 Systemy oświetlenia w Smart City.....	7
2.1 Architektura systemów inteligentnych miast	7
2.2 Fog/Edge computing.....	11
2.3 Dynamiczne sterowanie oświetleniem	13
2.4 Systemy adaptacyjnego sterowania oświetleniem.....	15
3 Metodyka monitorowania anomalii w oświetleniu	17
3.1 Monitorowanie stanu systemu oświetlenia.....	17
3.2 Wykrywanie anomalii w zużyciu energii	19
3.3 Rekurencyjne sieci neuronowe	25
4 Detekcja anomalii w systemie oświetleniowym	29
4.1 Baza danych pomiarów energii	29
4.2 Algorytmy detekcji stosujące okres jednogodzinny	34
4.3 Algorytmy detekcji stosujące okres 15 minutowy	54
4.4 Wnioski z badań symulacyjnych	67
5 Lampa do oświetlania inteligentnych miast.....	69
5.1 Koncepcja platformy sterowania	70
5.2 Konstrukcja.....	74
5.3 Oprogramowanie platformy	80
5.4 Integracja z systemem chmurowym	81
5.5 Przykładowe realizacje inteligentnych lamp	82
6 Podsumowanie i wnioski.....	85
Bibliografia.....	87
Spis rysunków	96
Spis tabel	99
Załączniki	101
Załącznik A: Szczegółowe wyniki eksperymentów.....	101

1 Wprowadzenie

1.1 Wstęp

Oświetlenie dróg, parków i innych miejsc publicznych to kluczowa usługa publiczna, która ma bezpośredni i pośredni wpływ na społeczny dobrobyt mieszkańców miast poprzez wspieranie mobilności mieszkańców, umożliwianie prawidłowych interakcji użytkowników dróg oraz zapobieganie przestępczości. Oświetlenie publiczne wspiera wzrost gospodarczy, zwiększając ilość czasu, który ludzie mogą spędzić rozrywce, uprawianiu sportu lub spotkaniach towarzyskich. Dlatego wiele miast na całym świecie wdraża programy oświetlenia publicznego mimo to, że jest to związane ze znaczącymi wydatkami z budżetu miasta, przy tym energia zużywana przez oświetlenie publiczne stanowi 30-50% energii zużywanej w europejskich miastach. Już dziś miasta zużywają około 75 procent globalnej energii i emitują od 50 do 60 procent wszystkich gazów cieplarnianych na świecie. Aktualnie ponad 50% światowej populacji żyje na obszarach miejskich i szacuje się, że wraz z ciągłym rozwojem miast do 2050 r. będzie w nich zamieszkiwało ponad dwie trzecie światowej populacji [1].

Jedną z odpowiedzi na te wyzwania jest koncepcja inteligentnych miast – Smart City. Smart City wykorzystuje technologie cyfrowe do skutecznego angażowania i aktywizowania mieszkańców, poprawiając wydajność miasta, społeczny dobrobyt jego mieszkańców oraz gości, zmniejszając koszty operacyjne i zużycie zasobów oraz tworząc nowe możliwości biznesowe, zwiększając ogólną atrakcyjność miasta. Inteligentne miasta wykorzystują rozwiązania oparte na technologiach informacyjnych i komunikacyjnych korzystających a danych pochodzących od mieszkańców, z sensorów, z urządzeń mobilnych oraz Internetu Rzeczy (IoT – Internet of Things).

Priorytetową kwestią dla miast w ich strategii na rzecz zrównoważonego rozwoju jest zarówno wymiana źródeł światła na bardziej wydajne, czyli wprowadzanie technologii LED, jak i wdrażanie inteligentnego oświetlenia publicznego, ponieważ przynosi to szybkie korzyści takie jak zwiększenie bezpieczeństwa, poprawa mobilności, zwiększenie atrakcyjności przestrzeni publicznych, poprawa jakości życia obywateli z ogromnym wpływem na oszczędność energii i koszty utrzymania infrastruktury oświetleniowej. Inteligentne oświetlenie jest naturalnym elementem Smart City, poprzez wykorzystanie danych z różnych źródeł do realizacji funkcji adaptacyjnego oświetlenia. Czujniki monitorujące lokalne warunki mogą umożliwić dostosowanie jasności oświetlenia ulicznego do natężenia ruchu: jeżeli nie ma ruchu lub natężenie ruchu jest bardzo małe, oświetlenie uliczne o pełnej jasności nie jest niezbędne. Czujniki pogodowe mogą również umożliwiać adaptację parametrów oświetlenia do deszczu, śniegu lub innych warunków. Inteligentne oświetlenie może również realizować funkcje związane z bezpieczeństwem publicznym, na przykład migające światła przed domem, który ratownicy próbują znaleźć, rozjaśnianie światła w miejscu wypadku lub przestępstwa, ostrzeganie kierowców w szkolnych strefach bezpieczeństwa.



Ważnym elementem systemów inteligentnego oświetlenia jest funkcja monitorowania zużycia energii. Dokładna informacja o zużyciu energii jest ważnym elementem w redukcji kosztów energii. Ponieważ oświetlenie uliczne staje się częścią bardziej złożonego systemu elektroenergetycznego, dokładne informacje o zużyciu energii w czasie rzeczywistym stają się coraz bardziej istotne z punktu widzenia optymalizacji i zarządzania siecią. W tym miejscu następuje połączenie z inną ważną technologią wdrażaną w ostatnich latach – inteligentne liczniki energii – Smart Meters, które zostały wdrożone na całym świecie w milionowych ilościach, na przykład w Europie wskaźnik penetracji inteligentnych liczników energii elektrycznej osiągnął 56% na koniec 2022 roku. Powszechna popularność inteligentnych liczników umożliwia gromadzenie ogromnej ilości szczegółowych danych dotyczących zużycia energii elektrycznej. Dane o wysokiej rozdzielczości z inteligentnych liczników dostarczają bogatych informacji na temat prawidłowego lub nie działania różnych urządzeń w zakresie zużycia energii elektrycznej. W przypadku gospodarstw domowych i biur dostarczają informacje na temat zachowań konsumentów oraz i stylu życia. Wykorzystanie ogromnej ilości danych z inteligentnych liczników niewątpliwie może służyć do promowania zrównoważonego rozwoju i zwiększania wydajności systemów energetycznych zarówno od strony konsumpcji, jak i wydajności.

Systemy sterowania oświetleniem, podobnie jak wszystkie systemy sterujące, wymagają monitoringu, w tym detekcji odstępstw od zaplanowanego działania, czyli anomalii. Rozwój funkcji inteligentnych systemów sprawia, że skuteczne wykrywanie anomalii w czasie rzeczywistym staje się wymagającym zadaniem. Dodatkowo dla systemów adaptacyjnych niezbędne jest by system nadzoru identyfikował złożone procesy i wzorce bez udziału człowieka udzielającego ścisłych wskazówek, czyli stosował nienadzorowane samouczenie, w którym modele trenowane są na danych, które nie mają etykiet, ani zdefiniowanego wyniku.

Sz szczególnie efektywnym sposobem wykrywania anomalii jest analiza zużycia energii, co jest problemem dobrze zbadanym w zakresie różnych typów odbiorców: mieszkania, budynki użytkowe, przemysł. Do analizy stosuje się różne metody probabilistyczne, statystyczne, modelowanie liniowe, maszynowe uczenie. Systemy oświetlenia drogowego nie były dotychczas przedmiotem takich badań, stąd taki temat podejmuje niniejsza rozprawa.

1.2 Cel badań

Celem badawczym niniejszej rozprawy jest zbadanie metod wykrywania anomalii w systemie oświetleniowym na podstawie danych z inteligentnych liczników energii oraz opracowanie algorytmów detekcji anomalii w czasie rzeczywistym. Wykrywanie anomalii będzie możliwe zarówno dla systemów inteligentnych, jak i tradycyjnych oraz stosujących zarówno nowoczesne LED-owe źródła światła, jak i starszego typu HPS. Wykrywanie anomalii ma służyć automatycznej diagnostyce, mającej na celu wykrywanie uszkodzeń zespołów lamp, jak również pojedynczej lampy, zasilanej we wspólnym obwodzie wraz z wieloma innymi

lampami, detekcji odstępstw od zaplanowanych scenariuszy oraz kradzieży energii z sieci zasilającej. W założeniu rozwiązanie będzie istotnym elementem koncepcji inteligentnych miast, ponieważ umożliwi oszczędzanie energii elektrycznej, zmniejszy koszty utrzymania infrastruktury oświetleniowej i wpłynie na poprawę bezpieczeństwa użytkowników dróg i mieszkańców. Opracowane algorytmy mogą być implementowane na urządzeniach stanowiących infrastrukturę oświetleniową realizując koncepcję Edge/Fog Computing, optymalizując transmisję danych, zmniejszając opóźnienia reakcji i polepszając bezpieczeństwo systemu. W niniejszej rozprawie zostanie również omówiona koncepcja lampy do oświetlania inteligentnych miast, jako praktyczne wdrożenie urządzenia typu Edge, na którym opracowane algorytmy mogą być implementowane.

1.3 Zakres

Wyznaczenie optymalnego algorytmu wymagało przeanalizowania charakteru danych pochodzących z inteligentnych liczników energii dla systemów oświetlenia jako szeregów czasowych. Kolejnym krokiem jest klasyfikacja sygnałów pomiaru energii i zdefiniowanie typów anomalii. Zbadano różne metody maszynowego uczenia, oparte o modele liniowe, wykorzystujące metody głębokiego uczenia stosujące sieci rekurencyjne oraz model Autoenkodera. Ewaluacja algorytmów detekcji anomalii została przeprowadzona w oparciu o dane pochodzące z systemu sterowania oświetleniem, zainstalowanego w jednym z polskich miast przez przedsiębiorstwo zatrudniające autora niniejszej pracy. System steruje ponad 4400 inteligentnymi lampami LED i wyposażony jest w ponad 90 inteligentnych liczników energii. Na potrzeby niniejszej pracy utworzono bazę danych pomiarów zużycia energii przez system oświetlenia. Ponadto, w ramach realizacji projektu badawczo rozwojowego INFOLIGHT (www.multimed.biz/infolight) w Politechnice Gdańskiej i w przedsiębiorstwie TSTRONIC sp. z o. o., zatrudniającym jest autora niniejszej pracy, skonstruowano inteligentną lampę, będącą elementem chmurowej platformy oświetleniowej. Autor zrealizował jako konstruktor i programista szereg związanych z tym projektem prac koncepcyjnych i konstrukcyjnych, które zostały krótko podsumowane w niniejszej pracy, gdyż opracowana inteligentna lampa jest docelową platformą umożliwiającą bezpośrednie zastosowanie przedstawionych wyników badań.

1.4 Konspekt pracy

Układ niniejszej rozprawy jest następujący. Rozdział 2 przedstawia przegląd zagadnień związanych z systemami oświetlenia jako elementu Inteligentnego Miasta. Omówiono architekturę systemów Smart City, koncepcję Edge/Fog Computing, z sposoby dynamicznego sterowania oświetleniem oraz systemy adaptacyjnego sterowania oświetleniem. W rozdziale 3 przedstawiono przegląd zagadnień teoretycznych dotyczących monitorowania anomalii w systemach oświetlenia. Rozdział 4 przedstawia porównanie wybranych algorytmów detekcji anomalii opartych o maszynowe uczenie. Rozdział 5 przedstawia koncepcję i elementy

konstrukcji lampy przeznaczonej do oświetlania inteligentnych miast. W rozdziale 6 zamieszczono podsumowanie oraz wnioski.

1.5 Tezy rozprawy

W ramach części badawczej niniejszej pracy udowodnione będą następujące tezy:

T1. Algorytmy oparte na uczeniu maszynowym nadają się do wykrywania anomalii w poborze mocy przez systemy oświetlania dróg, na podstawie analizy zbiorczych danych pochodzących z inteligentnych liczników energii.

T2. Opracowane algorytmy oparte na uczeniu maszynowym wykrywają anomalie w poborze mocy przez systemy oświetleniowe w sposób bardziej efektywny od metod dotychczas stosowanych w tym celu.

2 Systemy oświetlenia w Smart City

Koncepcja inteligentnego miasta - Smart City wyłoniła się z konwergencji różnych postępów technologicznych i pomysłów na rozwój miast już pod koniec XX wieku i rozwijała się jako wspólne idee naukowców, urbanistów i entuzjastów technologii dotyczące integracji technologii informacyjno-komunikacyjnych (ICT) w środowiskach miejskich [2], [3], [4], [5]. Mimo rozpowszechnionego posługiwania się pojęciem Smart City, w literaturze brakuje jednoznacznej i uniwersalnej jego definicji, między innymi dlatego, że obejmuje zarówno kwestie technologiczne, jak i społeczne. Choć koncepcja Smart City jest bardzo podatna na ewolucję idei i technologii leżących u jej podstaw, niezmiennie pozostało wykorzystanie analityki danych (Big Data), urządzeń Internetu Rzeczy (IoT), infrastruktury komunikacyjnej, planowania urbanistycznego oraz partycypacyjnego zarządzania w celu poprawy jakości życia mieszkańców, efektywnego zarządzania zasobami miast oraz zrównoważonego wzrostu gospodarczego. Należy tu wspomnieć o próbach ustanowienia standardów przez organizacje międzynarodowe [6] takie jak Międzynarodowa Organizacja Normalizacyjna (ISO) [7], [8], Międzynarodowy Związek Telekomunikacyjny (ITU) [9] czy Międzynarodowa Komisja Elektrotechniczna (IEC) [10]. Standardy te umożliwiają miastom osiągnięcie odpowiednich wskaźników technicznych, środowiskowych i społecznych, które mają znaczący wpływ na infrastrukturę, bezpieczeństwo i życie mieszkańców.

W niniejszym rozdziale wskazano na powiązania koncepcji i architektury systemów inteligentnych miast ze sterowanymi i monitorowanymi cyfrowo systemami oświetlenia, których dotyczy niniejsza praca.

2.1 Architektura systemów inteligentnych miast

Definiując umiejscowienie systemu inteligentnego oświetlenia wewnątrz Smart City można przyjąć wiele perspektyw [11]. Jedną z takich perspektyw są kategorie (wymiary) w jakich generowane i wdrażane są rozwiązania inteligentne, czyli „Smart”. W raporcie SRF, Vienna University of Technology [12] wprowadzono sześć kluczowych kategorii, służących do oceny wydajności miasta jako inteligentnego miasta:

- Ludzie (Smart People), określające takie czynniki jak poziom kwalifikacji, pluralizm społeczny i etniczny, kreatywność, kosmopolityzm, udział w życiu publicznym;
- Mobilność (Smart Mobility), określająca dostępność lokalną i międzynarodową, dostępność infrastruktury ICT, zrównoważone i bezpieczne systemy transportowe.
- Środowisko (Smart Environment), uwzględniająca atrakcyjność warunków naturalnych, zanieczyszczenia, ochronę środowiska, zrównoważone zarządzanie zasobami.

- Gospodarka (Smart Economy), uwzględniająca takie czynniki jak innowacyjność, przedsiębiorczość, znaki towarowe, wydajność, elastyczność rynku pracy, relacje międzynarodowe, zdolność do transformacji.
- Rządzenie (Smart Governance), obejmująca udział w podejmowaniu decyzji, usługi publiczne i społeczne, transparentność zarządzania, strategię i perspektywy polityczne.
- Jakość życia (Smart Living), określające takie czynniki jak obiekty kulturalne i edukacyjne, warunki zdrowotne, bezpieczeństwo indywidualne, jakość mieszkalnictwa, atrakcyjność turystyczna, spójność społeczna.

Inną perspektywą są konkretne konteksty i cele działania inteligentnego miasta. Najczęściej w inicjatywach inteligentnych miast pojawiają się następujące elementy czy też obszary [13], [14]:

- Usługi (Smart City Services). Inteligentne usługi dla obywateli koncentrują się na zapewnianiu obywatelom wydajnych i dostępnych usług. Może to obejmować portale usług online, aplikacje mobilne do świadczenia usług, cyfrowe systemy płatności i spersonalizowane platformy informacyjne. Usługi inteligentnego miasta obejmują działania, które obejmują zadania komunalne, takie jak dostarczanie wody, gospodarka odpadami, kontrola i monitorowanie środowiska [15], [16].
- Transport (Smart Transportation). Ten obszar obejmuje inteligentne zarządzanie ruchem, informacje transportowe w czasie rzeczywistym, inteligentne rozwiązania parkingowe, infrastrukturę pojazdów elektrycznych oraz platformy mobilności jako usługi. Wprowadzanie innowacji w tym obszarze ma na celu ulepszenie systemów transportowych, poprawę mobilności i promowanie zrównoważonych środków transportu. Przykładowe rozwiązania to śledzenie zachowania uczestników ruchu, tworzenie wzorców ruchu, mapowania tras w aplikacjach nawigacyjnych, planowanie podróży w transporcie publicznym, kierowanie kierowców do najbliższego wolnego miejsca parkingowego. Umożliwiają to takie technologie jako łączność „pojazd do pojazdu” (V2V), „pojazd do infrastruktury” (V2I), „pojazd do pieszego” (V2P) czy „pieszy do infrastruktury” (P2I) oraz systemy lokalizacji GPS [17].
- Energetyka (Smart Energy). Ten element obejmuje inteligentne sieci (Smart Grids), monitorowanie i optymalizację energii, integrację energii odnawialnej oraz mechanizmy reagowania na popyt. Inteligentna energetyka koncentruje się na efektywności energetycznej, odnawialnych źródłach energii i inteligentnych systemach zarządzania energią. Ponieważ inteligentne sieci dostarczają w czasie rzeczywistym dane o zużyciu energii przez konsumentów, możliwe jest

zarządzanie wytwarzaniem energii przy użyciu modeli predykcyjnych, opracowanych na podstawie indywidualnych profili. Smart Grids umożliwiają również nadzorowanie rozproszonego wytwarzania energii, zarówno po stronie prosumenta jak i zakładu energetycznego oraz wprowadzenia do sieci funkcji adaptacyjnych [18].

- Opieka zdrowotna (Smart Healthcare). W tym obszarze zawarte jest wykorzystanie technologii w celu poprawy usług i świadczenia opieki zdrowotnej. Obejmuje elektroniczną dokumentację medyczną, telemedycynę, zdalne monitorowanie pacjentów, urządzenia do noszenia i analizy opieki zdrowotnej w celu wydajnego świadczenia opieki zdrowotnej. Rozpowszechnienie telefonów komórkowych i urządzeń monitorujących stan zdrowia umożliwia rejestrowanie w czasie rzeczywistym danych dotyczące zdrowia ludzi, np.: EKG, temperatura, nasycenie tlenem ciała, aktywność fizyczna. Dane te mogą być przetwarzane w aplikacjach chmurowych wspierając podejmowanie decyzji dotyczących poprawy opieki zdrowotnej, zmniejszając ogólne koszty i obciążenie placówek służby zdrowia [19], [20].
- Infrastruktura (Smart Infrastructure). Ten obszar obejmuje inteligentne systemy gospodarki wodnej, rozwiązania w zakresie gospodarki odpadami oraz zaawansowane systemy monitorowania i utrzymania elementy infrastruktury krytycznej. Wykorzystanie czujników do pomiaru odkształceń stali, betonu i gruntu, czujników mikrofalowych, akustycznych, akcelerometrów oraz inteligentnych materiałów umożliwia monitorowania w czasie rzeczywistym stanu konstrukcji budynków, mostów, dróg i innych elementów infrastruktury. Dane zebrane przez te czujniki pozwalają na predykcyjną konserwację tych kluczowych jednostek w celu utrzymania normalnego funkcjonowania miasta [21], [22].
- Budynki (Smart Buildings). W tym obszarze zawarte są energooszczędne systemy, automatyka i sterowanie w budynkach użyteczności publicznej i mieszkalnych, sieci czujników do monitorowania aktywności użytkowników, urządzenia śledzące ruch i zużycie energii oraz zintegrowane systemy zarządzania budynkiem. Inteligentne budynki wykorzystują technologię do optymalizacji zużycia energii, zwiększania komfortu użytkowników i usprawniania konserwacji [23].
- Oświetlenie (Smart Lighting). Ten obszar obejmuje inteligentne oświetlenie uliczne i drogowe, czujniki ruchu i sterowanie adaptacyjne w celu optymalizacji zużycia energii poprzez dostosowywanie poziomów oświetlenia w oparciu o potrzeby w czasie rzeczywistym. Inteligentne systemy oświetleniowe umożliwiają zdalne monitorowanie i sterowanie, umożliwiając efektywne praktyki konserwacyjne i szybkie wykrywanie usterek lub awarii. Ponadto systemy te można zintegrować z

innymi technologiami, takimi jak monitoring wideo i systemy reagowania kryzysowego, tworząc bardziej wszechstronną infrastrukturę bezpieczeństwa [24].

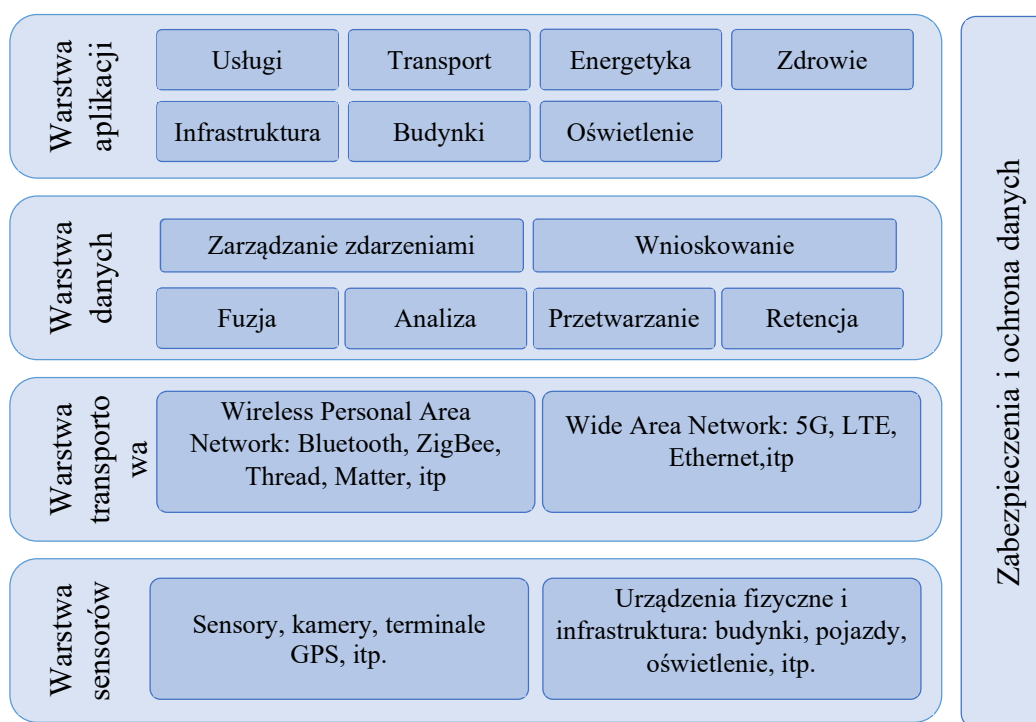
Kolejnym sposobem analizy Smart City jest określenie jego kluczowych warstw jako systemu ICT [25]:

- Infrastruktura ICT - podstawowa warstwa, na której opierają się wszystkie inne komponenty. Infrastruktura ICT obejmuje szybką łączność przewodową i bezprzewodową, chmurę obliczeniową, IoT, sieci czujników, elementy wykonawcze i wiele innych.
- Reguły (E-governance) - warstwa formułuje zasady, zasady i przepisy prawne w celu poprawy wydajności administracji i zaoferowania potencjalnych korzyści obywatelom.
- Usługi - w tej warstwie zlokalizowane są obszary funkcjonowania Smart City wymienione wcześniej: Smart Energy, Smart Healthcare, Smart Infrastructure, itp.
- Interesariusze - obejmuje wszystkie podmioty mające wpływ na działanie Smart City lub pozostają pod wpływem jego działania: obywatele, przedstawiciele administracji, operatorzy usług, turyści, itp.

Inny model warstwowy, który opisano w artykule [26], jest oparty o model wstępujący (bottom-up). Ta architektura składa się z czterech warstw: warstwy sensorów, warstwy transmisji, warstwy zarządzania danymi i warstwy aplikacji. Dodatkowo model wyróżnia moduły bezpieczeństwa, które są zintegrowane z każdą warstwą. Gromadzenie danych z urządzeń fizycznych jest głównym zadaniem warstwy sensorów, która znajduje się u dołu schematu architektury. Szereg różnych technologii komunikacyjnych tworzących warstwę transmisyjną przenosi dane do warstw wyższych. Warstwa zarządzania danymi przetwarza i przechowuje cenne informacje, które są przydatne do świadczenia usług oferowanych przez różne aplikacje w warstwie aplikacyjnej. Schemat opisanej architektury przedstawia Rys. 1.

Systemy inteligentnego oświetlenia powiązane są z innymi systemami tworzącymi Smart City na wiele sposobów i należy patrzeć na te powiązania z wielu perspektyw. Systemy takie przyczyniają się do poprawy bezpieczeństwa i ochrony na obszarach miejskich. Mogą być wyposażone w czujniki ruchu i inteligentne sterowanie, które automatycznie dostosowuje poziomy oświetlenia w oparciu o warunki w czasie rzeczywistym. Zapewnia to dobrze oświetlone ulice, przestrzenie publiczne i ciągi komunikacyjne, zmniejszając ryzyko wypadków i działalności przestępczej. Infrastruktura oświetleniowa może służyć jako podstawa dla innych aplikacji i usług inteligentnego miasta. Inteligentne systemy oświetleniowe można łączyć z siecią miejską w celu gromadzenia danych i obsługi różnych funkcji. Na przykład infrastruktura oświetleniowa może zawierać czujniki do zbierania danych środowiskowych, monitorowania jakości powietrza lub wykrywania korków. Dane te można wykorzystać do planowania miasta, optymalizacji alokacji zasobów i poprawy ogólnego

zarządzania miastem. Właściwie zaprojektowane oświetlenie w przestrzeni publicznej może poprawić ogólną jakość życia obywateli. Inteligentne systemy oświetleniowe można zaprogramować tak, aby tworzyły atrakcyjne wizualnie i komfortowe środowiska, pozytywnie wpływając na nastrój, produktywność i dobre samopoczucie ludzi. Oświetlenie adaptacyjne może również wspierać naturalny rytm dobowy poszczególnych osób, dostosowując temperaturę barwową i intensywność w ciągu dnia.



Rys. 1. Warstwowa architektura systemu Smart City

Podsumowując, inteligentne oświetlenie jest kluczowym elementem inteligentnych miast, oferującym efektywność energetyczną, oszczędność kosztów, bezpieczeństwo, zrównoważony rozwój i lepszą jakość życia. Dzięki integracji zaawansowanych technologii oświetleniowych i inteligentnego sterowania miasta mogą tworzyć bardziej wydajne, zrównoważone i przyjazne dla życia środowiska miejskie. Stąd istotne jest zapewnienie niezawodnego działania tych systemów, które można osiągnąć dzięki stosowaniu systemów nadzoru nad nimi. W niniejszej pracy zaproponowano i zweryfikowano nowe sposoby prowadzenia takiego nadzoru w sposób zautomatyzowany.

2.2 Fog/Edge computing

Bardzo ważną rolę w rozwoju i funkcjonowaniu Inteligentnych Miast odgrywa koncepcja chmury obliczeniowej (Cloud Computing). Wykorzystanie Big Data, czyli analizy masywnych danych jest jedną z kluczowych własności, ponieważ Inteligentne Miasta generują ogromne ilości danych za pośrednictwem różnych czujników, urządzeń i systemów. Platformy chmurowe mogą obsłużyć masowy napływ danych i zapewnić niezbędną infrastrukturę do ich

przetwarzania i analizowania w czasie rzeczywistym, zapewniając dużą skalowalność. Na platformach chmurowych dostępne są usługi umożliwiające wykorzystywanie algorytmów maszynowego uczenia, analizy predykcyjne i narzędzia do wizualizacji danych. Cloud Computing zapewnia możliwości łączności i integracji niezbędne do agregacji danych z różnych źródeł i urządzeń (IoT) oraz umożliwia rozwój usług i aplikacji zorientowanych na potrzeby mieszkańców i gości [27].

Rozwój aplikacji działających w czasie rzeczywistym i wrażliwych na opóźnienia takich jak usługi związane z połączonymi pojazdami, dronami, kamerami, itp. zwiększył wymagania na skrócenie czasu reakcji oraz na przeniesienie przetwarzania danych bliżej ich źródła. W związku z tym wprowadzono rozwiązanie Fog Computing (mgła obliczeniowa) [28], które przenosi moc obliczeniową, pamięć masową, komunikację, kontrolę i podejmowanie decyzji bliżej brzegu sieci, gdzie generowane są dane. Celem tego jest pokonywanie ograniczeń obecnej infrastruktury i umożliwienie przypadków użycia o znaczeniu krytycznym i dużej ilości danych. Dane są przetwarzane i analizowane w urządzeniach brzegowych (Edge Device) - zdecentralizowanych urządzeniach, takich jak na przykład sterowniki przemysłowe, przełączniki, routery i kamery do monitoringu wideo, stąd też druga nazwa rozwiązania – Edge Computing. Cloud Computing i Fog Computing wzajemnie się uzupełniają się, oferując zdyweryfikowane zasoby i usługi [29].

Zastosowanie Fog/Edge computing jest wskazane tam, gdzie wymagane są następujące własności:

- Minimalizacja opóźnień. Fog/Edge zapewnia przetwarzanie danych, sterowanie inne zadania wrażliwe na opóźnienia blisko użytkownika końcowego i w ten sposób pozwalają spełnić rygorystyczne wymagania czasowe wielu systemów IoT [30].
- Ograniczenie transmisji danych. Dane są przetwarzane na różnych poziomach infrastruktury, od lokalnych urządzeń na krawędzi sieci po bardziej centralne systemy chmurowe, co pozwala na zoptymalizowanie efektywności transmisji oraz minimalizację opóźnień w transmisji danych [31].
- Ograniczone zasoby urządzeń. Fog/Edge może wykonywać zadania wymagające dużych zasobów w imieniu urządzeń o ograniczonych zasobach, gdy takich zadań nie można przenieść do chmury z jakiegokolwiek powodu, zmniejszając w ten sposób złożoność tych urządzeń, koszty cyklu życia i zużycie energii [32].
- Odporność na przerwy w łączności z chmurą. Lokalny system Fog/Edge może działać autonomicznie, aby zapewnić nieprzerwane usługi, nawet jeśli ma przerywaną łączność sieciową z chmurą [33].
- Anonimizacja danych. Fog/Edge może zacierać powiązania między danymi z czujników a podmiotami, których dane dotyczą, na przykład poprzez generalizację, agregację lub syntezę [34].
- Optymalizacja uczenia maszynowego. Faza szkolenia to zadanie długoterminowe, wymagające dużej mocy obliczeniowej może być realizowana w chmurze

obliczeniowej, faza wnioskowania wymagająca znacznie mniejszej mocy obliczeniowej jest realizowana przez Fog/Edge [35].

Specyficzną implementacją przetwarzania brzegowego dostosowaną do sieci mobilnych jest Multi-access Edge Computing (MEC), znany również jako Mobile Edge Computing. MEC rozszerza możliwości przetwarzania brzegowego na sieci dostępu radiowego (RAN) sieci komórkowych, umożliwiając wdrażanie zasobów i usług obliczeniowych na obrzeżach infrastruktury komórkowej. W MEC serwery brzegowe są umieszczane na stacjach bazowych lub wieżach komórkowych sieci komórkowych, przybliżając zasoby obliczeniowe do urządzeń mobilnych i użytkowników. Pozwala to na szybsze przetwarzanie danych i zmniejszenie opóźnień aplikacji mobilnych, poprawiając wrażenia użytkownika i obsługując nowe zastosowania, takie jak rzeczywistość rozszerzona i aplikacje IoT o niskich opóźnieniach [36], [37].

Fog/Edge Computing jest rozwiązaniem, które ma szczególne zastosowania w nowoczesnych systemach inteligentnego oświetlenia. System oświetlenia obejmuje największą część obszaru miasta spośród wszystkich elementów Smart City a ponieważ lampy muszą być zasilane, więc mogą być wykorzystywane jako naturalna infrastruktura dla udostępniania usług Smart City. Inteligentne systemy oświetlenia oparte są to hierarchiczne sieci sterujące, których architektura odpowiada hierarchicznemu modelowi sieci komunikacyjnej [38]. Pierwszy poziom, to strefa obejmująca ograniczony obszar i ograniczoną ilość lamp, działających w celu spełnienia lokalnych celów oświetleniowych i podlegających lokalnym ograniczeniom, np. zasięg transmisji dla wybranej technologii, radiowej (ZigBee, Bluetooth) lub przewodowej (Power Line Communication) [39]. Każda taka strefa obsługiwana jest przez sterownik strefowy – koncentrator, który pełni między innymi funkcję mostu sieciowego (Bridge) i jest urządzeniem brzegowym (Edge Device) Koncentratory łączą się między sobą oraz serwerami usług poprzez sieć rozległą (WAN - Wide Area Network) i ta sieć może formować Fog Computing. Tworzenie zautomatyzowanych systemów nadzoru systemów oświetlenia powinno więc wykorzystywać tę architekturę i dlatego proponowane w niniejszej pracy algorytmy będą do tego dostosowane.

2.3 Dynamiczne sterowanie oświetleniem

Głównym zadaniem oświetlenia ulicznego jest zapewnienie oświetlenia dróg publicznych i ulic w porze nocnej. Najważniejszym celem jest poprawa widoczności pieszych i pojazdów zmniejszając ryzyko wypadków przez możliwość zauważania przeszkód czy elementów infrastruktury drogowej z odpowiedniej odległości, czyli takiej która pozwala na uniknięcie wypadku, a w przypadku pieszych potknięcia się i upadku. Oświetlenie dróg publicznych i ulic usprawnia ruch drogowy dzięki zapewnieniu widoczności znaków drogowych, sygnalizacji i oznaczeń pasów ruchu. Odpowiednie oświetlenie poprawia bezpieczeństwo zniechęcając do działań przestępczych, zwiększając postrzeganie ryzyka przez



potencjalnych sprawców, ponieważ istnieje większe prawdopodobieństwo, że zostaną zauważeni i zidentyfikowani. Ekonomizacja wszelkich działań publicznych wymaga by zadanie oświetlenia dróg publicznych i ulic w porze nocnej zostało zrealizowane przy najmniejszych kosztach instalacji i eksploatacji. Należy również uwzględnić negatywny wpływ oświetlenia na środowisko [40] i ludzi [41].

Podstawowe funkcje sterowania systemem oświetlenia to załączanie, wyłączenie oraz ustawienie poziomu świecenia. Zasady załączania i wyłączania oświetlenia dróg jest uregulowana przez różne przepisy, zazwyczaj stosuje się do tego pojęcia zmierzchu cywilnego i świtu cywilnego, formalnie zdefiniowanego jako momenty, w których słońce znajduje się 6 stopni poniżej linii horyzontu, odpowiednio po zachodzie lub przed wschodem. Innym sposobem sterowania jest stosowanie czujnika zmierzchowego i załączanie lamp w zależności od poziomu oświetlenia otoczenia. Wybór metody zależy od lokalnych uregulowań. Sposób realizacji metody może się różnić w zależności od dostępnych technicznych możliwości instalacji oświetleniowej. W przypadku sterowania w oparciu o zmierzchy i świty stosuje się zegar astronomiczny w szafie sterującej oświetleniem, który załącza i wyłącza całe obwody zasilania lamp. Współcześnie, w przypadku inteligentnych lamp, zegar astronomiczny może być już wbudowany w samą lampę. Pozwala to na jeszcze bardziej zaawansowane i autonomiczne zarządzanie oświetleniem, bez potrzeby zewnętrznej szafy sterującej. W przypadku sterowania w oparciu o poziom oświetlenia otoczenia stosuje się czujnik zmierzchowy umieszczony w szafie oświetleniowej, który załącza i wyłącza całe obwody lamp przy zadanym poziomie oświetlenia lub czujnik zmierzchowy w oprawie, bezpośrednio sterujący lampą. Choć załączanie i wyłączanie jest najprostszą metodą sterowania to i tą metodą można realizować funkcje adaptacyjnego systemu oświetlenia [42].

Poziom oświetlenia dróg publicznych i ulic jest określony normami, bazującymi na rekomendacjach opracowanych przez Międzynarodową Komisję Oświetleniową (CIE) [43]. Obecnie w Europie oświetlenie drogowe projektowane jest zgodnie z wymaganiami normy EN 13201-Road Lighting W Europie obowiązuje norma PN-EN 13201:2016 – „Oświetlenie dróg” [44]. Zgodnie z tą normą jest możliwość wyboru klasy oświetlenia (od M1 do M6) w zależności od warunków panujących na drodze i jej otoczeniu takich jak: natężenie ruchu, skład ruchu, rozgraniczenie jezdni, zaparkowane pojazdy, jasność otoczenia, itp. Poziom oświetlenia dla danej drogi może być zdefiniowany jednorazowo i pozostać niezmienny przez cały okres eksploatacji jednak jest bardzo nieefektywne i dlatego wprowadzono rozwiązania polegające na okresowym zmniejszeniu poziomu świecenia – redukcji. Redukcja jest dopuszczalna poprzez dynamiczną zmianę klasy oświetleniowej zarówno dla ruchu motorowego, jak i pieszego [45] i przynosi oszczędności minimum 15% [46]. Powszechne stosowanie redukcji jest możliwe dzięki stosowaniu źródeł światła typu LED, dla których sterowanie poziomem jest technicznie proste, w przeciwieństwie do źródeł światła starszego typu, np. HPS.

Najłatwiejszą i jedną z najczęściej wdrażanych strategii redukcji jest redukcja statyczna, według której ustawia się poziomy ściemniania lamp ulicznych zawsze zgodnie z ustalonymi godzinami. Przykładowy harmonogram to taki, który utrzymuje poziom mocy na poziomie 100% przez pół nocy i na poziomie 40% przez pozostałą połowę. Redukcja rozszerza funkcjonalność adaptacyjnego systemu oświetlenia, który w oparciu o dane z czujników lub innych źródeł dynamicznie dostosowuje natężenie światła lampy lub grupy lamp do warunków meteorologicznych lub do warunków panujących na drodze.

Ważnym parametrem oświetlenia jest temperatura barwowa, które wprawdzie nie podlega jeszcze ścisłym regulacjom jednak wiele badań dowiodło, że temperatura barwowa ma wpływ, w tym negatywny, na zachowania ludzi i zwierząt [47]. Głównym punktem odniesienia podczas projektowania oświetlenia zewnętrznego, w tym ulicznego, jest naturalne białe światło dzienne. Dlatego coraz częściej wykorzystuje się lampy oparte na diodach LED, które zapewniają właśnie to białe światło. Jest to szczególnie istotne w przypadku oświetlenia nocnego dróg, gdyż zapewnia ono znacznie lepszą widoczność niż światło ciepłe. Oświetlenie białe powoduje, że otoczenie jest postrzegane jako bardziej naturalne, co ma bezpośredni wpływ na zwiększenie bezpieczeństwa użytkowników dróg publicznych. Wysoka efektywność świetlna diod LED przyczynia się do podniesienia poziomu bezpieczeństwa na drogach miejskich. W związku z tym, w przypadku oświetlenia ulicznego, zazwyczaj stosuje się lampy o temperaturze barwowej około 5000 K. Natomiast w przestrzeniach miejskich, mieszkalnych oraz parkach preferowane jest użycie barwy ciepłej o temperaturze 3500 K lub 4000 K. Odpowiednie oświetlenie dróg i ulic ma kluczowy wpływ na komfort mieszkańców miast oraz turystów, a także przyczynia się do zwiększenia bezpieczeństwa w przestrzeni publicznej [48]. Dlatego wybór właściwej technologii oświetlenia i odpowiednich parametrów światła ma ogromne znaczenie dla jakości życia i użytkowania przestrzeni miejskiej. Dynamiczne sterowanie oświetleniem może obejmować również sterowanie temperaturą barwową źródła światła [49].

2.4 Systemy adaptacyjnego sterowania oświetleniem

Inteligentne systemy oświetlenia mają różne stopnie złożoności, które można zdefiniować jako cztery uzupełniające się poziomy. Poziom pierwszy - wbudowany, to poziom funkcji znajdujących się w silniku oświetleniowym, rozumianym jako źródło światła i układ zasilający. Drugi poziom - systemowy, to funkcje realizowane przez lampy i systemy oświetleniowe. Trzeci poziom – sieciowy, to zarządzanie i monitorowanie źródeł zasilania, elektrowni oraz mediów i urządzeń dystrybucyjnych. Czwarty poziom – adaptacyjny, to kompletne rozwiązania oświetleniowe z aplikacjami do monitorowania, sterowania i zarządzania.

System adaptacyjny umożliwia dostosowanie zachowania systemu oświetlenia do zachowania monitorowanych podmiotów i zmiennych będących przedmiotem zainteresowania w konkretnej sytuacji [50], [51], [52]. Możliwa jest również obsługa określonych sytuacji, takich jak święta, ciemne noce lub konkretne daty, w których konieczne lub odpowiednie jest specjalne oświetlenie w zależności od różnych wymagań, np. akcja „Godzina dla Ziemi”.



System adaptacyjny korzysta z całego szeregu czujników. Estymacja wymaganego poziomu świecenia lub temperatury barwowej odbywa się na podstawie analizy danych z grup czujników, które współpracują między sobą przekazując informacje o natężeniu ruchu, warunkach pogodowych przy drodze oraz ewentualnym pojawieniu się pieszych na przejściu bądź przy skrajni drogi. Sensory, które dostarczają informacji niezbędnych do analizy pod kątem dostosowania widoczności na drodze są sensory odpowiadające za dane meteorologiczne. Między innymi można wymienić tutaj: czujnik opadów i wilgotności, czujnik natężenia światła, czujnik zmierzchowy, czujnik stanu nawierzchni. Czujniki te mogą pracować w grupie odpowiadającej za dostosowanie odpowiednio wysokiego poziomu luminancji co umożliwi uzyskanie dobrej widoczności sylwetki pieszego przy jednoczesnym uzyskaniu odpowiedniego kontrastu oświetlenia pieszego bądź pojazdów. Jednoczesny pomiar parametrów z czujników pozwala na automatyzację dostosowania poziomu oświetlenia lamp ulicznych.

System adaptacyjny umożliwia dostosowanie parametrów oświetlenia do zróżnicowanych funkcji drogi czy ulicy, zapewniając bardzo dobre oświetlenie w miejscach niebezpiecznych, na przykład w miejscach przejść dla pieszych czy skrzyżowaniach, w pozostałych miejscach redukując poziom oświetlenia. Komunikacja dwukierunkowa w poszczególnych lampach systemu umożliwia nie tylko przekazywanie zebranych danych z czujników i ich transfer do późniejszej analizy, ale również otrzymywanie komunikatów o redukcji bądź wzroście mocy świetlnej i zmianie temperatury barwowej. Grupa inteligentnych lamp może współpracować ze sobą podejmując decyzję o zmianie wcześniej wymienionych parametrów bądź też przysyłać informację do użytkownika końcowego o podjęciu decyzji o ich zmianie, czego wynikiem może być poprawa warunków bezpieczeństwa na drodze dla pieszych i kierowców. Aspekty zmiany temperatury barwowej powinny być dostosowane do lokalizacji, w której znajdują się lampa, należy mieć na uwadze gęstość zabudowania domostw mieszkańców oraz występowanie zwierząt nocnych.

Oświetlenie uliczne często jest przewymiarowane, co przy szybkim poruszaniu pojazdu może stwarzać dyskomfort dla kierowcy i ryzyko oślnienia. Nadmiarowe oświetlenie może powodować chętniejsze spacerowanie w jasnych okolicach, lecz jest to wrażenie złudne, gdyż ilość światła nie zapewnia pełnego bezpieczeństwa. Nierzadko prowadzi to do potrąceń pieszych przez samochody, gdyż sztuczne światło zmniejsza czujność. Głównym czynnikiem wpływającym na poczucie bezpieczeństwa jest wysoki poziom oraz równomierność luminancji, którymi to parametrami może sterować system adaptacyjny.

W systemie adaptacyjnym występuje wiele przyczyn zmiany poziomu świecenia co sprawia, że monitorowanie pracy systemu oświetlenia jest jeszcze trudniejsze, dlatego niezbędne jest poszukiwanie metod skutecznego monitorowania.



3 Metodyka monitorowania anomalii w oświetleniu

Ze względu na znaczenie systemu oświetlenia dla miasta i jego mieszkańców musi on podlegać stałemu monitorowaniu. Monitorowaniu podlega przede wszystkim stan źródeł światła, awaria nawet pojedynczej lampy może wpływać na stan bezpieczeństwa lub płynność ruchu drogowego lub pieszego. Anomalia, czyli odstępstwo od oczekiwanego stanu lub zaplanowanego scenariusza działania jest uznawane za awarię i jest raportowane obsłudze systemu w celu podjęcia przez nią działań naprawczych. Monitorowaniu podlegać musi również ilość zużywanej energii elektrycznej przez system oświetlenia ze względu na znaczny udział systemu oświetlenia w całkowitym zużyciu energii oraz wysokie koszty energii.

W niniejszym rozdziale omówiono różne metody monitorowania systemów oświetlenia pod kątem wykrywania anomalii, w szczególności przy zastosowaniu metod analizy zużycia energii.

3.1 Monitorowanie stanu systemu oświetlenia

W tradycyjnych systemach, bez zdalnego nadzoru, wykrywanie awarii odbywa się poprzez bezpośrednią kontrolę polegającą na okresowej inspekcji systemu co jest związane z kosztami, które są znaczące w budżecie eksploatacyjnym. Ważnym elementem monitorowania są zgłoszenia od mieszkańców i użytkowników dróg, co jednak wymaga systemu przyjmowania i obsługi takich zgłoszeń.

Aby zmniejszyć koszty wykrywania awarii bez inwestowania w systemy nadzoru pojawiły się rozwiązania przejściowe, wykorzystujące technologię ICT bez modyfikacji infrastruktury. Jednym z takich rozwiązań jest specjalnie zaprojektowane urządzenie instalowane na dachu autobusów miejskich [53]. Urządzenie mierzy natężenie oświetlenia, rejestruje je wraz z danymi GPS oraz akcelerometru podczas nocnej jazdy autobusów. Dane porównywane są z danymi wzorcowymi zarejestrowanymi podczas testowej jazdy i na tej podstawie wykrywane są nie działające lampy oraz ich lokalizacja. Bardziej zaawansowane rozwiązanie zaproponowano w [54]. Urządzenie składa się z macierzy czujników oświetlenia montowanych na dachu samochodu, kamer video oraz modułu nawigacji inercyjnej IMU. Dodatkowo urządzenie wyposażono w interfejs ODB-II do odczytu prędkości pojazdu, a fuzja danych GPS, IMU i ODB zwiększyła dokładność danych o położeniu i ruchu samochodu. Poza wykrywaniem awarii lamp rozwiązanie umożliwiało pomiar wysokości montażu lamp oraz tworzenia map natężenia oświetlenia.

Opisane powyżej rozwiązania niewątpliwie usprawniają metodę okresowej inspekcji systemu, niemniej ich wadą jest to, że nie wykrywają awarii w czasie rzeczywistym. Między innymi z tego powodu zaczęto stosować systemy zdalnego nadzoru, tym bardziej, że rozwój technologii ICT w tym Internetu Rzeczy (IoT) sprawił, że instalacja systemów zdalnego nadzoru dla systemów oświetlenia stała się ekonomicznie uzasadniona. Systemy te



wykorzystują różne media komunikacyjne. Systemy oparte o komunikację przewodową stosują standard DALI [55] lub KNX [56]. W obu przypadkach komunikacja odbywa się przy użyciu dodatkowego przewodu dwużyłowego, co praktycznie uniemożliwia zastosowanie w przypadku już istniejących systemów oświetlenia i nadaje się tylko do nowych instalacji. Tych wad nie ma zastosowanie Power Line Communication (PLC) - komunikacja poprzez elektroenergetyczną sieć rozdzielczą [57]. Do zdalnego sterowania wykorzystuje się linie zasilające, tak więc nie jest potrzebna dodatkowa para przewodów. Możliwa jest więc modernizacja systemów bez wymiany okablowania. Jedną z wad komunikacji PLC jest stosunkowo duża energochłonność oraz zależność od topologii sieci energetycznej, dlatego też komunikację w systemach zdalnego zarządzania dominują technologie bezprzewodowe o różnych charakterystykach [58]:

- WPAN – Wireless Personal Area Network – sieci o małym zasięgu, np.: Bluetooth, Zigbee;
- WLAN – Wireless Local Area Network – sieci o średnim zasięgu, np.: Wifi, IEEE 802.11;
- WWAN – Wireless Wide Area Network – sieci o dużym zasięgu, np.: GSM, LTE, 5G;
- LPWAN - Low-Power Wide-Area Network – sieci dalekiego zasięgu o małej mocy, np.: LoraWan, SigFox.

Zdalny nadzór umożliwia monitorowanie pojedynczej lampy w czasie rzeczywistym, przy czym rozwiązania w tym zakresie są zróżnicowane w zależności od konstrukcji lampy. Każda lampa składa się ze źródła światła oraz elektronicznego urządzenia sterującego (Electronic Control Gear - ECG) i sposób monitorowania zależy od tego jakie są funkcje wbudowane w te elementy. Najbardziej zaawansowane rozwiązania wykorzystują funkcje wbudowane w ECG: monitorowanie prądu przepływającego przez źródło światła, monitorowanie napięcia na źródle światła, monitorowanie temperatury źródła światła. Opracowano szereg standardów takich jak D4i [55], które umożliwiają odczyt danych bezpośrednio z ECG przez system zdalnego nadzoru. Należy wspomnieć o standardach przemysłowych opracowywanych przez międzynarodową organizację Zhaga Consortium, w szczególności Księga 18 – „Smart interface between outdoor luminaires and sensing/communication modules” - Inteligentny interfejs między oprawami zewnętrznymi a modułami czujnikowymi/komunikacyjnymi. Skrótowo interfejs nazywa się Zhaga SR, co oznacza zarówno Sensor Ready (możliwość rozbudowy o czujniki) jak i System Ready (możliwość rozbudowy o system sterowania) [59].

Jeżeli brak tych funkcji, to monitoring jest możliwy na przykład przez wykorzystanie czujnika zmierzchowego wbudowanego w lampę [60], przy czym każda lampa musi być wyposażona w sterownik, który steruje załączeniem lampy w zależności od poziomu oświetlenia otoczenia a po załączeniu sprawdza czy poziom oświetlenia uległ odpowiedniej zmianie. Są też rozwiązania, które stosują dodatkowy fotorezystor instalowany poniżej źródła światła [61]. Jeśli poziom intensywności jest poniżej pewnego limitu, to interpretowane jest to

jako awaria. Inne sposoby monitorowania nieinwazyjnego to wykorzystanie miernika mocy, który mierzy pobór mocy przez grupę lamp, opisany w pracy [62]. Wprawdzie metoda detekcji jest prosta, progowa, ale po wykryciu znaczącej różnicy w stosunku do wartości referencyjnej, sprawdzany jest stan każdej z lamp poprzez indywidualne załączanie, wyłączenie i pomiar mocy. Bardziej zaawansowane metody opisano w [63], gdzie dane z liczników energii są przetwarzane za pomocą metod maszynowego uczenia. Zastosowane metody detekcji oparte o nienadzorowane klastrowanie pozwalają wykryć różne typy anomalii.

3.2 Wykrywanie anomalii w zużyciu energii

Stosunkowo nowym sposobem detekcji anomalii jest analiza poboru energii przez system oświetlenia. Wykorzystuje się do tego dane pochodzące z inteligentnych liczników energii elektrycznej (Smart Meter), które umożliwiają odczyt w czasie rzeczywistym parametrów sieci energetycznej: prądu, napięcia, mocy czynnej, współczynnika mocy. Inteligentne liczniki wraz z niezbędną infrastrukturą (AMI - Advanced Metering Infrastructure) są powszechnie instalowane przez operatorów sieci dystrybucyjnych. W Europie wskaźnik penetracji inteligentnych liczników energii elektrycznej osiągnął 56 % na koniec 2022 r [64], w Ameryce Północnej 74% na koniec 2021r. Dane pomiarowe o wysokiej rozdzielczości z liczników tworzą potężne zbiory danych Big Data i dostarczają bogatych informacji na temat zachowań konsumentów w zakresie zużycia energii elektrycznej. Gromadzenie danych z liczników nazywane jest również nieinwazyjnym monitorowaniem poboru energii (NILM – Non-Intrusive Load Monitoring) [65]. Ponieważ w domenie publicznej znajduje się wiele baz danych pomiarowych, umożliwiających testowanie modeli badawczych, są one przedmiotem licznych badań omówionych w pracy [66], w takich obszarach zastosowań jak tworzenie profili zużycia energii elektrycznej, prognozowanie zużycia energii oraz wykrywanie anomalii.

Profil zużycia energii elektrycznej to schemat przeciętnego zużycia energii elektrycznej przez odbiorcę końcowego w poszczególnych godzinach doby i reprezentuje przebieg zapotrzebowanie na energię w ciągu dnia, tygodnia lub roku. Profile zdefiniowane są z określoną rozdzielczością czasową, im częstsze pomiary, tym bardziej szczegółowy profil zużycia energii, typowe okresy to 15, 30 i 60 minut. Schemat obciążenia definiuje obciążenie szczytowe, czyli okres, w którym zapotrzebowanie na energię elektryczną osiąga najwyższą wartość, obciążenie podstawowe, czyli minimalny poziom zapotrzebowania na energię elektryczną, który utrzymuje się niezmiennie przez cały dzień, oraz wahania sezonowe w zużyciu energii elektrycznej. Różne typy konsumentów wykazują różne kształty obciążenia w oparciu o ich wzorce konsumpcji, takie jak klienci indywidualni, komercyjni lub przemysłowi [67].

Prognozowanie zużycia energii, to proces przewidywania przyszłego zapotrzebowania na energię elektryczną lub obciążenia na podstawie historycznych danych o zużyciu zebranych

z inteligentnych liczników. W oparciu o profile o zużycia energii elektrycznej szacuje się, ile energii elektrycznej będzie potrzebne w różnych momentach w przyszłości, zwykle w perspektywie krótkoterminowej - godzinowej lub dziennej oraz długoterminowej - miesięcznej lub rocznej [68], [69]. Prognozowanie zużycia energii jest powszechnie stosowane przez przedsiębiorstwa branży energetycznej, aby skutecznie planować i zarządzać wytwarzaniem, przesyłaniem i dystrybucją energii elektrycznej.

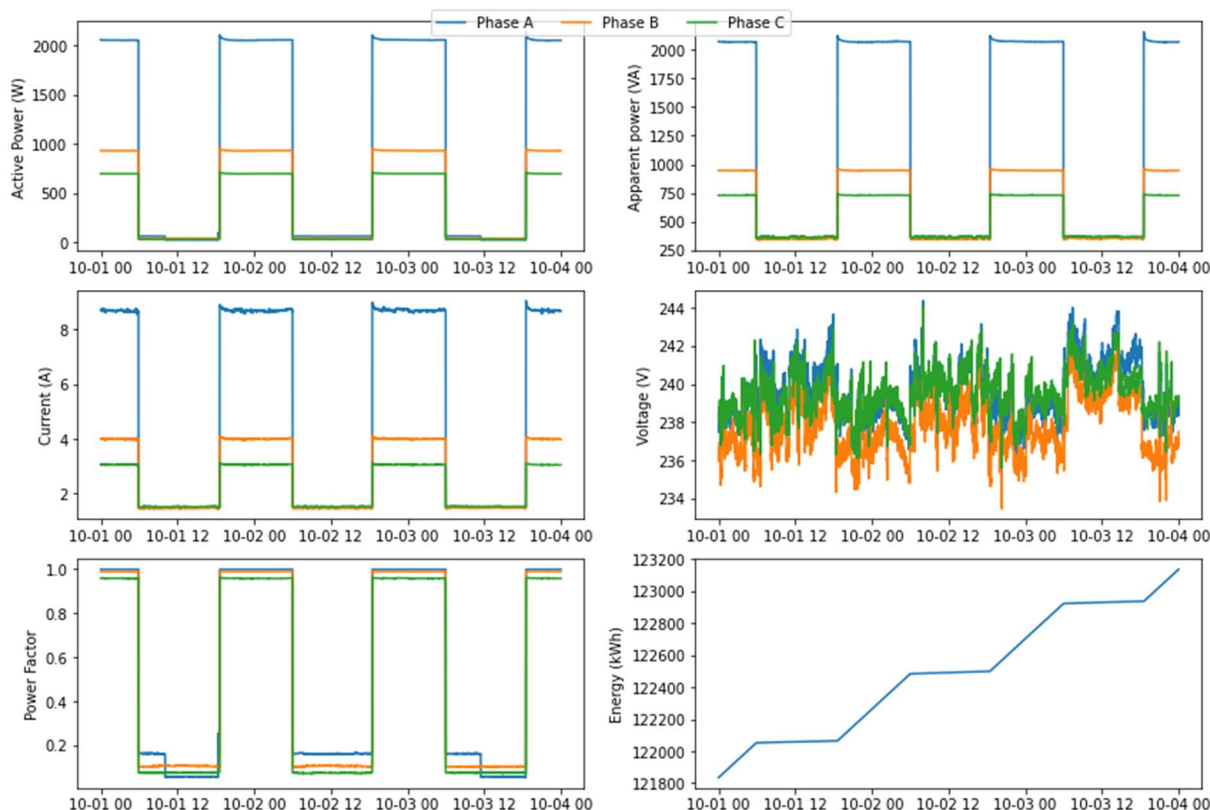
Anomalia w zużyciu energii odnosi się do nieregularnych lub nieoczekiwanych odchyłeń od normalnych wzorców zużycia energii. Praktycznie można przyjąć, że anomalia występuje, gdy rzeczywiste zużycie energii znacznie różni się od tego, czego zwykle oczekuje się na podstawie danych historycznych, wahań sezonowych lub innych ustalonych wzorców. Anomalie mogą wystąpić z powodu różnych czynników, takich jak awarie sprzętu, błędy podczas gromadzenia danych, komunikacji lub wprowadzania danych, warunki pogodowe, błędy ludzkie lub zmiany w zachowaniu użytkowników [70]. Należy tu wspomnieć, że błędne dane o zużyciu energii mogą się pojawić również w wyniku intencjonalnego przestępczego ingerowania w infrastrukturę AMI oraz manipulowaniu odczytami inteligentnych liczników w celu zmniejszenia zużycia energii przez klientów i rachunków za prąd [71].

Wykrywanie anomalii w zużyciu energii jest problemem dobrze zbadanym w zakresie różnych typów odbiorców: mieszkania [72], budynki użytkowe [73], przemysł [74]. Na podstawie danych z inteligentnych liczników można wykryć zmianę zachowania mieszkańców, np. w pracy [75] zbadano możliwość prognozowania bezrobocia wśród osób mieszkających w jednym gospodarstwie domowym.

Aby omówić metody stosowane w analizie anomalii w zużyciu energii należy określić charakter danych rejestrowanych przez inteligentne liczniki. Typowy licznik trójfazowy odczytuje następujące parametry sieci energetycznej:

- Całkowita energia
- Napięcie fazowe V_A , V_B , V_C
- Prąd I_A , I_B , I_C
- Moc czynna P_A , P_B , P_C
- Moc pozorna S_A , S_B , S_C
- Współczynnik mocy PF_A , PF_B , PF_C

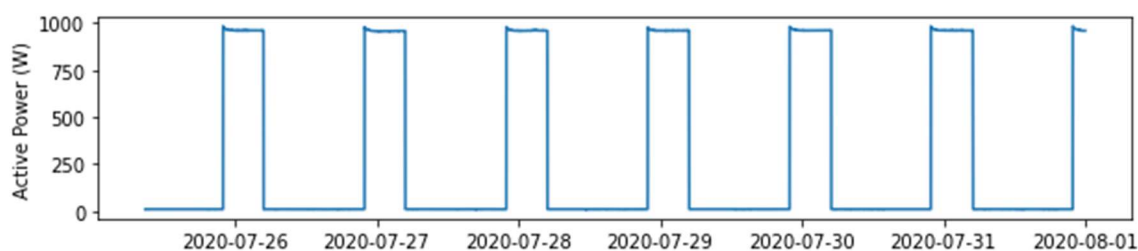
Na rys. 2 przedstawiono przykład odczytów danych z rzeczywistego licznika mierzącego energię systemu oświetleniowego, na osiach poziomych są uwidocznione daty, a na osiach pionowych wielkości pomiarowe.



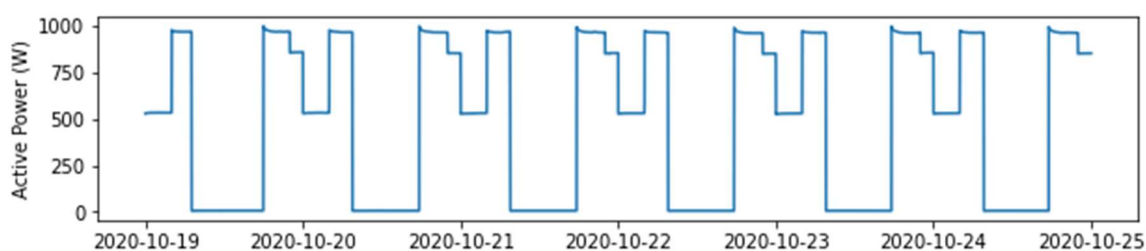
Rys. 2. Przykładowe zapisy inteligentnego licznika

Odczyty danych z liczników energii dokonywane są ze stałą częstotliwością, której wartość zależy od zastosowania, typowo są to 4 odczyty na godzinę, ponieważ 15 minut to standardowy okres dla ustalania profili zużycia energii dla instalacji użytkowych. Odczytane dane rejestrowane są ze znacznikiem czasowym, czyli tworzą sekwencję danych w czasie dyskretnym co oznacza, że mają charakter szeregów czasowych. Do analizy zużycia energii wykorzystuje się przede wszystkim zapisy mocy czynnej, która reprezentuje czas i poziom świecenia lamp. Pozostałe parametry energii elektrycznej mogą być również wykorzystane do analizy anomalii, na przykład na podstawie wartości napięcia można wykryć, czy wystąpiła awaria zasilania (blackout) lub zakłócenie zasilania (brownout) ale nie jest to przedmiotem niniejszej pracy. Przykładowy wykres mocy czynnej jednej z faz dla lamp bez redukcji, czyli załączanych o zmierzchu i wyłączanych o świcie przedstawiono na rys. 3, zaś na rys. 4 przedstawiono wykres mocy czynnej dla lamp, w których zastosowano redukcję poziomu oświetlenia według następującego harmonogramu:

- Zmierzch – załączenie lamp 100 %
- 22:00 – redukcja do 70 %
- 00:00 – redukcja do 40 %
- 04:00 – wyłączenie redukcji, powrót do 100 %
- Świt - wyłączenie

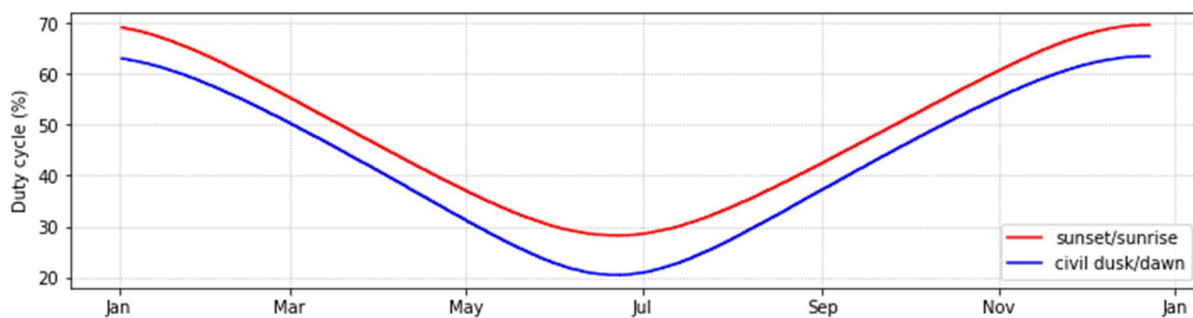


Rys. 3. Wykres mocy czynnej jednej fazy dla lamp bez redukcji



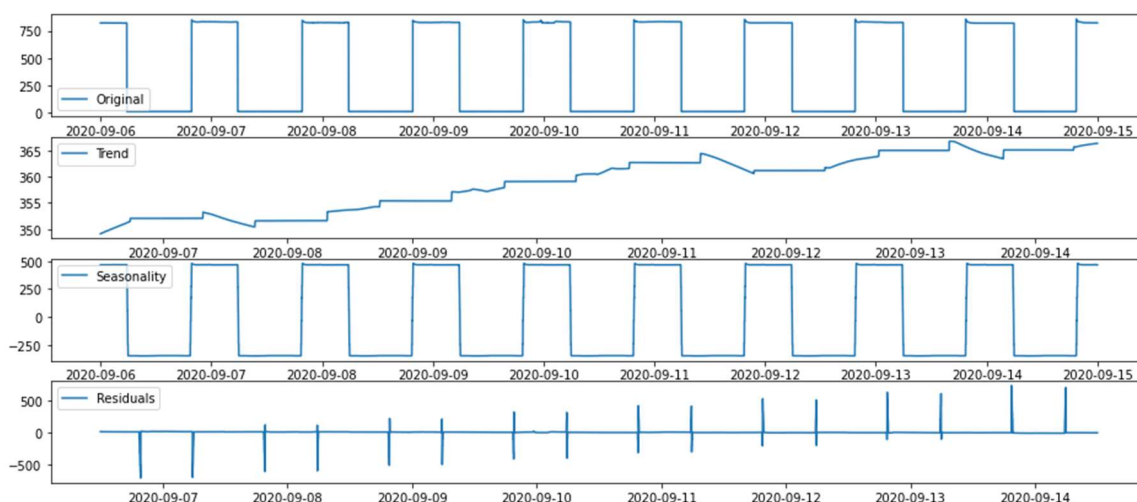
Rys. 4. Wykres mocy czynnej jednej fazy dla lamp z redukcją

Szereg czasowy mocy czynnej jest jednowymiarowym szeregiem czasowym (Univariate Time Series). Dla systemów oświetlenia szeregi czasowe mocy czynnej wykazują istotną cechę – okresowość, ponieważ bez względu na schemat sterowania systemem oświetlenia, czy jest on prosty czy adaptacyjny, lampy powinny być załączone po zmierzchu i wyłączone po świcie. Dla prostego sposobu sterowania oświetleniem drogowym, bez redukcji, przebieg ma charakter unipolarnej fali prostokątnej z okresem 24 h, ze zmiennym współczynnikiem wypełnienia, ponieważ długość dnia i nocy ulega zmianie w ciągu roku. Współczynnik wypełnienia jest proporcjonalny do długości nocy (czy też dnia), co oznacza, że wartość średnia przebiegu ma charakter okresowy z okresem jednego roku. Ponieważ długość nocy jest związana z położeniem we współrzędnych geograficznych, dla każdej lokalizacji zmiany wartości średniej będą się różniły. Na rys. 5 przedstawiono wykres współczynnika wypełnienia poboru mocy dla miasta Gdańska, dla początku i końca świecenia wyliczonego według zachodów i wschodów słońca, czyli momentów przejścia tarczy słońca pod horyzont. Na rysunku tym przedstawiono również wykres współczynnika wypełnienia według zmierzchów i świtów cywilnych, czyli takich momentów, kiedy środek tarczy słonecznej znajdzie się w punkcie położonym 6 stopni poniżej horyzontu. Do wyliczeń zastosowano algorytm opisany przez Earth System Research Laboratory agencji NOAA [76], bazujący na metodzie zdefiniowanej w literaturze [77]. Algorytm ten jest powszechnie stosowany w systemach sterowania oświetleniem [78].

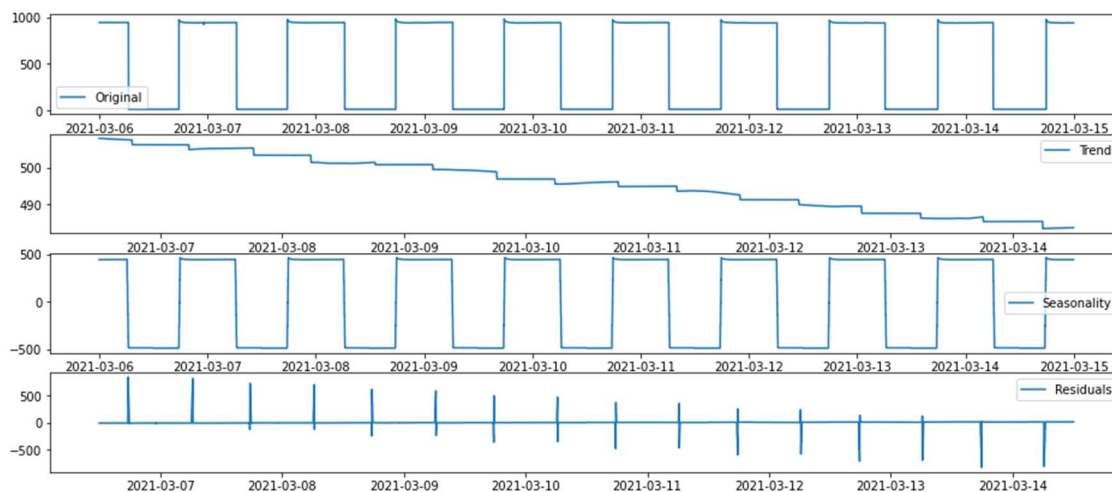


Rys. 5. Współczynnik wypełnienia impulsu dla poboru mocy dla miasta Gdańska

Ponieważ szereg czasowy mocy czynnej jest sygnałem okresowym, to potrzebne jest określenie składników: trendu, szumu i składowej okresowej. Stosując metodę dekompozycji w oparciu o średnią kroczącą i addytywny model (Seasonal decomposition using moving averages) uzyskujemy rezultat jak na rys. 6. Dla innej pory roku (rys. 7) rezultat dekompozycji będzie się różnił trendem. Wynika to z tego, że w zależności od pory roku, pora nocna, a w związku z tym długość świecenia, wydłuża się lub skraca. Taka zmienność poboru mocy sprawia, że mamy do czynienia z niestacjonarnym szeregiem czasowym.



Rys. 6. Dekompozycja szeregu czasowego mocy czynnej – okres jesienny



Rys. 7. Dekompozycja szeregu czasowego mocy czynnej – okres wiosenny



Omawiając metody detekcji anomalii należy wprowadzić podział na typy anomalii [79]. Podstawowy typ to anomalie punktowe, które odnoszą się do poszczególnych punktów danych, które znacznie różnią się od reszty zbioru danych. Te anomalie wyróżniają się jako pojedyncze instancje danych, które wykazują nietypowe zachowanie. Dla systemu oświetlenia może to być ekstremalna wartość mocy czynnej. Kolejny typ to anomalie kontekstowe, które występują, gdy punkt danych jest uważany za nieprawidłowy tylko w określonym kontekście lub podzbiorze danych. Te anomalie mogą być normalne w jednej sytuacji, ale nienormalne w innej. W przypadku systemu oświetlenia wartość mocy czynnej bliska zeru jest normalna w dzień, w nocy może być anomalią. W przypadku szeregów czasowych szczególne znaczenie mają anomalie czasowe, które występują, gdy dane poprzez analizę składnika czasowego wykazują nietypowe zachowanie w porównaniu z wzorcami historycznymi. Dla systemu oświetlenia anomalia czasowa wystąpi, gdy na przykład zmienia się godziny stosowania redukcji mocy. Są jeszcze inne typy anomalii: grupowe, przestrzenne, kontradiktoryjne, ale nie mają one tu zastosowania.

Istnieje wiele metod wykrywania anomalii w szeregach czasowych [80], [81]. Metody te można podzielić na metody nadzorowane i nienadzorowane, w zależności od tego czy dane historyczne zostały zaetykietowane, czyli wskazano, które dane są poprawne, czyli „normalne”, a które są odstępstwem. Etykietowanie może być zadaniem czasochłonnym zwłaszcza w przypadkach, gdy anomalie występują stosunkowo rzadko i potrzebna jest analiza zbiorów danych z długich okresów. W praktycznych rozwiązaniach systemu monitorowania wskazane jest stosowanie metod nienadzorowanych, samouczących na podstawie treningu opartego na danych, które nie mają etykiet, ani konkretnego, zdefiniowanego wyniku. System uczy się identyfikować złożone procesy i wzorce bez udziału człowieka udzielającego ścisłych, stałych wskazówek.

W algorytmach wykrywania anomalii stosuje się metody probabilistyczne, statystyczne, metody uczenia maszynowego (ML – Machine Learning) oraz metody wykorzystujące rekurencyjne sieci neuronowe (RNN – Recurrent Neuron Networks). W ostatnim czasie najczęściej badane są metody ML i RNN [82], [83].

W niniejszej pracy przebadano następujące metody detekcji anomalii stosujące maszynowe uczenie:

- Autoregresja oparta o model AutoRegressive Integrating Moving Average w wersji okresowej (SARIMA),
- RNN stosującej sieć Long Short Term Memory (LSTM)
- RNN stosującej sieć Gated Recurrent Units (GRU)
- Autoenkoder oparty o sieć LSTM
- Autoenkoder oparty o sieć GRU
- Autoenkoder oparty o sieć splotową jednowymiarową (1D Convolutional).

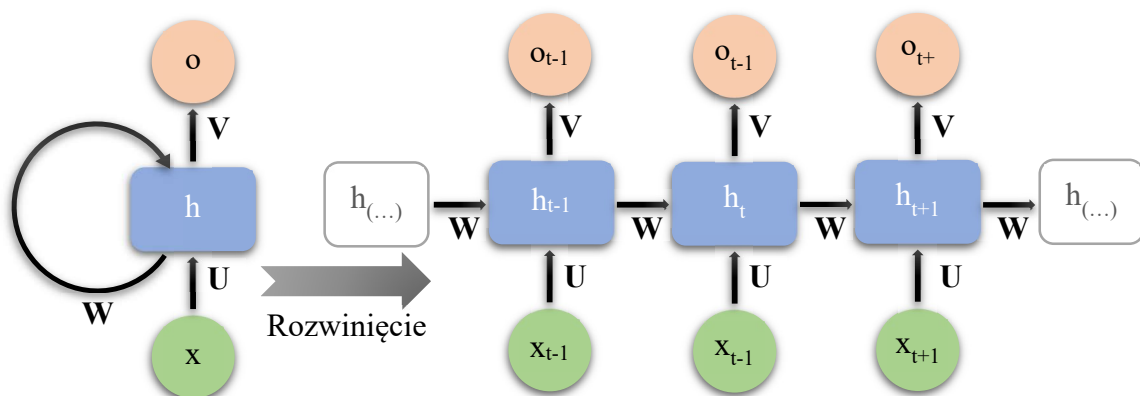
3.3 Rekurencyjne sieci neuronowe

Najnowsze metody modelowania danych oparte są o techniki głębokiego uczenia maszynowego - Deep Learning [84], stosujące wielowarstwowe sieci neuronowe. Do analizy danych sekwencyjnych, takich jak szeregi czasowe, stosuje się rekurencyjne sieci neuronowe (RNN – Recurrent Neural Network), przeznaczone do pracy z danymi sekwencyjnymi, takimi jak dane szeregów czasowych, wyposażone w zdolność do uwzględniania kolejności i zależności czasowych w danych. Osiąga się to przez połączenia zwrotne, które pozwalają zachować w sieci ukryte stany lub pamięć wcześniejszych danych wejściowych. Połączenia zwrotne które tworzą pętlę, dzięki czemu stan ukryty w kroku czasowym „t” zależy nie tylko od wejścia w tym kroku czasowym, ale także od stanu ukrytego z poprzedniego kroku czasowego „t-1”. Schemat takiej sieci przedstawiono na rys. 8, zaś na rys. 9 przedstawiono standardową sieć RNN z powielającym się modułem zawierającym jedną warstwę ukrytą. W każdym kroku czasowym RNN oblicza stan ukryty przy użyciu bieżącego wejścia i stanu ukrytego z poprzedniego kroku czasowego jak przedstawiono to w równaniach 3.1 ... 3.3, gdzie: \mathbf{x} – sekwencja wejściowa, \mathbf{o} – sekwencja wyjściowa, \mathbf{h} – ukryty stan, \mathbf{W} - macierz wag dla połączenia wejście-ukryty, \mathbf{U} - macierz wag dla połączenia ukryty-wyjście, \mathbf{V} – macierz wag dla połączenia ukryty-ukryty, \mathbf{b} i \mathbf{c} wektory stałych przesunięć.

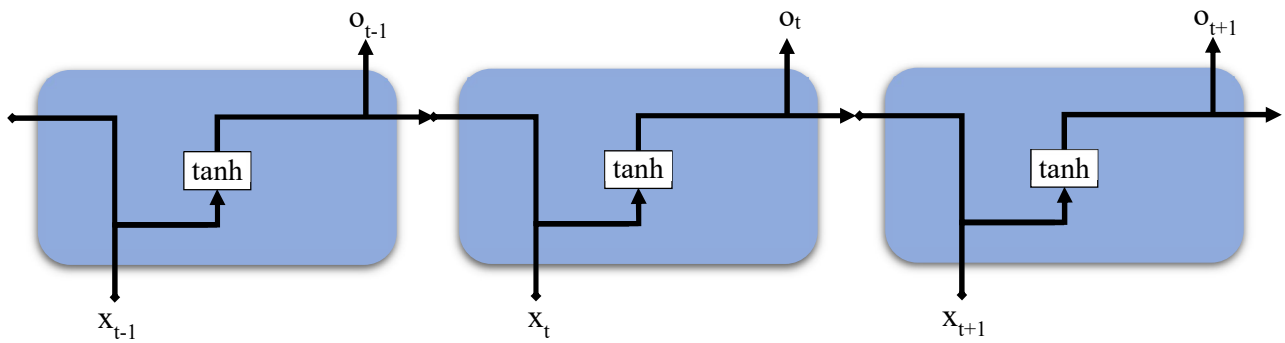
$$\mathbf{h}_t = \tanh(\mathbf{b} + \mathbf{W} \cdot \mathbf{h}_{t-1} + \mathbf{U} \cdot \mathbf{x}_t) \quad (3.1)$$

$$\mathbf{o}_t = \mathbf{c} + \mathbf{V} \cdot \mathbf{h}_t \quad (3.2)$$

$$\hat{\mathbf{y}}_t = \sigma(\mathbf{o}_t) \quad (3.3)$$



Rys. 8. Sieć RNN w postaci symbolicznej notacji i rozwinięcia. Oznaczenia: \mathbf{x} – sekwencja wejściowa, \mathbf{o} – sekwencja wyjściowa, \mathbf{h} – ukryty stan, \mathbf{W} , \mathbf{U} , \mathbf{V} – macierze wag.



Rys. 9. Standardowa sieć RNN z powielającym się modulem zawierającym jedną warstwę ukrytą.

Trenowanie sieci RNN dokonuje się przez wsteczną propagację w czasie - Back Propagation Through Time (BPTT). Dla dłuższych sekwencji występuje zjawisko zanikającego gradientu, polegające na tym gradienty używane do aktualizacji parametrów sieci stają się z czasem bardzo małe. Aby rozwiązać ten problem, opracowano bardziej zaawansowane architektury RNN, takie jak sieci Long Short-Term Memory (LSTM) i Gated Recurrent Units (GRU), które mają mechanizmy bramkowania, które pozwalają im zatrzymywać i aktualizować informacje w dłuższych sekwencjach.

Sieci LSTM [85] składają się ze specyficznych komórek połączonych warstwami, wyposażonymi w pamięć (komórka stanu) i 3 nieliniowe bramki: wejściową - decydującą w jaki sposób wejście aktualizuje stan pamięci, zapominania - decydującą jak wartości z poprzedniego stanu aktualizują stan pamięci, wyjściową - decydującą o tym, jakie wartości powinny znaleźć się na wyjściu, wyznaczone w oparciu o dane wejściowe i stan pamięci. Na rys. 10 przedstawiono schemat bloku konstrukcyjnego sieci LSTM. Komórka LSTM działa jak mini-maszyna stanów, która wykorzystuje wewnętrzną komórkę pamięci do utrzymywania wartości stanu przez dłuższy czas, a bramki posiadają wagi, które wyliczane są podczas procedury szkoleniowej. Sposób obliczania stanów przedstawiono to w równaniach 3.4 ... 3.9, gdzie: f - stan bramki zapominania i - stan bramki wejściowej, o - stan bramki wyjściowej, C - stan komórki stanu, W_f, W_i, W_o, W_C - macierze wag poszczególnych warstw, b_f, b_i, b_o, b_C , wektory stałych przesunięć dla poszczególnych warstw.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.4)$$

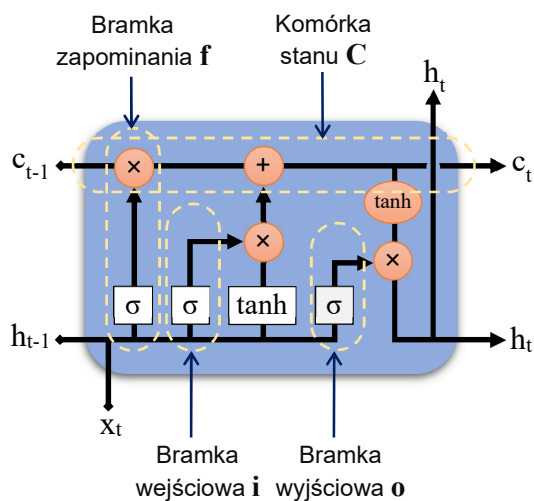
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.7)$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \quad (3.8)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad (3.9)$$



Rys. 10. Schemat bloku konstrukcyjnego sieci LSTM

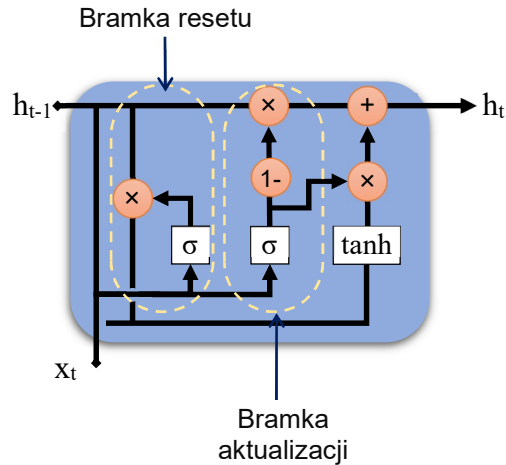
Sieci GRU są uproszczoną wersją LSTM, ponieważ wyposażone są tylko w dwie bramki: resetu i aktualizacji. Ze względu na prostszą architekturę z mniejszą liczbą parametrów modele GRU są generalnie szybsze w trenowaniu i wymagają mniej pamięci w porównaniu z modelami LSTM. Na rys. 10 przedstawiono schemat bloku konstrukcyjnego sieci GRU. Sposób obliczania stanów przedstawiono to w równaniach 3.10 ... 3.13, gdzie: \mathbf{z} – stan bramki aktualizacji, \mathbf{r} – stan bramki resetowania.

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_z) \quad (3.10)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_r) \quad (3.11)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_h) \quad (3.12)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad (3.13)$$

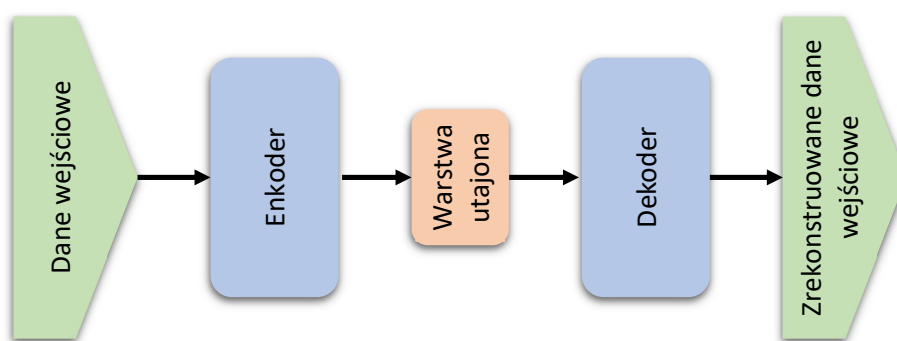


Rys. 11. Schemat bloku konstrukcyjnego sieci GRU

W zadaniach uczenia maszynowego bez nadzoru bardzo ważną rolę odgrywa rodzaj sztucznej sieci neuronowej zwanej Autoenkoder. Architektura Autoenkodera przedstawia się składa się z następujących części:

- enkodera, która pobiera dane wejściowe i odwzorowuje je na reprezentację przestrzeni utajonej o niższych wymiarach; enkoder wykonuje zadanie redukcji wymiarowości i odpowiada za stworzenie skompresowanej reprezentacji danych;
- dekodera, która pobiera skompresowaną reprezentację i próbuje zrekonstruować z niej oryginalne dane; dekodek rzutuje dane z powrotem do ich oryginalnej wymiarowości;
- warstwy utajonej zawierającej skompresowaną reprezentację danych wejściowych.

Do analizy danych sekwencyjnych, takich jak szeregi czasowe, koder i dekodek mają architekturę rekurencyjnej sieci neuronowej, najczęściej stosuje się sieć typu LSTM lub GRU.



Rys. 12. Schemat Autoenkodera

Autoenkoder jest szkolony w sposób nienadzorowany, dlatego podczas uczenia celem jest zminimalizowanie błędu rekonstrukcji między oryginalnymi danymi wejściowymi a wyjściem dekodera. Jednym z głównych zastosowań jest wykrywanie anomalii, ponieważ można trenować go na normalnych danych i używać do wykrywania anomalii lub wartości odstających, gdy prezentowane są dane znacznie odbiegające od normalnych wzorców.

4 Detekcja anomalii w systemie oświetleniowym

W niniejszym rozdziale omówiona zostanie zbadana praktyczna możliwość wykrywania w czasie rzeczywistym anomalii w systemie oświetlenia drogowego w oparciu o analizę danych z inteligentnych liczników energii. Do analizy algorytmów zostaną zastosowane rzeczywiste dane pochodzące z odczytów z rzeczywistego systemu inteligentnego oświetlenia drogowego.

Zastosowanie uczenia maszynowego do detekcji anomalii polega na stworzeniu modelu, który z maksymalną dokładnością odzwierciedla badany proces. W oparciu o stworzony model generowana jest prognoza, która następnie jest porównywana z rzeczywistym pomiarem. Różnica jest wskazaniem, że nadzorowany proces odbiega od wzorca, czyli jest anomalią. Do oceny różnicy służą różne metryki, a w przypadku metod nienadzorowanych stosuje się próg detekcji.

Algorytmy wykrywania anomalii można podzielić na algorytmy typu „offline” [87] i „online”. Podstawowa różnica między nimi jest następująca: dla algorytmu offline zakłada się, że pełny zestaw danych jest dostępny, zaś wykrycie anomalii jest równoznaczne z odnalezieniem wszystkich istniejących punktów spełniających przyjęte kryterium. Algorytm online przyjmuje, że dane będą dostępne punkt po punkcie w czasie rzeczywistym, zaś wykrycie awarii powinno nastąpić w określonym czasie. Praktyczne zastosowanie detekcji awarii na podstawie odczytu liczników energii wymaga, aby był to algorytm typu online. Chodzi bowiem o to, aby w chwili pojawienia się nowego pomiaru można było podjąć decyzję czy wartość jest zgodna z oczekiwaniem, czy jest niezgodna i w związku z tym należy zaalarmować obsługę. Podjęcie decyzji musi zapaść w ograniczonym czasie, nie przekraczającym okresu pojawiania się pomiarów.

Praktyczne podejście do wykrywania anomalii stawia też wymaganie, aby algorytmy były oparte o nienadzorowane samouczenie, czyli analizowały nieetykietowanym dane, ponieważ wstępna klasyfikacja procesów przez użytkownika systemów monitoringu może być dla niego zbyt obciążająca. W nienadzorowanym wykrywaniu anomalii dane są klasyfikowane wyłącznie na podstawie wewnętrznych właściwości zbioru danych, dzięki temu nadaje się do scenariuszy, w których anomalie są rzadkie i trudne do oznaczenia.

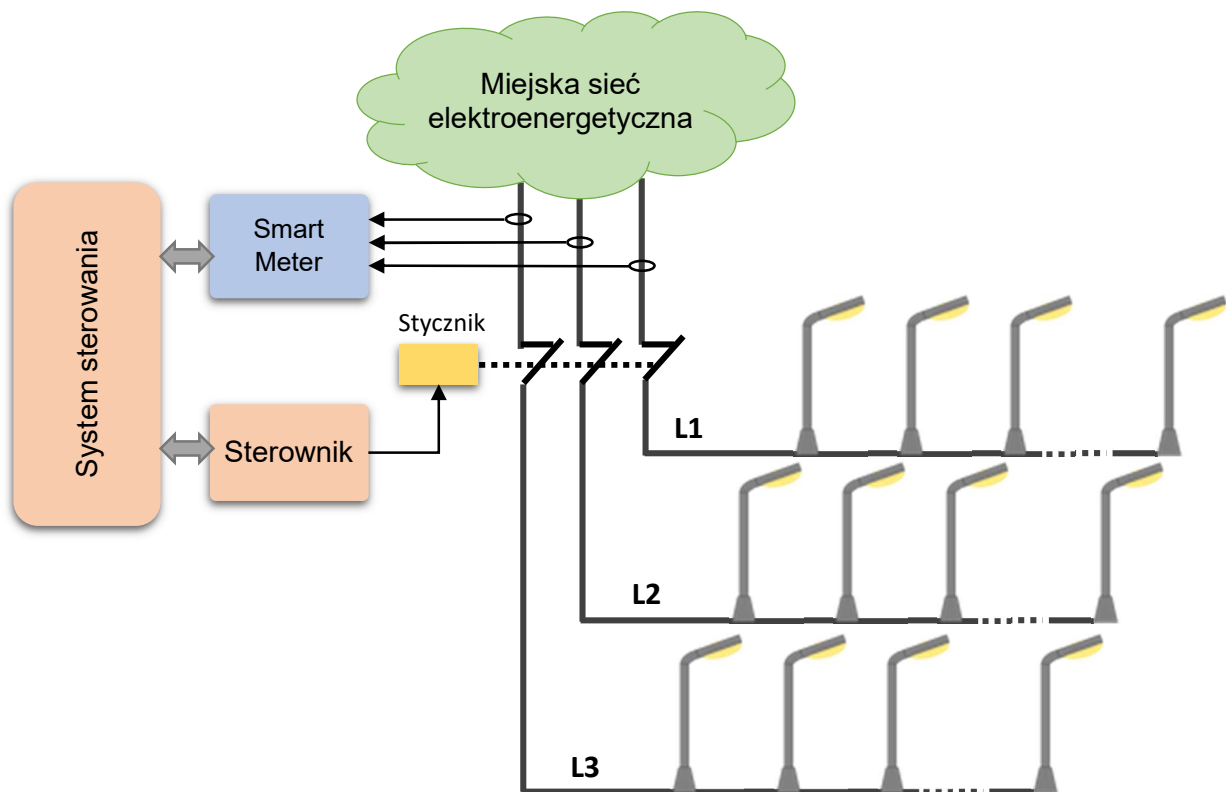
Algorytmy też muszą zakładać ograniczone zasoby wykorzystywane do przetwarzania danych, aby można było implementować te algorytmy w urządzeniach przemysłowych, działających w trudnych warunkach.

4.1 Baza danych pomiarów energii

Bazy danych pomiarów z inteligentnych liczników są w większości administrowane przez firmy energetyczne. Ponieważ bazy te mogą zawierać dane wrażliwe oraz powiązane z

bezpieczeństwem, wiele firm waha się przed udostępnieniem tych baz opinii publicznej. W ciągu ostatnich kilku lat udostępniono publicznie kilka zanonimizowanych lub częściowo zanonimizowanych zbiorów danych dotyczących gospodarstw domowych. Wykazy takich zestawów danych dostępnych publicznie zawarto np. w pracach [66], [86], [88], [89]. Jednak dotychczas nie udostępniono baz danych zużycia energii przez systemy oświetlenia drogowego, dlatego też dla potrzeby niniejszej pracy utworzono taką bazę danych. Baza powstała w oparciu o dane pochodzące z systemu sterowania oświetleniem, zainstalowanego w jednym z polskich miast i została udostępniona poprzez stronę związaną z projektem inteligentnego chmurowego oświetlenia pod adresem www.multimed.biz/infolight.

System sterowania oświetleniem dróg i miejsc publicznych jest zaprojektowany w reprezentatywny sposób dla tego typu instalacji, to znaczy oparty jest o rozdzielnice sterowania oświetleniem - szafy oświetleniowe, które służą do rozdziału energii, sterowania momentem włączenia i wyłączenia oświetlenia oraz do zabezpieczenia elementów przed zwarciami i przeciążeniami. Lampy grupowane są w obwody podłączone do tych szaf, a ponieważ zasilanie szaf jest trójfazowe, więc każda szafa ma trzy obwody, jak to przedstawiono na rys. 13. Do każdego obwodu podłączona jest pewna liczba lamp, zależna od konfiguracji danej ulicy, drogi czy parku. Optymalne rozwiązanie to równomierne rozłożenie lamp pomiędzy obwodami, jednak często nie jest to możliwe. Najczęściej do jednego obwodu jest podłączone od kilkunastu do kilkudziesięciu lamp.



Rys. 13. Ilustracja sposobu prowadzenia pomiarów mocy w szafie sterującej oświetleniem ulicznym

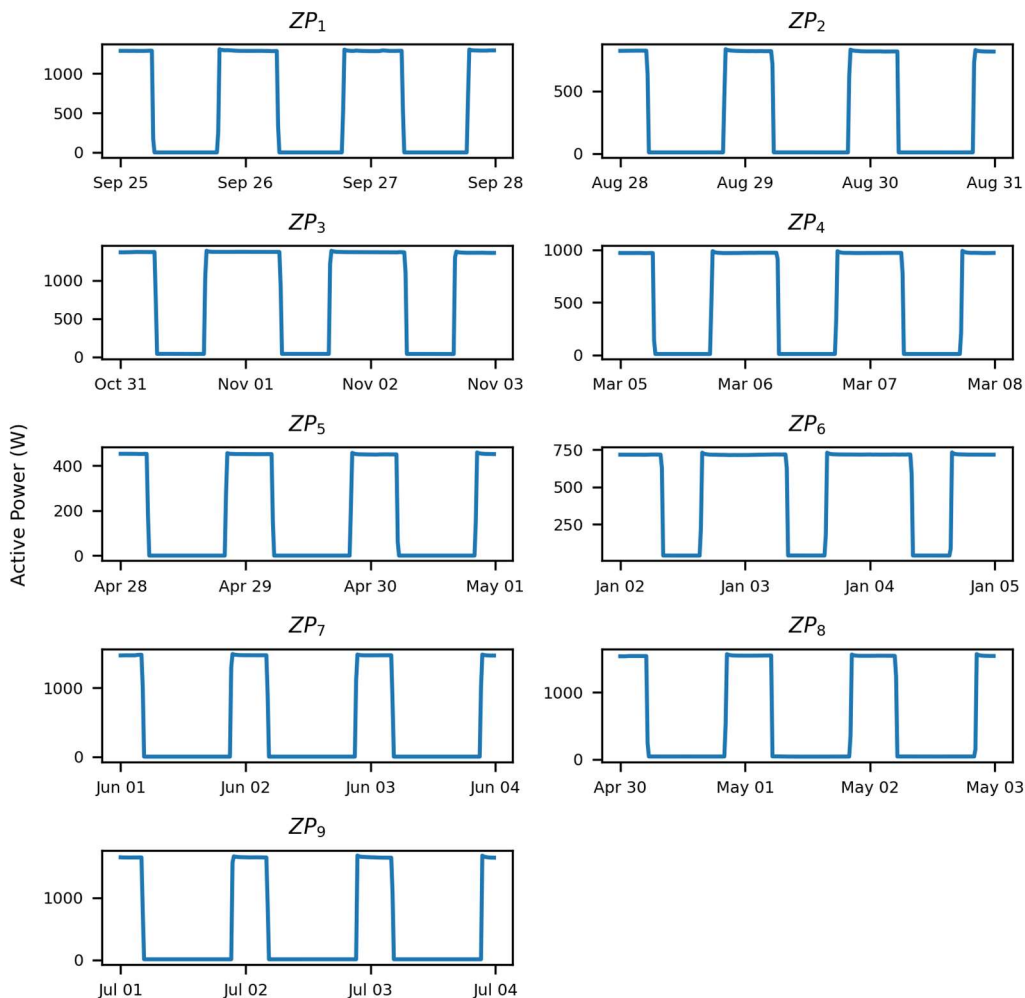
Do obwodów przyporządkowanych poszczególnym fazom podłączonych jest od kilku do kilkudziesięciu lamp. W opisywanej instalacji średnia liczba lamp na obwód, to 20, a ponad 90% obwodów ma mniej niż 50 lamp. Lampy połączone do danego obwodu mogą mieć różne moce, w tej instalacji zakres mocy wynosi od 52W do 143W. Średnio do jednego obwodu połączone są lampy o sumarycznej mocy znamionowej 1200 W. Dla takiej mocy znamionowej wyłącznie lampy o mocy 52 W powoduje zmniejszenie odczytu mocy czynnej o 4.3%, co jest wartością przewyższającą rozdzielczość liczników energii. W tej instalacji zainstalowano liczniki, które realizują pomiar energii czynnej w klasie 0.5S, zgodnie z normą IEC/EN 62053-22.

System, zainstalowany w Słupsku przez przedsiębiorstwo zatrudniające autora rozprawy zarządza ponad 4400 inteligentnymi lampami LED i wyposażony jest w 90 inteligentnych trójfazowych liczników energii, zainstalowanych w szafach sterujących. Dane z liczników odczytywane są z 60-sekundowym okresem i przekazywane są do centralnej bazy danych. Każdy rekord zawiera następujące dane:

- Identyfikator licznika
- Data/czas
- Całkowita energia
- Napięcie fazowe V_A , V_B , V_C
- Prąd I_A , I_B , I_C
- Moc czynna P_A , P_B , P_C
- Moc pozorna S_A , S_B , S_C
- Współczynnik mocy PF_A , PF_B , PF_C

W bazie zarejestrowano ponad 90 mln. rekordów z okresu od czerwca 2020 do grudnia 2022. Dla potrzeb niniejszej pracy dokonano przeglądu tych danych i wybrano dwie grupy zestawów danych, jednej zawierającej dane regularne bez anomalii oraz drugiej, zawierającej dane z anomaliami. Dane w zestawach poddano operacji czyszczenia danych (data cleaning) polegającej na uzupełnieniu ciągu próbek o brakujące zapisy. Brakujące zapisy wynikają z tego, że zapisywane są w czasie rzeczywistym i każda przerwa w działaniu urządzenia, wynikająca przykładowo z restartu, skutkuje okresowym brakiem zapisów.

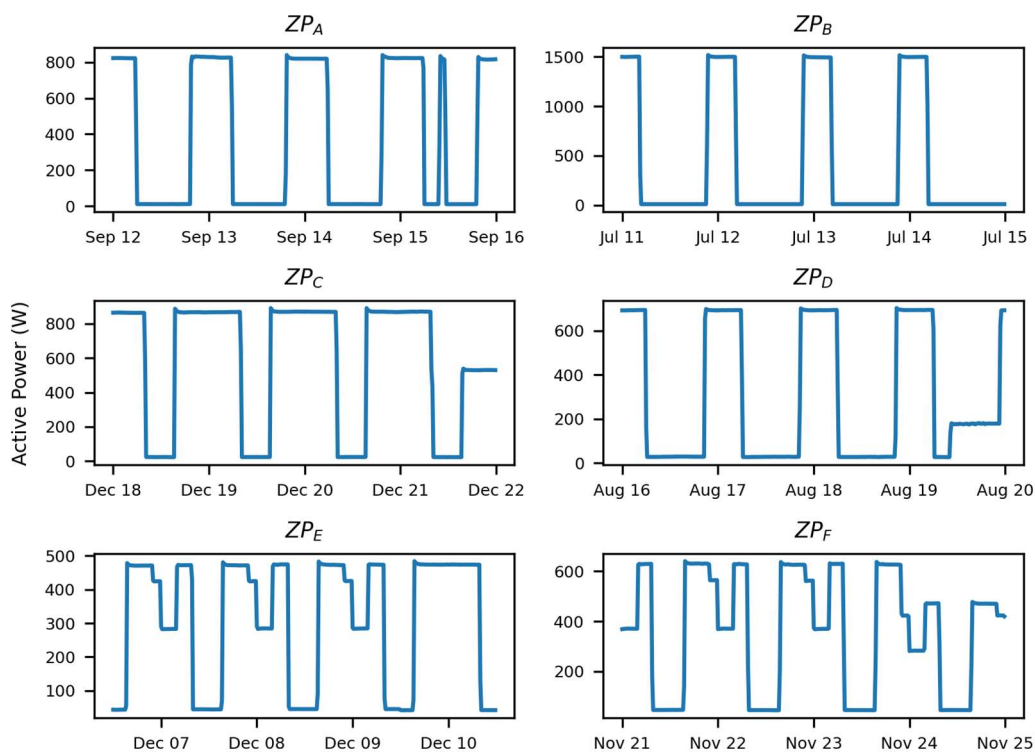
Grupa danych regularnych składa się 9 zbiorów danych oznaczanych **ZP₁**, **ZP₂**, ... **ZP₉**. Dane te pochodzą z różnych liczników dla różnych pór roku, zawierają zróżnicowane wartości amplitud pomiaru mocy, gdyż mierzą różne obwody. Na rys. 14 zwizualizowano fragmenty tych zbiorów danych.



Rys. 14. Przykładowe wizualizacje danych z pomiarów mocy dla ZP_1 , ZP_2 , ... ZP_9

Grupa danych zawierających anomalie składa się sześciu zbiorów danych oznaczonych ZP_A , ZP_B , ... ZP_F . Na rys. 15 zwizualizowano fragmenty tych zbiorów danych. Wykresy A, B, C, D przedstawiają anomalie dla sterowania bez redukcji, wykresy E i F dla sterowania z redukcją:

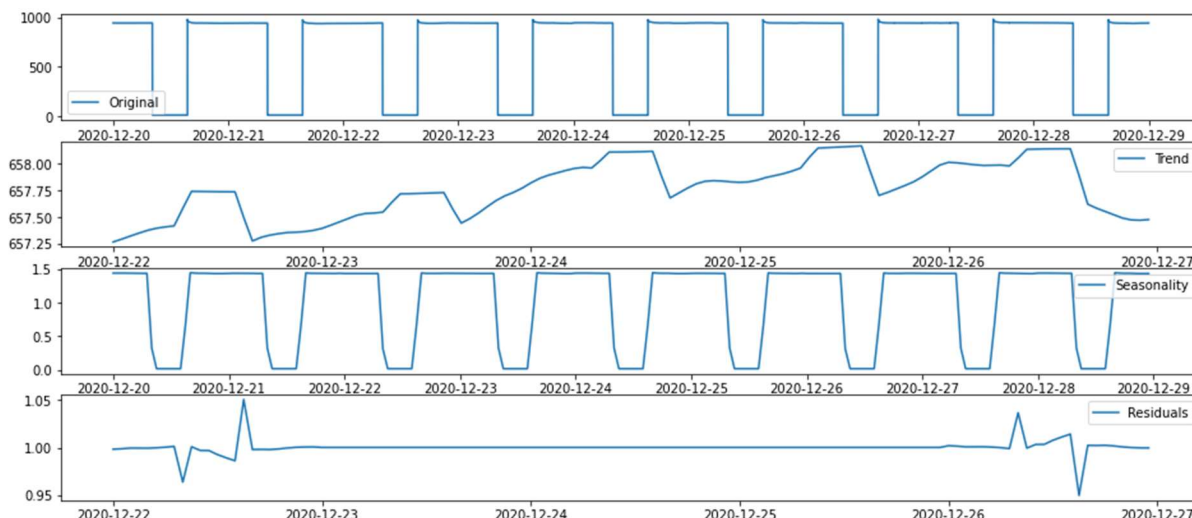
- A – załączenie lamp w dzień
- B – brak załączenie lamp w nocy
- C – część lamp nie działa – nastąpił spadek pobieranej mocy
- D – załączenie grupy lamp w dzień lub kradzież energii
- E – wyłączenie harmonogramu redukcji
- F - część lamp nie działa – nastąpił spadek pobieranej mocy



Rys. 15. Przykładowe dane z pomiarami mocy dla ZP_A, ZP_B, ... ZP_F

Wybór właściwej rozdzielczości czasowej w analizie szeregów czasowych jest ważną decyzją, która ma wpływ na dokładność oraz na wnioski, które można uzyskać na podstawie danych. Ponieważ analiza danych o wyższej częstotliwości wymaga większych zasobów obliczeniowych więc potrzebny jest kompromis pomiędzy wymaganymi zasobami, a poziomem szczegółowości, który to wynika z praktycznego przeznaczenia algorytmów. System oświetlenia charakteryzuje się wolnym zmianami więc okres pomiarowy 60 sekund jest z punktu widzenia potrzeb detekcji anomalii okresem zbyt krótkim, a dodatkowo pojawiają się ograniczenia niektórych metod obliczeniowych.

Rozdzielczość czasowa metod analizy pomiarów zużycia energii była analizowana w wielu pracach, między innymi w [90], [91], [92]. Najczęściej przyjmuje się rozdzielczość czasową równą 15 minut lub 60 minut. Obie te wartości są standardowymi okresami dla ustalania profili zużycia energii przez przedsiębiorstwa energetyczne. W związku z tym, do tworzenia modelu szeregu czasowego zdecydowano się na „downsampling” strumienia przez wyliczenie średniej dla okresu wejściowego do rozdzielczości czasowej 15 minut, co daje okres sezonowości (Seasonal Order) równy 96 oraz do rozdzielczości czasowej 1 godziny z okresem sezonowości 24. Mniejsza rozdzielczość czasowa ma zastosowanie do metody opartej o model autoregresyjny, dla którego okres sezonowości równy 96 jest zbyt duży. Rys. 16 przedstawia dekompozycję dla szeregu z okresem 60 minut.



Rys. 16. Dekompozycja szeregu czasowego mocy czynnej, downsampling 60 minut

4.2 Algorytmy detekcji stosujące okres jednogodzinny

W niniejszym rozdziale przebadane będą algorytmy uczenia maszynowego stosujące modele regresyjne oraz rekurencyjne sieci neuronowe LSTM i GRU. Ze względu na ograniczenia metody ARIMA przyjęty został 60-minutowy okres pomiarowy. Model ARIMA i RNN TSM były analizowane w artykule [93].

4.2.1 Algorytm detekcji anomalii oparty o SARIMA

Pierwszą zbadaną metodą detekcji anomalii używającą maszynowe uczenie jest metoda oparta o model AutoRegressive Integrating Moving Average. Ze względu na okresowy charakter szeregu zastosowana będzie wersja okresowa – SARIMA - Seasonal Autoregressive Integrated Moving Average.

Model SARIMA określają następujące parametry:

- (p) parametr autoregresyjny,
- (d) rząd różnicowania,
- (q) parametr średniej ruchomej,
- (P) parametr autoregresyjny dla okresu,
- (D) rząd różnicowania dla okresu,
- (Q) parametr średniej ruchomej dla okresu,
- (m) okres sezonowości,

zapisywane jako $(p,d,q)(P,D,Q)[m]$.

W rozważanym przypadku ustalony jest parametr $m = 24$, natomiast pozostałe parametry wymagają doboru. Klasyczne metody doboru parametrów modelu opierają się na analizie funkcji autokorelacji, autokorelacji cząstkowej sezonowości oraz trendów [94]. Ponieważ opracowywany algorytm powinien być samouczący, to do wyznaczenia parametrów modelu użyty będzie algorytm automatycznego doboru parametrów. Jednym z przykładowych

algorytmów do automatycznego doboru parametrów jest algorytm krokowy (step-wise), zaproponowany w pracy [95]. Algorytm jest powszechnie stosowany, ponieważ posiada wiele implementacji, w tym w językach R i Python. Algorytm opiera się na minimalizacji kryterium informacyjnego Akaikego (AIC), co prowadzi do preferowania modeli o niższym stopniu złożoności kosztem dokładności prognozowania.

Istnieje jeszcze jedno uzasadnienie dla stosowania automatycznego doboru parametrów - jest nim zjawisko zmiany charakterystyki danych w czasie. Zmiana ta może być powodowana różnymi czynnikami, takimi jak zmiana długości nocy i dnia w ciągu roku, zmiany w ilości zainstalowanych lamp, zmiana typów lamp lub zmiany w harmonogramie redukcji. Te zmiany powodują, że parametry modelu, które były początkowo dobrze dobrane do danych, mogą stać się nieaktualne i wymagać ponownego dostosowania. To zjawisko jest znane jako dryf koncepcyjny (Concept Drift) [96]. Ten rodzaj zmiany charakterystyki danych jest również obserwowany w krótkoterminowych prognozach zużycia energii (STLF) w budynkach mieszkalnych i użytkowych [97]. Autorzy zwracają uwagę na to, że tradycyjne modele ARIMA, które są powszechnie stosowane do STLF, nie mają przyrostowego mechanizmu uczenia się, polegającego na zapominaniu nieaktualnych danych i dostosowaniu się do najnowszych pomiarów. Metody tradycyjne uczą się parametrów danego modelu ARIMA tylko jednokrotnie, przy użyciu ustalonego zestawu treningowego, a następnie stosują ten model do wszystkich przyszłych pomiarów. Autorzy zaproponowali algorytm przyrostowy, który okresowo odbudowuje model predykcyjny za pomocą koncepcji okna przesuwającego (Sliding Window). Podobny mechanizm zostanie zaimplementowany w opracowywanym algorytmie detekcji anomalii, z tą różnicą, że ulega zmianie cel algorytmu. W algorytmie OLIN celem było wyznaczenie aktualnego profilu zużycia energii, podczas gdy w proponowanym rozwiązaniu celem jest wykrycie anomalii. Ponadto, zmieniony został sposób walidacji modelu, aby dostosować go do wymagań detekcji anomalii.

Do walidacji i testowania modelu odwzorowującego szeregi czasowe stosuje się wiele metod, najczęściej stosowane to MAPE (Mean Absolute Percentage Error) - średni bezwzględny błąd procentowy jest określony w równaniu 4.1:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - y_t^p}{y_t} \right| * 100\% \quad (4.1)$$

Wadą tej miary jest to, że przyjmuje niezdefiniowane wartości, podczas gdy rzeczywiste dane są zerowe, oraz przyjmuje wartości ekstremalne, gdy rzeczywiste dane są bardzo bliskie zeru, a tak jest w przypadku danych pochodzących z systemu oświetlenia.

Wady tej pozbawiona jest miara MAE (Mean Absolute Error), dla której średni błąd bezwzględny określony jest równaniem 4.2:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - y_t^P| \quad (4.2)$$

Miara ta informuje, ile będzie wynosiło odchylenie od wartości rzeczywistej, średnio w okresie prognozy. Podstawową miarą błędu między zmienną prognozowaną a prognozą jest błąd bezwzględny, oznaczany jako AE (Absolute Error), równanie 4.3:

$$AE_t = |y_t - y_t^P| \quad (4.3)$$

W przypadku problemu detekcji anomalii istotna jest również miara MaAE (Maximum Absolute Error) – maksymalny błąd bezwzględny dla uzyskanego zbioru błędów (4.4):

$$MaAE = \max\{AE_t\}_{t=0}^{N-1} \quad (4.4)$$

Aby można było dokonywać porównań dla przebiegów o różnych amplitudach, wprowadzamy miarę znormalizowanego MaAE_{norm}, proporcjonalnego do wartości międzyszczytowej (4.5):

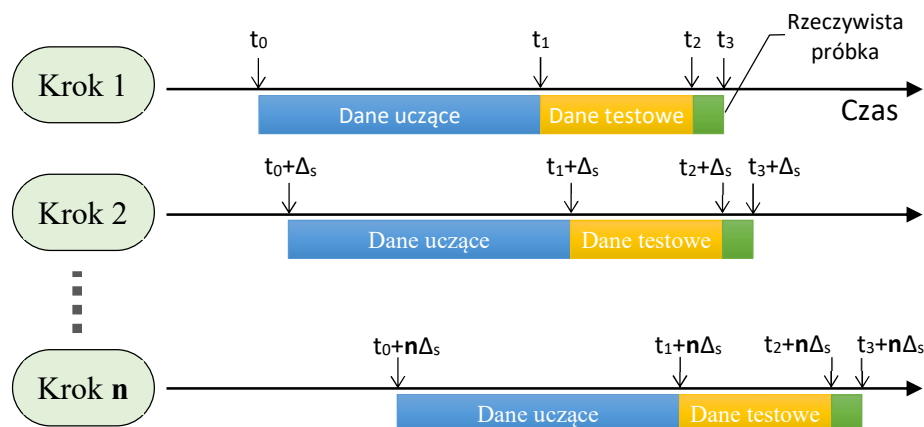
$$MaAE_{norm} = 100\% * \frac{MaAE}{Y_{max} - Y_{min}} \quad (4.5)$$

Do wyszukiwania parametrów modelu zastosowano algorytm wyszukiwania kratowego (grid search) – Algorytm 1. Algorytm wymaga na wejściu zbioru treningowego Y_{tr} , zbioru walidacyjnego Y_{val} oraz zestawu zbiorów dopuszczalnych wartości dla parametrów SARIMA: $\{p_1...p_p\}$, $\{d_1...d_d\}$, $\{q_1...q_q\}$, $\{P_1...P_P\}$, $\{D_1...D_D\}$, $\{Q_1...Q_Q\}$. Z zestawu zbiorów parametrów tworzony jest zbiór wektorów parametrów $(p_m, d_m, q_m, P_m, D_m, Q_m)$. Dla każdego wektora zestawu parametrów tworzony jest model SARIMA na podstawie próbek ze zbioru treningowego. Na podstawie modelu wyliczana jest prognoza „Out-of-sample” o długości równej wielkości zbioru walidacyjnego. W oparciu o prognozę oraz o próbki ze zbioru walidacyjnego wyliczany jest błąd MAE, który wraz z wektorem parametrów dodawany jest do listy. Po utworzeniu listy wybierany jest wektor parametrów, dla którego wartość MAE jest najmniejsza.

Dla każdego zbioru próbek algorytm może wybrać inne parametry modelu. Dobór liczby próbek dla zbiorów treningowego i walidacyjnego są istotne dla działania algorytmu detekcji anomalii. Większa liczba próbek powoduje zwiększenie dokładności modeli, ale z drugiej strony wydłuża czas reakcji systemu na potencjalne anomalie. Dla zbioru treningowego minimalna liczba próbek jest zdefiniowana na przykład w pracy [98]. W proponowanym algorytmie przyjęto dwie doby, czyli 48 próbek. Wielkość prognozy nie powinna przekraczać okresu sezonowości.

Algorytm detekcji anomalii działa cyklicznie w rytm napływających danych - odczytów z licznika Smart Meter. W każdym kroku wyznaczany jest błąd bezwzględny AE, który posłuży do podjęcia decyzji o wykryciu anomalii. Algorytm 3 przedstawia sposób wyznaczania zbioru AE na potrzeby symulacji. Idea algorytmu detekcji anomalii oparta o SARIMA została

pokazana na rys. 17. W pierwszej fazie następuje kompletacja próbek tworzących zbiór treningowy i walidacyjny, które posłużą do wytworzenia inicjalnego modelu przy użyciu algorytmu Grid Search. Następnie algorytm w pętli wykonuje następujące są operacje: utworzenie i wytrenowanie modelu SARIMA, wyliczenie prognozy w oparciu o utworzony model, a następnie wyznaczenie błędu prognozy **AE** w oparciu o rzeczywisty pomiar, który posłuży do podjęcia decyzji o wykryciu anomalii. Następnie przesuwane jest okno pomiarowe o jedną próbkę, wyznaczane są nowe zbiory treningowy i walidacyjny i w oparciu o nie dokonuje się weryfikacji, czy wystąpiło zjawisko Concept Drift – Algorytm 2. Jeżeli warunek ten jest spełniony, to ponownie wykonywany jest algorytm Grid Search.



Rys. 17. Idea algorytmu detekcji anomalii oparta o SARIMA

Do wyliczeń przyjęto następujące dane wejściowe: liczby próbek zbioru treningowego $N_{tr} = 48$, zbioru walidacyjnego $N_{val} = 23$, próg konieczności wyszukania nowych parametrów modelu $D_{Th} = 1.1$. Do poszukiwania parametrów modelu przyjęto następujące zbiory wartości parametrów SARIMA: $p = \{0, 1, 2\}$, $d = \{0, 1\}$, $q = \{0, 1, 2\}$, $P = \{0, 1\}$, $D = \{0, 1\}$, $Q = \{0, 1, 2\}$, co daje 216 kombinacji.

Algorytm 1. Wyszukiwanie kratowe (Grid search) parametrów modelu SARIMA

Input:

- Training set— Y_{tr}
- Validating set— Y_{val} ,
- Parameters sets— $\{p_1...p_p\}$, $\{d_1...d_d\}$, $\{q_1...q_q\}$, $\{P_1...P_P\}$, $\{D_1...D_D\}$, $\{Q_1...Q_Q\}$

Output:

- MAE for model
- parameters of model (p, d, q, P, D, Q)

1: Generate cartesian product for parameter sets:

$$M = \{p_1...p_p\} \times \{d_1...d_d\} \times \{q_1...q_q\} \times \{P_1...P_P\} \times \{D_1...D_D\} \times \{Q_1...Q_Q\}$$

2: **for each** $(p_m, d_m, q_m, P_m, D_m, Q_m)$ **in** M :

- 3: create model: $\text{model} = \text{ARIMA}(Y_{\text{tr}}, (p_m, d_m, q_m), (P_m, D_m, Q_m), 24)$
 - 4: fit the parameters of the model: $\text{model_fit} = \text{model.fit}()$
 - 5: make out-of-sample forecast: $Y_p = \text{model_fit.forecast}(\text{len}(Y_{\text{val}}))$
 - 6: calculate MAE: $\text{MAE}_m = \text{MAE}(Y_{\text{val}}, Y_p)$
 - 7: add $(\text{MAE}_m, (p_m, d_m, q_m, P_m, D_m, Q_m))$ to list $\{\text{ML}\}$
 - 8: **return** $\text{MAE}_x, (p_x, d_x, q_x, P_x, D_x, Q_x)$ for $\min(\text{MAE})$ in $\{\text{ML}\}$
-

Algorytm 2. Detekcja Concept Drift.

Input:

- Training set— Y_{tr}
- Validating set— Y_{val} ,
- Parameters of SARIMA model $(p_i, d_i, q_i, P_i, D_i, Q_i)$
- Current model MAE— MAE_i
- Concept drift threshold— D_{Th}

Output:

- new MAE_o for model
 - new parameters of SARIMA model $(p_o, d_o, q_o, P_o, D_o, Q_o)$
- 1: Create model: $\text{model} = \text{ARIMA}(Y_{\text{tr}}, (p_o, d_o, q_o), (P_o, D_o, Q_o), 24)$
 - 2: Fit the parameters of the model: $\text{model_fit} = \text{model.fit}()$
 - 3: Make forecast: $Y_p = \text{model_fit.forecast}(\text{len}(Y_{\text{val}}))$
 - 4: Calculate MAE: $\text{MAE}_m = \text{MAE}(Y_{\text{val}}, Y_p)$
 - 5: **If** $\text{MAE}_m / \text{MAE}_i > D_{\text{Th}}$ **then**
 - 6: Find parameters and MAE (Algorithm 1): $\text{MAE}_o, (p_o, d_o, q_o, P_o, D_o, Q_o) = \text{GridSearch}(Y_{\text{tr}}, Y_{\text{val}})$
 - 7: **Else**
 - 8: $\text{MAE}_o = \text{MAE}_i$
 - 9: $(p_o, d_o, q_o, P_o, D_o, Q_o) = (p_i, d_i, q_i, P_i, D_i, Q_i)$
 - 10: **return** $\text{MAE}_o, (p_o, d_o, q_o, P_o, D_o, Q_o)$
-

Algorytm 3. Wyliczanie AE przy użyciu SARIMA.

Input:

- Number of test steps— N_s
- Number of training samples— N_{tr}
- Number of validating samples— N_{val}
- Set of samples of length = $N_{\text{tr}} + N_{\text{val}} + N_s$

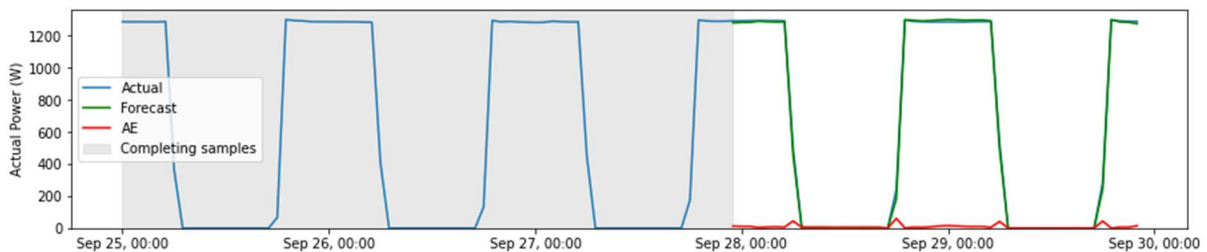
Output:

- Calculated absolute errors $\{\text{AE}_1, \text{AE}_2, \dots, \text{AE}_{N_s}\}$
- 1: Calculate $t_0 = t_{\text{start}}, t_1 = t_{\text{start}} + N_{\text{tr}}, t_2 = t_1 + N_{\text{val}}, t_3 = t_2 + 1$
 - 2: Prepare training set $Y_{\text{tr}} [t_0 ; t_1]$, validating set $Y_{\text{val}} [t_1 ; t_2]$, actual value y_{t_3}
 - 3: Find initial parameters and MAE (Algorithm 1):

- $$MAE_c, (p_c, d_c, q_c, P_c, D_c, Q_c) = \text{GridSearch}(Y_{tr}, Y_{val})$$
- 4: **for** $i = 1$ **to** N_s
 - 5: Create model: $\text{model} = \text{ARIMA}(Y_{tr}, (p_c, d_c, q_c), (P_c, D_c, Q_c), 24)$
 - 6: Fit the parameters of the model: $\text{model_fit} = \text{model.fit}()$
 - 7: Make forecast: $y_p = \text{model_fit.forecast}(1)$
 - 8: Calculate absolute error: $AE_i = |y_p - y_{t3}|$
 - 9: Add AE_i to list $\{AE\}$
 - 10: Calculate new window: $t_0 = t_0 + 1, t_1 = t_1 + 1, t = t_{22} + 1, t_3 = t_3 + 1$
 - 11: Prepare training set $Y_{tr}[t_0 ; t_1]$, validating set $Y_{val}[t_1 ; t_2]$, actual value y_{t3}
 - 12: Check concept drift: $MAE_c, (p_c, d_c, q_c, P_c, D_c, Q_c) = \text{ConceptDriftCheck}(Y_{tr}, Y_{val}, (p_c, d_c, q_c, P_c, D_c, Q_c), MAE_c, D_{Th})$
 - 13: **return** $\{AE\}$

Wszystkie algorytmy zaimplementowano w języku Python w wersji 3.10.5. Użyto następujących bibliotek: pandas 1.3.5, NumPy 1.21.5, statsmodels 0.13.1, scikit-learn 1.0.2, TensorFlow 2.9.1, Keras 2.9.0.

Wynik symulacji dla zestawu pomiarów **ZP₁** czyli zapisów bez anomalii przedstawiono na rys. 18. Szare pole oznacza okres gromadzenia próbek. Okres ten wynosi 71 godziny ponieważ potrzeba 48 godzin próbek treningowych i 23 próbki walidacyjne. Dopiero po tym czasie rozpoczyna się detekcja (białe pole). Etykietą „Forecast” oznaczono wyliczoną prognozę, „AE” – błąd bezwzględny prognozy.



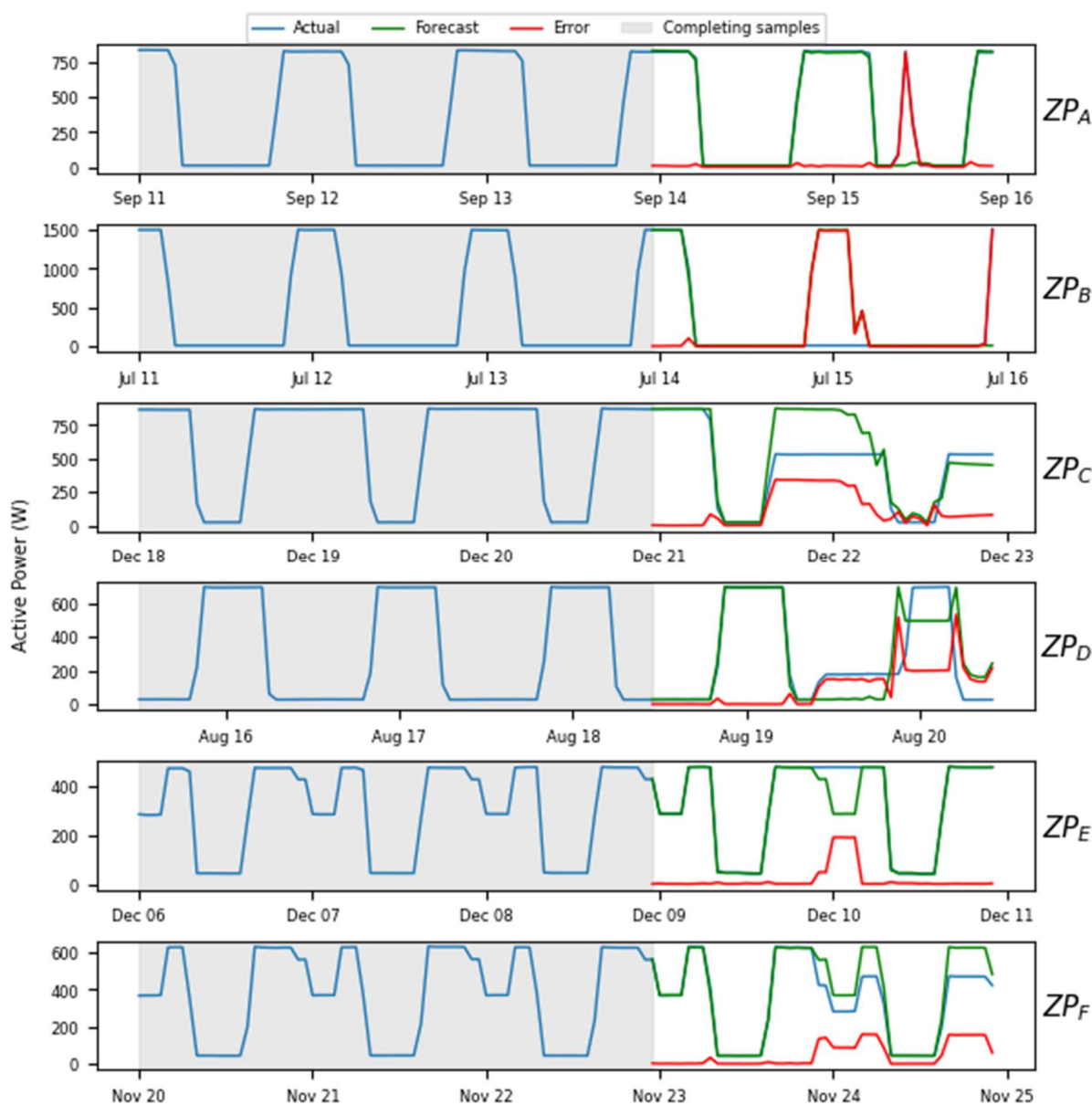
Rys. 18. Wynik symulacji przy zastosowaniu SARIMA dla zestawu pomiarów **ZP₁** – pomiary bez awarii

Następnie symulację powtórzono dla wszystkich zestawów **ZP₁ ... ZP₉**, wyliczając mierniki MAE oraz MaAE. Ponieważ zestawy mają różne amplitudy mocy czynnej, to dodatkowo wyliczono znormalizowaną wartość $MaAE_{norm}$, - wyniki zawiera tab. 1. Wynika z niej, że dla tych zestawów maksymalna wartość znormalizowana $MaAE_{norm}$ jest równa 5 % i może to być próg odstępstwa od normalnego przebiegu, czyli anomalii.

Tabela 1. Wyniki symulacji algorytmem SARIMA dla zestawów **ZP₁ ... ZP₉**

Dataset	ZP ₁	ZP ₂	ZP ₃	ZP ₄	ZP ₅	ZP ₆	ZP ₇	ZP ₈	ZP ₉
MAE	8,51	8,75	4,72	3,25	2,89	2,18	2,29	4,53	2,10
MaAE	60,27	68,33	42,35	52,60	27,75	34,89	24,34	34,93	20,97
MaAE _{norm}	4 %	5 %	5 %	5 %	5 %	5 %	1 %	2 %	1 %

Następnie wykonano symulacje dla zestawów z anomaliami ZP_A , ZP_B , ... ZP_F . Wynik symulacji przedstawiono na rys. 19.



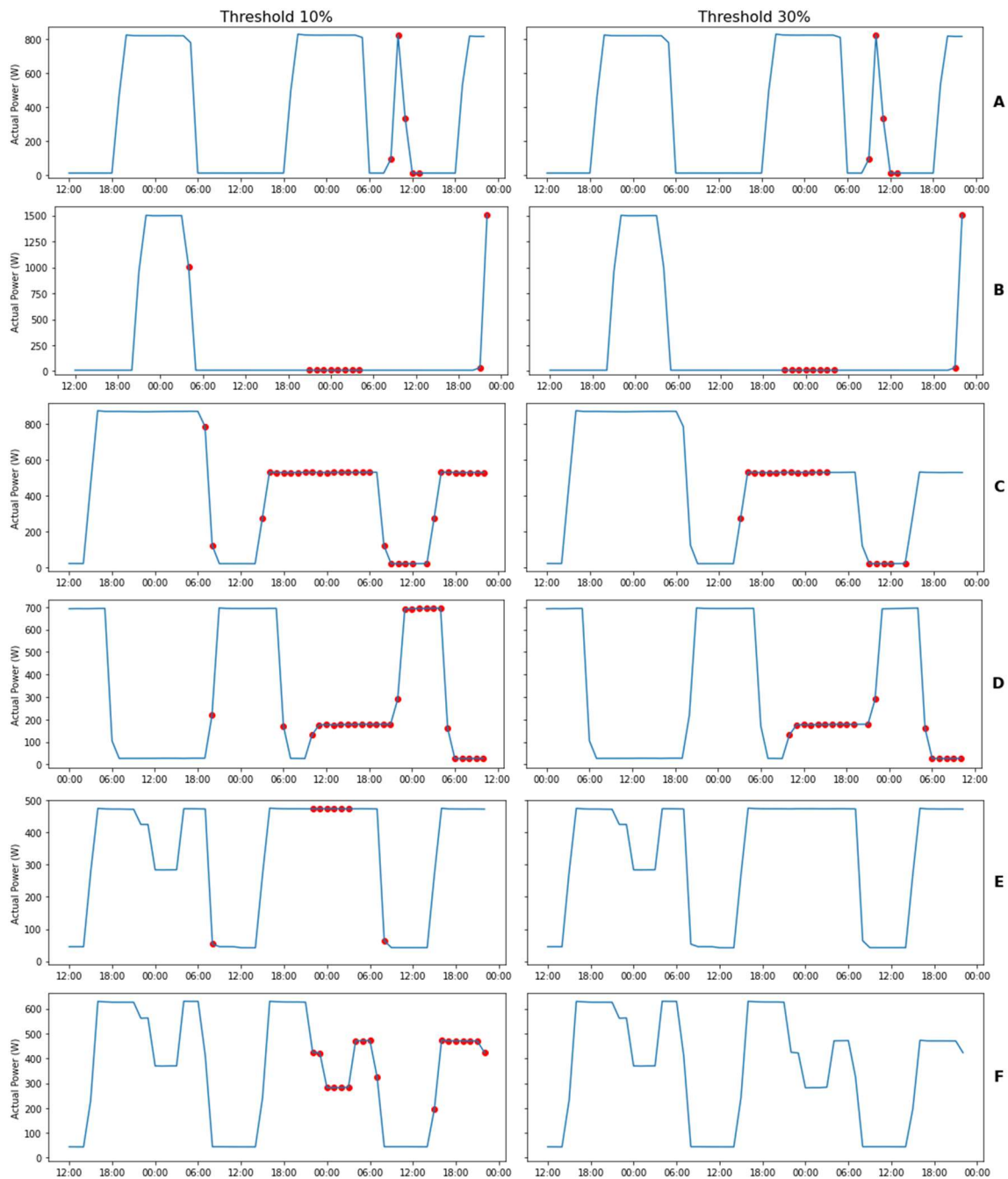
Rys. 19. Wyniki symulacji przy zastosowaniu SARIMA dla poszczególnych typów anomalii. Podwykresy A-F odpowiadają anomalii przedstawionym na rys. 15

Tab. 2 zawiera wyliczone wartości MAE, MaAE oraz $MaAE_{norm}$ dla zestawów ZP_A , ZP_B , ... ZP_F . Z wyliczonych danych wynika, że najmniejsza wartość $MaAE_{norm}$ jest równa 27 %. Można więc dobrać tak próg detekcji błędu w przedziale $\langle 5\%, 27\% \rangle$, aby skutecznie wykryć anomalie.

Tabela 2. Wyniki symulacji algorytmem SARIMA dla zestawów ZP_A, \dots, ZP_F

Dataset	ZP_A	ZP_B	ZP_C	ZP_D	ZP_E	ZP_F
MAE	30,89	221,99	119,97	101,49	19,30	48,03
MaAE	813,76	1499,33	342,00	532,44 </td <td>189,79</td> <td>159,56</td>	189,79	159,56
MaAE _{norm}	99 %	100 %	40 %	79 %	43 %	27 %

Dobór progów umożliwia określenie czułości algorytmu i jest związany z konkretną implementacją algorytmu. Na rys. 20 przedstawiono symulację algorytmu detekcji dla progów 10% i 30%.

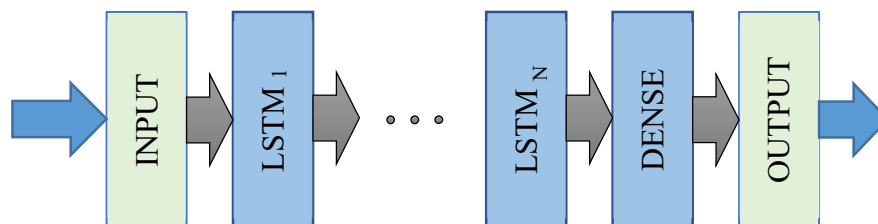


Rys. 20. Zastosowanie różnych progów do detekcji anomalii dla algorytmu SARIMA. Podwykresy A-F odpowiadają anomalii przedstawionym na rys. 15

Przebadany algorytm wykazał się skutecznością w zakresie detekcji anomalii w analizowanych zbiorach danych.

4.2.2 Algorytm detekcji anomalii oparty o LSTM

Kolejną metodą analizy szeregu czasowego pomiarów mocy czynnej jest wykorzystanie do maszynowego uczenia sieci LSTM, jedno- i wielowarstwowych, strukturę których przedstawiono na Rys. 21.



Rys. 21. Sieć LSTM zastosowana do algorytmu detekcji anomalii

Użycie szeregu czasowego jako strumienia wejściowego do uczenia maszynowego, wymaga takiego zorganizowania danych, aby można było zastosować tryb uczenia nadzorowanego. Dane wejściowe (X) i wyjściowe (Y) są podawane do sieci jednocześnie, dzięki czemu algorytm może nauczyć się przewidywać i minimalizować różnice między wartościami oczekiwanymi i wyuczonymi. Dlatego konieczne jest przekształcenie szeregów czasowych z sekwencji jednowymiarowej do dwuwymiarowej macierzy, której jeden wymiar reprezentuje dane wejściowe (cechy – „features”) a drugi dane wyjściowe (etykiety – „labels”). Dla ciągu pomiarów $t_0, t_1, \dots, t_{N-1}, t_N$ wartości odczytane od t_0, t_1, \dots, t_{N-1} stanowią cechy, a wartość t_N jest etykietą.

Jak już wspomniano, szeregi czasowe pomiarów mocy czynnej charakteryzują się dużą periodycznością związaną z rytmem dobowym, dlatego naturalną rzeczą jest, aby budowana sieć była w stanie przewidzieć kolejną wartość mocy czynnej na podstawie danych z dnia poprzedniego. Oznacza to, że potrzebne są 24 ostatnie przedziały czasu, aby przewidzieć wartość w następnym przedziale czasu, zakładając okres próbkowania wynoszący 60 minut.

Konfigurując sieć LSTM, należy określić liczbę ukrytych warstw sieci oraz wielkość wektora danych przekazywanych przez poszczególne warstwy. Dodatkowo należy określić wielkość zbioru uczącego sieć. Ponieważ w przypadku algorytmu on-line mamy do czynienia z ciągłym napływem nowych próbek, istnieje pewnego rodzaju dowolność w wyborze wielkości zbioru uczącego. Z jednej strony im większy zbiór, tym większe prawdopodobieństwo uzyskania dopasowanego modelu, z drugiej strony większy zbiór oznacza dłuższe opóźnienie startu algorytmu detekcji oraz większy nakład obliczeniowy. Aby móc porównać algorytm detekcji anomalii oparty o LSTM z algorytmem opartym o SARIMA, wskazane byłoby zastosowanie takiego samego okresu – 3 x 24 godziny, jednakże nie jest to

możliwe, ponieważ sieć trenowana jest wektorem, który tworzony jest z 24 pomiarów. Aby uzyskać N_{tr} wektorów uczących, to potrzeba $N_{tr} + N_F + N_L$ pomiarów, gdzie N_F to liczba cech, a N_L to liczba etykiet. W rozważanym przypadku dla zbioru uczącego złożonego z 48 wektorów potrzeba 73 próbki. Analogicznie dla zbioru testowego, aby uzyskać zbiór testowy o wielkości 24 wektorów, potrzeba 49 próbek, jednak aby zredukować całkowitą wielkość zbioru, do wytworzenia wektorów testujących zostanie użytych N_F ostatnich próbek ze zbioru uczącego. Całkowita liczba próbek dla obu zbiorów w związku tym wynosi: $N_{tr} + N_F + N_L + N_{val} + N_F + N_L$ próbek, w badanym przypadku jest to 98 pomiarów.

Ponieważ brak jednoznacznych metodologii doboru architektury sieci LSTM, to do wyszukiwania optymalnych parametrów zastosowano algorytm wyszukiwania kratowego, analogiczny do algorytmu zastosowanego w metodzie opartej o SARIMA. Algorytm wymaga na wejściu zbioru treningowego Y_{tr} , zbioru walidacyjnego Y_{val} oraz zestawu zbiorów dopuszczalnych wartości dla liczby ukrytych warstw sieci $\{l_1 \dots l_l\}$ oraz wielkość wektora przekazywanego przez poszczególne warstwy $\{o_1 \dots o_o\}$. Z zestawu zbiorów parametrów tworzony jest zbiór wektorów parametrów (l_m, o_m) . Dla każdego wektora zestawu parametrów tworzony jest model o odpowiedniej liczbie warstw oraz wielkości wektorów. Model trenowany jest stałą liczbą iteracji (epok), równą 100, z użyciem próbek ze zbioru uczącego. Z użyciem modelu wyliczana jest prognoza dla zbioru testowego, w oparciu o którą oraz o próbki ze zbioru testowego wyliczany jest błąd MAE, który wraz z wektorem parametrów dodawany jest do listy. Po utworzeniu listy wybierany jest wektor parametrów dla którego wartość MAE jest najmniejsza.

Algorytm 4. Wyszukiwanie kratowe (Grid search) parametrów sieci LSTM.

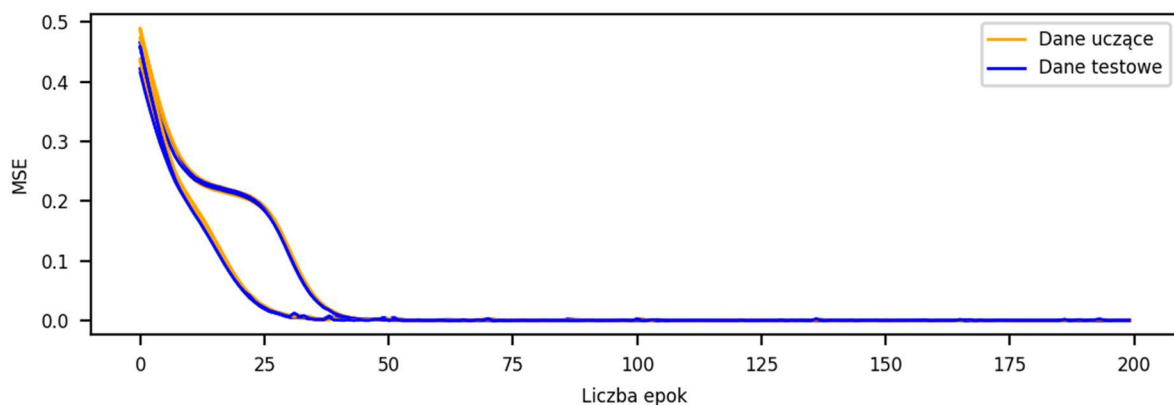
Input:

- Number of training samples— N_{tr}
- Number of validating samples— N_{val}
- Set of samples $N_{tr} + N_{val}$
- Parameters sets— $\{l_1 \dots l_l\}$, $\{o_1 \dots o_o\}$

Output:

- MAE for model
 - parameters of the model (layers, output space dimension)
- 1: Split samples set to Training set— Y_{tr} and Validating set— Y_{val}
 - 2: Generate cartesian product for parameter sets: $M = \{l_1 \dots l_l\} \times \{o_1 \dots o_o\}$
 - 3: **for each** (l_m, o_m) **in** M :
 - 4: create model: LSTM(layers = l_m , output_space = o_m)
 - 5: train model: model.fit(Y_{tr})
 - 6: make prediction: $Y_p = \text{model.predict}(Y_{val})$
 - 7: calculate MAE: $MAE_m = \text{MAE}(Y_{val}, Y_p)$
 - 8: add $(MAE_m, (l_m, o_m))$ to list $\{ML\}$
 - 9: **return** $MAE_x, (l_x, o_x)$ **for** min(MAE) **in** $\{ML\}$
-

W powyższym algorytmie przyjęto stałą liczbę epok, wyznaczoną przez obserwację przebiegu funkcji straty podczas walidacji skrośnej. Część zbioru uczącego przeznaczona jest do przeprowadzania w trakcie uczenia okresowej walidacji, tak aby kontrolować proces uczenia. Celem kontroli jest osiągnięcie pożądanego poziomu błędu oraz zapobieganie zjawisku przeuczenia - overfitting. Na rys. 22 przedstawiono wykres funkcji straty zdefiniowanej jako błąd MSE (Mean Squared Error) dla zbiorów trenującego i walidacyjnego. Na wykresie wykreślono funkcje dla wszystkich testowanych zbiorów ZP_1, ZP_2, \dots, ZP_9 . Jak widać, funkcja straty szybko się stabilizuje i nawet przy 200 iteracjach, nie występuje efekt przetrenowania (overfitting).



Rys. 22. Funkcja straty dla zbiorów trenującego i walidacyjnego dla zbiorów ZP_1, ZP_2, \dots, ZP_9

Wyniki działania algorytmu 4 dla zbiorów ZP_1, ZP_2, \dots, ZP_9 przedstawiono w tab. 3. Przyjęto następujące zbiory parametrów: warstwy sieci $l = \{1, 2, 3, 4, 5\}$, przestrzeń wyjściowa $o = \{1, 2, 3, \dots, 10\}$. Tabela zawiera wyznaczoną konfigurację sieci, odpowiadającą jej wartości MAE (minimum), maksymalną i średnią wartość MAE, która wystąpiła przy testowaniu wszystkich konfiguracji.

Tabela 3. Wyniki symulacji wyszukiwanie kratowego parametrów sieci LSTM

Dataset	ZP_1	ZP_2	ZP_3	ZP_4	ZP_5	ZP_6	ZP_7	ZP_8	ZP_9
Layers	5	5	5	5	4	5	5	1	5
Output space	1	1	1	1	1	1	1	7	1
Minimum MAE	3,87	6,58	3,24	4,27	3,63	3,07	6,57	10,86	5,88
Maximum MAE	32,44	33,54	21,49	33,86	19,83	24,01	27,11	31,11	24,45
Mean MAE	13,58	15,40	9,28	11,93	8,61	8,45	12,71	18,44	11,44

W przypadku głębokiego uczenia również występuje zjawisko dryfu koncepcyjnego [99], więc potrzebny jest mechanizm wykrywania takiej sytuacji. W projektowanym algorytmie

zastosowano mechanizm kontroli polegający na wyliczaniu wartości MAE dla zbioru treningowego i sprawdzaniu w każdym kroku, czy błąd nie wzrasta powyżej założonego progu.

Algorytm detekcji anomalii, wykorzystujący model LSTM działa analogicznie do algorytmu opartego o SARIMA. Działanie rozpoczyna się od zgromadzenia próbek tworzących zbiór treningowy i walidacyjny, na podstawie których wyznacza się konfigurację sieci przy użyciu algorytmu wyszukiwania kratowego, a następnie tworzy się model i trenuje się go połączonymi zbiorami treningowym i walidacyjnym. Następnie algorytm w pętli wykonuje następujące operacje: wyliczenie prognozy w oparciu o utworzony model, wyznaczenie błędu prognozy **AE** w oparciu o rzeczywisty pomiar, który posłuży do podjęcia decyzji o wykryciu anomalii. Następnie przesuwane jest okno pomiarowe o jedną próbkę, wyznaczany jest nowy zbiór treningowy i dokonuje się weryfikacji, czy wyliczony wskaźnik MAE dla prognozy w oparciu o ten zbiór nie przekracza założonego progu. Jeżeli warunek ten jest spełniony, to ponownie wykonywany jest algorytm wyszukiwania kratowego i wyszukiwane są nowe parametry sieci.

Algorytm 5. Wyliczanie AE przy użyciu LSTM.

Input:

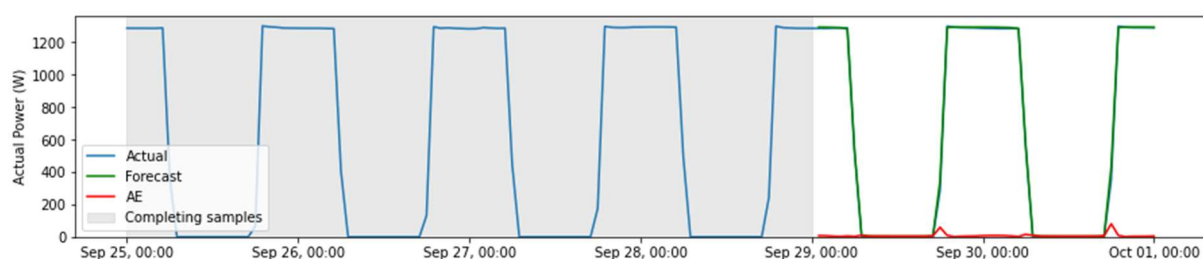
- Number of test steps— N_s
- Number of training samples— N_{tr}
- Set of samples of length = $N_{tr} + N_s$
- Concept drift threshold— D_{Th}

Output:

- Calculated absolute errors $\{AE_1, AE_2, \dots, AE_{N_s}\}$
- 1: Calculate $t_0 = t_{start}$, $t_1 = t_{start} + N_{tr}$, $t_2 = t_1 - 24$, $t = t_{32} + N_s$, $t_4 = t_1 + 1$
 - 2: Split samples set to Training set Y_{tr} [$t_0 ; t_1$] and Testing set Y_s [$t_2 ; t_3$]
 - 3: Find network configuration (Algorithm 4): $(l_c, o_c) = \text{GridSearch}(Y_{tr})$
 - 4: Create model: LSTM(layers = l_c , output_space = o_c)
 - 5: Train model: model.fit(Y_{tr})
 - 6: Make prediction on training set: $Y_p = \text{model.predict}(Y_{tr})$
 - 7: Calculate MAE: $MAE_m = \text{MAE}(Y_{tr}, Y_p)$
 - 8: **for** $i = 1$ **to** N_s
 - 9: Create model: model = ARIMA(Y_{tr} , (p_c, d_c, q_c), (P_c, D_c, Q_c), 24)
 - 10: Make prediction: $y_p = \text{model.predict}(Y_{val}[i])$
 - 11: Calculate absolute error: $AE_i = |y_p - y_{t4}|$
 - 12: Add AE_i to list $\{AE\}$
 - 13: Calculate new window:
 $t_0 = t_0 + 1$, $t_1 = t_1 + 1$, $t = t_{22} + 1$, $t_3 = t_3 + 1$, $t = t_{44} + 1$
 - 14: Prepare training set Y_{tr} [$t_0 ; t_1$]
 - 15: Make prediction: $Y_p = \text{model.predict}(Y_{tr})$
 - 16: Calculate $MAE_i = \text{MAE}(Y_{tr}, Y_p)$
 - 17: **If** $MAE_i / MAE_m > D_{Th}$ **then**
 - 18: Find network configuration (Algorithm 4): $(l_c, o_c) = \text{GridSearch}(Y_{tr})$
 - 19: Create model: LSTM(layers = l_c , output_space = o_c)
 - 20: Train model: model.fit(Y_{tr})
 - 21: Make prediction: $Y_p = \text{model.predict}(Y_{tr})$
 - 22: Calculate $MAE_m = \text{MAE}(Y_{tr}, Y_p)$
 - 23: **Else**

24: $MAE_m = MAE_i$
 25: **return** {AE}

Symulację według zdefiniowanego algorytmu wykonano dla wszystkich zestawów $ZP_1 .. ZP_9$ wyliczając mierniki MAE, MaAE oraz $MaAE_{norm}$. Okres gromadzenia próbek jest dłuższy niż w przypadku algorytmu SARIMA i wynosi 98 godzin. Wynik symulacji dla zestawu pomiarów ZP_1 przedstawiono na rys. 23, natomiast wyliczone wskaźniki zawiera tab. 4.



Rys. 23. Wynik symulacji przy użyciu algorytmu LSTM dla pomiarów bez anomalii ZP_1

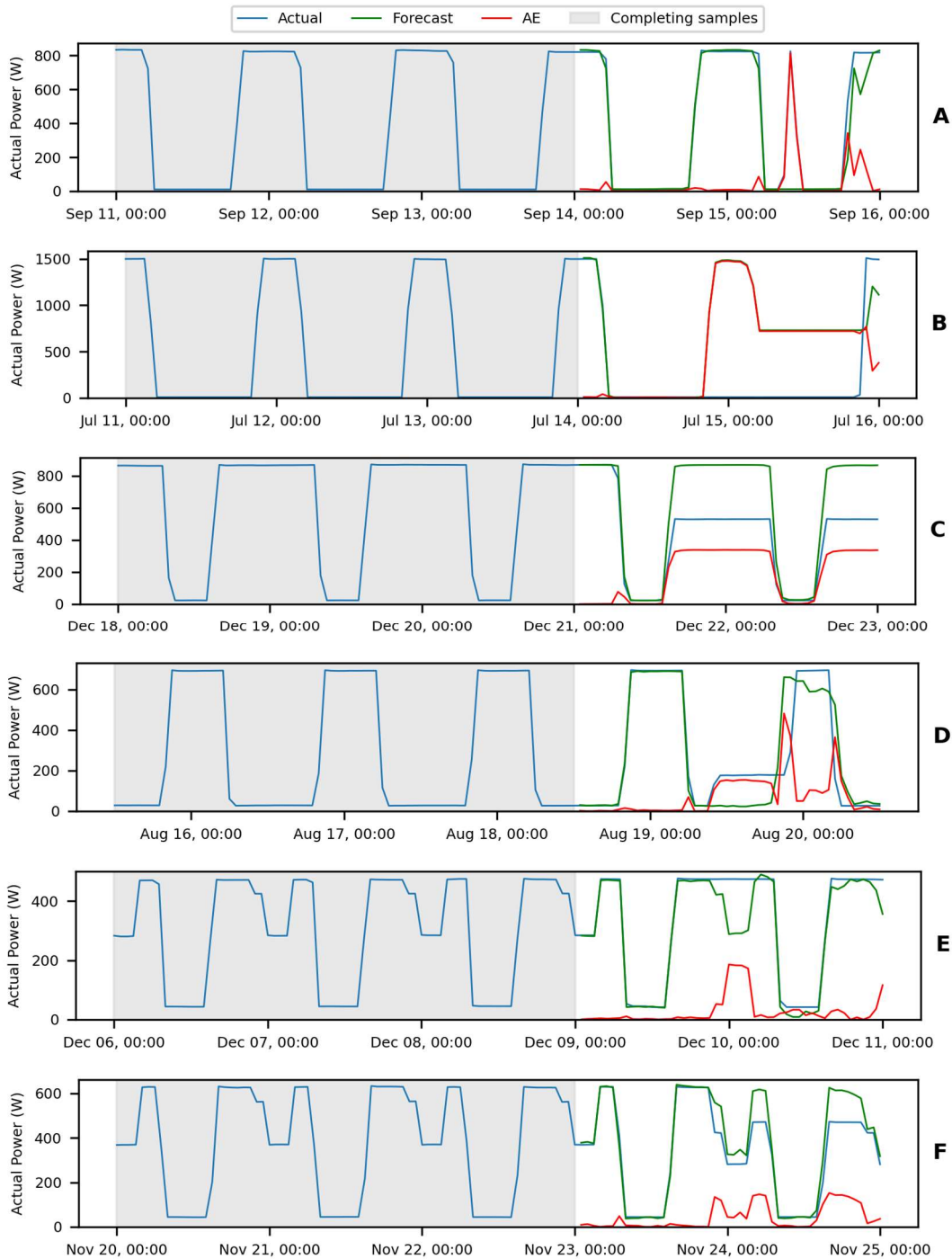
Tabela 4. Wyniki symulacji algorytmem LSTM dla zestawów $ZP_1 .. ZP_9$

Dataset	ZP_1	ZP_2	ZP_3	ZP_4	ZP_5	ZP_6	ZP_7	ZP_8	ZP_9
MAE	7,44	11,47	7,79	6,71	3,90	4,59	5,92	8,02	5,38
MaAE	80,46	101,61	71,16	59,36	37,98	41,34	65,94	69,21	59,79
$MaAE_{norm}$	6%	7%	8%	6%	6%	6%	4%	4%	4%

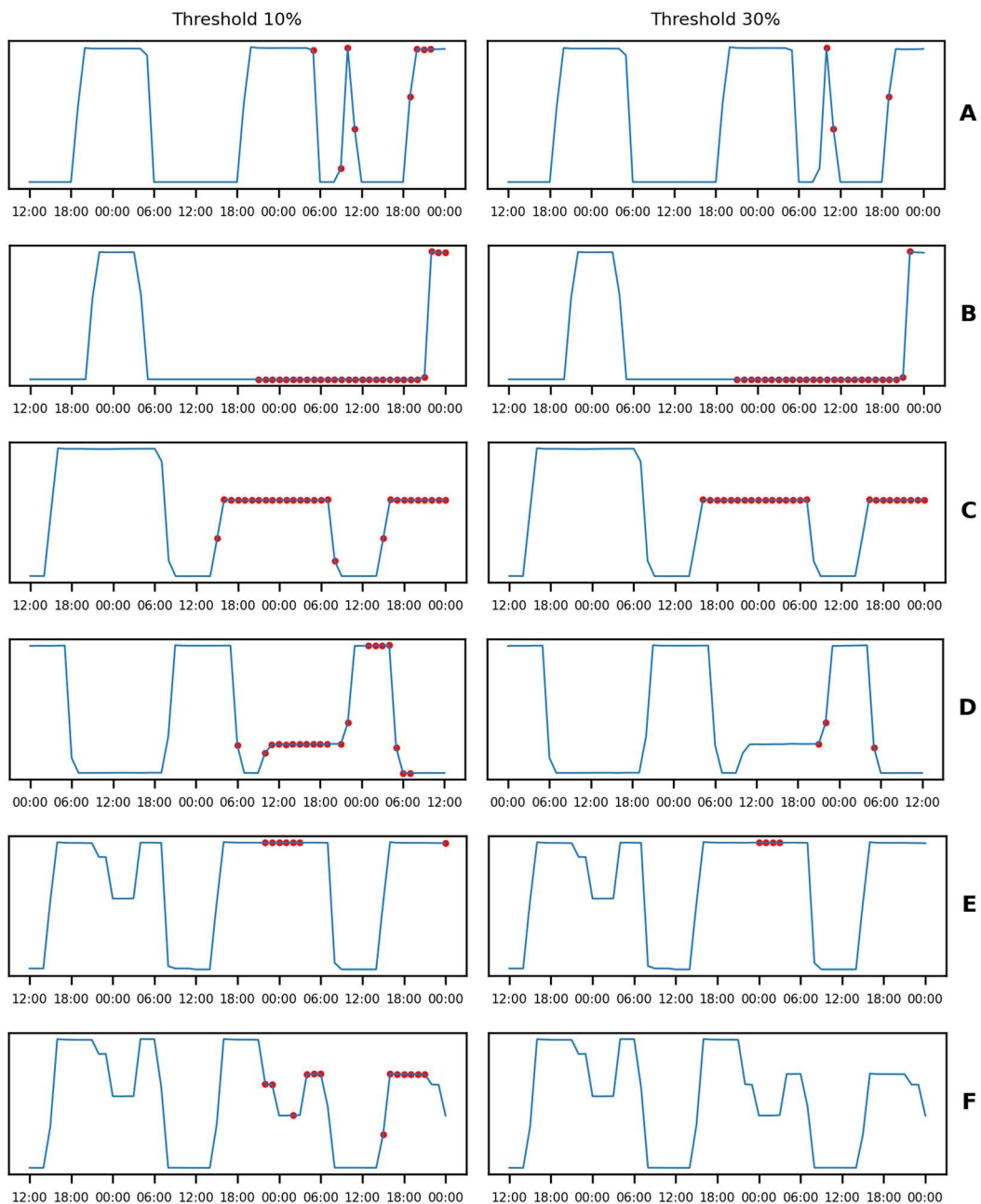
Następnie wykonano symulacje dla zestawów z anomaliami $ZP_A, ZP_B, \dots ZP_F$. Wynik symulacji przedstawiono na rys. 24. Tab. 5 zawiera wyliczone wartości MAE, MaAE oraz $MaAE_{norm}$ dla zestawów $ZP_A, ZP_B, \dots ZP_F$. Z wyliczonych danych wynika, że najmniejsza wartość $MaAE_{norm}$ jest równa 26 %. Można więc przyjąć próg detekcji błędu w przedziale $<8\%, 26\%>$. Przedział jest więc węższy niż w przypadku algorytmu SARIMA. Na rys. 25 przedstawiono symulację algorytmu detekcji dla progów 10% i 30%.

Tabela 5. Wyniki symulacji algorytmem LSTM dla zestawów $ZP_A, \dots ZP_F$

Dataset	ZP_A	ZP_B	ZP_C	ZP_D	ZP_E	ZP_F
MAE	49,04	515,68	189,91	75,52	28,53	43,17
MaAE	812,69	1476,98	338,83	482,01	186,01	152,79
$MaAE_{norm}$	99%	99%	40%	72%	42%	26%



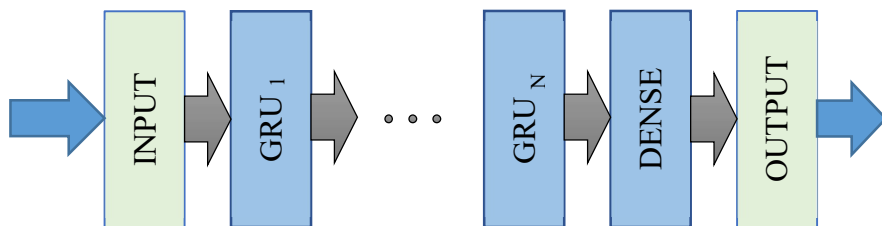
Rys. 24. Wyniki symulacji w oparciu o algorytm LSTM dla poszczególnych typów anomalii. Podwykresy A-F odpowiadają anomaliiom przedstawionym na rys. 15. Dla zwiększenia czytelności okres kolekcjonowania próbek jest przedstawiony jako krótszy, niż w rzeczywistości



Rys. 25. Zastosowanie różnych progów do detekcji anomalii dla algorytmu LSTM. Podwykresy A-F odpowiadają anomalii przedstawionym na rys. 15

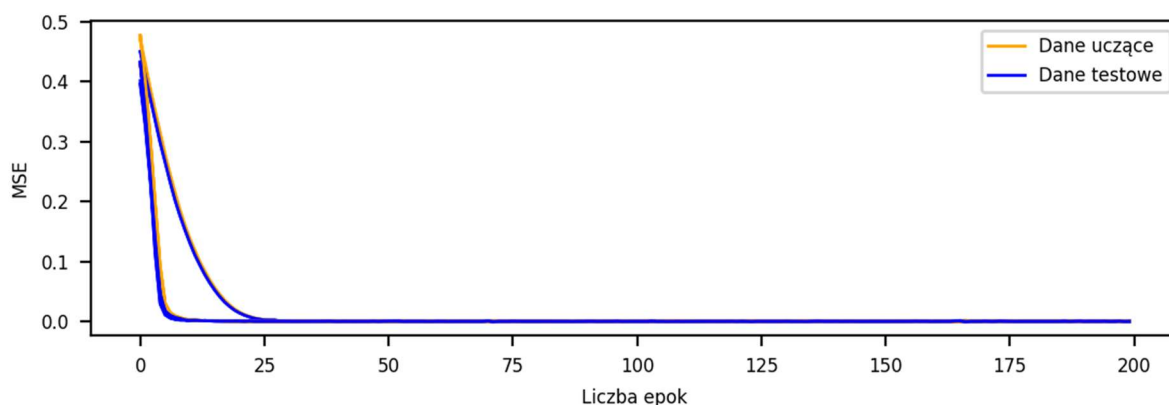
4.2.3 Algorytm detekcji anomalii oparty o GRU

Wyliczenia przeprowadzone w poprzednim rozdziale powtórzono dla innego typu rekurencyjnej sieci neuronowej – GRU, również w konfiguracji jedno i wielowarstwowej, przedstawionych na rys. 26.



Rys. 26. Sieć GRU zastosowana do algorytmu detekcji anomalii

Model ten wymaga takiej samej organizacji danych jak LSTM oraz określenia identycznych parametrów architektury sieci: liczby ukrytych warstw sieci oraz wielkości wektora przekazywanego przez poszczególne warstwy. W związku z tym do wyszukiwania parametrów sieci użyto ponownie algorytmu wyszukiwania kratowego (Algorytm 4), z następującymi zbiorami parametrów: warstwy sieci $l = \{1, 2, 3, 4, 5\}$, przestrzeń wyjściowa $o = \{1, 2, 3, \dots, 10\}$. Na rys. 27 przedstawiono wykres funkcji straty przebiegu uczenia dla modelu GRU, dla wszystkich testowanych zbiorów ZP_1, ZP_2, \dots, ZP_9 . Funkcja straty stabilizuje się szybciej niż dla modelu LSTM.



Rys. 27. Funkcja straty dla zbiorów trenujących i walidacyjnych dla modelu GRU

Wyniki działania algorytmu wyszukiwania kratowego dla modelu GRU przedstawiono w tab. 6.

Tabela 6. Wyniki symulacji wyszukiwanie kratowego parametrów sieci GRU

Dataset	ZP ₁	ZP ₂	ZP ₃	ZP ₄	ZP ₅	ZP ₆	ZP ₇	ZP ₈	ZP ₉
Layers	4	4	4	4	3	4	3	1	3
Output space	6	6	6	6	1	6	5	10	5
Minimum MAE	7,16	8,88	5,37	6,97	5,28	4,98	9,44	11,99	8,57
Maximum MAE	46,43	50,51	31,75	45,31	27,41	32,20	32,87	36,34	29,82
Mean MAE	16,69	18,27	10,95	14,29	10,60	10,07	16,56	21,80	14,94

Do wyliczania AE przy użyciu sieci GRU zastosowano ten sam algorytm jak dla LSTM (Algorytm 5). Wynik symulacji dla zestawów pomiarów bez anomalii zawiera tab. 7. Z wyliczonych danych wynika, że największa wartość $MaAE_{norm}$ jest równa 13 %.

Tabela 7. Wyniki symulacji wyliczania AE dla sieci GRU, zestawy pomiarów bez anomalii

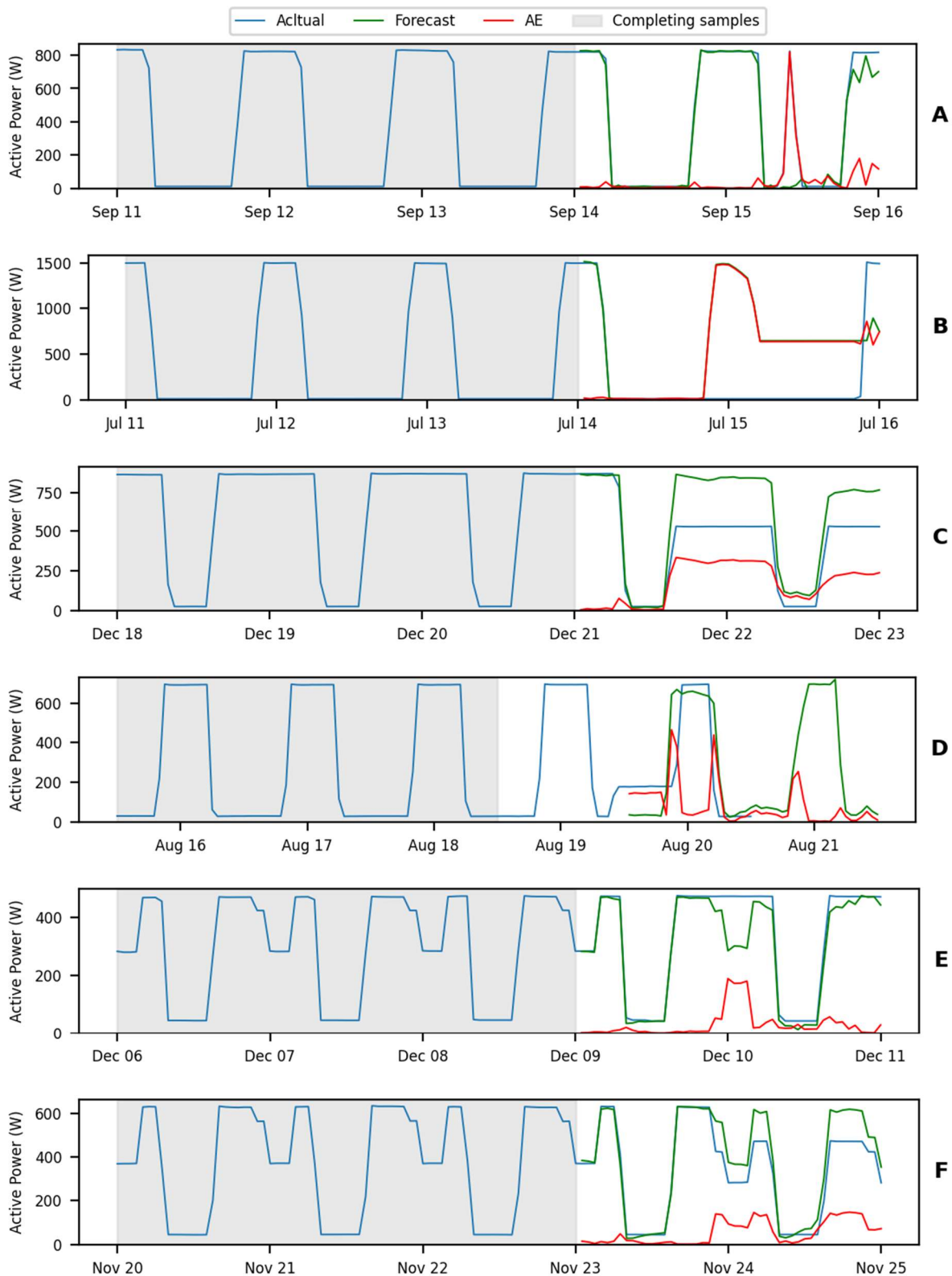
Dataset	ZP ₁	ZP ₂	ZP ₃	ZP ₄	ZP ₅	ZP ₆	ZP ₇	ZP ₈	ZP ₉
MAE	11,17	14,66	10,88	7,92	8,82	5,87	7,67	10,31	6,93
MaAE	71,61	128,99	93,40	71,02	75,40	53,61	82,44	60,48	73,97
$MaAE_{norm}$	6%	9%	11%	7%	13%	8%	5%	4%	5%

Następnie wykonano symulacje dla zestawów z anomaliami ZP_A, ZP_B, ... ZP_F. Wynik symulacji przedstawiono na rys. 28. Tab. 8 zawiera wyliczone wartości MAE, MaAE oraz $MaAE_{norm}$ dla zestawów danych z anomaliami. Z wyliczonych danych wynika, że najmniejsza wartość $MaAE_{norm}$ jest równa 25 %. Można więc przyjąć próg detekcji błędu w przedziale <13%, 25%>. Przedział jest znowu węższy niż w przypadku algorytmu SARIMA.

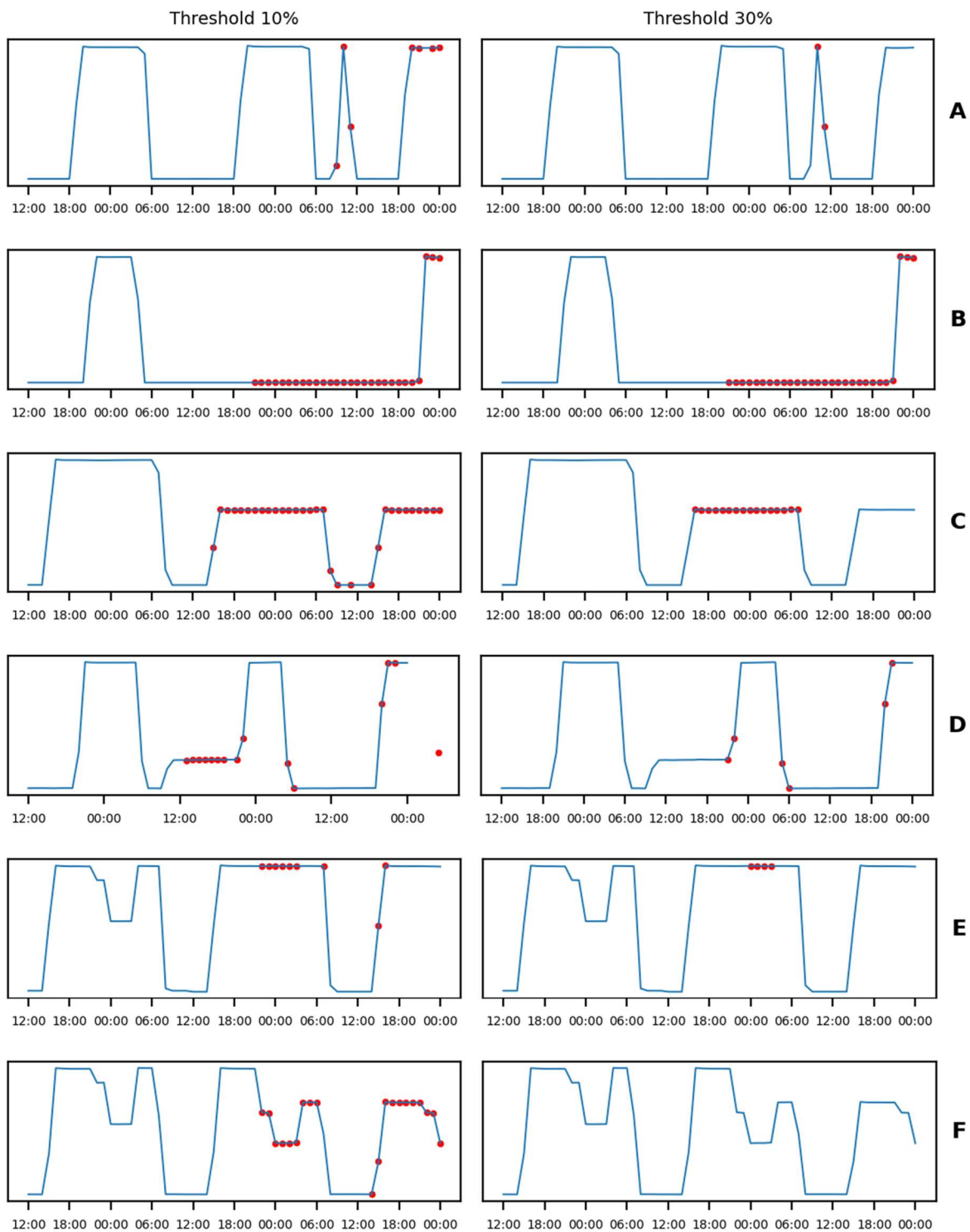
Tabela 8. Wyniki symulacji wyliczania AE dla sieci GRU, zestawy pomiarów z anomaliami

Dataset	ZP _A	ZP _B	ZP _C	ZP _D	ZP _E	ZP _F
MAE	48,91	494,97	172,28	84,00	29,61	54,04
MaAE	820,14	1483,11	333,92	464,93	188,97	146,89
$MaAE_{norm}$	100%	99%	39%	69%	43%	25%

Na rys. 29 przedstawiono wyniki symulacji uzyskane przy pomocy algorytmu detekcji dla progów 10% i 30%.



Rys. 28. Wyniki symulacji w oparciu o algorytm GRU dla poszczególnych typów anomalii. Podwykresy A-F odpowiadają anomalii przedstawionym na rys. 15. Z uwagi na potrzebę zwiększenia czytelności okres kolekcjonowania próbek jest przedstawiony jako krótszy, niż w rzeczywistości

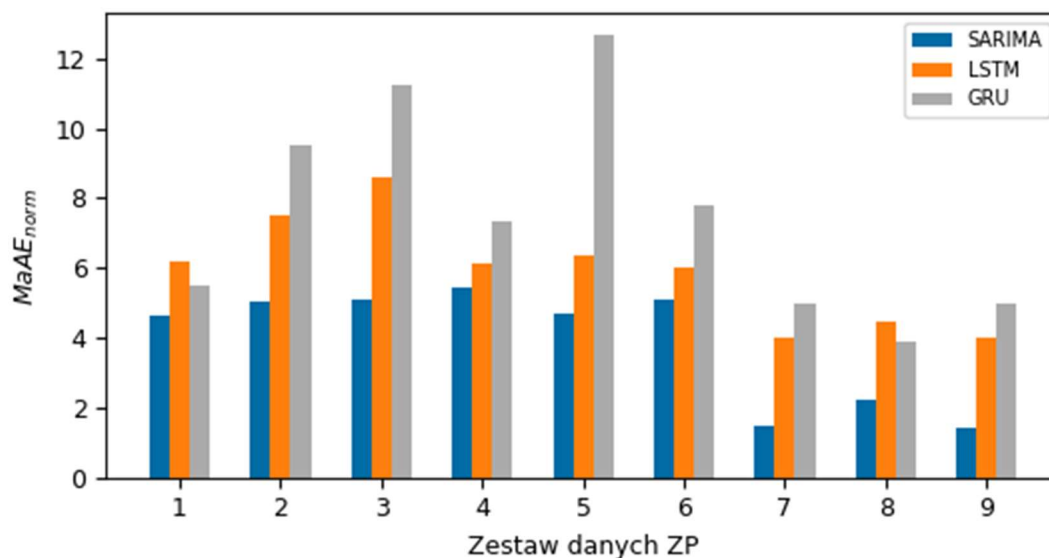


Rys. 29. Zastosowanie różnych progów do detekcji anomalii dla algorytmu GRU. Podwykresy A-F odpowiadają anomaliiom przedstawionym na rys. 15

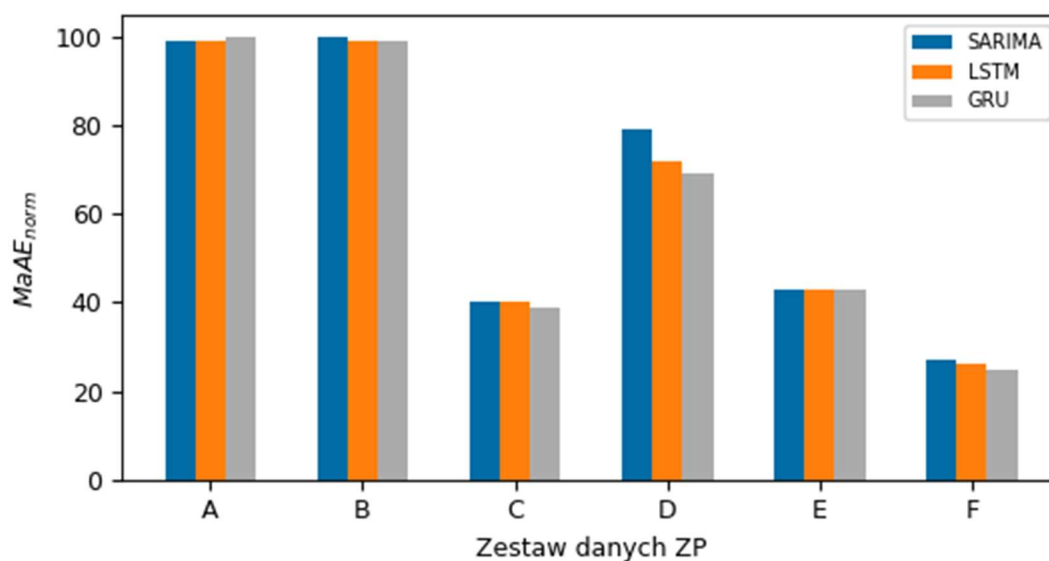
4.2.4 Porównanie wyników

Wszystkie zbadane algorytmy wykazały się skutecznością w zakresie detekcji anomalii w analizowanych zbiorach danych. Na rys. 30 przedstawiono porównanie wartości $MaAE_{norm}$ dla zestawów danych bez anomalii. Z porównania wynika, że algorytm oparty o model

SARIMA charakteryzuje się najlepszą dokładnością odwzorowania szeregu czasowego zużycia energii. Trochę gorzej zachowuje się algorytm oparty o sieć GRU, a najgorzej algorytm oparty o sieć LSTM. Na rys. 31 przedstawiono porównanie wartości $MaAE_{norm}$ dla zestawów danych z anomaliami. Dla tej metryki algorytmy wykazują zbliżone charakterystyki.



Rys. 30. Porównanie wartości $MaAE_{norm}$ dla zestawów danych bez anomalii



Rys. 31. Porównanie wartości $MaAE_{norm}$ dla zestawów danych bez anomalii

Do porównania czasów wykonywania algorytmów zastosowano miniaturowy komputer przemysłowy Raspberry PI Compute Module 4, ponieważ w warunkach rzeczywistych przy stosowaniu koncepcji Edge Computing tego typu wydajność obliczeniowa jest reprezentatywna. Porównanie czasów zawarto w tab. 9.

Tabela 9. Porównanie czasów trwania etapów algorytmów

Etap	SARIMA	LSTM	GRU
Czas kolekcjonowania próbek [h]	71	98	98
Czas wyszukiwania parametrów modelu [s]	113	2400	1980
Czas wyliczania prognozy dla jednej próbki [s]	41	1,9	1,6

Z porównania wynika, że algorytm oparty o SARIMA wymaga mniej czasu do osiągnięcia gotowości wykrywania anomalii. Wprawdzie algorytm oparty o LSTM ma 20-krotnie krótszy czas wyliczania prognozy, jednak czas wyszukiwania hiper-parametrów jest ponad 20-krotnie dłuższy. Oba algorytmy są w stanie działać w czasie rzeczywistym przy zastosowaniu platformy sprzętowej porównywalnej z Raspberry Pi, ponieważ czas analizy dla pojedynczego pomiaru jest mniejszy, niż okres próbkowania wynoszący 60 minut.

Krótszy czas startu algorytmu i większa dokładność odwzorowania szeregu czasowego pomiarów mocy czynnej wskazują ostatecznie na przewagę algorytmu opartego o SARIMA. Dodatkowy wniosek wypływający z badań to to, że algorytm wyszukiwania kratowego jest etapem wymagającym znacznie większych zasobów niż trenowanie modelu i wyliczanie prognoz.

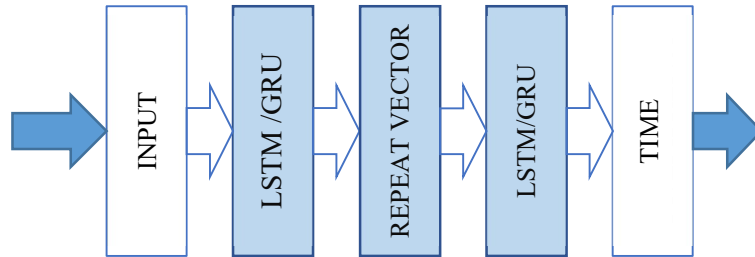
4.3 Algorytmy detekcji stosujące okres 15 minutowy

Ze względu na ograniczenia metody ARIMA/SARIMA nie ma ona zastosowania dla okresu 15 minutowego. Z kolei opisana w poprzednim rozdziale metoda wyszukiwania kratowego parametrów ma ograniczone zastosowanie dla algorytmów opartych o sieci RNN, czyli LSTM i GRU, ponieważ w obu przypadkach czas wyszukiwania parametrów modelu przekracza 15 minut.

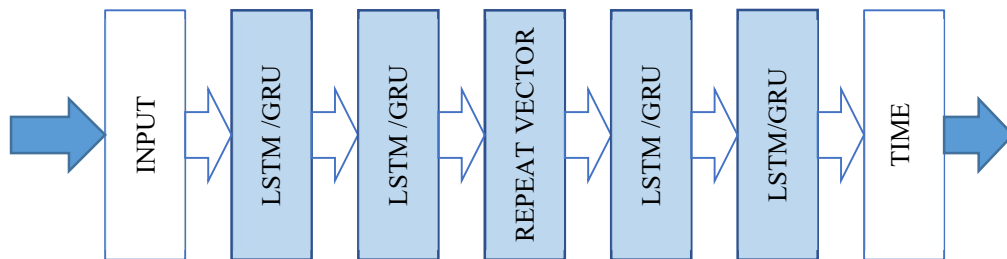
Dla okresu 15 minutowego przede wszystkim zostaną zbadane sieci typu autoenkoder, które powszechnie stosowane są do wykrywania anomalii [100], [101], [102]. Podstawowym celem autoenkodera jest rekonstrukcja danych wejściowych a pomiar błędu rekonstrukcji służy do wskazania na wystąpienie anomalii. Nietypowe punkty danych prowadzą do wyższych błędów rekonstrukcji.

Do badania wybrano modele oparte o rekurencyjną sieć neuronową typu LSTM i GRU, ponieważ została ona zaprojektowana do obsługi danych sekwencyjnych. Dla tego typu sieci przebadano podstawowy model złożony z dekodera i enkodera, zwany dalej modelem jednowarstwowym oraz model głęboki z dodatkową warstwą ukrytą, zwany dalej modelem dwuwarstwowym. Na rys. 32 i rys. 33 przedstawiono strukturę analizowanych sieci używających LSTM i GRU. Dla porównania przeprowadzono również wyliczenia dla AE z siecią splotową jednowymiarową - 1D Convolutional, której strukturę sieci przedstawiono na Rys. 34. Dodatkowo dla celów porównawczych do algorytmu detekcji anomalii użyto

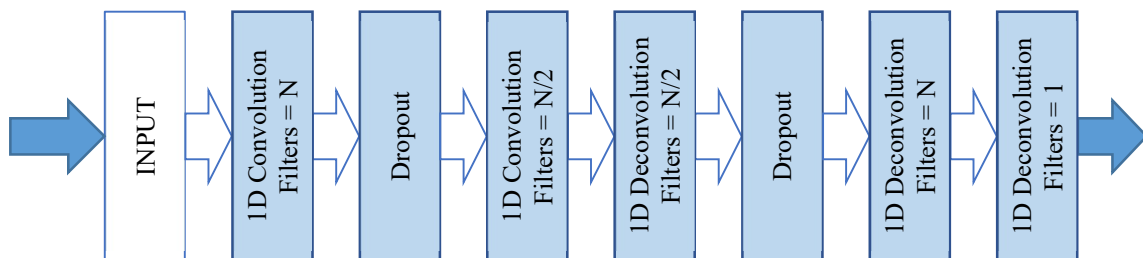
klasycznych sieci LSTM, strukturę których przedstawiono na rys. 21, oraz sieci GRU, strukturę których przedstawiono na rys. 26.



Rys. 32. Autoenkoder jednowarstwowy z siecią LSTM



Rys. 33. Autoenkoder dwuwarstwowy z siecią LSTM



Rys. 34. Autoenkoder z siecią splotową 1D Convolution

Konstruując model autoenkodera należy zdefiniować szereg hiperparametrów, których dobór decydować będzie zarówno o poprawności modelu jak i o złożoności obliczeniowej. Ponieważ algorytm wyszukiwania kratowego parametrów jest zbyt obciążający poszukiwana jest taka architektura, która zapewni jakiś poziom odporności na zjawisko dryfu koncepcyjnego.

Podstawowe parametry wybranych sieci i oznaczenia używane dalej do oznaczania symulacji:

dla AE LSTM jednowarstwowego:

- AEL1U32 - Output space = 32
- AEL1U64 - Output space = 64
- AEL1U96 - Output space = 96

- AEL1U128 - Output space = 128

dla AE LSTM dwuwarstwowego:

- AEL2U32 - Output space 1st layer = 32, 2nd layer = 16
- AEL2U64 - Output space 1st layer = 64, 2nd layer = 32
- AEL2U96 - Output space 1st layer = 96, 2nd layer = 32
- AEL2U128 - Output space 1st layer = 128, 2nd layer = 32

dla AE GRU jednowarstwowego:

- AEG1U32 - Output space = 32
- AEG1U64 - Output space = 64
- AEG1U96 - Output space = 96
- AEG1U128 - Output space = 128

dla AE LSTM dwuwarstwowego:

- AEG2U32 - Output space 1st layer = 32, 2nd layer = 16
- AEG2U64 - Output space 1st layer = 64, 2nd layer = 32
- AEG2U96 - Output space 1st layer = 96, 2nd layer = 32
- AEG2U128 - Output space 1st layer = 128, 2nd layer = 32

dla AE 1D Convolution:

- AEC7U32 - output space 1st layer = 32, 2nd layer = 16, convolution window = 7
- AEC7U64 - output space 1st layer = 64, 2nd layer = 32, convolution window = 7
- AEC9U64 - output space 1st layer = 64, 2nd layer = 32, convolution window = 9

dla sieci LSTM:

- LSTxUyy – x : number of layers = 1,2,3,4; yy : output space = 16, 32, 64

dla sieci GRU:

- GRUxUyy – x : number of layers = 1,2,3,4; yy : output space = 16, 32, 64

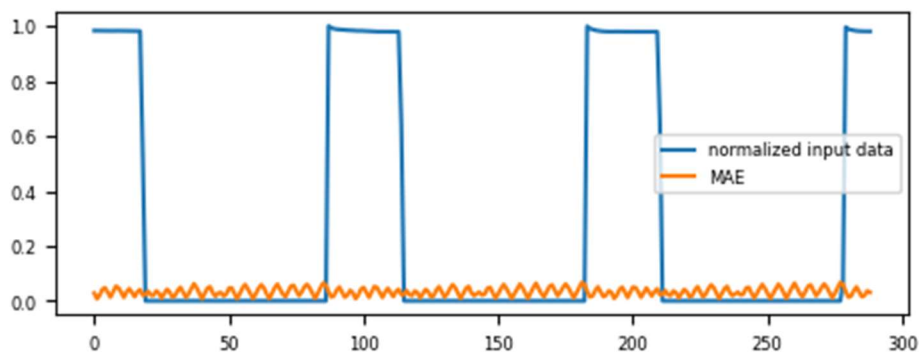
Ponieważ algorytm detekcji anomalii ma być algorytmem czasu rzeczywistego, szczególnie istotne jest określenie minimalnego okresu uczenia modeli. Modele wymagające dużej ilości danych treningowych, czyli długiego czasu, miałyby małe zastosowanie praktyczne. Do badań wybrano, zatem, okresy treningowe o długości od 3 do 7 dób.

Do rozwiązywania problemu detekcji anomalii można podejść jako do metody klasyfikacji binarnej. Wprawdzie wartości analizowanych sygnałów mają wartości ciągłe, ale zdefiniowanie wartości progowej (cutoff value) sprawi, że wartości można klasyfikować jako dodatnie lub ujemne, w zależności od tego, czy wynikowa wartość jest wyższa czy niższa od wartości progowej.

4.3.1 Analiza danych regularnych bez anomalii

Ponieważ założono wykorzystanie autoenkodera do detekcji anomalii, istotna jest odpowiedź na pytanie, w jaki sposób skonstruowany model dokonuje rekonstrukcji sygnałów

„poprawnych”, czyli bez anomalii. Dane wejściowe autoenkodera stanowią sekwencje o ustalonej długości, odpowiadającej okresowości sygnału analizowanego. W przypadku pomiarów mocy okres równa się 24 godziny, co przy okresie próbkowania 15 minut daje 96 próbek na sekwencję. W każdym kroku działania autoenkodera generowana jest sekwencja o takiej długości. Do oceny zgodności sygnału zrekonstruowanego z sygnałem wejściowym używa się miary MAE – (równanie 4.2), wyliczanej dla każdej pary sekwencji wejściowej i wyjściowej. Przykładowy przebieg wartości MAE dla danych treningowych przedstawia rys. 35.



Rys. 35. Przykładowy przebieg wartości MAE dla Autoenkodera

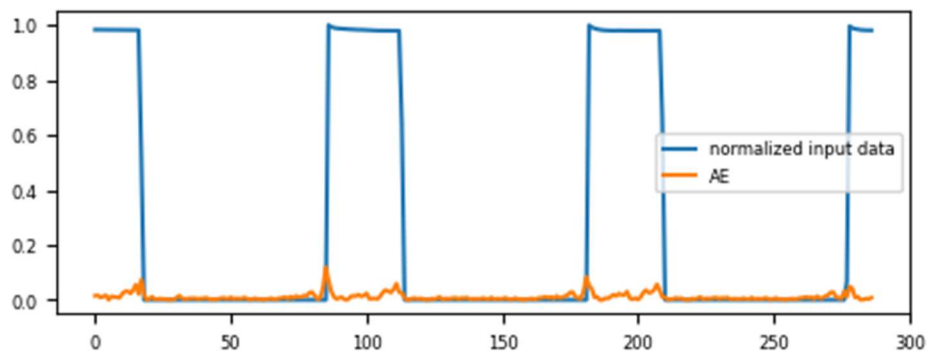
Dane wejściowe przed przetwarzaniem przez algorytm AE poddawane są normalizacji min-max, w związku z czym wartości MAE mogą być porównywane dla różnych zestawów danych wejściowych. Algorytm detekcji anomalii wyznacza maksymalną wartość MAE w okresie treningowym, która to wartość stanowi następnie próg odniesienia dla MAE w okresie testowym. Zakłada się, że im większa zgodność zrekonstruowanego sygnału dla tym większa możliwość wykrywania anomalii, ponieważ próg detekcji będzie relatywnie niski.

Celem symulacji jest wykrycie zależności pomiędzy konfiguracją modelu autoenkodera oraz długością okresu treningowego, a zdolnością do rekonstruowania sygnału wejściowego. Do oceny zdolności rekonstrukcji zastosowano wskaźnik: maksymalna wartość MAE - MAE_{max} . Oczekiwana jest jak najniższa wartość tego wskaźnika, zarówno dla danych treningowych, jak i testowych. Do oceny potencjału algorytmu te wskaźniki nie są wystarczające. Ponieważ progami detekcji ma być wartość MAE_{max} dla danych treningowych, to pożądane jest by wartość MAE_{max} dla danych testowych była jak najbardziej do niej zbliżona. Dlatego wprowadza się (równanie 4.6) dodatkowy wskaźnik MR wyliczany jako wartość bezwzględna różnicy liczby 1 i stosunku MAE_{max} dla okresu testowego do MAE_{max} dla okresu treningowego:

$$MR = \left| \frac{Test\ MAE_{max}}{Train\ MAE_{max}} - 1 \right| \quad (4.6)$$

Oczekiwane jest, aby wartość MR dla sygnałów regularnych, bez anomalii, była zbliżona do zera.

Dla symulacji opartych o model LSTM i GRU sposób wyliczania wskaźnika MR musi się różnić, ponieważ w jego przypadku w jednym kroku dla danej sekwencji wejściowej wykonywana jest predykcja jednej wartości. Dla LSTM i GRU miarą oceny jakości rekonstrukcji jest błąd bezwzględny, oznaczany jako AE (równanie 4.3). Przykładowy przebieg wartości AE dla danych treningowych dla LSTM przedstawia rys. 36.



Rys. 36. Przykładowy przebieg wartości AE dla sieci LSTM

Dla modelu LSTM do wyliczania MR używa się średnich wartości za dany okres, czyli MAE (równanie 4.7).

$$MR = \left| \frac{Test\ MAE}{Train\ MAE} - 1 \right| \quad (4.7)$$

Dla każdego z modeli przeprowadzono wyliczenie zdefiniowane poprzez Algorytm 6.

Algorytm 6. Wyliczanie MAE_{max} .

Input:

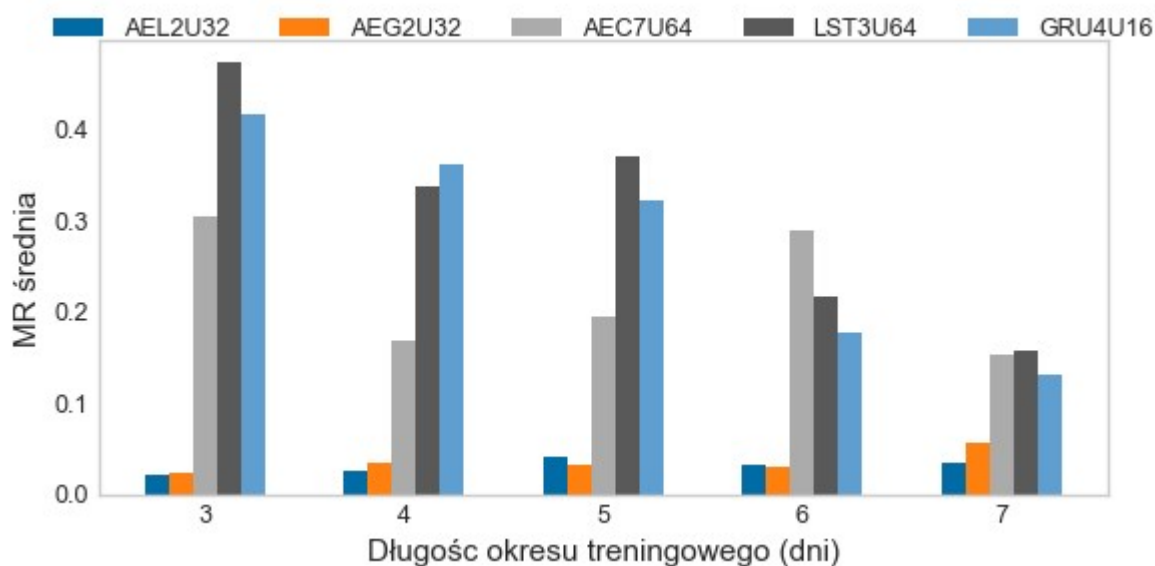
- data set— $\{ZP_1...ZP_9\}$
- train period set— $\{3d...7d\}$

- 1: **for each** train_period **in** $\{3d...7d\}$:
- 2: **for each** data_set **in** $\{ZP_1...ZP_9\}$:
- 3: Spilt data_set into train data and test data
- 4: create model
- 5: train model
- 6: make prediction for train data
- 7: calculate Train MAE for prediction
- 8: make prediction for test data
- 9: calculate Test MAE for prediction
- 10: calculate Train MAE_{max} , Test MAE_{max} and MR

Aby uniknąć zjawiska przetrenowania (overfitting), przy trenowaniu modeli zastosowano metodę wcześniejszego zatrzymywania treningu „early stopping”. Przyjęto MAE jako miarę monitorowania wydajności modelu, zbiór danych walidacyjnych odpowiadający 24 godz. Pomiarów, minimalną zmianę wskaźnika równą 0.0001, liczbę epok bez zmiany wskaźnika równą 50 i maksymalną liczbę epok równą 500.

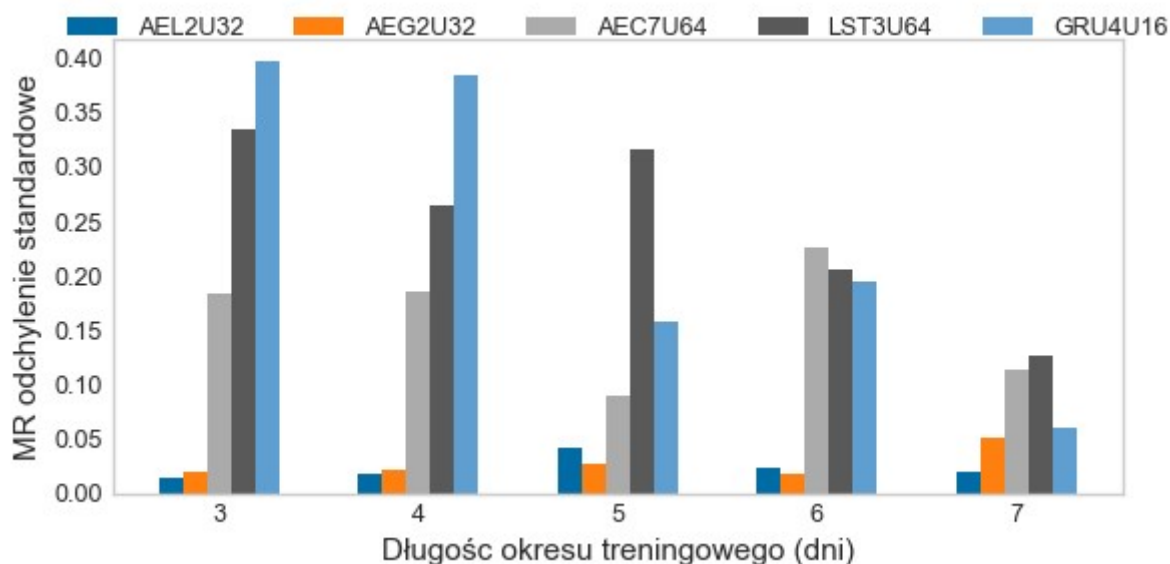
Wyniki symulacji następnie pogrupowano dla różnych długości okresu treningowego oraz dla różnych zestawów danych i wyliczono wskaźnik MR. Wartość średnia wskaźnika MR uśrednioną dla wszystkich zestawów danych ZP₁...ZP₉ w zależności od długości okresu treningowego przedstawiono w Tabeli A.1 (załącznik A) dla Autoenkodera, zaś w Tabeli A.2 dla sieci LSTM i GRU. Dla Autoenkodera najmniejszą wartość, czyli najkorzystniejszą, osiągnęła symulacja dla modelu AEL2U32 i okresu treningowego równego trzem dniom. Bardzo zbliżoną wartość osiągnął model AEG2U32 też dla okresu 3-dniowego bo tylko o 15% większą. Należy zwrócić uwagę, że zwiększenie długości okresu treningowego nie poprawia wskaźnika MR, co jest korzystne, ponieważ zapewnia krótszy czas gotowości algorytmu na detekcję anomalii. Symulacje dla Autoenkodera z siecią spłotową osiągnęły wyższe wartości wskaźnika MR.

Jeszcze gorsze wyniki osiągnęły sieci LSTM i GRU. Najmniejszą wartość MR osiągnęła symulacja dla modelu GRU2U64 i okresu treningowego równego siedmiu dniom. Istotną cechą algorytmów używających tych sieci jest fakt, że w przeciwieństwie do Autoenkodera, zwiększenie długości okresu treningowego zmniejsza wskaźnik MR, czyli te modele wymagają dłuższego okresu treningowego, co jest niewątpliwie wadą. Rys. 37 przedstawia porównanie wartości średniej wskaźnika MR dla wybranych modeli, dla tych dla których wartość średnia MR jest najmniejsza.



Rys. 37. Porównanie wartości średniej wskaźnika MR dla wybranych modeli

W Tabeli A.3 przedstawiono odchylenie standardowe wskaźnika MR dla Autoenkodera, w Tabeli A.4 dla modeli LSTM. Najmniejsza wartość odchylenia standardowego osiągnęła symulacja dla modelu L2U32 i okresu 3 dni. Na Rys. 38 przedstawiano porównanie odchylenia standardowego wskaźnika MR dla wybranych modeli.



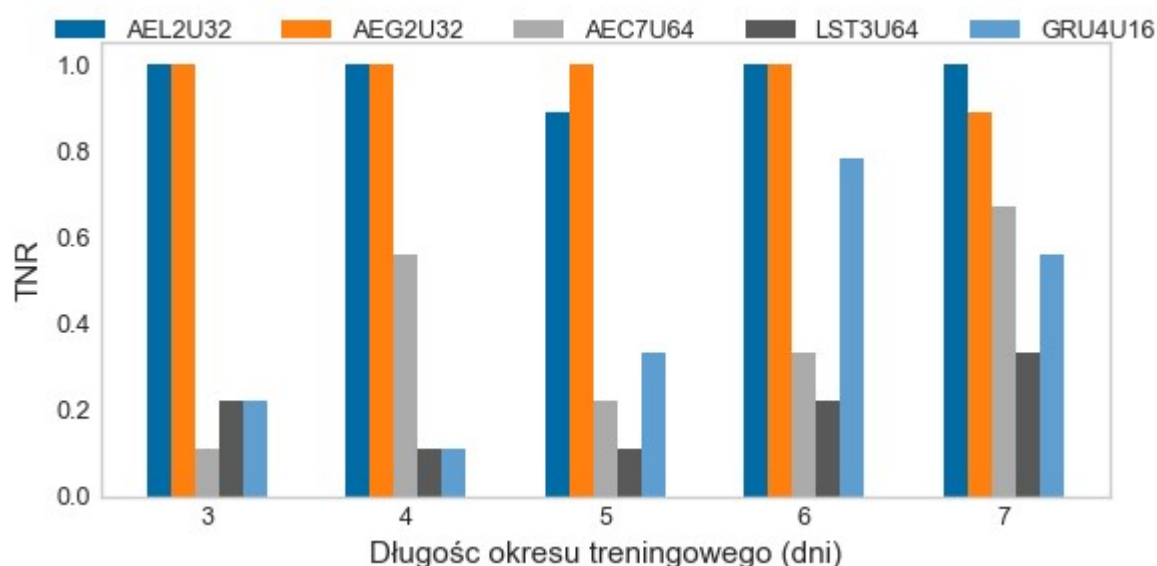
Rys. 38. Porównanie wartości średniej wskaźnika MR dla wybranych modeli

Małe wartości odchylenia standardowego wskaźnika MR wskazują, że dany model poprawnie rekonstruuje dane odpowiadające sytuacjom bez anomalii dla wszystkich zastawów danych. Według powyższych wyników model AEL2U32 osiągnął zarówno najmniejszą wartość średnią wskaźnika MR, jak i najmniejsze odchylenie standardowe. Dla Autoenkodera używających LSTM lub GRU odchylenie standardowe jest niższe, niż dla pozostałych modeli. Oznacza to, modele AE z LSTM lub GRU są najbardziej odporne na zjawisko dryfu koncepcyjnego.

Do oceny jakości modeli użyto również rachunek macierzy błędów (Confusion Matrix). Aby można było zastosować klasyfikację binarną, to należy określić wartość progową. Naturalnym wyborem jest wartość MAE_{max} dla okresu treningowego, dodatkowo wyliczenia przeprowadzono dla progów większego o 5% i 10% od $Train MAE_{max}$. Dla takich progów wyznaczono macierze błędów dla wszystkich kombinacji model/okres treningowy.

Ponieważ na tym etapie analizowane są sygnały regularne, bez anomalii, to wartości True Positives (TP) i False Negative (FN) będą zawsze równe zero. W związku z tym nie można wyznaczyć wartości F1-score, natomiast można wyliczyć wskaźnik True Negative Rate (TNR), zwany też swoistością (specifity). Wyliczone wskaźniki dla Autoenkodera dla różnych okresów treningowych zawarto w Tabeli A.5, dla sieci LSTM i GRU w tab. A.6.

Wyliczone wartości wskazują, że dla progu równego Train MAE_{max} żaden z modeli nie dokonuje klasyfikacji bezbłędnie. Natomiast dla progu większego o 10% kilka konfiguracji Autoenkodera potrafi klasyfikować dane bezbłędnie. Ponownie model AEL2U32 wykazuje najlepsze własności w zakresie rekonstruowania sygnału. Nieco gorsze wyniki osiągnął model AEL2U32. Na rys. 39 przedstawiono porównanie wartości wskaźnika TNR dla wybranych modeli dla progu $Cutoff = 1.1 * MAE_{max}$.

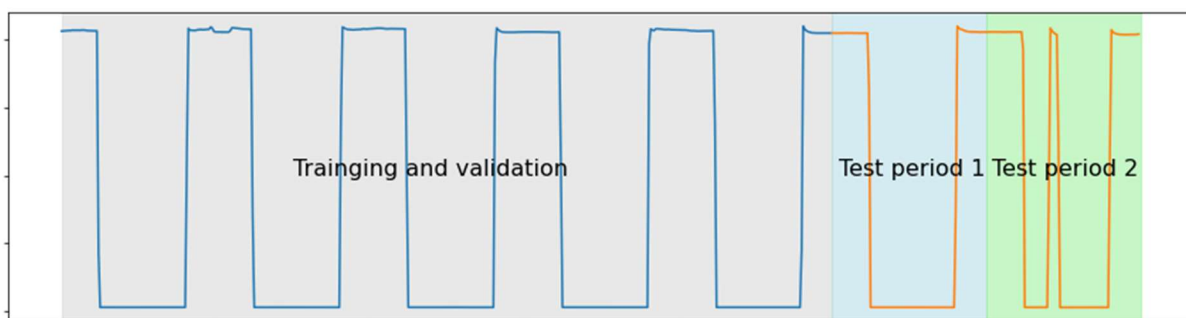


Rys. 39. Porównanie wartości wskaźnika True Negative Rate (TNR) dla wybranych modeli dla progu $Cutoff = 1.1 * MAE_{max}$

4.3.2 Analiza danych z anomaliami

Kolejnym krokiem na drodze do weryfikacji wpływu konfiguracji Autoenkodera na skuteczność algorytmu detekcji jest przeprowadzenie testu, w którym wyliczany będzie błąd rekonstrukcji dla sygnałów z anomaliami. Ponieważ dla algorytmu detekcji istotna jest zdolność do rozróżniania sygnału poprawnego od zaburzonego, szereg czasowy pomiarów zawierający anomalię dzielony jest na 3 odcinki następujące po sobie. Pierwszy to okres treningowy, drugi – odcinek bez anomalii (T1), trzeci – odcinek z anomalią (T2), jak przedstawiono to na rys. 40. Analogicznie do wyliczeń dla zestawów danych bez anomalii ZP_1, ZP_2, \dots, ZP_9 wyliczane są wskaźniki MAE_{max} oraz MR dla okresu T1. Dodatkowo wyliczany jest wskaźnik AR (równanie 4.8), który przedstawia amplitudę anomalii.

$$AR = \frac{T2 MAE_{max}}{T1 MAE_{max}} \quad (4.8)$$



Rys. 40. Zasada podział szeregu czasowego pomiarów zawierających anomalie

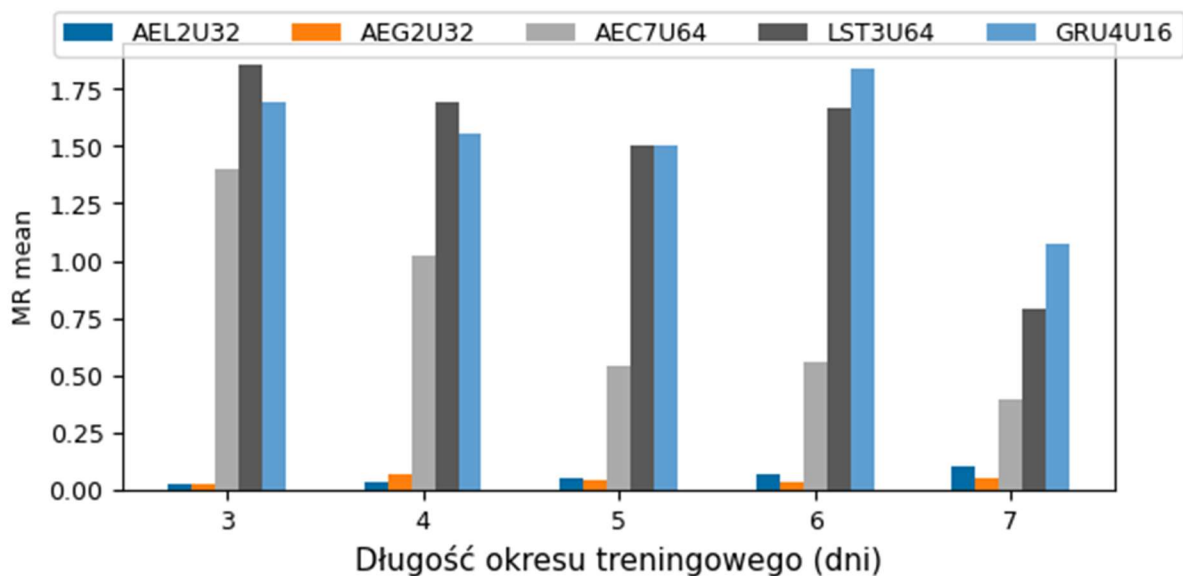
Wyliczenia przeprowadzono dla 6 zestawów pomiarów zawierających anomalie oznaczone ZP_A , ZP_B , ... ZP_F i które zostały przedstawione na rys. 15. Dla każdego z modeli przeprowadzono wyliczenie zdefiniowane przez Algorytm 7.

Algorytm 7. Wyliczanie MAE_{max} dla okresu treningowego, testowego T1 i T2

Input:

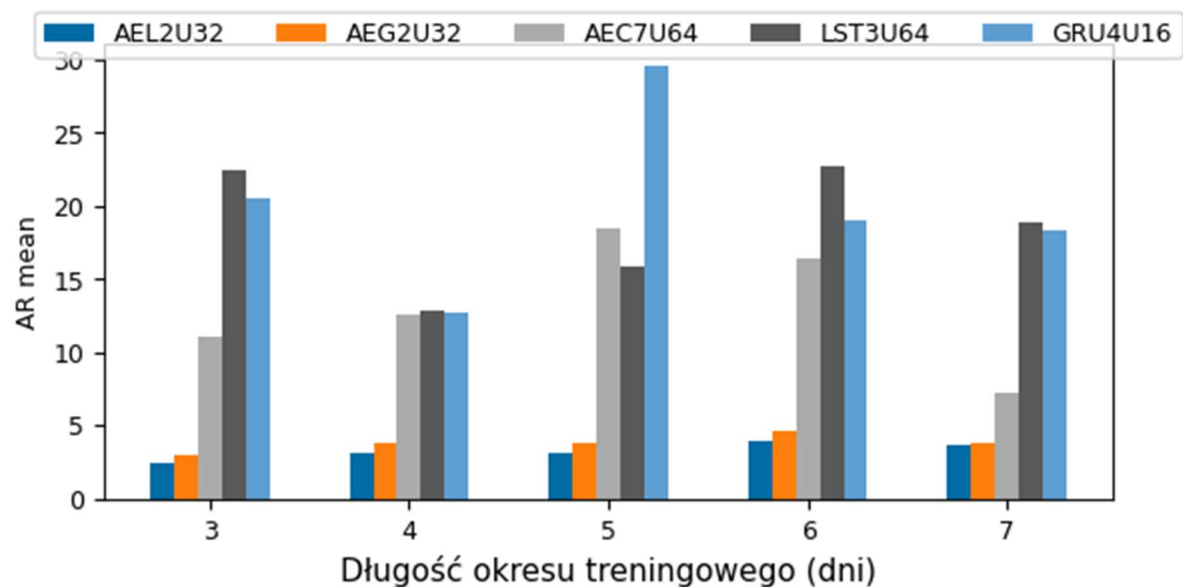
- data set— $\{ZP_A...ZP_F\}$
 - train period set— $\{3d...7d\}$
- 1: **for each** train_period in $\{3d...7d\}$:
 - 2: **for each** data_set in $\{ZP_1...ZP_9\}$:
 - 3: Spilt data_set into train period and test period T1 and T2
 - 4: create model
 - 5: train model
 - 6: make prediction for train data
 - 7: calculate Train MAE for prediction
 - 8: make prediction for test period T1
 - 9: calculate T1 MAE for prediction
 - 10: make prediction for test period T2
 - 11: calculate T2 MAE for prediction
 - 12: calculate Train MAE_{max} , T1 MAE_{max} , T2 MAE_{max} , MR, AR

Wyniki symulacji pogrupowano dla zestawów danych i wyliczono wartość średnią MR i AR. Wartości średnie wskaźnika MR przedstawiono w Tabeli A.7 dla modeli Autoenkodera zaś w Tabeli A.8, dla modeli LSTM i GRU. Ponownie wskaźniki MR są zdecydowanie gorsze dla modeli LSTM i GRU, niż dla modeli Autoenkodera. Ponownie różnica pomiędzy modelami Autoenkodera z siecią LSTM i GRU jest niewielka. Na rys. 41 przedstawiano porównanie wartości średniej wskaźnika MR dla okresu T1 dla wybranych modeli.



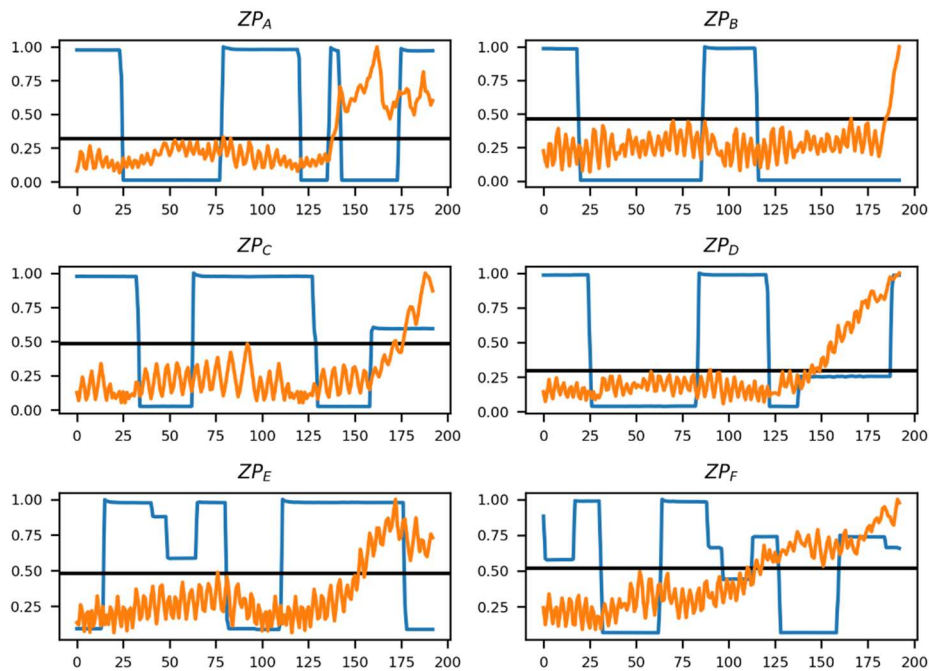
Rys. 41. Porównanie wartości średniej wskaźnika MR dla okresu T1 dla wybranych modeli

Wartości średnie wskaźnika AR przedstawiono w Tabeli A.9 dla modeli Autoenkodera zaś w Tabeli A.10 dla modeli LSTM i GRU. Tym razem wskaźniki dla modeli LSTM są lepsze niż dla modeli Autoenkodera – im większa wartość AR, tym lepsze wyniki można uzyskać. Warto zauważyć, że najgorszy wskaźnik AR osiągnął model AEL2U32, które przewyższał pozostałe modele pod względem innych wskaźników. Jedną z najniższych wartości AR osiągnął również model AEG2U32. Jednak dla wszystkich typów anomalii wartość wskaźnika AR jest większa niż 1 więc powinna być wystarczająca do poprawnej detekcji. Na rys. 42 przedstawiono porównanie wartości średniej wskaźnika AR dla wybranych modeli.

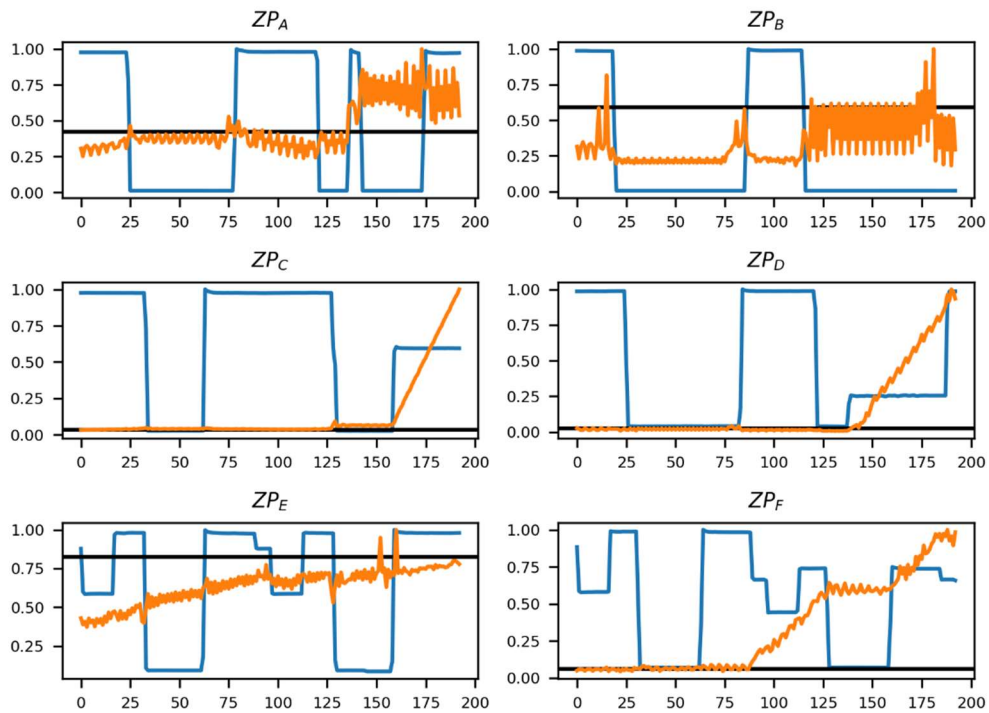


Rys. 42. Porównanie wartości średniej wskaźnika AR dla wybranych modeli

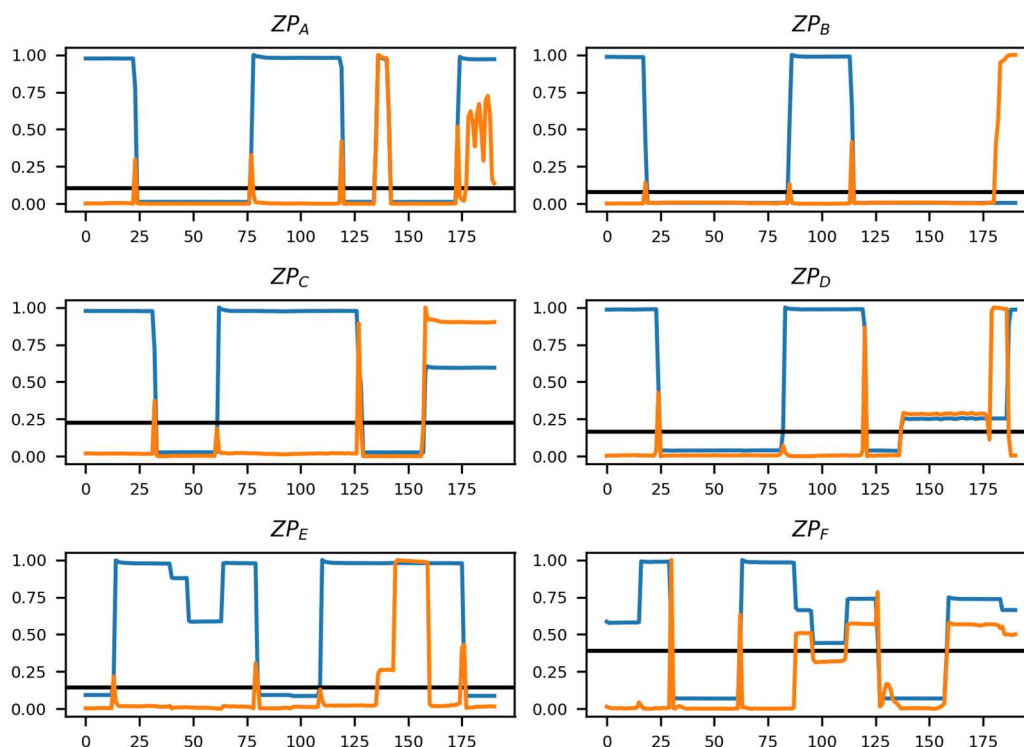
Przykładowe przebiegi wartości MAE w trakcie symulacji dla Autoenkodera LSTM przedstawiono na rys. 43 i rys. 44. Przykładowe przebiegi wartości AE w trakcie symulacji dla modeli LSTM przedstawiono na rys. 45.



Rys. 43. Przebieg wartości MAE podczas symulacji dla modelu AEL2U32 i okresu treningowego równego trzem dniom

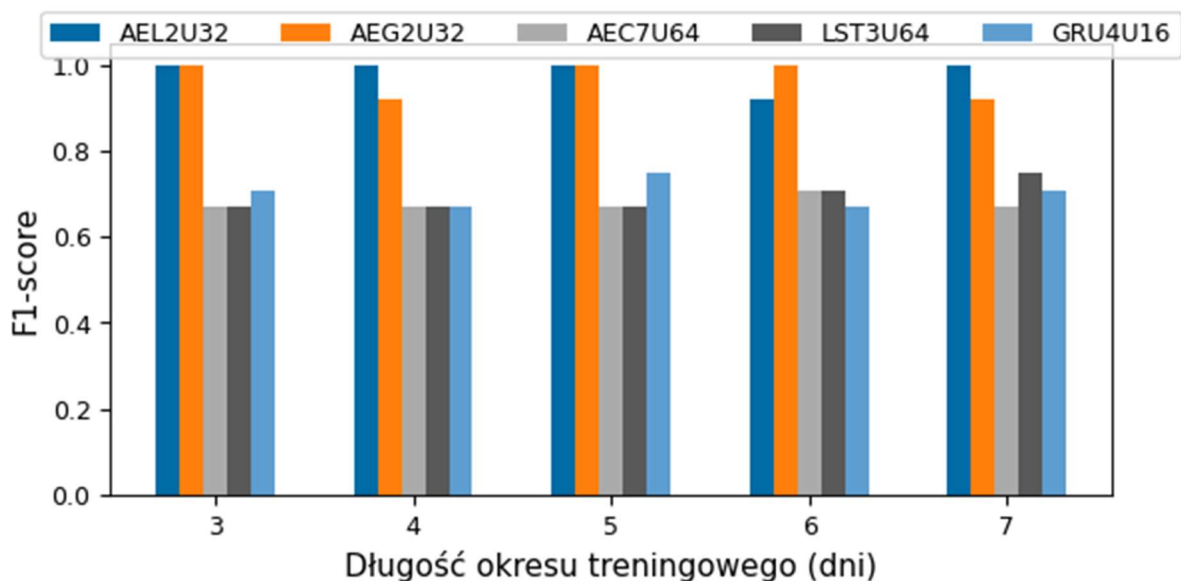


Rys. 44. Przebieg wartości MAE podczas symulacji dla modelu AEC9U64 i okresu treningowego równego trzem dniom



Rys. 45. Przebieg wartości AE podczas symulacji dla modelu LST2U32 i okresu treningowego równego czterem dniom

Analogicznie do analizy danych regularnych, do oceny jakości modeli użyto rachunek macierzy błędów i użyto tych samych wartości progów. Tym razem do dyspozycji jest pełen zestawem wartości, ponieważ okres T1 odpowiada wartości negatywnej, zaś okres T2 odpowiada wartości pozytywnej. Jeżeli sygnał w okresie T1 jest poniżej progu, to jest to przypadek True Negative, jeżeli powyżej to jest to przypadek False Positive. Jeżeli sygnał w okresie T2 jest poniżej progu to jest to False Negative, jeżeli powyżej to jest to True Positive. Bazując na tej zależności można wyznaczyć wartości F1-score. Wyliczone wskaźniki dla Autoenkodera dla różnych okresów treningowych zawarto w Tabeli A.11, dla sieci LSTM i GRU w tabeli A12. Wyliczone wartości wskazują, że dla progu równego Train MAE_{\max} żaden z modeli nie osiąga wartości F1-score równej 1. Natomiast dla progu większego o 10% kilka Autoenkodera z LSTM potrafi osiąga wartości F1-score równe 1. Ponownie model AEL2U32 wykazuje najlepsze własności w zakresie rekonstruowania sygnału. Porównanie wartości wskaźnika F1-score dla wybranych modeli dla progu $\text{Cutoff} = 1.1 * \text{MAE}_{\max}$ przedstawiono na rys. 46.



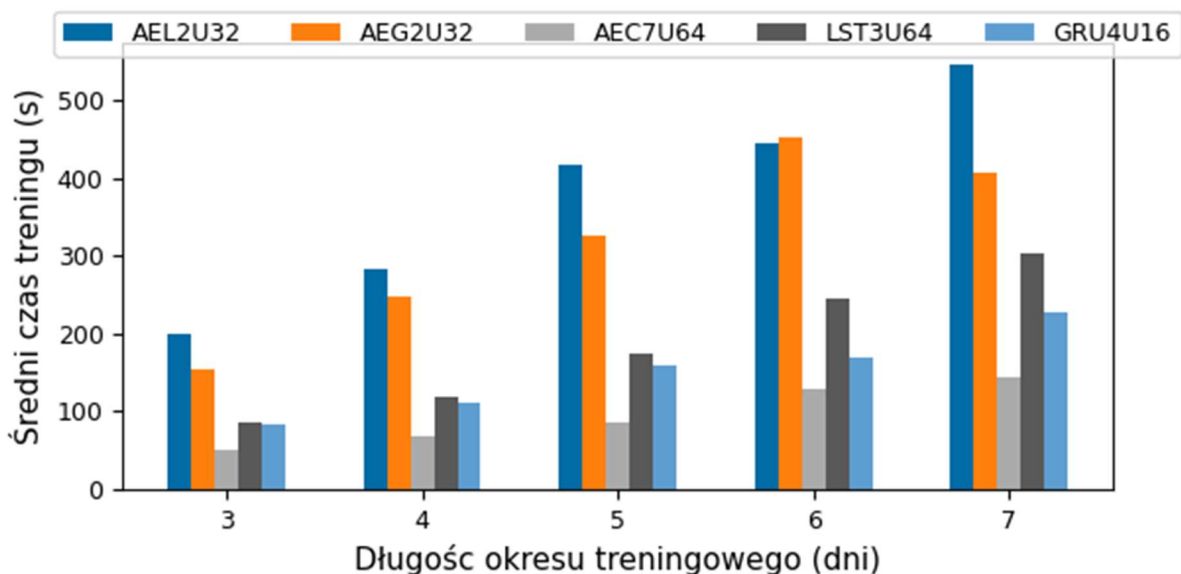
Rys. 46. Porównanie wartości wskaźnika F1-score dla wybranych modeli dla progu $\text{Cutoff} = 1.1 * \text{MAE}_{\text{max}}$

4.3.3 Porównanie wyników

W oparciu o powyższe wyniki można stwierdzić, że najlepsze wyniki w zakresie detekcji anomalii osiągnęły algorytmy oparte o Autoenkoder z siecią LSTM. Po pierwsze algorytmy te osiągnęły najlepszą wartość F1-score, po drugie najlepszy wskaźnik TNR. Po drugie osiągnęły najniższe wartości MR. Po trzecie wymagają one najkrótszych okresów treningowych.

W oparciu o metodę Autoenkodera można skonstruować algorytm dokonujący poprawnej detekcji anomalii w pomiarach mocy systemów oświetlenia. Możliwe jest takie skonstruowanie modelu, aby algorytm działał poprawnie przez cały okres rejestracji danych, niezależnie od pory roku. Okres trenowania modelu może być równy trzem dniom, co oznacza, że algorytm po tym czasie jest gotowy do detekcji anomalii.

Ważnym miarą dla badanych algorytmów jest czas obliczeniowy, niezbędny do trenowania modelu. Wyliczenia takie przeprowadzono dla wszystkich zestawów danych ZP₁...ZP₉, dla różnych okresów treningowych. Uśrednione wartości czasów zawarto w Tabeli A.13 dla Autoenkodera, w Tabeli A.14 dla LSTM i GRU. Porównanie czasów dla wybranych modeli przedstawiono na rys. 47.



Rys. 47. Średni czas treningu sieci dla wybranych modeli

Najkrótszy czas treningu wykazał model Autoenkodera z siecią spłotową. Nieco dłuższy czas zanotowały modele oparte o sieć LSTM i GRU, przy czym sieć GRU wykazuje krótsze czasy wraz ze wzrostem długości okresu treningowego. Najgorsze czasy zanotowały modele Autoenkodera z siecią LSTM oraz GRU, przy czym różnica pomiędzy nimi jest najwyższa dla okresu 7 dni i wynosi 25%.

Dla porównania z algorytmami omówionymi w rozdz. 4.2 wykonano symulacje algorytmów wykorzystujących Autoenkoder, które wykazały najlepsze wyniki, na miniaturowym komputerze przemysłowym Raspberry PI Compute Module 4. Wyniki przedstawiono w tab. 10. Zastosowanie architektury GRU wymaga około 11% mniej czasu niż LSTM, czasy prognozowania są identyczne. Dla obu modeli czas trenowania nie przekracza okresu 15 minut, a więc są w stanie działać w czasie rzeczywistym.

Tabela 10. Porównanie czasów trwania etapów algorytmów Autoenkodera

Etap	AEL2U32	AEG2U32
Czas kolekcjonowania próbek [h]	72	72
Czas trenowania modelu [s]	780	690
Czas wyliczania prognozy dla jednej próbki [s]	2	2

4.4 Wnioski z badań symulacyjnych

Wyniki badań symulacyjnych wykazały, że najlepszym rozwiązaniem dla detekcji anomalii w czasie rzeczywistym w systemie oświetlenia drogowego w oparciu o analizę danych zużycia energii jest zastosowanie Autoenkodera stosującego sieci typu LSTM lub GRU. Rozwiązanie takie wymaga najkrótszego czasu trenowania modelu co ma istotne znaczenie dla praktycznego zastosowania, ponieważ system monitorujący może szybciej rozpocząć detekcję

anomalii. Symulacja algorytmów na komputerze przemysłowym typu Edge Computing wykazała, że są w stanie działać w czasie rzeczywistym.

Rozwiązanie stosujące maszynowe uczenie z sieciami rekurencyjnymi przewyższa algorytmy stosujące algorytmy regresji. Przede wszystkim, nie ma takich ograniczeń, można stosować większą rozdzielczość do analizy danych. Czas prognozy dla modeli SARIMA z okresem jednogodzinnym jest porównywalny z czasem prognozy dla Autoenkodera z okresem 15-minutowym. Dodatkowym argumentem za stosowaniem rozwiązań z sieciami neuronowymi jest dynamicznie rozwijające się wsparcie sprzętowe poprzez wyposażanie komputerów wbudowanych do zastosowań przemysłowych w akceleratory maszynowego uczenia.

5 Lampa do oświetlania inteligentnych miast

Dynamiczny rozwój półprzewodnikowych źródeł światła, przede wszystkim opartych o technologię LED (Light-Emitting Diode) zrewolucjonizował konstrukcje lamp i opraw oświetleniowych we wszystkich zastosowaniach, np.: oświetlenie mieszkań, domów, biur, ulic, przemysł motoryzacyjny, itp. Rewolucja dotyczy możliwości użycia nowych form zewnętrznych, wynikającej z cech technologii LED takich jak miniaturyzacja czy niższe temperatury pracy. Rewolucja dotyczy również zastosowania urządzeń elektronicznych stanowiących układy zasilające. W wyładowczych źródłach światła układ zasilający oparty był o układy indukcyjne, które ostatnio były stopniowo zastępowane przez układy elektroniczne. Technologia LED wymusiła stosowanie układów elektronicznych, a pojawiające się stale nowe funkcjonalności wymagają użycia w konstrukcji układów mikroprocesorowych, dlatego coraz częściej w konstrukcji lamp stosuje się inteligentne układy zasilające. To wszystko sprawia, że lampę rozumianą jako oprawa ze źródłem światła, wbudowanym lub wymiennym, coraz częściej traktuje się jako urządzenie elektroniczne a nawet inteligentne [103]. W szczególności dotyczy to lamp przeznaczonych do oświetlania ulic, przestrzeni publicznych i ciągów komunikacyjnych, które stają się urządzeniami Internetu Rzeczy, co oznacza urządzenia wyposażone w czujniki i elementy wykonawcze, zaprogramowane do realizacji określonej aplikacji i komunikujące się z innymi urządzeniami oraz aplikacją chmurową poprzez dedykowaną sieć lub Internet. Inteligentna lampa może sama przetwarzać pozyskiwane dane realizując ideę Edge Computing, przykłady takich rozwiązań opisano w publikacjach [104] i [105]. Przetwarzanie może obejmować analizę zdarzeń, strumienia pomiarów, danych z kamer wizyjnych. Do przetwarzania danych mogą być zastosowane algorytmy uczenia maszynowego, realizowane zarówno lokalnie, jak i w aplikacji chmurowej.

Inteligentna lampa drogowa może realizować takie zadania jak: monitorowanie ruchu pojazdów i przechodniów, monitorowanie jakości powietrza i hałasu, monitorowanie bezpieczeństwa publicznego, np. przez wykrywanie wystrzałów i zgromadzeń, obsługa punktów ładowania pojazdów elektrycznych, monitorowanie śmieci i zaśmiecania, itd. Lamy drogowe wraz ze słupami mogą być wykorzystywane jako platformy reklamowe wykorzystującą wyświetlacze typu Digital Signage. Poprzez wbudowane urządzenia komunikacyjne Vehicle-to-everything (V2X) lampy drogowe stają się również elementami inteligentnego transportu. Inteligentne lampy drogowe przestają być postrzegane jako statyczna infrastruktura, ale raczej jako „inteligentna sieć”, w pełni zorientowana w sytuacji, kontekście i parametrach środowiska.

Realizacja powyższych zadań wymaga, aby układy sterujące lamp inteligentnych mogły obsługiwać różne kanały komunikacyjne wymienione w rozdz. 3.1, przede wszystkim łączności bezprzewodowej. Układy sterujące lamp inteligentnych muszą być wyposażone w zasoby

umożliwiającej integrację z chmurowymi systemami Smart City oraz umożliwić implementację algorytmów uczenia maszynowego.

Szczególnym wyzwaniem dla układów sterujących inteligentnych lamp jest obsługa różnych typów sensorów i urządzeń wejściowych. Przykładowe sensory, które mają zastosowanie w systemach Smart City:

- czujnik temperatury, wilgotności, ciśnienia atmosferycznego
- czujnik opadów
- czujnik natężenia światła
- czujnik promieniowania UV
- czujnik natężenia dźwięku
- czujnik zanieczyszczeń powietrza
- czujnik lotnych związków organicznych

Przykładowe urządzenia wejściowe:

- kamera wizyjna
- kamera IR
- LIDAR
- radar
- sonda akustyczna

Wdrażanie tak wysoce heterogenicznych systemów, jak systemy inteligentnego oświetlenia sprawia, że stosowane są różne podejścia i technologie, a próby definiowania standardów przemysłowych i ujednoczenia rozwiązań będą odbywały się w najbliższej przyszłości. Należy stwierdzić, że istnieje konieczność zdefiniowania architektury uniwersalnej platformy sterowania elementami infrastruktury oświetleniowej. W ramach projektu badawczo rozwojowego INFOLIGHT (www.multimed.biz/infolight) w Politechnice Gdańskiej i w przedsiębiorstwie TSTRONIC sp. z o. o. powstała taka platforma sterowania. Rozwiązanie zostało stworzone pod nazwą INFOLIGHT Edge Device, w skrócie ILED.

5.1 Koncepcja platformy sterowania

Różnorodność konfiguracji i scenariuszy, w jakich może funkcjonować inteligentna lampa wymaga stworzenia platformy sterowania rozumianej jako zestaw specjalizowanych modułów, w oparciu, o które może powstać dostosowana do potrzeb implementacja. Platforma jest integrowana z lampą uliczną, więc zapewnia sterowanie źródłem światła LED w zakresie

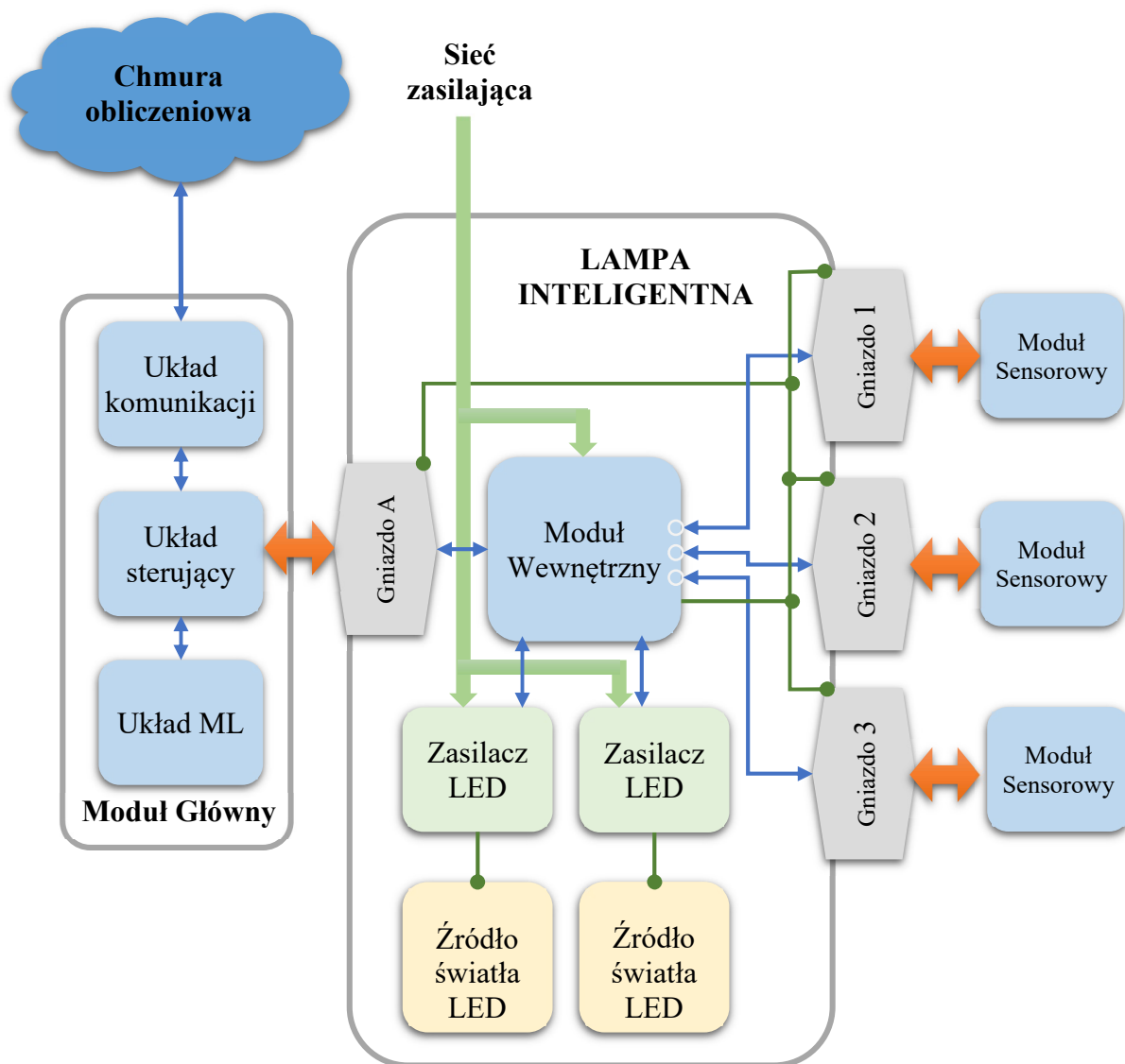
poziomu natężenia, barwy światła i kształtu strumienia. Oczywistym ograniczeniem jest konstrukcja lampy drogowej, która ma skończone rozmiary i pracuje w warunkach zewnętrznych, narażona na zmiany temperatury, opady deszczu i wibracje. Należy również wziąć pod uwagę możliwość rekonfigurowania lampy inteligentnej w warunkach polowych, tak by główne moduły można było dobierać i wymieniać w trakcie eksploatacji lampy.

Opracowana koncepcja zakłada, że podstawowa funkcjonalność obejmująca sterowanie źródłem światła, zasilanie oraz wewnętrzną sieć komunikacyjną obsługiwana jest przez moduł wbudowany w lampę. Moduł sterujący, sensory i urządzenia wejściowe są modułami wymiennymi, montowanymi na zewnątrz oprawy, poprzez specjalistyczne interfejsy. Interfejsy muszą zapewniać transmisję szerokopasmową, na przykład cyfrowy sygnał wideo o rozdzielczości 640*480 pikseli, 30 klatek na sekundę to przy kodowaniu MJPEG to około 10 Mbp/s, przy kodowaniu H.264 to około 1 Mbp/s. Interfejsy muszą zapewniać również odpowiednie zasilanie dla modułów zewnętrznych.

Architektura platformy ma charakter modułowy na dwóch poziomach. Na wyższym poziomie funkcjonalność występują moduły grupujące funkcjonalności, połączone magistralą transmisji danych. Na niższym poziomie, to wymiennosc specjalizowanych układów, w szczególności dotyczy to Modułu Głównego, który może występować z różnymi układami komunikacyjnymi. Dzięki modułowości konstrukcja platformy ILED jest otwarta, konstruując nowe typy modułów można rozszerzać typy sensorów i układów wykonawczych. Modułowość umożliwia również stosowanie zróżnicowanych konfiguracji sterowników w ramach jednej instalacji, np. w grupie lamp na jednej ulicy sąsiednie lampy mają różne sensory. Wielomodalność systemu jest zapewniona przez zróżnicowanie modułów w ramach instalacji. Schemat blokowy platformy sterowania inteligentnej lampy przedstawiono na rys. 48. Poszczególne elementy platformy sterowania inteligentnej lampy to:

- Moduł Główny, składający się z Układu sterującego, Układu Komunikacji i Układu ML. Moduł główny jest obligatoryjnym elementem platformy sterowania.
- Układ Sterujący odpowiedzialny za funkcje inteligentnej lampy oraz za sterowanie interfejsem chmury obliczeniowej.
- Układ Komunikacji odpowiedzialny za połączenie Modułu Głównego z chmurą obliczeniową lub ze stanowiskiem zdalnego zarządzania. Najczęściej wykorzystywane technologie GSM w standardzie LTE lub 5G, możliwe jest również wykorzystanie komunikacji w standardzie IEEE 802.11. Układ Komunikacji odpowiada również za komunikację WPAN: Zigbee, Bluetooth i WiFi MESH.
- Układ ML to akcelerator uczenia maszynowego, układ wspomagający Układ Sterujący w zakresie przetwarzania informacji pochodzących z sensorów i urządzeń wejściowych.

- Moduł Wewnętrzny jest odpowiedzialny za dostarczenie zasilania do modułów zewnętrznych, sterowanie źródłem światła oraz za wewnętrzną sieć komunikacyjną, łączącą Moduły Sensorów z Modułem Głównym. Sterowanie źródłem światła odbywa się poprzez wykorzystanie standardu sterowania zasilaczami DALI, wymienionym w rozdz. 3.1. DALI umożliwia łączenie kilku zasilaczy LED na jednej magistrali, przez co możliwe jest konstruowanie lamp z kilkoma źródłami światła.
- Moduł Sensorów to różne czujniki, np. czujniki temperatury, wilgotności, ciśnienia atmosferycznego lub urządzenia wejściowe takie kamera video, radar, LIDAR.
- Gniazdo M – dedykowane gniazdo z interfejsem do Modułu Głównego
- Gniazdo 1...4 - gniazda z interfejsami do Modułów Sensorów



Rys. 48. Schemat blokowy platformy sterowania inteligentnej lampy

Wymiennosc Modulu Glownego umożliwia zmianę technologii komunikacyjnej wykorzystywanej przez inteligentną lampę w trakcie eksploatacji. Jest to szczególnie istotne w warunkach wdrażania nowych technologii komunikacyjnych, takich jak 5G. Po uruchomieniu nowej transmisji na obszarze funkcjonowania inteligentnych lamp, można będzie zastąpić Moduł Główny z układem komunikacji LTE na nowy z układem komunikacji 5G.

Moduły zewnętrzne (Moduł Główny i Moduły Sensorów) są połączone z inteligentną lampą za pomocą gniazd z interfejsem komunikacyjnym i zasilaniem. Gniazda te muszą zapewnić odporność na warunki zewnętrzne i jednocześnie być proste w obsłudze. Wymagania takie spełnia standard przemysłowy opracowanych przez międzynarodową organizację Zhaga Consortium, o których mowa była w rodz. 3.1. W specyfikacji Zhaga Book 18 zdefiniowano gniazdo z mocowaniem bagnetowym – Zhaga SR. Gniazdo montowane jest w otworze w obudowie oprawy a zalecenia Book 18 definiują wszystkie parametry mechaniczne i elektryczne co zapewnia współpracę konstrukcji różnych producentów. Gniazdo wyposażone jest w 4-biegunowy interfejs elektryczny, z napięciem znamionowym 24 V DC i maksymalnym prądzie 500 mA. Oryginalnie interfejs jest dedykowany do standardu DALI, zapewniającym transmisję z prędkością 1200 Bodów, co jest dalece niewystarczające na potrzeby sensorów szerokopasmowych. Dlatego też w platformie ILED zastosowano nowatorskie rozwiązanie, wykorzystujące połączenie standardu USB 2.0 z podwyższonym napięciem zasilania w stosunku do napięcia zgodnego z tym standardem 5V do napięcia 24V. Zmienione parametry zasilania umożliwiają stosowanie modułów o większym zapotrzebowaniu na energię. W ten sposób powstała magistrała ILED-BUS będąca połączeniem standardu USB 2.0 i mechanicznego interfejsu Zhaga SR. Najważniejsze jej cechy:

- Topologia komunikacyjna: gwiazda z wyróżnionym urządzeniem Modułem Głównym.
- Obsługa do 3 modułów sensorowych.
- Interfejs mechaniczny Zhaga SR.
- Interfejs elektryczny 4-biegunowy.
- Transmisja danych jest zgodny ze standardem USB 2.0 opisanym przez IEC 62680-1, zapewniając maksymalną szybkością transmisji danych 480 Mbit/s.
- Napięcie zasilania: 24 V DC, max prąd: 500 mA, maksymalne obciążenie interfejsu 12 W.

Interfejs elektryczny to 4 przewody:

1. DC+ dodatni biegun zasilania
2. DA- masa zasilania
3. LSI transmisja danych Data-
4. DA+ transmisja danych Data+

Opisane powyżej rozwiązanie jest przedmiotem zgłoszenia patentowego: Śmiałkowski T., Czyżewski A. - „Lampa uliczna do oświetlenia inteligentnych miast”, numer P.441273 z 25 maja 2022 r.

Ważną rolę w skonstruowanej platformie pełni Układ ML, czyli akcelerator uczenia maszynowego, ponieważ układ sterujący może wykorzystywać mechanizmy uczenia maszynowego. W fazie uczenia się zdarzenia wykrytego przez czujniki, informacje są wysyłane w formie komunikatów za pośrednictwem modułu komunikacyjnego do chmury obliczeniowej, gdzie podlegają przetwarzaniu i są wykorzystywane do dotrenowania modelu uczenia maszynowego, przy zastosowaniu dedykowanych lub publicznie dostępnych algorytmów. Po wytworzeniu model dostosowywany jest do pracy w środowisku wbudowanym, pobierany jest w formie reprezentacji kodowej przez moduł sterujący i umieszczany w akceleratorze uczenia maszynowego. W trakcie normalnego działania moduł sterujący kieruje strumień danych z sensorów do akceleratora, który realizuje funkcje wnioskowania wytwarzając dane wyjściowe. Moduł sterujący odczytuje te dane, które służą do lokalnego sterowania lub są przesyłane do chmury obliczeniowej.

5.2 Konstrukcja

Dostosowanie do standardu przemysłowego Zhaga SR wprowadza pewne ograniczenia konstrukcyjne głównie związane z wielkością obudowy. Standardowa typoszereg obudów obejmuje obudowy o średnicy 40 i 80 mm i wysokości od 24 mm do 58 mm. Obudowy takie oferowane są komercyjnie przez kilku producentów, przykładowa obudowa firmy TE Connectivity przedstawiona jest na rys. 49.



Rys. 49. Obudowa modułu Zhaga SR firmy TE Connectivity [źródło: <https://www.te.com/usa-en/products/connectors/lighting-connectors/street-lighting-controls/zhaga-book-18-street-lighting-connectors.html>, dostęp 2023-07-13]

W ramach projektu INFOLIGHT wyprodukowano prototyp platformy ILED. Prototyp składa się z Modułu Głównego, Modułu Wewnętrznego oraz szeregu Modułów Sensorów: Modułu Kamery, Modułu Sondy Akustycznej, Modułu Radaru, Modułu Sensora Środowiskowego, Modułu Casambi.

5.2.1 Moduł Główny

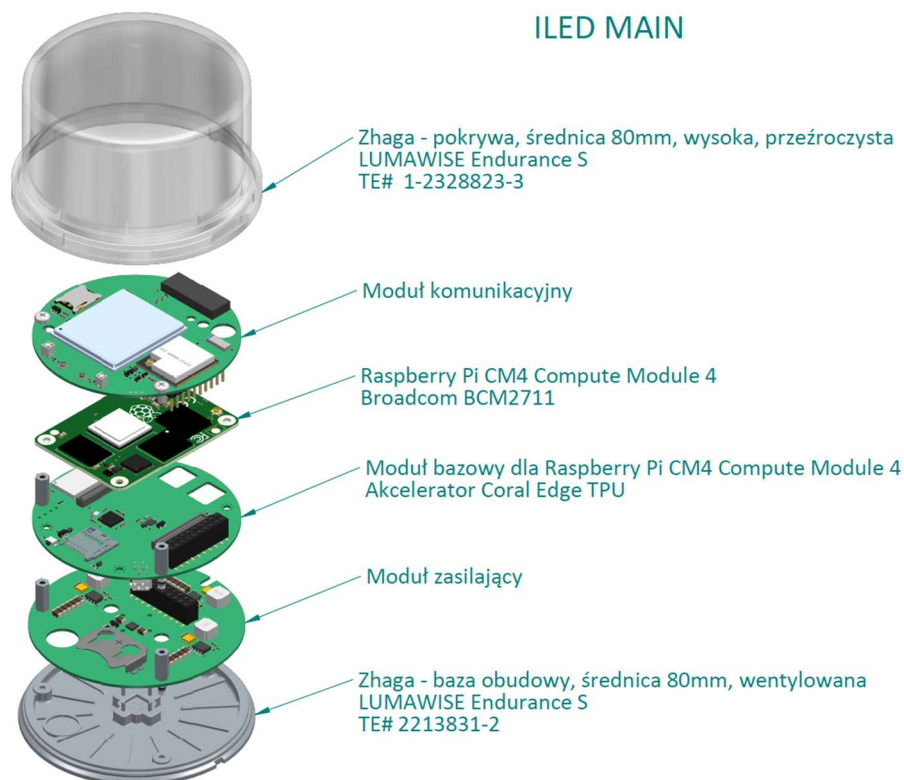
Moduł Główny (ILED-MAIN) jest modułem złożonym z kilku płyt PCB połączonych kołkami dystansowymi w strukturę warstwową. Taka konstrukcja zapewnia elastyczność i skalowalność modułu. Na przykład: moduł komunikacyjny jest zaprojektowany w dwóch wariantach: jeden z modem LTE i modułem WiFi MESH, drugi z modemem 5G.

Moduł wykorzystuje standardową obudowę Zhaga SR o średnicy 80 mm i wysokości 58 mm. Konstrukcję modułu zaprezentowano na rys. 50. Układ sterujący oparty jest o komputer przemysłowy Raspberry PI Compute Module 4.

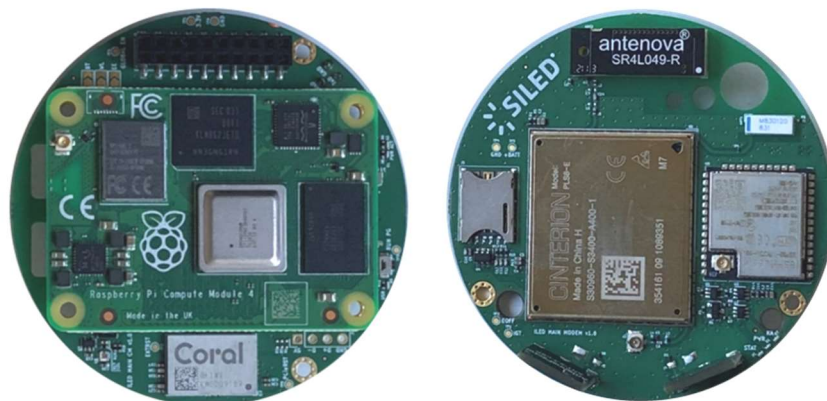
Układ akceleratora maszynowego uczenia to koprocesor Edge TPU firmy Coral, oferujący wydajność 4 TOPS (int8) przy poborze 2W mocy.

Na fot. 1 pokazano prototypy Układu Sterującego oraz Układ Komunikacyjny w wersji LTE. W tej wersji układ komunikacyjny zapewnia transmisję LTE Cat. 3, download: max. 100 Mbps, upload: max. 50 Mbps. Dodatkowo układ wyposażono w modem WiFi, realizujący sieć kratową WiFiMesh (ESP32) w pasmie 2.4 GHz. Dodatkowe kanały komunikacyjne obsługiwane są przez Układ Sterujący: transmisja bezprzewodowa IEEE 802.11 b/g/n/ac w pasmach 2.4 GHz i 5 GHz oraz Bluetooth 5.0, BLE.

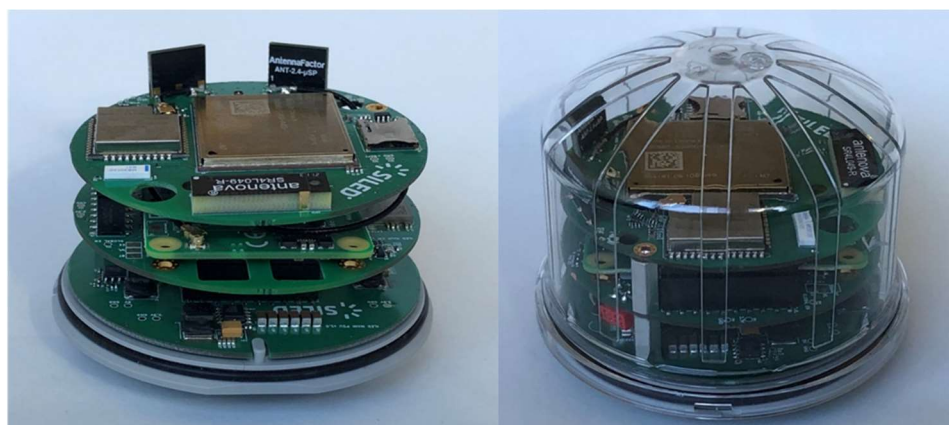
Na fot. 2 pokazano prototyp Modułu Głównego. W trakcie testów eksploatacyjnych okazało się, że dla osiągnięcia pełnej wydajności potrzebna jest wersja Modułu Głównego z układem zarządzania energią ciepłą. Taką wersję zaprezentowano na fot. 3.



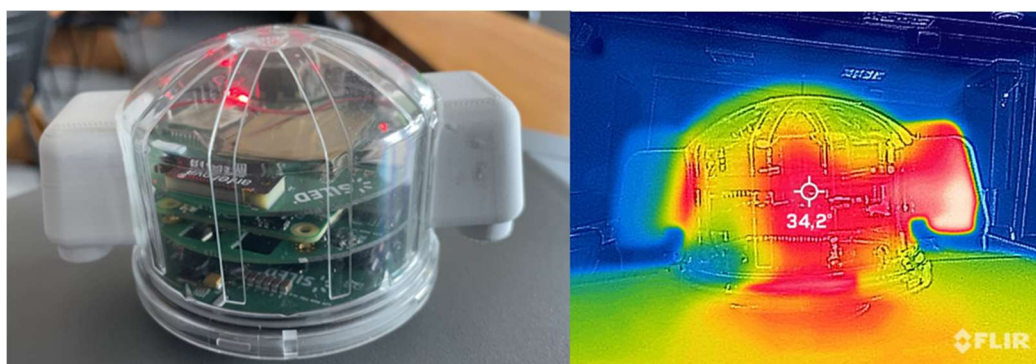
Rys. 50. Konstrukcja Modułu Głównego (materiały własne firmy TSTRONIC Sp. z o.o.)



Fot. 1. Układ Sterujący oraz Układ Komunikacyjny Modułu Głównego



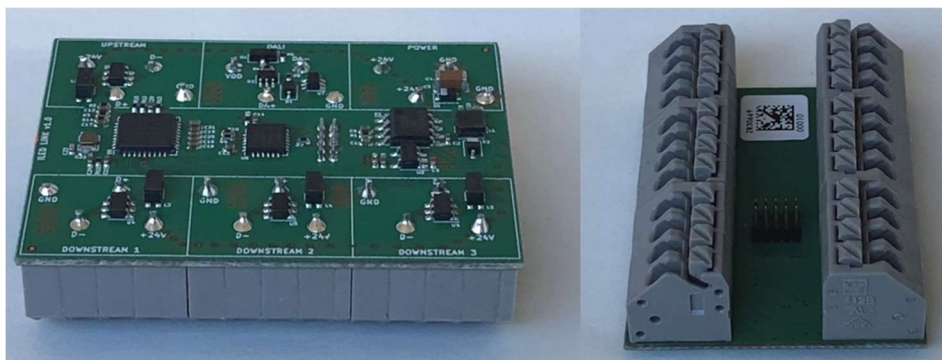
Fot. 2. Kompletny Moduł Główny, bez obudowy i w obudowie



Fot. 3. Moduł Główny z układem zarządzania energią cieplną. Po prawej stronie mapa cieplna modułu wskazująca na efektywność rozwiązania

5.2.2 Moduł Wewnętrzny

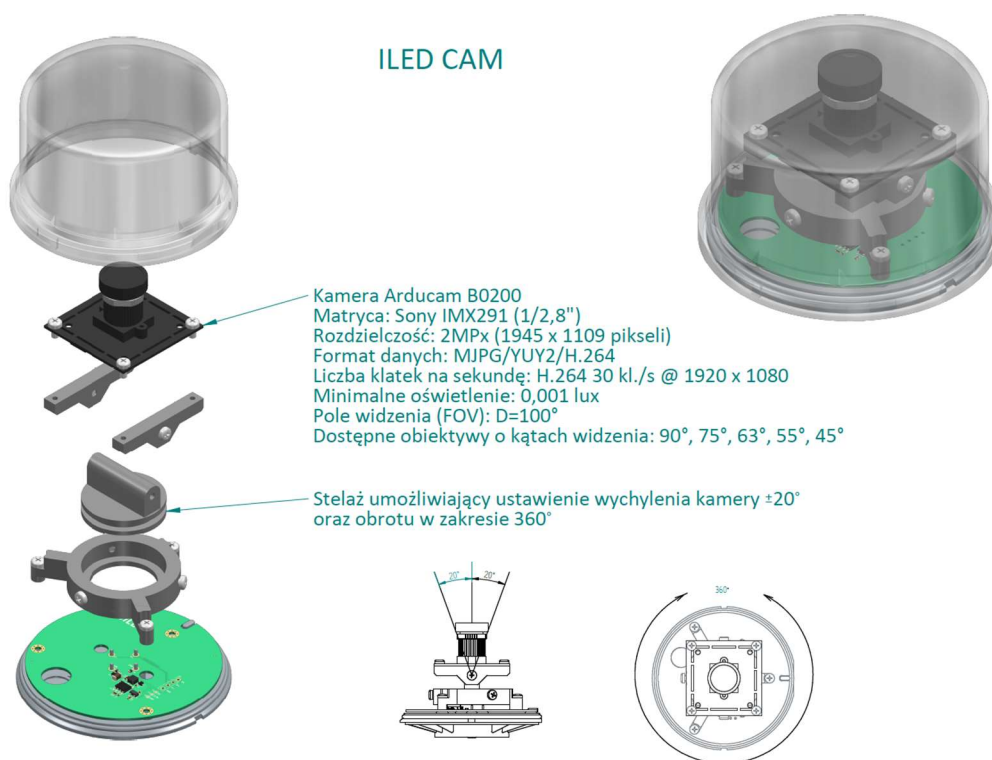
Moduł Wewnętrzny ILED LINK jest instalowany w obudowie lampy. Moduł umożliwia sterowanie zasilaczami LED i obsługuje standardy DALI, DALI-2, D4I. Moduł wyposażono w szereg szybkozłącznych ułatwiających okablowanie wewnętrzne lampy. Na rys. 4 pokazano prototyp Modułu Wewnętrznego.



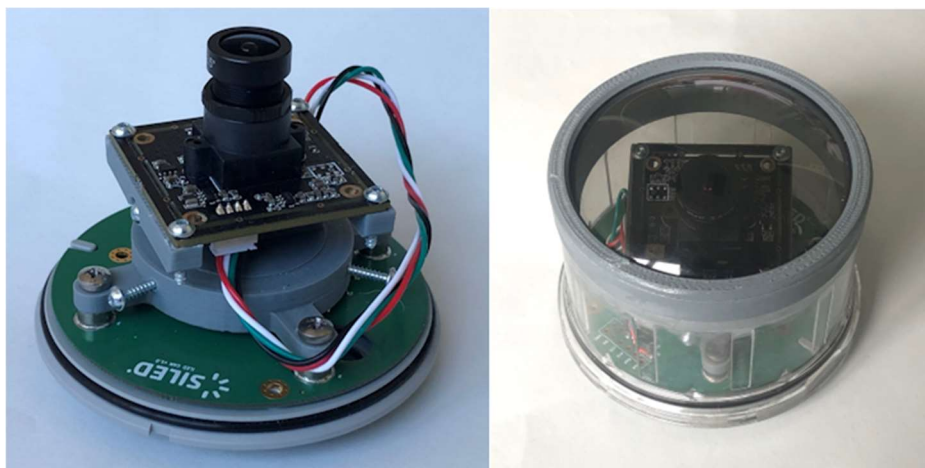
Fot. 4. Prototyp Modułu Wewnętrznego

5.2.3 Moduł Kamery

Moduł Kamery wykorzystuje elementy standardowej obudowy Zhaga SR o średnicy 80 mm i wysokości 58 mm. Ponieważ standardowa obudowa ma niewystarczającą transparentność została wykonana dedykowana obudowa, W module zastosowano kamerę Arducam B0200 wyposażoną w sensor o rozdzielczości 2 MPx, wbudowany filtr IR, wspierającą format danych H.264, MJPG i YUY2. Konstrukcję modułu zaprezentowano na rys. 51, na fot. 5 prototyp Modułu.



Rys. 51. Konstrukcja Modułu Kamery (materiały własne firmy TSTRONIC Sp. z o.o.)



For. 5. Prototyp Modułu Kamery, bez obudowy i w obudowie

5.2.4 Moduł Sondy Akustycznej

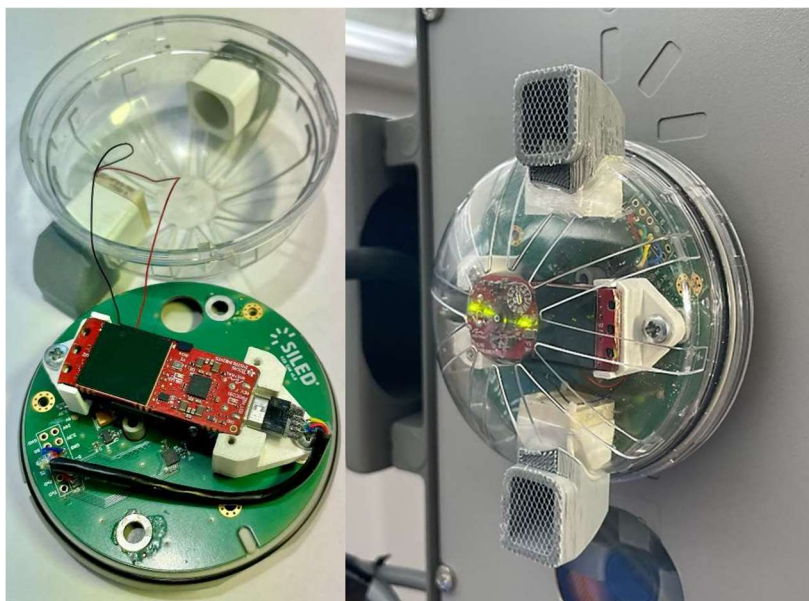
Moduł Sondy Akustycznej wykorzystuje rozwiązanie stworzone Katedrze Systemów Multimedialnych Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej. Jest to akustyczny czujnik wektorowy, czyli trójwymiarowa akustyczna sonda natężeniowa. Sonda to sześciąt złożony z cyfrowych mikrofonów MEMS, połączonych konwerterem I²S-USB. W celu zapewnienia ochrony czujnika przed czynnikami pogodowymi oraz mechanicznymi, na czujnik akustyczny nakładana jest obudowa z tworzywa sztucznego, dodatkowo, na obudowę nakładana jest osłona przeciwwiatrowa, tłumiąca wpływ wiatru na sygnały rejestrowane przez czujnik. Prototyp Modułu przedstawiono na fot. 6.



Fot. 6. Prototyp Modułu Sondy Akustycznej: bez osłony, z osłoną ochronną oraz z osłoną przeciwwiatrową

5.2.5 Moduł Radaru

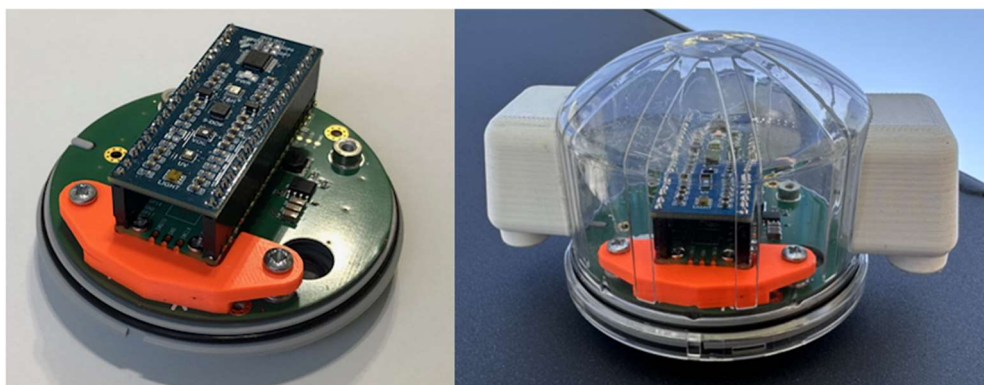
Konstrukcja modułu radaru oparta jest o czujnik IWR6843AOP firmy Texas Instruments. Czujnik pracuje w paśmie 60-64GHz, a więc w pasmie milimetrowym, co daje możliwość miniaturyzacji i integracji anten. Czujniki mmWave pozwalają na szybkie i dokładne wykrywanie odległości, kąta i prędkości obiektów. Prototyp Modułu przedstawiono na fot. 7.



Fot. 7. Prototyp Modułu Radaru: bez obudowy i z obudową

5.2.6 Moduł Sensora Środowiskowego

Moduł realizuje funkcje sensora środowiskowego. Moduł zawiera następujące czujniki: światła, temperatury, wilgotności, ciśnienia atmosferycznego, promieniowania UV, LZO – lotnych związków organicznych, akcelerometr, żyroskop, and magnetometr. Dokonano również adaptacji standardowej obudowy, aby umożliwić poprawnie działanie czujnika. Prototyp Modułu przedstawiono na fot. 8.



Fot. 8. Prototyp Modułu Sensora Środowiskowego: bez obudowy i z obudową

5.2.7 Moduł Casambi

Moduł rozszerza funkcjonalność platformy ILED o możliwość sterowania lampami wykorzystującymi technologię Casambi. Rozwiązanie Casambi opiera się na technologii Bluetooth Low Energy i zapewnia sieć mesh, czyli samoorganizującą się sieć kratową. Prototyp Modułu przedstawiono na fot. 9.



Fot. 9. Prototyp Modułu Casambi, bez obudowy i z obudową

5.3 Oprogramowanie platformy

Na potrzeby platformy sterownia ILED został opracowany system operacyjny wbudowany (embedded), zawierający między innymi sterowniki sensorów oraz interfejsów komunikacyjnych. System bazowy oparty jest o system Debian 11.2 (codename "bullseye") w dystrybucji Raspbian. System jest 64 bitowy co pozwala w pełni wykorzystać zainstalowaną pamięć RAM (do 8GB). Najważniejsze sterowniki użyte w systemie:

- sterownik modułów komunikacyjnych GSM LTE i WiFi użyto systemowych sterowników. Dodatkowo zainstalowano pakiet OpenVPN umożliwiający pracę w prywatnej sieci wirtualnej.
- sterowanie modułem ESP 32 użyto oprogramowania WiFi Mesh opartego o ESP-WIFI-MESH Development Framework
- do obsługi akceleratora Coral EdgeTPU użyto Coral Gasket Driver.
- obsługa wbudowanych sensorów oraz RTC odbywa się przez interfejs I2C z driverem systemowym.
- do sterowania elementami wykonawczymi lampy używa się konwertera USB/DALI wbudowanego w moduł ILED_LINK.
- do obsługi modułu kamery wykorzystywany jest USB video device class (UVC).

Sterowanie elementami wykonawczymi lampy, odczyt danych z Modułów Sensorowych oraz zarządzanie podległymi urządzeniami IoT zrealizowane zostało przez dedykowaną aplikację, w całości napisaną w języku C++. Sterowanie elementami wykonawczymi lampy, np. zasilaczami, odbywa się przez interfejs DALI umieszczony w Module Wewnętrznym. Zarządzanie podległymi urządzeniami IoT odbywa się przez protokół Bluetooth, Zigbee lub WiFi Mesh. Aplikacja komunikuje się z serwerem w warstwie chmurowej poprzez protokół opracowany na potrzeby projektu.

5.4 Integracja z systemem chmurowym

W ramach prac nad projektem INFOLIGHT opracowano protokół komunikacyjny warstwy Fog Computing, nazywany dalej ILFP – InfoLight Fog Protocol. Fog/Edge Computing zostało omówione w rozdz. 2.2. Protokół ten obsługuje podstawowe zdarzenia generowane i obsługiwane przez elementy systemu takie jak:

- odczyty sensorów
- sterowanie aktuatorów
- monitorowanie pracy elementów sieci
- konfigurowanie elementów sieci

Do definiowania relacji pomiędzy elementami sieci systemu INFOLIGHT przyjęto architekturę zdefiniowaną przez konsorcjum OpenFog. Organizacja ta określiła referencyjną architekturę dla Fog Computing: „OpenFog Reference Architecture for Fog Computing” [29]. Protokół obsługuje następujące modele tam zdefiniowane: Model 2 – w którym warstwa sieci Fog jest hierarchiczna i część elementów pośredniczy w dostępie do usług Cloud Computing oraz Model 3 – w którym warstwa sieci Fog jest jednopoziomowa a elementy sieci FOG bezpośrednio kontaktują się z usługami Cloud Computing.

Na potrzeby protokołu zdefiniowano prosty model zasobów, w którym każda informacja udostępniona przez Fog Element jest zasobem (Resource). Zasoby są logicznie zorganizowane w obiekty. Każdy zasób otrzymuje unikalny identyfikator.

- obiekt to zbiór logicznie powiązanych zasobów (informatycznie: Klasa)
- instancja obiektu to jedno wystąpienie obiektu (informatycznie: Obiekt)
- zasób to atomowa jednostka informacji (informatycznie: Atrybut)

Aby ułatwić integrację systemu INFOLIGHT z innymi systemami w ramach projektów Smart City zdecydowano się nie tworzyć od zera własnego modelu, a odwołać się do istniejących zdefiniowanych, otwartych (Open Source) modeli, o największym potencjale ustanowienia standardu. Wybrano model opracowany przez konsorcjum uCIFI - uCIFI Device Data Model, dedykowany model dla potrzeb Smart City [106].

Na potrzeby protokołu obiekty używane przez system INFOLIGHT zostały zdefiniowane w rejestrze OMA [107] i zostały opisane za pomocą standardu JSON. Definicja tych obiektów znajduje się na serwerze GitLab Politechniki Gdańskiej <https://git.pg.edu.pl/infolight/datamodel/>.

Protokół ILFP oparty jest o MQTT v 3.1.1 i o wiadomości w formacie JSON. Każdy element systemu podczas produkcji podlega wstępnej konfiguracji poprzez zdefiniowanie adresu brokera MQTT. Protokół określa sposób identyfikacji elementów systemu, tworzenia instalacji oraz grup. Protokół wyposażono w zaawansowane funkcje:

- „Check-in – rejestracja, każde urządzenie pierwszy raz pojawiające się w sieci musi przejść proces rejestracji „check-in”. Element sieciowy po restarcie sprawdza w

pamięci nieulotnej czy ma ustawioną flagę „Check-in”. Jeżeli nie, to publikuje specjalną wiadomość a element nadrzędny odpowiada na takie zgłoszenie potwierdzeniem lub odrzuceniem zgłoszenia

- „Discovery” - odkrywanie zasobów informacyjnych elementów sieci. Ten mechanizm pozwala elementowi nadrzędnemu odpytać podległy element o jego listę zdefiniowanych obiektów.
- „Bootstrapping” - inicjalna konfiguracji elementów sieci. Przykładowy scenariusz: element sieci traci konfigurację więc zwraca się do elementu nadrzędnego o przesłanie aktualnie obowiązujących.
- „Observe” - funkcja polega na wyznaczenie przez obiekt nadrzędny innym obiektom jakie informacje i kiedy mają przesyłać jako zdarzenia aktualizujące status. Obiekt nadrzędny może z dokładnością do pojedynczego atrybutu w obiekcie określić czy informacja ma być tylko w przypadku zmiany wartości czy też przesyłana okresowo a jeżeli tak to z jaką częstotliwością. Możliwe jest połączenie obu mechanizmów.

Protokół zaimplementowano w prototypach lamp inteligentnych oraz na serwerach chmurowych obsługujących instalację projektu INFOLIGHT.

5.5 Przykładowe realizacje inteligentnych lamp

Prototyp platformy sterowania ILED wykorzystano do skonstruowania szeregu inteligentnych lamp, w ramach projektu INFOLIGHT. Zbadano kilka konfiguracji, z różnymi zestawami modułów. Na fot. 10 do fot. 13 przedstawiono kilka z tych konfiguracji.



Fot. 10. Lampa inteligentna drogowa ILED z Modułem Głównym i Modułem Kamery



Fot. 11. Lampa inteligentna drogowa ILED z Modułem Głównym, Modułem Sensora Środowiskowego i modułami sieci WAPN typu ZigBee



Fot. 12. Lampa inteligentna z Modułem Głównym i czujnikiem PIR



Fot. 13. Lampa inteligentna drogowa ILED z Modułem Głównym, Modułem Radarowym, Modułem Sondy Akustycznej i modułem czujnika temperatury barwowej oświetlenia otoczenia

6 Podsumowanie i wnioski

Systemy oświetlenia drogowego nie były dotychczas przedmiotem badań w zakresie wykrywania anomalii w oparciu o analizę danych z inteligentnych liczników energii. Zbadanie możliwości wykrywania w czasie rzeczywistym anomalii w systemie oświetlenia drogowego w oparciu o analizę danych z inteligentnych liczników energii jest, zatem, oryginalnym wkładem niniejszej pracy.

Przebadano metody oparte o uczenie maszynowe, zarówno stosujące metody statystyczne jak i sztuczne sieci neuronowe. Przeprowadzono szereg symulacji, wykonując obliczenia na rzeczywistych danych. Na potrzeby niniejszej pracy utworzono bazę danych pomiarów inteligentnych liczników energii, pochodzących z systemu sterowania oświetleniem, zainstalowanego w jednym z polskich miast.

Wybierając rozdzielczość czasową dla metod analizy zdecydowano się użyć standardowe okresy dla ustalania profili zużycia energii: 15 minutowy i 60 minutowy. Dla metod statystycznych, ze względu na ograniczenia metody ARIMA przyjęty został 60-minutowy okres pomiarowy. Z tego też względu analiza algorytmów jest podzielona na dwie grupy, dla algorytmów jednogodzinnych przebadano metody statystyczne oraz zastosowanie rekurencyjnych sieci neuronowych, dla algorytmów 15-minutowych przebadano rekurencyjne sieci neuronowe oraz Autoenkoder.

Algorytmy stosujące okres jednogodzinny wykazały się skutecznością w zakresie detekcji anomalii w analizowanych zbiorach danych oraz są w stanie działać w czasie rzeczywistym. Algorytm oparty o model SARIMA charakteryzuje się najlepszą dokładnością odwzorowania szeregu czasowego zużycia energii oraz wymaga mniej czasu do osiągnięcia gotowości wykrywania anomalii w stosunku do algorytmów LSTM i GRU.

W przypadku algorytmów stosujących okres 15-minutowy najlepsze wyniki w zakresie detekcji anomalii osiągnęły algorytmy oparte o Autoenkoder z siecią LSTM i GRU. Algorytm działa poprawnie przez cały okres rejestracji danych, niezależnie od pory roku. Ponadto okres trenowania modelu może być krótki, co oznacza, że algorytm szybko osiąga gotowość do detekcji anomalii. Model Autoenkodera wymaga wprawdzie więcej czasu na trening, ale symulacje na komputerze przemysłowym odpowiadającym warunkom Edge Computing wykazały, że, czas trenowania nie przekracza okresu 15 minut, a więc algorytm może działać w czasie rzeczywistym.

Badania obu grup algorytmów pozwoliły na udowodnienie tezy nr 1: „Algorytmy oparte na uczeniu maszynowym nadają się do wykrywania anomalii w poborze mocy przez systemy oświetlenia dróg, na podstawie analizy zbiorczych danych pochodzących z inteligentnych liczników energii”.

Algorytmy maszynowego uczenia z sieciami rekurencyjnymi przewyższają algorytmy z metodami statystycznymi pod względem elastyczności konfiguracji, czasów obliczeń oraz odporności na zjawisko dryfu koncepcyjnego. Algorytmy te otrzymują dynamiczne wsparcie sprzętowe w postaci akceleratorów uczenia maszynowego, wpisując się w koncepcję Edge Computing. Przykładem rzeczywistego rozwiązania, na której opracowane algorytmy mogą działać, jest przedstawiona platforma sterowania inteligentną lampą. W ten sposób wykazano zasadność tezy nr 2: „Opracowane algorytmy oparte na uczeniu maszynowym wykrywają anomalie w poborze mocy przez systemy oświetleniowe w sposób bardziej efektywny od metod dotychczas stosowanych w tym celu”.

Opracowane algorytmy zostaną wdrożone w produktach firmy TSTRONIC, rozszerzając funkcjonalność systemu sterowania oświetleniem oferowanym przez tę firmę. Planowane są również prace nad rozwojem tych algorytmów dla potrzeb wykrywania innych typów anomalii występujących w systemach Smart City. Prace badawcze i badawczo-rozwojowe zrealizowane przez autora rozprawy w ramach programu Doktorat Wdrożeniowy mają szansę przyczynić się do uzyskania nowych funkcjonalności w systemach oświetlania inteligentnych miast i dróg.

Bibliografia

- [1] UN DESA, United Nations, Department of Economic and Social Affairs, Population Division. World Urbanization Prospects: The 2018 Revision, 2015.
<http://esa.un.org/unpd/wup/Publications/Files/WUP2014-Report.pdf>, dostęp 2023-05-13.
- [2] Batty, Michael, et al. "Smart cities of the future." The European Physical Journal Special Topics 214 (2012): 481-518.
- [3] Hall, R. & Bowerman, B. & Braverman, Joseph & Taylor, J. & Todosow, Helen & Wimmersperg, U.. (2000). The vision of a smart city. 2nd Int. Life.
- [4] Hollands, Robert G. "Will the real smart city please stand up? Intelligent, progressive or entrepreneurial?." City 12.3 (2008): 303-320.
- [5] Caragliu, A., Del Bo, C.F., & Nijkamp, P. (2011). Smart Cities in Europe. Journal of Urban Technology, 18, 65 - 82.
- [6] Lai, Chun Sing, et al. "A review of technical standards for smart cities." Clean Technologies 2.3 (2020): 290-310.
- [7] ISO 37120:2018 Sustainable cities and communities - Indicators for city services and quality of life: <https://www.iso.org/standard/68498.html>, dostęp 2023-05-13.
- [8] ISO 37122:2019 Sustainable cities and communities - Indicators for smart cities <https://www.iso.org/standard/69050.html>, dostęp 2023-05-13.
- [9] ITU-T SG20: Internet of things (IoT) and smart cities and communities (SC&C): <https://www.itu.int/en/ITU-T/studygroups/2022-2024/20/Pages/default.aspx>, dostęp 2023-05-13.
- [10] IEC 63152:2020 Smart cities - City service continuity against disasters - The role of the electrical supply
- [11] Albino, Vito, Umberto Berardi, and Rosa Maria Dangelico. "Smart cities: Definitions, dimensions, performance, and initiatives." Journal of urban technology 22.1 (2015): 3-21.
- [12] Giffinger R, Fertner C, Kramar H, Kalasek R, Pichler-Milanovi N, Meijers E (2007) Smart cities. Ranking of European medium-sized cities. Wien, Hrsg. Centre of Regional Science, TU Wien. www.smart-cities.eu/download/smart_cities_final_report.pdf, dostęp 2023-05-13
- [13] Syed, Abbas Shah, et al. "IoT in smart cities: A survey of technologies, practices and challenges." Smart Cities 4.2 (2021): 429-475.
- [14] Belli, Laura, et al. "IoT-enabled smart sustainable cities: Challenges and approaches." Smart Cities 3.3 (2020): 1039-1071.

- [15] Rojek, Izabela, and Jan Studzinski. "Detection and localization of water leaks in water nets supported by an ICT system with artificial intelligence methods as a way forward for smart cities." *Sustainability* 11.2 (2019): 518.
- [16] Dutta, Joy, et al. "Towards smart city: sensing air quality in city based on opportunistic crowd-sensing." *Proceedings of the 18th international conference on distributed computing and networking*. 2017.
- [17] Wang, Yun, et al. "A big data approach for smart transportation management on bus network." *2016 IEEE international smart cities conference (ISC2)*. IEEE, 2016.
- [18] Shirazi, Elham, and Shahram Jadid. "Autonomous self-healing in smart distribution grids using agent systems." *IEEE Transactions on Industrial Informatics* 15.12 (2018): 6291-6301.
- [19] Andreão, Rodrigo Varejão, et al. "Raspicare: a telemedicine platform for the treatment and monitoring of patients with chronic diseases." *Assistive Technologies in Smart Cities*. London, UK: IntechOpen, 2018.
- [20] Trencher, Gregory, and Andrew Karvonen. "Stretching “smart”: Advancing health and well-being through the smart city agenda." *Smart and Sustainable Cities?*. Routledge, 2020. 54-71.
- [21] Farag, Siham G. "Application of smart structural system for smart sustainable cities." *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*. IEEE, 2019.
- [22] Silik, A., Noori, M., Altabey, W.A., Ghiasi, R., Wu, Z. (2021). Analytic Wavelet Selection for Time–Frequency Analysis of Big Data Form Civil Structure Monitoring. In: Rainieri, C., Fabbrocino, G., Caterino, N., Ceroni, F., Notarangelo, M.A. (eds) *Civil Structural Health Monitoring. CSHM 2021. Lecture Notes in Civil Engineering*, vol 156. Springer, Cham. https://doi.org/10.1007/978-3-030-74258-4_29
- [23] Bolchini, Cristiana, Angela Geronazzo, and Elisa Quintarelli. "Smart buildings: A monitoring and data analysis methodological framework." *Building and environment* 121 (2017): 93-105.
- [24] Sikder, Amit Kumar, et al. "IoT-enabled smart lighting systems for smart cities." *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018.
- [25] Bawany, Narmeen Zakaria, and Jawwad A. Shamsi. "Smart city architecture: Vision and challenges." *International Journal of Advanced Computer Science and Applications* 6.11 (2015).

- [26] Silva, Bhagya Nathali, Murad Khan, and Kijun Han. "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities." *Sustainable cities and society* 38 (2018): 697-713.
- [27] Jiang, Dingfu. "The construction of smart city information system based on the Internet of Things and cloud computing." *Computer Communications* 150 (2020): 158-166.
- [28] Bonomi, Flavio, et al. "Fog computing and its role in the internet of things." *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012.
- [29] OpenFog Consortium Architecture Working Group. "OpenFog Reference Architecture for Fog Computing. 2017".
https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf,
dostęp 2023-05-13.
- [30] Yi, Shanhe, et al. "Fog computing: Platform and applications." 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb). IEEE, 2015.
- [31] Cheng, Bin, et al. "Fog function: Serverless fog computing for data intensive iot services." 2019 IEEE International Conference on Services Computing (SCC). IEEE, 2019.
- [32] Chiang, Mung, and Tao Zhang. "Fog and IoT: An overview of research opportunities." *IEEE Internet of things journal* 3.6 (2016): 854-864.
- [33] Atlam, Hany F., Robert J. Walters, and Gary B. Wills. "Fog computing and the internet of things: A review." *big data and cognitive computing* 2.2 (2018): 10.
- [34] Eckhoff, David, and Isabel Wagner. "Privacy in the smart city—applications, technologies, challenges, and solutions." *IEEE Communications Surveys & Tutorials* 20.1 (2017): 489-516.
- [35] Chen, Jiasi, and Xukan Ran. "Deep learning with edge computing: A review." *Proceedings of the IEEE* 107.8 (2019): 1655-1674.
- [36] Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys and Tutorials*, 19(3), 1657-1681. [7931566]. <https://doi.org/10.1109/COMST.2017.2705720>
- [37] Filali, Abderrahime, et al. "Multi-access edge computing: A survey." *IEEE Access* 8 (2020): 197017-197046.
- [38] Volosencu, C., Curiac, D. I., Baniias, O., Ferent, C., Pescaru, D., & Doboili, A. (2008, May). Hierarchical approach for intelligent lighting control in future urban environments. In *2008 IEEE International Conference on Automation, Quality and Testing, Robotics* (Vol. 1, pp. 158-163). IEEE.

- [39] Talari, S., Shafie-Khah, M., Siano, P., Loia, V., Tommasetti, A., & Catalão, J. P. (2017). A review of smart cities based on the internet of things concept. *Energies*, *10*(4), 421.
- [40] Gaston, K. J., & Sánchez de Miguel, A. (2022). Environmental impacts of artificial light at night. *Annual Review of Environment and Resources*, *47*, 373-398.
- [41] Zielinska-Dabkowska, K. M., Schernhammer, E. S., Hanifin, J. P., & Brainard, G. C. (2023). Reducing nighttime light exposure in the urban environment to benefit human health and society. *Science*, *380*(6650), 1130-1135.
- [42] Galatanu, C. D. (2020). On/off optimization of public lighting systems depending on the road class. *Procedia Manufacturing*, *46*, 378-383.
- [43] Commission Internationale de l'Éclairage. (2010). Lighting of Roads for Motor and Pedestrian Traffic: CIE 115: 2010. CIE.
- [44] European Committee for Standardization. Standard EN-13201: Road Lighting; European Committee for Standardization: Bruxelles, Belgium, 2015.
- [45] Górczewska, M.; Redukcja poziomu oświetlenia drogowego - możliwości i ograniczenia. *Wiadomości Elektrotechniczne* 2014 nr 10, strony 6-9.
- [46] Fryc, I., Czyżewski, D., Fan, J., & Gălăţanu, C. D. (2021). The Drive towards Optimization of Road Lighting Energy Consumption Based on Mesopic Vision—A Suburban Street Case Study. *Energies*, *14*(4), 1175.
- [47] Sweater-Hickcox, K., Narendran, N., Bullough, J. D., & Freyssinier, J. P. (2013). Effect of different coloured luminous surrounds on LED discomfort glare perception. *Lighting Research & Technology*, *45*(4), 464-475.
- [48] Davidovic, M., Djokic, L., Cabarkapa, A., Djuretic, A., Skerovic, V., & Kostic, M. (2019). Drivers' preference for the color of LED street lighting. *IEEE access*, *7*, 72850-72861.
- [49] Tripathy, A. K., Mishra, A. K., & Das, T. K. (2017, July). Smart lighting: Intelligent and weather adaptive lighting in street lights using IOT. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)* (pp. 1236-1239). IEEE.
- [50] Marino, F., Leccese, F., & Pizzuti, S. (2017). Adaptive street lighting predictive control. *Energy Procedia*, *111*, 790-799.
- [51] Gagliardi, G., Lupia, M., Cario, G., Tedesco, F., Cicchello Gaccio, F., Lo Scudo, F., & Casavola, A. (2020). Advanced adaptive street lighting systems for smart cities. *Smart Cities*, *3*(4), 1495-1512.

- [52] Mohandas, P., Dhanaraj, J. S. A., & Gao, X. Z. (2019). Artificial neural network based smart and energy efficient street lighting system: A case study for residential area in Hosur. *Sustainable Cities and Society*, 48, 101499.
- [53] Lee, H. C., & Huang, H. B. (2014). A low-cost and noninvasive system for the measurement and detection of faulty streetlights. *IEEE Transactions on Instrumentation and Measurement*, 64(4), 1019-1031.
- [54] Kumar, S., Deshpande, A., Ho, S. S., Ku, J. S., & Sarma, S. E. (2016). Urban street lighting infrastructure monitoring using a mobile sensor platform. *IEEE Sensors Journal*, 16(12), 4981-4994.
- [55] [DALI. Standard. International Electrotechnical Commission IEC 62386. 2020. <https://www.digitalilluminationinterface.org/dali/standards.html>, dostęp 2023-06-18.
- [56] KNX. <https://www.knx.org/>. dostęp 2023-06-18.
- [57] "IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications," in IEEE Std 1901-2020 (Revision of IEEE Std 1901-2010), vol., no., pp.1-1622, 19 Jan. 2021, doi: 10.1109/IEEESTD.2021.9329263.
- [58] Mendalka, M., Gadaj, M., Kulas, L., & Nyka, K. (2010, June). WSN for intelligent street lighting system. In *2010 2nd International Conference on Information Technology,(2010 ICIT)* (pp. 99-100). IEEE.
- [59] ZHAGA Consortium. <https://www.zhagastandard.org/>. dostęp 2023-06-18.
- [60] Karthikeyan, P., Karthik, M., Deepikapriya, V., Briya, S. D., Dharanishwarma, R., & Janakirthick, S. (2022, January). Design and Implementation of Smart Street Light Automation and Fault Detection System. In *2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)* (pp. 1-7). IEEE.
- [61] Saha, D., Sorif, S. M., & Dutta, P. (2021, August). Weather adaptive intelligent street lighting system with automatic fault management using boltuino platform. In *2021 International Conference on ICT for Smart Society (ICISS)* (pp. 1-6). IEEE.
- [62] Adriansyah, A., Dani, A. W., & Nugraha, G. I. (2017, August). Automation control and monitoring of public street lighting system based on internet of things. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)* (pp. 231-236). IEEE.
- [63] Ali, M., Scandurra, P., Moretti, F., & Blaso, L. (2023, March). Architecting a big data-driven software architecture for smart street lighting. In *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)* (pp. 1-10). IEEE.
- [64] Berg Insight. "Smart electricity meter penetration rate in Europe reached 56 percent at the end of 2022" <https://www.berginsight.com/>. dostęp 2023-06-18.

- [65] Zoha, A., Gluhak, A., Imran, M. A., & Rajasegarar, S. (2012). Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12), 16838-16866.
- [66] Wang, Y., Chen, Q., Hong, T., & Kang, C. (2018). Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10(3), 3125-3148.
- [67] McLoughlin, F., Duffy, A., & Conlon, M. (2015). A clustering approach to domestic electricity load profile characterisation using smart metering data. *Applied energy*, 141, 190-199.
- [68] Quilumba, F. L., Lee, W. J., Huang, H., Wang, D. Y., & Szabados, R. L. (2014). Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Transactions on Smart Grid*, 6(2), 911-918.
- [69] Fekri, M. N., Patel, H., Grolinger, K., & Sharma, V. (2021). Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Applied Energy*, 282, 116177.
- [70] Rivas, A. E. L., & Abrao, T. (2020). Faults in smart grid systems: Monitoring, detection and classification. *Electric Power Systems Research*, 189, 106602.
- [71] Syed, D., Abu-Rub, H., Refaat, S. S., & Xie, L. (2020, December). Detection of energy theft in smart grids using electricity consumption patterns. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 4059-4064). IEEE.
- [72] Wang, X., & Ahn, S. H. (2020). Real-time prediction and anomaly detection of electrical load in a residential community. *Applied Energy*, 259, 114145.
- [73] Himeur, Y., Ghanem, K., Alsalemi, A., Bensaali, F., & Amira, A. (2021). Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy*, 287, 116601.
- [74] Kaymakci, C., Wenninger, S., & Sauer, A. (2021). Energy Anomaly Detection in Industrial Applications with Long Short-term Memory-based Autoencoders. *Procedia CIRP*, 104, 182-187.
- [75] Montañez, C. A. C., & Hurst, W. (2020). A machine learning approach for detecting unemployment using the smart metering infrastructure. *IEEE Access*, 8, 22525-22536.
- [76] Solar Calculation Details: <https://gml.noaa.gov/grad/solcalc/calcdetails.html> (dostęp 01-06-2022)
- [77] van Flinders, T. C., Pulkkinen, K. F.: Low-precision formulae for planetary positions. *Astrophysical Journal Supplement Series*, vol. 41, Nov. 1979, p. 391-411

- [78] Gaston, K. J., Davies, T. W., Bennie, J., & Hopkins, J. (2012). Reducing the ecological consequences of night-time light pollution: options and developments. *Journal of Applied Ecology*, 49(6), 1256-1266.
- [79] Foorhuis, R. (2021). On the nature and types of anomalies: a review of deviations in data. *International journal of data science and analytics*, 12(4), 297-331.
- [80] Aggarwal, C. C., & Aggarwal, C. C. (2017). *An introduction to outlier analysis* (pp. 1-34). Springer International Publishing.
- [81] Wang, H., Bah, M. J., & Hammad, M. (2019). Progress in outlier detection techniques: A survey. *Ieee Access*, 7, 107964-108000.
- [82] Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2021). A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3), 1-33.
- [83] Aligholian, A., Farajollahi, M., & Mohsenian-Rad, H. (2019, August). Unsupervised learning for online abnormality detection in smart meter data. In *2019 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1-5). IEEE.
- [84] Goodfellow I., Bengio Y., Courville A.: "Deep Learning Systemy uczące się", 2018, Wydawnictwo Naukowe PWN
- [85] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [86] Murray, D., Stankovic, L., & Stankovic, V. (2017). An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. *Scientific data*, 4(1), 1-12.
- [87] Cook, A. A., Mısırlı, G., & Fan, Z. (2019). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal*, 7(7), 6481-6494.
- [88] Pereira, L., Costa, D., & Ribeiro, M. (2022). A residential labeled dataset for smart meter data analytics. *Scientific Data*, 9(1), 134.
- [89] Jung, S. M., Park, S., Jung, S. W., & Hwang, E. (2020). Monthly electric load forecasting using transfer learning for smart cities. *Sustainability*, 12(16), 6364.
- [90] Rhodes, J. D., Cole, W. J., Upshaw, C. R., Edgar, T. F., & Webber, M. E. (2014). Clustering analysis of residential electricity demand profiles. *Applied Energy*, 135, 461-471.
- [91] Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74, 902-924.

- [92] Khan, W., Liao, J. Y., Walker, S., & Zeiler, W. (2022). Impact assessment of varied data granularities from commercial buildings on exploration and learning mechanism. *Applied Energy*, 319, 119281.
- [93] Śmiałkowski, T., & Czyżewski, A. (2022). Detection of Anomalies in the Operation of a Road Lighting System Based on Data from Smart Electricity Meters. *Energies*, 15(24), 9438.
- [94] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- [95] Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of statistical software*, 27, 1-22.
- [96] Oliveira, G. H., Cavalcante, R. C., Cabral, G. G., Minku, L. L., & Oliveira, A. L. (2017, November). Time series forecasting in the presence of concept drift: A pso-based approach. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 239-246). IEEE.
- [97] Penya, Y. K., Borges, C. E., Agote, D., & Fernández, I. (2011, June). Short-term load forecasting in air-conditioned non-residential Buildings. In *2011 IEEE International Symposium on Industrial Electronics* (pp. 1359-1364). IEEE.
- [98] Hyndman, R. J., & Kostenko, A. V. (2007). Minimum sample size requirements for seasonal forecasting models. *foresight*, 6(Spring), 12-15.
- [99] Priya, S., & Uthra, R. A. (2021). Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex & Intelligent Systems*, 1-17.
- [100] Fan, C., Xiao, F., Zhao, Y., & Wang, J. (2018). Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied energy*, 211, 1123-1135.
- [101] Fazlipour, Z., Mashhour, E., & Joorabian, M. (2022). A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism. *Applied Energy*, 327, 120063.
- [102] Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018, July). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 387-395).
- [103] Toh, C. K., Sanguesa, J. A., Cano, J. C., & Martinez, F. J. (2020). Advances in smart roads for future smart cities. *Proceedings of the Royal Society A*, 476(2233), 20190439.

- [104] Kotus, J., Szwoch, G., Czyzewski, A., & Kostek, B. (2022). Assessment of road surface state with acoustic vector sensor. *The Journal of the Acoustical Society of America*, 152(4), A193-A193.
- [105] Marciniuk, K., & Kostek, B. (2022). Machine learning applied to acoustic-based road traffic monitoring. *Procedia Computer Science*, 207, 1087-1095.
- [106] The uCIFI® Alliance: <https://ucifi.org/>. dostęp 2023-08-10.
- [107] OMA SpecWorks: <https://omaspecworks.org/>. dostęp 2023-08-10.

Spis rysunków

Rys. 1. Warstwowa architektura systemu Smart City	11
Rys. 2. Przykładowe zapisy inteligentnego licznika	21
Rys. 3. Wykres mocy czynnej jednej fazy dla lamp bez redukcji	22
Rys. 4. Wykres mocy czynnej jednej fazy dla lamp z redukcją	22
Rys. 5. Współczynnik wypełnienia impulsu dla poboru mocy dla miasta Gdańska	23
Rys. 6. Dekompozycja szeregu czasowego mocy czynnej – okres jesienny.....	23
Rys. 7. Dekompozycja szeregu czasowego mocy czynnej – okres wiosenny.....	23
Rys. 8. Sieć RNN w postaci symbolicznej notacji i rozwinięcia.	25
Rys. 9. Standardowa sieć RNN z powielającym się modułem zawierającym jedną warstwę ukrytą.....	26
Rys. 10. Schemat bloku konstrukcyjnego sieci LSTM	27
Rys. 11. Schemat bloku konstrukcyjnego sieci GRU.....	28
Rys. 12. Schemat Autoenkodera.....	28
Rys. 13. Ilustracja sposobu prowadzenia pomiarów mocy w szafie sterującej oświetleniem ulicznym	30
Rys. 14. Przykładowe wizualizacje danych z pomiarów mocy dla ZP ₁ , ZP ₂ , .. ZP ₉	32
Rys. 15. Przykładowe dane z pomiarami mocy dla ZP _A , ZP _B , .. ZP _F	33
Rys. 16. Dekompozycja szeregu czasowego mocy czynnej, downsampling 60 minut.....	34
Rys. 17. Idea algorytmu detekcji anomalii oparta o SARIMA	37
Rys. 18. Wynik symulacji przy zastosowaniu SARIMA dla zestawu pomiarów ZP ₁ – pomiary bez awarii	39
Rys. 19. Wyniki symulacji przy zastosowaniu SARIMA dla poszczególnych typów anomalii.....	40
Rys. 20. Zastosowanie różnych progów do detekcji anomalii dla algorytmu SARIMA	41
Rys. 21. Sieć LSTM zastosowana do algorytmu detekcji anomalii	42
Rys. 22. Funkcja straty dla zbiorów trenującego i walidacyjnego dla zbiorów ZP ₁ , ZP ₂ , .. ZP ₉	44
Rys. 23. Wynik symulacji przy użyciu algorytmu LSTM dla pomiarów bez anomalii ZP ₁	46
Rys. 24. Wyniki symulacji w oparciu o algorytm LSTM dla poszczególnych typów anomalii.....	47
Rys. 25. Zastosowanie różnych progów do detekcji anomalii dla algorytmu LSTM	48

Rys. 26. Sieć GRU zastosowana do algorytmu detekcji anomalii	49
Rys. 27. Funkcja straty dla zbiorów trenującego i walidacyjnego dla modelu GRU.....	49
Rys. 28. Wyniki symulacji w oparciu o algorytm GRU dla poszczególnych typów anomalii.....	51
Rys. 29. Zastosowanie różnych progów do detekcji anomalii dla algorytmu GRU.	52
Rys. 30. Porównanie wartości $MaAE_{norm}$ dla zestawów danych bez anomalii	53
Rys. 31. Porównanie wartości $MaAE_{norm}$ dla zestawów danych bez anomalii	53
Rys. 32. Autoenkoder jednowarstwowy z siecią LSTM	55
Rys. 33. Autoenkoder dwuwarstwowy z siecią LSTM	55
Rys. 34. Autoenkoder z siecią splotową 1D Convolution.....	55
Rys. 35. Przykładowy przebieg wartości MAE dla Autoenkodera	57
Rys. 36. Przykładowy przebieg wartości AE dla sieci LSTM.....	58
Rys. 37. Porównanie wartości średniej wskaźnika MR dla wybranych modeli.....	59
Rys. 38. Porównanie wartości średniej wskaźnika MR dla wybranych modeli.....	60
Rys. 39. Porównanie wartości wskaźnika True Negative Rate dla wybranych modeli	61
Rys. 40. Zasada podział szeregu czasowego pomiarów zawierających anomalię	62
Rys. 41. Porównanie wartości średniej wskaźnika MR dla okresu T1 dla wybranych modeli	63
Rys. 42. Porównanie wartości średniej wskaźnika AR dla wybranych modeli.....	63
Rys. 43. Przebieg wartości MAE podczas symulacji dla modelu AEL2U32 i okresu treningowego równego trzem dniom.....	64
Rys. 44. Przebieg wartości MAE podczas symulacji dla modelu AEC9U64 i okresu treningowego równego trzem dniom.....	64
Rys. 45. Przebieg wartości AE podczas symulacji dla modelu LST2U32 i okresu treningowego równego czterem dniom	65
Rys. 46. Porównanie wartości wskaźnika F1-score dla wybranych modeli.....	66
Rys. 47. Średni czas treningu sieci dla wybranych modeli	67
Rys. 48. Schemat blokowy platformy sterowania inteligentnej lampy	72
Rys. 49. Obudowa modułu Zhaga SR firmy TE Connectivity.....	74
Rys. 50. Konstrukcja Modułu Głównego	75
Rys. 51. Konstrukcja Modułu Kamery	77



Spis fotografii

Fot. 1. Układ Sterujący oraz Układ Komunikacyjny Modułu Głównego	76
Fot. 2. Kompletny Moduł Główny, bez obudowy i w obudowie	76
Fot. 3. Moduł Główny z układem zarządzania energią cieplną	76
Fot. 4. Prototyp Modułu Wewnętrznego	77
For. 5. Prototyp Modułu Kamery, bez obudowy i w obudowie	78
Fot. 6. Prototyp Modułu Sondy Akustycznej: bez osłony, z osłoną ochronną oraz z osłoną przeciwwiatrową	78
Fot. 7. Prototyp Modułu Radaru: bez obudowy i z obudową	79
Fot. 8. Prototyp Modułu Sensora Środowiskowego: bez obudowy i z obudową.....	79
Fot. 9. Prototyp Modułu Casambi, bez obudowy i z obudową	80
Fot. 10. Lampa inteligentna drogowa ILED z Modułem Głównym i Modułem Kamery.....	82
Fot. 11. Lampa inteligentna drogowa ILED z Modułem Głównym, Modułem Sensora Środowiskowego i modułami sieci WAPN typu ZigBee	83
Fot. 12. Lampa inteligentna z Modułem Głównym i czujnikiem PIR	83
Fot. 13. Lampa inteligentna drogowa ILED z Modułem Głównym, Modułem Radarowym, Modułem Sondy Akustycznej i modułem czujnika temperatury barwowej oświetlenia otoczenia.....	84

Spis tabel

Tabela 1. Wyniki symulacji algorytmem SARIMA dla zestawów ZP₁ .. ZP₉	39
Tabela 2. Wyniki symulacji algorytmem SARIMA dla zestawów ZP_A, ... ZP_F	41
Tabela 3. Wyniki symulacji wyszukiwanie kratowego parametrów sieci LSTM	44
Tabela 4. Wyniki symulacji algorytmem LSTM dla zestawów ZP₁ .. ZP₉	46
Tabela 5. Wyniki symulacji algorytmem LSTM dla zestawów ZP_A, ... ZP_F	46
Tabela 6. Wyniki symulacji wyszukiwanie kratowego parametrów sieci GRU	49
Tabela 7. Wyniki symulacji wyliczania AE dla sieci GRU, zestawy pomiarów bez anomalii	50
Tabela 8. Wyniki symulacji wyliczania AE dla sieci GRU, zestawy pomiarów z anomaliami	50
Tabela 9. Porównanie czasów trwania etapów algorytmów.....	54
Tabela 10. Porównanie czasów trwania etapów algorytmów Autoenkodera	67
Tabela A.1. Wartości średnia wskaźnika MR dla Autoenkodera w zależności od długości okresu treningowego	101
Tabela A.2. Wartości średnia wskaźnika MR dla sieci LSTM i GRU w zależności od długości okresu treningowego	101
Tabela A.3. Odchylenie standardowe wskaźnika MR dla Autoenkodera w zależności od długości okresu treningowego.....	102
Tabela A.4. Odchylenie standardowe wskaźnika MR dla LSTM i GRU w zależności od długości okresu treningowego.....	103
Tabela A.5. Wartości wskaźnika True Negative Rate (TNR) dla różnych modeli i okresów treningowych dla Autoenkodera.....	104
Tabela A.6. Wartości wskaźnika True Negative Rate (TNR) dla różnych modeli i okresów treningowych dla sieci LSTM i GRU	105
Tabela A.7. Wartości średnia wskaźnika MR dla okresu T1 dla Autoenkodera.....	106
Tabela A.8. Wartości średnia wskaźnika MR dla okresu T1 dla sieci LSTM i GRU.....	106
Tabela A.9. Wartości średnia wskaźnika AR dla Autoenkodera	107
Tabela A.10. Wartości średnia wskaźnika AR dla sieci LSTM i GRU	108
Tabela A.11. Wartości wskaźnika F1-score dla Autoenkodera dla różnych okresów treningowych.....	108
Tabela A.12. Wartości wskaźnika F1-score dla sieci LSTM i GRU dla różnych okresów treningowych.....	109



Tabela A.13. Średni czas uczenia sieci w zależności od długości okresu treningowego dla Autoenkodera	110
Tabela A.14. Średni czas uczenia sieci w zależności od długości okresu treningowego dla sieci LSTM i GRU.....	111

Załączniki

Załącznik A: Szczegółowe wyniki eksperymentów

Tabela A.1. Wartości średnia wskaźnika MR dla Autoenkodera w zależności od długości okresu treningowego. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

Model	MR mean				
	3	4	5	6	7
AEL1U32	0,03972	0,03988	0,06690	0,15267	0,04380
AEL1U64	0,07119	0,08117	0,10567	0,09804	0,09911
AEL1U96	0,08823	0,06369	0,10800	0,10358	0,15012
AEL1U128	0,03922	0,09449	0,12566	0,16576	0,15726
AEL2U32	0,02053	0,02553	0,04144	0,03200	0,03422
AEL2U64	0,03174	0,02733	0,05909	0,05837	0,07681
AEL2U96	0,03566	0,05354	0,06531	0,11873	0,05653
AEL2U128	0,02751	0,07802	0,07531	0,05615	0,08491
AEG1U32	0,06604	0,02511	0,05684	0,09496	0,03529
AEG1U64	0,04103	0,11495	0,07997	0,09332	0,13352
AEG1U96	0,05018	0,06867	0,12352	0,26290	0,15204
AEG1U128	0,08810	0,11391	0,17177	0,23516	<i>0,28650</i>
AEG2U32	0,02420	0,03331	0,03141	0,03068	0,05606
AEG2U64	0,03136	0,03743	0,03963	0,06538	0,09234
AEG2U96	0,03759	0,06773	0,04747	0,07369	0,06444
AEG2U128	0,05660	0,04309	0,03623	0,10889	0,13438
AEC7U32	<i>0,39140</i>	0,22284	0,27419	0,15188	0,18893
AEC7U64	<i>0,30362</i>	0,16735	0,19461	<i>0,28809</i>	0,15175
AEC9U64	<i>0,43728</i>	<i>0,39176</i>	0,15731	0,16575	<i>0,27560</i>

Tabela A.2. Wartości średnia wskaźnika MR dla sieci LSTM i GRU w zależności od długości okresu treningowego. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

Model	MR mean				
	3	4	5	6	7
LST1U16	0,48965	0,25068	0,43803	0,29908	0,38373
LST2U16	0,49094	0,32658	0,45108	0,32358	0,19015
LST3U16	<i>0,57576</i>	0,27486	0,44164	0,22017	0,23561
LST4U16	0,48872	0,18265	0,35482	0,32419	0,26679
LST1U32	<i>0,68413</i>	0,35629	0,42631	0,38450	0,28177
LST2U32	0,55786	0,48913	0,27319	0,32616	0,18758
LST3U32	0,42473	0,29295	0,45153	0,31814	0,22275

LST4U32	0,44101	0,32394	0,43125	0,15984	0,29952
LST1U64	<i>0,57964</i>	0,50902	0,43103	0,44499	0,21942
LST2U64	0,56145	0,37017	0,38252	0,19887	0,25431
LST3U64	0,47298	0,33755	0,37009	0,21653	0,15768
LST4U64	0,45209	0,42922	0,29879	0,21309	0,37715
GRU1U16	0,52345	0,47490	0,43828	0,53853	0,29865
GRU2U16	<i>0,68038</i>	0,26651	0,40708	0,29553	0,17969
GRU3U16	0,47719	0,44460	0,35864	0,27058	0,36656
GRU4U16	0,41579	0,36027	0,32119	0,17731	0,12995
GRU1U32	<i>0,59074</i>	0,39569	0,44699	0,35616	0,29803
GRU2U32	0,45669	0,32886	0,36514	0,32268	0,24273
GRU3U32	<i>0,64521</i>	0,36896	0,33199	0,32805	0,17949
GRU4U32	0,36796	0,40978	0,48187	0,34608	0,27885
GRU1U64	0,53254	0,38719	0,54877	0,47950	0,48928
GRU2U64	0,56948	0,33017	0,41245	0,18039	0,12679
GRU3U64	0,30832	0,25688	0,33634	0,33971	0,20711
GRU4U64	<i>0,68332</i>	0,27790	0,23723	0,30992	0,22391

Tabela A.3. Odchylenie standardowe wskaźnika MR dla Autoenkodera w zależności od długości okresu treningowego. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

Model	MR standard deviation				
	3	4	5	6	7
AEL1U32	0,04054	0,02323	0,06963	0,21441	0,03664
AEL1U64	0,06641	0,04982	0,11830	0,07804	0,13563
AEL1U96	0,06520	0,04581	0,05190	0,10848	0,23210
AEL1U128	0,03392	0,07674	0,12720	0,15669	0,07744
AEL2U32	0,01413	0,01772	0,04198	0,02302	0,02032
AEL2U64	0,02758	0,01933	0,03695	0,04999	0,08505
AEL2U96	0,03508	0,06703	0,05219	0,07404	0,03729
AEL2U128	0,01915	0,06084	0,04978	0,06018	0,07241
AEG1U32	0,02416	0,02214	0,03277	0,09908	0,03966
AEG1U64	0,03027	0,10835	0,06904	0,05310	0,17987
AEG1U96	0,03799	0,03897	0,07501	<i>0,41436</i>	0,13098
AEG1U128	0,05834	0,08068	0,18231	<i>0,24246</i>	<i>0,42412</i>
AEG2U32	0,02033	0,02094	0,02723	0,01779	0,05105
AEG2U64	0,02890	0,03554	0,03270	0,06839	0,05403

AEG2U96	0,02906	0,03545	0,04566	0,05086	0,04147
AEG2U128	0,04540	0,04114	0,03488	0,07050	0,09160
AEC7U32	0,25389	0,15225	0,28044	0,11570	0,14204
AEC7U64	0,18280	0,18554	0,08944	0,22632	0,11392
AEC9U64	0,38679	0,24987	0,09925	0,13623	0,20721

Tabela A.4. Odchylenie standardowe wskaźnika MR dla LSTM i GRU w zależności od długości okresu treningowego. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

Model	MR standard deviation				
	3	4	5	6	7
LST1U16	0,29052	0,17758	0,34742	0,25714	0,20520
LST2U16	0,24895	0,21375	0,27734	0,43845	0,16660
LST3U16	0,64618	0,19682	0,22013	0,15577	0,14379
LST4U16	0,25356	0,17409	0,37181	0,38517	0,15508
LST1U32	0,45274	0,29368	0,22723	0,26143	0,11404
LST2U32	0,56283	0,26621	0,15845	0,25481	0,21186
LST3U32	0,41948	0,11085	0,29760	0,21608	0,14232
LST4U32	0,24379	0,24489	0,18590	0,08492	0,21785
LST1U64	0,53526	0,33809	0,20905	0,28912	0,14422
LST2U64	0,37516	0,27956	0,27083	0,16193	0,09611
LST3U64	0,33479	0,26347	0,31550	0,20626	0,12649
LST4U64	0,34028	0,38413	0,15491	0,12602	0,31797
GRU1U16	0,41813	0,34079	0,22836	0,30211	0,26577
GRU2U16	0,47073	0,14890	0,13913	0,29983	0,09423
GRU3U16	0,53536	0,29159	0,31109	0,25755	0,27483
GRU4U16	0,39716	0,38420	0,15836	0,19367	0,05957
GRU1U32	0,33277	0,35406	0,28707	0,29412	0,14671
GRU2U32	0,33107	0,30292	0,14180	0,23872	0,17515
GRU3U32	0,51269	0,24053	0,28135	0,33341	0,12371
GRU4U32	0,23672	0,38896	0,29907	0,34881	0,23900
GRU1U64	0,28149	0,29518	0,26788	0,35850	0,36903
GRU2U64	0,40132	0,26282	0,30482	0,20951	0,09351
GRU3U64	0,22454	0,23531	0,23421	0,28159	0,12984
GRU4U64	0,61750	0,14821	0,21407	0,32601	0,21764

Tabela A.5. Wartości wskaźnika True Negative Rate (TNR) dla różnych modeli i okresów treningowych dla Autoenkodera. Pogrubioną czcionką wyróżniono wartości równe 1.0, oznaczające bezbłędną klasyfikację zaś kursywą wartości równe 0.89, odpowiadające jednej błędnej klasyfikacji.

Model	TNR - True Negative Rate														
	Cutoff = MAE _{max}					Cutoff = 1,05 * MAE _{max}					Cutoff = 1,1 * MAE _{max}				
	3	4	5	6	7	3	4	5	6	7	3	4	5	6	7
AEL1U32	0,33	0,44	0,56	0,33	0,44	0,78	0,89	0,67	0,56	0,67	0,89	1,00	0,67	0,67	0,89
AEL1U64	0,11	0,33	0,44	0,22	0,44	0,56	0,44	0,56	0,56	0,78	0,67	0,67	0,78	0,56	0,89
AEL1U96	0,44	0,22	0,44	0,22	0,22	0,56	0,56	0,44	0,56	0,44	0,78	0,78	0,67	0,56	0,67
AEL1U128	0,56	0,44	0,22	0,11	0,11	0,67	0,56	0,33	0,33	0,22	0,89	0,67	0,44	0,56	0,44
AEL2U32	0,11	0,44	0,44	0,33	0,22	0,89	1,00	0,89	0,89	0,78	1,00	1,00	0,89	1,00	1,00
AEL2U64	0,44	0,56	0,44	0,33	0,33	0,78	0,78	0,67	0,78	0,67	1,00	1,00	0,89	0,78	0,78
AEL2U96	0,78	0,44	0,56	0,44	0,22	0,89	0,67	0,67	0,56	0,56	1,00	0,89	0,89	0,56	0,89
AEL2U128	0,00	0,67	0,56	0,22	0,33	0,89	0,67	0,56	0,78	0,56	1,00	0,67	0,78	0,89	0,67
AEG1U32	0,11	0,33	0,56	0,44	0,44	0,33	0,89	0,78	0,56	0,89	1,00	1,00	0,89	0,78	0,89
AEG1U64	0,22	0,33	0,33	0,33	0,56	0,78	0,56	0,56	0,44	0,78	0,89	0,56	0,78	0,78	0,89
AEG1U96	0,33	0,22	0,11	0,33	0,22	0,67	0,33	0,33	0,33	0,44	0,78	0,67	0,33	0,44	0,44
AEG1U128	0,22	0,22	0,22	0,22	0,00	0,44	0,44	0,44	0,22	0,33	0,67	0,67	0,44	0,44	0,44
AEG2U32	0,44	0,22	0,56	0,33	0,67	0,89	1,00	0,89	1,00	0,78	1,00	1,00	1,00	1,00	0,89
AEG2U64	0,56	0,44	0,56	0,56	0,33	0,89	0,89	0,67	0,67	0,56	1,00	0,89	1,00	0,78	0,67
AEG2U96	0,33	0,11	0,33	0,44	0,33	0,89	0,33	0,67	0,56	0,56	0,89	0,78	0,89	0,78	0,78
AEG2U128	0,56	0,44	0,44	0,44	0,22	0,89	0,78	0,89	0,56	0,44	0,89	0,89	0,89	0,67	0,56
AEC7U32	0,11	0,00	0,11	0,33	0,67	0,11	0,11	0,22	0,56	0,78	0,22	0,22	0,44	0,56	0,78
AEC7U64	0,00	0,33	0,22	0,33	0,44	0,11	0,44	0,22	0,33	0,67	0,11	0,56	0,22	0,33	0,67
AEC9U64	0,00	0,00	0,11	0,56	0,56	0,00	0,00	0,22	0,56	0,56	0,11	0,11	0,44	0,67	0,56

Tabela A.6. Wartości wskaźnika True Negative Rate (TNR) dla różnych modeli i okresów treningowych dla sieci LSTM i GRU. Pogrubioną czcionką wyróżniono wartości równe 1.0, oznaczające bezbłędną klasyfikację zaś kursywą wartości równe 0.89, odpowiadające jednej błędnej klasyfikacji.

Model	TNR - True Negative Rate														
	Cutoff = MAE _{max}					Cutoff = 1,05 * MAE _{max}					Cutoff = 1,1 * MAE _{max}				
	3	4	5	6	7	3	4	5	6	7	3	4	5	6	7
LST1U16	0,11	0,00	0,11	0,11	0,00	0,11	0,00	0,22	0,33	0,22	0,11	0,22	0,22	0,33	0,22
LST1U32	0,00	0,00	0,11	0,00	0,11	0,00	0,11	0,11	0,00	0,22	0,00	0,22	0,11	0,33	0,44
LST1U64	0,11	0,11	0,00	0,11	0,44	0,33	0,22	0,00	0,22	0,44	0,33	0,22	0,00	0,33	0,44
LST2U16	0,00	0,22	0,11	0,56	0,56	0,00	0,44	0,22	0,56	0,56	0,00	0,56	0,33	0,56	0,56
LST2U32	0,11	0,00	0,11	0,22	0,22	0,11	0,11	0,11	0,22	0,22	0,11	0,22	0,11	0,22	0,22
LST2U64	0,00	0,00	0,11	0,22	0,44	0,22	0,00	0,11	0,22	0,56	0,33	0,00	0,22	0,22	0,56
LST3U16	0,33	0,00	0,00	0,11	0,56	0,33	0,00	0,00	0,11	0,56	0,33	0,00	0,11	0,33	0,56
LST3U32	0,11	0,11	0,00	0,11	0,44	0,11	0,11	0,00	0,22	0,44	0,11	0,22	0,00	0,33	0,44
LST3U64	0,11	0,00	0,11	0,22	0,11	0,22	0,11	0,11	0,22	0,22	0,22	0,11	0,11	0,22	0,33
LST4U16	0,11	0,00	0,11	0,22	0,33	0,22	0,00	0,11	0,44	0,33	0,22	0,00	0,11	0,44	0,33
LST4U32	0,00	0,11	0,00	0,22	0,44	0,11	0,33	0,22	0,44	0,56	0,22	0,33	0,22	0,56	0,67
LST4U64	0,22	0,11	0,22	0,33	0,44	0,22	0,22	0,33	0,33	0,44	0,33	0,22	0,33	0,44	0,44
GRU1U16	0,00	0,11	0,00	0,00	0,11	0,00	0,11	0,00	0,00	0,33	0,22	0,22	0,11	0,11	0,44
GRU2U16	0,00	0,22	0,00	0,44	0,33	0,11	0,33	0,00	0,44	0,33	0,11	0,33	0,00	0,56	0,44
GRU3U16	0,11	0,22	0,00	0,22	0,33	0,44	0,22	0,11	0,22	0,33	0,44	0,22	0,11	0,22	0,33
GRU4U16	0,22	0,33	0,22	0,56	0,44	0,33	0,33	0,22	0,67	0,44	0,33	0,44	0,22	0,67	0,67
GRU1U32	0,11	0,11	0,00	0,11	0,33	0,22	0,22	0,00	0,22	0,33	0,22	0,22	0,00	0,22	0,33
GRU2U32	0,11	0,11	0,11	0,33	0,56	0,11	0,11	0,11	0,33	0,56	0,22	0,11	0,11	0,44	0,56
GRU3U32	0,11	0,11	0,00	0,33	0,33	0,11	0,11	0,11	0,33	0,56	0,11	0,11	0,22	0,33	0,67
GRU4U32	0,22	0,00	0,11	0,22	0,33	0,33	0,00	0,22	0,22	0,33	0,33	0,11	0,22	0,33	0,33
GRU1U64	0,00	0,00	0,00	0,00	0,11	0,00	0,11	0,00	0,11	0,11	0,00	0,11	0,00	0,11	0,11
GRU2U64	0,00	0,11	0,22	0,67	0,11	0,11	0,11	0,22	0,78	0,33	0,22	0,11	0,33	0,78	0,56
GRU3U64	0,00	0,33	0,11	0,00	0,56	0,11	0,44	0,33	0,00	0,56	0,33	0,44	0,33	0,22	0,67
GRU4U64	0,22	0,22	0,33	0,22	0,56	0,22	0,22	0,56	0,33	0,56	0,33	0,22	0,56	0,44	0,67

Tabela A.7. Wartości średnia wskaźnika MR dla okresu T1 dla Autoenkodera. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

	MR mean for different training periods				
Model	3	4	5	6	7
AEL1U32	0,05113	0,02915	0,04854	0,11875	0,07110
AEL1U64	0,04808	0,04426	0,10781	0,05398	0,17951
AEL1U96	0,03402	0,10697	0,08853	0,21281	0,25065
AEL1U128	0,04834	0,09466	0,23444	0,17984	0,24968
AEL2U32	0,02498	0,03160	0,04816	0,07021	0,10062
AEL2U64	0,04222	0,07969	0,05141	0,10844	0,08041
AEL2U96	0,03135	0,06051	0,07953	0,10300	0,06142
AEL2U128	0,02739	0,08884	0,07176	0,08669	0,10745
AEG1U32	0,03054	0,04707	0,03479	0,03458	0,05003
AEG1U64	0,06654	0,12098	0,10286	0,09059	0,29596
AEG1U96	0,05245	0,14744	0,18061	0,22583	0,12887
AEG1U128	0,16379	0,16272	0,14953	0,28233	0,32069
AEG2U32	0,02741	0,06602	0,03803	0,03596	0,04618
AEG2U64	0,05042	0,08027	0,09171	0,10251	0,11920
AEG2U96	0,02831	0,08448	0,09364	0,14589	0,12798
AEG2U128	0,04431	0,05253	0,11629	0,17285	0,29888
AEC7U32	<i>0,83450</i>	<i>0,93544</i>	0,68598	0,59947	0,54827
AEC7U64	<i>1,39895</i>	<i>1,01770</i>	0,54210	0,55952	0,39343
AEC9U64	<i>0,93681</i>	<i>0,99969</i>	0,33812	<i>0,77687</i>	0,49274

Tabela A.8. Wartości średnia wskaźnika MR dla okresu T1 dla sieci LSTM i GRU. Pogrubioną czcionką wyróżniono 7 najmniejszych, kursywą 7 największych wartości.

	MR mean				
Model	3	4	5	6	7
LST1U16	1,95641	1,26665	0,88221	1,52120	0,91190
LST2U16	1,87311	1,71382	1,17654	1,57622	0,96567
LST3U16	<i>2,09356</i>	1,91868	1,38721	1,35630	1,06257
LST4U16	1,52183	1,69565	1,81549	1,76192	0,89908
LST1U32	1,87048	1,14084	1,04336	1,22610	1,18020
LST2U32	1,48922	1,93748	1,60217	1,75640	0,93197
LST3U32	1,15805	1,63450	1,34003	1,36031	0,78087
LST4U32	1,48923	1,51632	1,45469	1,63770	1,30821
LST1U64	2,01765	1,09728	0,98244	1,44700	1,03788
LST2U64	<i>2,19303</i>	<i>2,06107</i>	1,63591	1,64868	0,77278
LST3U64	1,85623	1,69333	1,50508	1,66571	0,79131
LST4U64	1,44792	1,70508	1,94945	1,67431	1,10373
GRU1U16	1,34468	1,21157	1,26170	1,32720	0,96774

GRU2U16	2,09532	1,61057	1,62638	1,78258	0,84317
GRU3U16	2,02849	1,94169	1,24672	1,60874	0,83215
GRU4U16	1,69211	1,55088	1,50157	1,83904	1,07541
GRU1U32	1,53480	1,40497	0,94562	1,24363	1,23242
GRU2U32	1,69706	1,96725	1,43312	1,38798	0,92232
GRU3U32	2,28569	2,01920	1,62620	1,77769	1,32294
GRU4U32	1,94381	1,70156	1,54696	1,53838	1,38054
GRU1U64	1,26503	1,24707	0,87537	1,36086	0,95946
GRU2U64	2,13551	1,81779	1,24373	1,51838	0,80422
GRU3U64	1,51824	1,82940	1,58507	1,60563	1,16102
GRU4U64	1,90168	1,23740	1,63545	1,43756	1,73220

Tabela A.9. Wartości średnia wskaźnika AR dla Autoenkodera. Pogrubioną czcionką wyróżniono 7 największych, kursywą 7 najmniejszych wartości.

Model	AR mean				
	3	4	5	6	7
AEL1U32	3,05012	3,31733	3,90280	3,68182	3,70649
AEL1U64	3,48337	4,08550	4,33475	5,44829	4,89508
AEL1U96	3,87193	4,88916	7,92353	8,21517	9,92131
AEL1U128	3,51464	4,87596	4,28970	6,63162	6,40066
AEL2U32	2,44564	3,15305	3,12341	3,97364	3,62779
AEL2U64	2,80578	4,32591	5,71755	7,18621	4,86811
AEL2U96	2,89398	4,33993	4,76396	5,49804	6,29359
AEL2U128	3,43460	4,87736	5,55112	6,26768	9,19711
AEG1U32	2,65289	3,77810	3,58904	3,80917	3,94956
AEG1U64	2,84461	4,51759	4,41866	4,62980	4,12331
AEG1U96	3,07513	4,66967	4,58195	4,25550	5,18991
AEG1U128	3,81447	4,86979	5,31262	5,32923	6,01575
AEG2U32	3,02085	3,83825	3,78791	4,57109	3,76800
AEG2U64	3,56324	5,12673	4,11762	5,22187	5,67083
AEG2U96	4,40338	4,15576	5,24033	6,48129	5,29186
AEG2U128	3,16179	4,55219	6,30958	5,81350	5,30679
AEC7U32	8,15378	11,70802	13,26902	10,08079	18,67025
AEC7U64	11,03933	12,60973	18,46674	16,43032	7,23564
AEC9U64	7,51256	11,44228	16,86993	16,92520	16,23037

Tabela A.10. Wartości średnia wskaźnika AR dla sieci LSTM i GRU. Pogrubioną czcionką wyróżniono 7 największych, kursywą 7 najmniejszych wartości.

Model	AR mean				
	3	4	5	6	7
LST1U16	15,60151	18,62488	15,08596	20,87465	18,20012
LST2U16	21,34068	18,52103	28,68775	22,20669	20,19267
LST3U16	19,36079	17,16739	18,87715	28,23481	19,80395
LST4U16	19,50385	<i>12,37200</i>	16,37253	20,15143	18,81375
LST1U32	19,76128	17,79041	17,15843	17,74367	15,84883
LST2U32	15,03598	13,97635	15,73592	17,71605	18,59864
LST3U32	19,71739	13,18574	20,58818	23,77156	22,06210
LST4U32	22,48515	<i>11,60377</i>	28,84765	19,59814	19,99510
LST1U64	18,12847	19,39665	17,41461	21,13131	14,49876
LST2U64	20,37969	15,16072	16,75528	25,24467	22,21257
LST3U64	22,49238	<i>12,84961</i>	15,90037	22,69671	18,94497
LST4U64	16,03103	14,71438	<i>12,65234</i>	24,79009	15,35840
GRU1U16	18,23402	17,93658	14,63856	19,10022	17,88572
GRU2U16	19,17594	17,20031	15,98972	19,11381	19,32305
GRU3U16	21,43348	15,05487	25,74074	20,17543	18,85330
GRU4U16	20,57914	<i>12,66616</i>	29,58220	18,99519	18,38115
GRU1U32	16,37971	19,81512	18,47404	21,02208	19,88314
GRU2U32	19,33216	13,21185	19,27205	24,18998	24,33848
GRU3U32	22,03094	14,36661	25,39562	19,17371	22,68090
GRU4U32	17,77667	13,42662	24,75046	28,91928	14,25464
GRU1U64	19,26374	19,07829	15,48613	20,21881	17,94711
GRU2U64	21,04956	14,87759	19,78735	20,72757	23,06974
GRU3U64	20,75630	<i>13,13004</i>	17,02775	25,29052	22,56636
GRU4U64	19,22817	<i>11,08577</i>	16,70148	20,20152	14,62902

Tabela A.11. Wartości wskaźnika F1-score dla Autoenkodera dla różnych okresów treningowych. Pogrubioną czcionką wyróżniono wartości równe 1.0, oznaczające bezbłędną klasyfikację zaś kursywą wartości równe 0.92, odpowiadające jednej błędnej klasyfikacji.

Model	F1-score														
	Cutoff = MAE _{max}					Cutoff = 1,05 * MAE _{max}					Cutoff = 1,1 * MAE _{max}				
	3	4	5	6	7	3	4	5	6	7	3	4	5	6	7
AEL1U32	0,75	0,71	0,75	0,75	0,80	0,86	<i>0,92</i>	0,86	0,80	<i>0,92</i>	<i>0,92</i>	1,00	<i>0,92</i>	<i>0,92</i>	<i>0,92</i>
AEL1U64	0,71	0,71	0,67	0,71	0,86	0,86	0,86	0,80	0,80	<i>0,92</i>	1,00	<i>0,92</i>	0,86	<i>0,92</i>	<i>0,92</i>
AEL1U96	0,86	0,80	0,75	0,75	0,75	1,00	0,86	0,86	0,80	0,75	1,00	<i>0,92</i>	<i>0,92</i>	0,80	0,80
AEL1U128	0,80	0,71	0,75	0,80	0,75	<i>0,92</i>	0,75	0,75	0,80	0,80	<i>0,92</i>	0,80	0,80	0,86	0,80

AEL2U32	0,80	0,75	0,92	0,75	0,92	1,00	1,00	1,00	0,92	1,00	1,00	1,00	1,00	0,92	1,00
AEL2U64	0,75	0,75	0,75	0,80	0,80	0,86	0,92	0,86	0,86	0,86	1,00	0,92	1,00	0,86	0,92
AEL2U96	0,86	0,80	0,80	0,86	1,00	0,92	0,86	0,80	0,92	1,00	1,00	0,92	1,00	0,92	1,00
AEL2U128	0,80	0,86	0,75	0,71	0,86	0,92	0,86	0,86	0,80	0,92	1,00	0,92	0,92	0,86	1,00
AEG1U32	0,80	0,75	0,86	0,71	0,75	0,86	0,86	1,00	0,92	0,92	1,00	1,00	1,00	1,00	0,92
AEG1U64	0,75	0,71	0,71	0,75	0,71	0,92	0,75	0,86	0,92	0,80	0,92	0,86	0,86	0,92	0,80
AEG1U96	0,75	0,71	0,71	0,71	0,67	0,86	0,80	0,80	0,71	0,71	1,00	0,86	0,80	0,75	0,80
AEG1U128	0,71	0,71	0,75	0,71	0,67	0,80	0,71	0,80	0,75	0,75	0,86	0,75	0,86	0,80	0,75
AEG2U32	0,75	0,75	0,80	0,80	0,80	0,92	0,92	0,92	1,00	0,92	1,00	0,92	1,00	1,00	0,92
AEG2U64	0,80	0,80	0,71	0,71	0,86	0,92	0,80	0,86	0,80	0,86	0,92	0,86	0,92	0,92	0,86
AEG2U96	0,75	0,75	0,67	0,80	0,67	1,00	0,86	0,75	0,80	0,71	1,00	0,92	0,86	0,86	0,86
AEG2U128	0,80	0,75	0,67	0,71	0,75	0,92	0,86	0,75	0,75	0,75	0,92	1,00	0,80	0,75	0,86
AEC7U32	0,67	0,67	0,71	0,71	0,75	0,67	0,67	0,71	0,71	0,71	0,67	0,67	0,62	0,62	0,77
AEC7U64	0,67	0,67	0,67	0,67	0,62	0,67	0,67	0,67	0,71	0,62	0,67	0,67	0,67	0,71	0,67
AEC9U64	0,67	0,67	0,67	0,71	0,75	0,67	0,67	0,67	0,71	0,75	0,67	0,67	0,67	0,71	0,80

Tabela A.12. Wartości wskaźnika F1-score dla sieci LSTM i GRU dla różnych okresów treningowych.

Model	F1-score														
	Cutoff = MAE _{max}					Cutoff = 1,05 * MAE _{max}					Cutoff = 1,1 * MAE _{max}				
	3	4	5	6	7	3	4	5	6	7	3	4	5	6	7
LST1U16	0,71	0,71	0,67	0,67	0,67	0,71	0,71	0,67	0,71	0,71	0,71	0,71	0,71	0,75	0,71
LST1U32	0,67	0,71	0,75	0,67	0,71	0,67	0,71	0,75	0,67	0,75	0,67	0,71	0,75	0,67	0,75
LST1U64	0,67	0,67	0,71	0,71	0,71	0,71	0,67	0,71	0,75	0,71	0,71	0,67	0,71	0,75	0,71
LST2U16	0,67	0,67	0,67	0,67	0,75	0,71	0,67	0,67	0,67	0,75	0,71	0,67	0,67	0,67	0,80
LST2U32	0,71	0,67	0,67	0,71	0,67	0,75	0,67	0,67	0,71	0,67	0,75	0,67	0,71	0,71	0,67
LST2U64	0,71	0,67	0,67	0,67	0,71	0,71	0,67	0,67	0,67	0,71	0,71	0,67	0,67	0,67	0,71
LST3U16	0,67	0,67	0,75	0,71	0,75	0,67	0,67	0,75	0,71	0,75	0,67	0,67	0,75	0,75	0,80
LST3U32	0,71	0,67	0,75	0,67	0,75	0,71	0,67	0,75	0,67	0,75	0,71	0,67	0,75	0,67	0,75
LST3U64	0,75	0,67	0,71	0,67	0,67	0,75	0,67	0,71	0,67	0,67	0,75	0,67	0,71	0,67	0,67
LST4U16	0,67	0,67	0,67	0,71	0,67	0,67	0,67	0,67	0,71	0,67	0,67	0,67	0,67	0,71	0,67
LST4U32	0,67	0,67	0,67	0,67	0,75	0,67	0,67	0,67	0,67	0,75	0,67	0,67	0,67	0,71	0,75
LST4U64	0,67	0,67	0,67	0,67	0,75	0,67	0,67	0,67	0,67	0,75	0,67	0,67	0,67	0,67	0,75
GRU1U16	0,67	0,71	0,71	0,67	0,71	0,67	0,71	0,71	0,67	0,71	0,67	0,75	0,71	0,67	0,71
GRU2U16	0,67	0,67	0,67	0,67	0,71	0,67	0,67	0,67	0,67	0,71	0,71	0,67	0,67	0,67	0,75

GRU3U16	0,67	0,67	0,71	0,67	0,67	0,67	0,67	0,71	0,67	0,67	0,71	0,67	0,71	0,67	0,67
GRU4U16	0,71	0,67	0,71	0,67	0,71	0,71	0,67	0,75	0,67	0,71	0,71	0,67	0,75	0,67	0,71
GRU1U32	0,71	0,71	0,67	0,67	0,71	0,71	0,71	0,71	0,67	0,71	0,71	0,71	0,71	0,67	0,71
GRU2U32	0,71	0,67	0,67	0,71	0,75	0,71	0,67	0,67	0,71	0,75	0,71	0,67	0,67	0,71	0,75
GRU3U32	0,67	0,67	0,71	0,67	0,71	0,67	0,67	0,71	0,67	0,71	0,71	0,67	0,71	0,71	0,71
GRU4U32	0,67	0,67	0,71	0,71	0,67	0,67	0,67	0,71	0,71	0,67	0,67	0,67	0,71	0,75	0,67
GRU1U64	0,71	0,71	0,67	0,71	0,71	0,71	0,71	0,67	0,71	0,71	0,71	0,71	0,71	0,67	0,71
GRU2U64	0,71	0,67	0,71	0,71	0,75	0,71	0,67	0,71	0,71	0,75	0,71	0,67	0,71	0,71	0,80
GRU3U64	0,71	0,67	0,67	0,71	0,75	0,71	0,67	0,67	0,71	0,75	0,71	0,67	0,67	0,71	0,75
GRU4U64	0,71	0,67	0,67	0,75	0,67	0,71	0,67	0,67	0,75	0,67	0,71	0,67	0,67	0,75	0,71

Tabela A.13. Średni czas uczenia sieci w zależności od długości okresu treningowego dla Autoenkodera.

Model	Średni czas uczenia				
	3	4	5	6	7
AEL1U32	123	181	231	318	420
AEL1U64	151	230	280	316	384
AEL1U96	149	231	292	423	568
AEL1U128	156	229	327	444	719
AEL2U32	200	282	418	446	546
AEL2U64	189	296	433	492	619
AEL2U96	232	357	447	660	771
AEL2U128	182	295	405	572	762
AEG1U32	145	175	246	343	330
AEG1U64	123	203	272	373	353
AEG1U96	159	214	278	330	404
AEG1U128	151	235	390	438	589
AEG2U32	153	248	327	453	407
AEG2U64	189	258	365	507	579
AEG2U96	199	317	381	498	560
AEG2U128	179	288	447	579	616
AEC7U32	62	69	90	116	150
AEC7U64	50	68	85	129	143
AEC9U64	54	76	92	116	143

Tabela A.14. Średni czas uczenia sieci w zależności od długości okresu treningowego dla sieci LSTM i GRU

Model	MR mean for different training periods				
	3	4	5	6	7
LST1U16	58	91	116	141	156
LST2U16	63	93	112	145	198
LST3U16	84	102	136	193	259
LST4U16	87	117	161	238	285
LST1U32	70	91	117	153	174
LST2U32	70	96	114	141	211
LST3U32	76	120	139	192	229
LST4U32	89	110	161	211	290
LST1U64	64	92	120	152	191
LST2U64	74	99	135	196	261
LST3U64	85	119	175	245	304
LST4U64	101	139	193	243	358
GRU1U16	69	96	110	134	158
GRU2U16	68	92	105	161	197
GRU3U16	80	105	135	207	275
GRU4U16	87	120	151	253	290
GRU1U32	69	100	107	154	218
GRU2U32	77	106	120	166	214
GRU3U32	84	113	138	266	263
GRU4U32	88	132	192	261	330
GRU1U64	83	110	158	170	228
GRU2U64	84	115	146	213	238
GRU3U64	82	136	155	256	355
GRU4U64	100	136	208	304	342