POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

Wydział Informatyki

Katedra Systemów Inteligentnych i Data Science
Inteligentne systemy przetwarzania danych

Błażej Bartkiewicz
S16752

System wykrywający niezrównoważone zużycie energii w biurowcach

Praca inżynierska
Promotor: Dr. Sinh Hoa Nguyen

Warszawa, wrzesień, 2024

POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

Faculty of Computer Science

Department of Intelligent Systems and Data Science

Błażej Bartkiewicz
S16752

System wykrywający niezrównoważone zużycie energii w biurowcach

Praca inżynierska
Promotor: Dr. Sinh Hoa Nguyen

Warszawa, wrzesień, 2024

Spis treści

1 Wprowadzenie 8
2 Cel i zakres pracy 11
3 Stan wiedzy 12
4 Sieci neuronowe 17
4.1 Budowa sieci neuronowych 17
4.1.1 Neuron 17
4.1.2 Funkcje aktywacji 18
4.1.3 Warstwy w sieci neuronowej 19
4.2 Hiperparametry 20
5 Opis problemu 26
6 Implementacja 28
6.1 Wykorzystane dane 28
6.2 Preprocessing danych 34
6.2.1 Analiza danych 34
6.2.2 Podział danych 39
6.3 Zaimplementowane modele 39
6.3.1 Struktura otrzymanych modeli 41
6.3.2 Analiza procesu trenowania modeli 43
6.4 Aplikacja 47
6.5 Narzędzia i biblioteki 48
7 Analiza wyników 50
7.1 Cooling 50
7.2 Heating 52
7.3 Total building 53
8 Dyskusja 56

9 Podsumowanie i wnioski 58
10 Bibliografia 60
11 Spis rysunków 62

Wykaz skrótów

ASHRAE - ang. American Society of Heating, Refrigerating and Air-Conditioning Engineers - Amerykańskie Stowarzyszenie Inżynierów Ogrzewnictwa, Chłodnictwa i Klimatyzacji

BEMS - ang. Building Energy Management System – system automatyki budynkowej optymalizujący zużycie energii

BMS - ang. Building Management System – system automatyki budynkowej

HVAC - ang. heating, ventilation, air conditioning – system ogrzewania, wentylacji I klimatyzacji

IOT - ang. Internet of Things – Internet rzeczy

IMGW - Instytut Meteorologii i Gospodarki Wodnej

Streszczenie

Celem niniejszej pracy była ocena możliwości wykorzystania sieci neuronowych do przewidywania zużycia energii w budynkach, a w szczególności wykrywania potencjalnych anomalii niezrównoważonego zużycia energii w budynkach biurowych. Praca na podstawie przeglądu dostępnej literatury przybliża problem predykcji wybranych zagadnień opisujących funkcjonowanie systemu ogrzewania, klimatyzacji oraz całkowitego rocznego zużycia energii przez budynek. Opisane metody, pozwoliły na stworzenie i wytrenowanie modeli sieci neuronowych. Dla wybranego przypadku budynku biurowego wykonano analizę danych, prototypowanie oraz finalne modele sieci neuronowych przy użyciu języka Python i przy wykorzystaniu środowiska Jupyter Notebook oraz Pycharm. Opracowany model został zaaplikowany do analizy funkcjonowania budynku i pozwolił na uzyskanie wiarygodnych predykcji zapotrzebowania energii dla kolejnego roku. Wyniki potwierdziły możliwość wykorzystania sieci neuronowych w inteligentnym budownictwie.

Abstract

The goal of this project was to assess the possibility of using neural networks to predict energy consumption in buildings, and in particular to detect potential anomalies of unsustainable energy consumption in office buildings. Based on a review of available literature, the work brings closer the problem of prediction of selected issues describing the functioning of the heating, air conditioning system and the total annual energy consumption of the building. The described methods allowed to create and train neural network models. For the selected case of an office building, data analysis, prototyping and final neural network models were performed using the Python language and using the Jupyter Notebook and Pycharm environments. The developed model was applied to the analysis of the building's functioning and allowed to obtain reliable predictions of energy demand for the next year. The results confirmed the possibility of using neural networks in intelligent construction.

Słowa kluczowe:

Sztuczna inteligencja, Sieć neuronowa, BMS, Predykcja zużycia energii w budownictwie

Keywords:

Artificial intelligence, Neural network, Building management system, Building energy use prediction

1 Wprowadzenie

Zmiany klimatyczne stale zmieniają świat, w którym żyjemy. Emisja gazów cieplarnianych, która jest pośrednio odpowiedzialna za światowe zmiany klimatu stała się globalnym problemem, a ich redukcja wyznaczyła kierunek zmian gospodarczych, szczególnie w Europie. Dekarbonizacja poszczególnych sektorów gospodarki stała się zagadnieniem kluczowym. Zgodnie z Dyrektywą Parlamentu Europejskiego i Rady (UE) 2024/1275 w sprawie charakterystyki energetycznej budynków wskazano, że „*Budynki odpowiadają za 40 % zużycia energii końcowej w Unii i za 36 % unijnej emisji gazów cieplarnianych związanych z energią*" [1]. Oznacza to konieczność analizy funkcjonowania budynków które odpowiadają za emisje gazów cieplarnianych w trakcie ich eksploatacji. *Wizja na 2050 r. dotycząca zdekarbonizowanych zasobów budowlanych wykracza poza kwestię operacyjnych emisji gazów cieplarnianych, na której skupiano się do tej pory*. Wspomniane działania w zakresie funkcjonowania budynków nałożyły się promocję budownictwa „zrównoważonego". Trendem ostatnich 20 lat w budownictwie stało się dążenie do tworzenia obiektów zgodnie z zasadami zrównoważonego rozwoju. W tym celu powstały certyfikacje budynków, które oceniały zgodność z wielokryterialnymi założeniami budownictwa zrównoważonego. Poza zagadnieniami nowoczesnego projektowania i wykonawstwa budynków podkreślano w nich znaczenie fazy eksploatacji budynków stanowiącej dominujący wpływ na całkowity koszt obiektu (liczonego w całym cyklu życia) ale także na zużycie energii i związanej z nią emisją gazów cieplarnianych. Dodatkowo, pojęcie energii zużywanej przez budynek jest błędnie przypisywane budynkowi, gdyż energię zużywa jego użytkownik oraz zainstalowane w nim systemy. Oznacza to konieczność zwiększenia uwagi na sposób w jaki pracują systemy budynkowe, szczególnie, że dwie trzecie energii zużywanej do ogrzewania i chłodzenia budynków nadal pochodzi z paliw kopalnych [1]. Systemy budynkowe stały się zatem głównym celem optymalizacji w nowoczesnym budownictwie. Wymagania dotyczące charakterystyki energetycznej systemów technicznych budynku powinny mieć zastosowanie do całych systemów zainstalowanych w budynkach, a nie do charakterystyki pojedynczych komponentów, które wchodzą w zakres rozporządzeń dotyczących poszczególnych produktów. Aby to było możliwe do spełnienia, niezbędne staje się wyposażenie budynków w odpowiednie systemy do sterowania systemami w budynkach takimi jak ogrzewanie, wentylacja, klimatyzacja (HVAC), oświetlenie itp.

Ostatnie 30 lat to znaczący rozwój systemów automatyki i sterowania. W dużych budynkach wykorzystuje się systemy BMS (Building Management Systems). Systemy te wykorzystują sieci czujników oraz sterowników do sterowania elementami systemów HVAC i ich pracy w zależności od chwilowego zapotrzebowania. Obecnie rozbudowane systemy BMS, w których priorytetem staje się optymalna praca pod kątem energii określane są pojęciem BEMS (Building Energy Management Systems). Rozwój systemów automatyki pozwolił na tworzenie inteligentnych budynków, co pozwala na realizację wizji Smart City.

W 2018 r wprowadzono zapisy Dyrektywy [1] wskazując, że „*Automatyka budynków i elektroniczne monitorowanie systemów technicznych budynku okazały się skutecznymi środkami zastępczymi dla przeglądów, w szczególności w przypadku dużych systemów, i mają ogromny potencjał opłacalnego uzyskania znacznych oszczędności energii zarówno dla konsumentów, jak i dla przedsiębiorstw. Instalację takich urządzeń należy uznać za najbardziej opłacalną alternatywę dla przeglądów w dużych budynkach niemieszkalnych i budynkach wielorodzinnych o dostatecznej wielkości, które umożliwiają uzyskanie okresu zwrotu nieprzekraczającego trzech lat, gdyż dzięki tym rozwiązaniom można podejmować działania na podstawie dostarczonych informacji, a tym samym uzyskiwać oszczędności energii na przestrzeni czasu*. Kluczowym zatem staje się instalacja systemów pozwalających na ciągły pomiar i analizę funkcjonowania systemów. Rozwój takich systemów (np. BMSCare) oznacza wykorzystanie najnowszych algorytmów prognozowania działania na podstawie danych aktualnych jak i historycznych.

Coraz częstszemu wykorzystywaniu odnawialnych źródeł energii towarzyszy problem związany z dużą zmiennością produkcji energii np. ilość energii pozyskiwana przez panele fotowoltaiczne, czy też turbiny wiatrowe jest silnie zależna od warunków pogodowych. Oznacza to konieczność precyzyjnego zarządzania zasobami energetycznymi.

Tak złożone wymagania spowodowały zainteresowanie pojęciem cyfrowego bliźniaka budynku. Oznacza on stworzenie cyfrowego modelu budynku, który po skalibrowaniu pozwala na podejmowanie decyzji o sposobie działania systemów w budynkach. Dlatego w Dyrektywie [1] zostało zawarte wymaganie, aby jak najszerzej wykorzystywać w przyszłości technologię cyfrowego bliźniaka: „*Cyfrowy bliźniak budynku to interaktywna i dynamiczna symulacja odzwierciedlająca stan i zachowanie budynku fizycznego w czasie rzeczywistym. Dzięki wykorzystaniu danych w czasie rzeczywistym z czujników, inteligentnych liczników i innych*

źródeł cyfrowy bliźniak budynku zapewnia całościowy obraz charakterystyki budynku, w tym m. in. zużycia energii, temperatury, wilgotności, i poziomów obłożenia, i może być wykorzystywany do monitorowania zużycia energii w budynku i zarządzania nim.". W tym kontekście możliwość przewidywania funkcjonowania budynku, jego systemów staje się krytycznie ważnym zagadnieniem. Do jego realizacji można wykorzystać szereg narzędzi informatycznych pozwalających na predykcję, która jest niezbędna do podejmowania optymalnych decyzji o sposobie sterowania systemami budynkowymi.

Kluczowym wydaje się zatem możliwość weryfikacji i wdrożenia nowych algorytmów do systemów sterowania systemami w nowoczesnych budynkach zrównoważonych.

2 Cel i zakres pracy

Przedstawione powyżej podejście do budownictwa zrównoważonego pozwoliło na określenie celu i zakresu pracy. Biorąc pod uwagę znaczenie zrównoważonego zarządzania budynkiem cennym stało się ocenienie możliwości wykorzystania nowoczesnych algorytmów do przewidywania stanu funkcjonowania systemów budynkowych (HVAC).

Celem niniejszej pracy stała się zatem ocena możliwości wykorzystania sieci neuronowych do przewidywania zużycia energii w budynkach bazując na danych historycznych. Na podstawie tak dokonanej oceny możliwe stanie się opracowanie testowego systemu wykrywającego niezrównoważone zużycie energii w budynkach biurowych.

Zakresem pracy objęto funkcjonowanie systemów HVAC w nowoczesnych budynkach biurowych. Jest to związane wykorzystaniem systemu, który wymaga złożonej sieci czujników, sterowników. W takich budynkach można znaleźć zaimplementowane złożone systemy HVAC oraz systemy automatyki BMS lub BEMS co gwarantuje dostęp do odpowiedniej ilości danych o funkcjonowaniu systemów. Zakłada się, że w niedalekiej przyszłości, po wdrożeniu Internetu Rzeczy (IoT) znacząco wzrośnie liczba urządzeń generujących i przetwarzających dane co pozwoli na szersze wykorzystanie przedstawionego systemu poza budynki biurowe – np. w budynkach mieszkaniowych czy handlowych. Zakres pracy także obejmował wykorzystanie systemu w przypadku posiadania danych o funkcjonowaniu budynku w ciągu całego roku. Jest to założenie umożliwiające analizę systemów ogrzewania i klimatyzacji w pełnym zakresie pogodowym.

3 Stan wiedzy

Od lat dziewięćdziesiątych ubiegłego wieku naukowcy tworzyli różne narzędzia symulacyjne do przewidywania zapotrzebowania energetycznego budynków [2]. Można je podzielić na trzy kategorie pod względem złożoności: metody inżynierskie, metody wykorzystujące algorytmy sztucznej inteligencji i metody hybrydowe [3]. Metoda inżynierska szacuje zużycie energii wykorzystując równania termodynamiczne konstrukcji budynku połączone z jej instalacjami w celu oszacowania zużycia energii przez pojedynczy element budynku lub cały budynek [5]. Metoda ta, ze względu na znajomość jej wewnętrznej logiki jest powszechnie określana jest jako "biała skrzynka". W odróżnieniu od podejścia inżynierskiego, metoda wykorzystująca narzędzia sztucznej inteligencji jest określana jako "czarna skrzynka", ponieważ przewiduje zużycie energii bez znajomości wewnętrznych elementów budynku. Metoda hybrydowa, określana również jako "szara skrzynka", wykorzystuje metody "czarnej skrzynki", jak i "białej skrzynki" w celu eliminacji ograniczeń związanych z wyżej wymienionymi metodami. Szczegółowy przegląd narzędzi do przewidywania zużycia energii w budynkach, w tym metod „białej skrzynki", „czarnej skrzynki" i „szarej skrzynki", jest dostępny w [3].

Metoda predykcyjna oparta na sztucznej inteligencji przewiduje zużycie energii w budynku na podstawie zmiennych, takich jak warunki środowiskowe, charakterystyka budynku i sposób jego użytkowania, które łączy korelacja. Ze względu na skuteczność predykcji, metody wykorzystujące narzędzia sztucznej inteligencji są szeroko stosowane w dziedzinie przewidywania zużycia energii w budynkach [2]. W poprzednich badaniach porównano skuteczność metod wykorzystujących narzędzia AI z innymi metodami predykcyjnymi zużycia energii w budynkach. Przykładem prac obejmujących zastosowanie w kompleksowej analizie budynków wraz z systemami są prace, w których: Neto i Fiorelli [6] porównali sztuczną sieć neuronową (ANN) z EnergyPlus [7], oprogramowaniem do szacowania zużycia energii w całym budynku, do przewidywania zużycia energii w budynku; Turhan i in. [8] porównali sieć neuronową z propagacją wsteczną (BPNN) z KEP-IYTEESS [9], innym narzędziem do symulacji energetycznej, służącym do przewidywania obciążenia ogrzewania budynków. Badania te wykazały, że podejścia oparte na sztucznej inteligencji, z zaletami takimi jak prostota modelu, szybkość obliczeń i zdolność uczenia się w porównaniu z metodami inżynieryjnymi i hybrydowymi, są najbardziej odpowiednią metodą przewidywania zużycia energii w istniejących zasobach budowlanych. Ponadto, wykorzystując dane szeregów czasowych, modele oparte na sztucznej inteligencji mogą być wykorzystywane do

przewidywania przyszłych wartości zużycia energii, podczas gdy oprogramowanie do modelowania energii jako klasyczne podejście przyszłościowe, oferuje szacowanie energii w ujęciu godzinowym, miesięcznym lub rocznym. Prowadzi to do jednej z istotnych zalet modeli opartych na sztucznej inteligencji, ponieważ wymagają one niewielkiej liczby parametrów, które odpowiednio reprezentują funkcjonowanie budynku wraz z systemami, w porównaniu z algorytmami symulacji energetycznej całego budynku, które wymagają szczegółów konstrukcyjnych, materiałowych, projektowych i instalacyjnych, ponieważ są poddawane jako zmienne wejściowe do obliczeń.

Zgodnie z przeglądem literatury tematu [10] szczególne znaczenie ma obszar implementacji algorytmów. Analizując przeprowadzone prace badawcze wykazano, że ok. 57% stanowi analiza całkowitego zużycia energii w budynku, 11% dotyczy wspólnego działania systemów ogrzewania i klimatyzacji, 13% samej klimatyzacji, 11% samego ogrzewania.

Z. Wang, R.S. Srinivasan

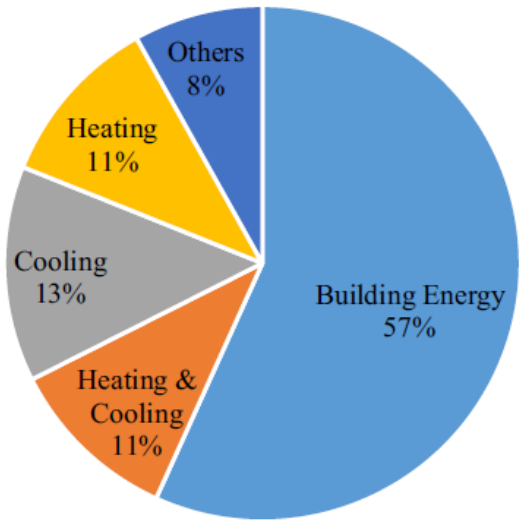


Fig. 2. The composition of energy type.

Rysunek 1: Wykres obszarów analizy w pracach badawczych

Zgodnie z przeglądem literatury tematu [11] najważniejszymi obszarami, w których szuka się optymalizacji są systemy HVAC (39,8%), ogrzewanie i klimatyzacja (23%), oświetlenie (16,8%), obudowa budynku (14%), oraz zużycie ciepłej i zimnej wody (6,2%).

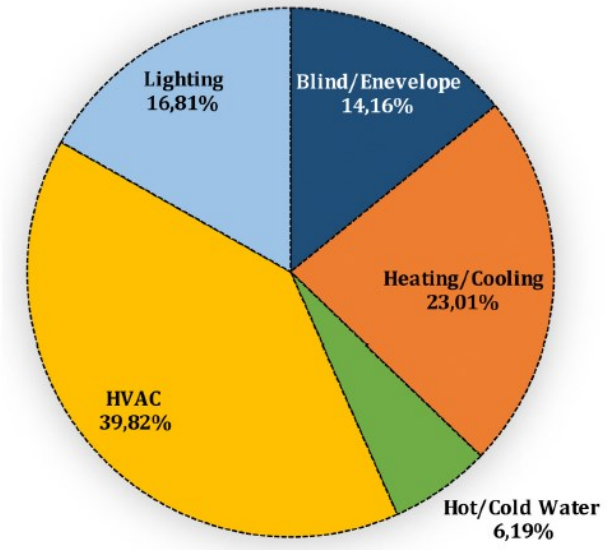


Fig. 11. The relative distribution of AI tools used in building control systems among the reviewed works.

Rysunek 2: Wykres obszarów analizowanych przy pomocy algorytmów AI

Najczęściej analizowanymi danymi budynku przez algorytmy sztucznej inteligencji pochodzą z systemów HVAC oraz systemów ogrzewania i chłodzenia, co potwierdza celowość zajęcia się nimi w niniejszej pracy.

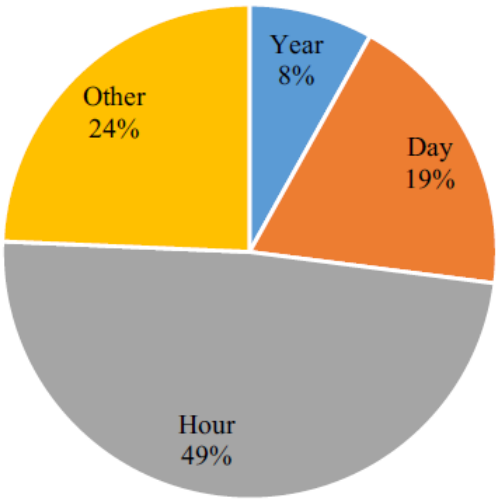


Fig. 3. The composition of prediction time scale.

Rysunek 3: Wykres przedziałów czasowych wykorzystywanych do analizy

Powyższy wykres wskazuje że [10] analizy energetyczne budynków obejmują różne przedziały czasowe. W części analiz kluczowym aspektem jest analiza szczegółowa, wspomagająca proces zarządzania parametrami instalacji i systemów realizowana metodami godzinowymi. Do analiz opisujących charakter pracy w różnych profilach dziennych (np. dzień pracy i weekend) stosuje się analizę dzienną. Do analizy zapotrzebowania energetycznego budynku, w kontekście oszacowania ilości zużywanej energii lub ciepła, a szczególnie kosztów z nimi związanych uzasadnione wydaje się przeprowadzenie bilansu rocznego. Dane wykorzystywane do analizy to najczęściej, bo aż w 49% mają postać godzinową.

W trendzie coraz częściej instalowanych systemów BEMS, kluczowym staje się efektywne przetworzenie zebranych danych pochodzących z budynków na w trakcie eksploatacji. Na przestrzeni ostatnich 30 lat, wskutek wykorzystania różnych algorytmów optymalizacyjnych udało się zredukować zużycie energii średnio od 21.8 do 44.3% oraz zwiększenie komfortu średnio o 21.6 do nawet 85.7% [11]. W zależności od skomplikowania problemu, implementowane były różne algorytmy. Począwszy od systemów wykorzystujących proste algorytmy realizujące regresję liniową, przez te wykorzystujące logikę rozmytą, SVR, kończąc na sieciach neuronowych. Obecnie sieci neuronowe stanowią 41% [10] wszystkich wykorzystywanych algorytmów. Algorytmy sztucznej inteligencji wykorzystywane są zatem nie tylko do prognozowania zużycia energii, ale także znajdują zastosowanie w zagadnieniach zarządzania wodą, monitorowania jakości środowiska wewnętrznego, wykrywania anomalii, optymalizacji zużycia energii przez systemy.

W dostępnych przeglądach literatury [2-10] wskazano na bogaty dorobek naukowy w zakresie prezentowanych zagadnień. Obejmują one kilkaset źródeł naukowych podejmujących szereg szczegółowych zagadnień, w szczególności implementacji algorytmów opisanych poniżej.

Opis algorytmów

W dobie szybko rosnącego zapotrzebowania na optymalne funkcjonowanie bardzo skomplikowanych systemów, coraz częściej okazuje się, że proste reguły decyzyjne nie przynoszą zadowalających efektów. W połączeniu z rosnącą mocą obliczeniową komputerów, coraz częściej wykorzystuje się różnego rodzaju narzędzia sztucznej inteligencji. Narzędzia sztucznej inteligencji, często nazywane w skrócie AI (ang. artificial intelligence), są to programy, algorytmy które na podstawie dużych zbiorów danych początkowo uczą się

dostosowując swoje parametry, a następnie, w zależności od tego do czego zostały stworzone: pozdejmują decyzje, tworzą predykcje, analizują dane, przetwarzają język naturalny, czy też optymalizują procesy. W zależności od zadania do którego będzie wykorzystywana sztuczna inteligencje, dobiera się odpowiednie narzędzie.

Algorytmy uczenia maszynowego (ML, machine learning)

Algorytmy uczenia maszynowego można podzielić na trzy różne kategorie względem sposobu nauki modelu:

Uczenie Nadzorowane (supervised learning)

Uczenie nadzorowane polega na dostarczeniu modelowi etykietowanych danych, czyli takich w których każda próbka zawiera informacje o tym jaki powinien być wynik predykcji. Do najczęściej używanych algorytmów uczenia maszynowego wykorzystujących uczenie nadzorowane zaliczamy: Regresję linową, drzewa decyzyjne, sieci neuronowe oraz SVM

Uczenie nienadzorowane(unsupervised learning)

Kolejną kategorią wśród algorytmów uczenia maszynowego charakteryzującą się odmiennym podejściem do uczenia niż wyżej wymieniona. Algorytmy z tej kategorii uczą się dostosowując swoje parametry próbując znaleźć ukryte wzorce pomiędzy danymi. Przykładami algorytmów wykorzystujących ten sposób uczenia są: K-means lub PCA

Uczenie ze wzmocnieniem (Reinforced learning)

Ostatnią kategorią którą cechuje odmiennie podejście na etapie treningu jest reinforced learning. Jest to technika implementująca metodę nagród i kar. Model metodą prób i błędów próbuje osiągnąć cel maksymalizując nagrody jednocześnie minimalizując kary.

4 Sieci neuronowe

Sieci neuronowe są elementem algorytmów sztucznej inteligencji, przypominającym w budowie i działaniu ludzki mózg. Tak samo jak ludzki mózg, sztuczne sieci neuronowe są w stanie przetwarzać i interpretować skomplikowane informacje na podstawie przekazywanych sygnałów pomiędzy neuronami.

4.1 Budowa sieci neuronowych

4.1.1 Neuron

Kluczowym aspektem do zrozumienia budowy sieci neuronowej jest poznanie jej elementów. Najmniejszym lecz najbardziej istotnym elementem jest neuron, naśladuje on działanie biologicznego neuronu w mózgu. Neuron odbiera sygnały, przetwarza je a następnie przekazuje dalej.

Neuron składa się z wielu wejść i jednego wyjścia. Na wejściu neuron otrzymuje sygnały z innych neuronów lub z danych wejściowych. Każde z wejść ma przypisaną wagę, która definiuje jak duży wpływ na neuron ma dane wejście. Sygnał na wyjściu jest sumą ważoną iloczynów sygnałów wejścia oraz ich wag.

$$z = \sum_{i=1}^{n} x_i * w_i + b$$

Gdzie:

z to wynik sumy ważonej

x_i to wartość wejścia

w_i to waga przypisana do danego wejścia

b to bias pozwalający na przesunięcie wartości sumy ważonej niezależnie od wejść

Następnie wartość sumy ważonej iloczynów sygnałów wejść przekazywany jest do funkcji aktywacji.

Neuron na wyjściu przekazuje zmodyfikowaną wartość, w zależności od budowy sieci, do następnych połączonych neuronów lub jako finalny wynik sieci.

4.1.2 Funkcje aktywacji

Funkcje aktywacji stanowią bardzo ważny element sieci neuronowej ponieważ dzięki nim zostaje wprowadzona nieliniowość w działaniu neuronu. Jest to bardzo istotna cecha, ponieważ to dzięki niej sieci neuronowe są w stanie modelować skomplikowane zależności pomiędzy danymi. Funkcje aktywacji decydują jakie sygnały będą przekazywane na wyjście neuronu co ma bezpośredni wpływ na wynik działania modelu. Najczęściej wykorzystywanymi w sieciach neuronowych funkcjami aktywacji są:

Funkcja aktywacji ReLU (Rectified Linear Unit)

$$f(z) = \max(0, z)$$

Funkcja ReLU zwraca 0 dla wartości mniejszych od zera, dla pozostałych zwraca ich wartość. Jest to prosta, a zarazem skuteczna funkcja aktywacji. Prostota jest główną zaletą, ponieważ nie jest wymagająca obliczeniowo, ale spełnia najważniejszą role, którą jest eliminacja liniowości i rozwiązuje problem zanikania gradientów. Wadą ReLU jest możliwość utworzenia martwego neuronu, czyli takiego który nie będzie się uczył przyjmując stale wartości ujemne oraz funkcja nie jest różniczkowa w punkcie 0.

Funkcja Sigmoidalna

$$f(z) = \frac{1}{1 + e^{-z}}$$

Funkcja sigmoidalna mapuje wartości na wyjściu na przedział (0,1), co wyjątkowo dobrze sprawdza się przy rozwiązywaniu problemu klasyfikacji binarnej. Jej potencjalnym problemem może być problem zanikania gradientu, co skutecznie utrudnia uczenie głębokich sieci.

Funkcja aktywacji SoftMax

$$f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

SoftMax jest to funkcja matematyczna odwzorowująca wektor liczb w wektor prawdopodobieństw. Każda wartość w wektorze podobieństw jest z przedziału (0,1), a ich suma jest zawsze równa 1. Stosowana w warstwach wyjściowych modeli klasyfikujących. Wektor na wyjściu traktowany jest jak prawdopodobieństwa przynależności do poszczególnych klas. Zaletą jest to że idealnie pasuje do problemów klasyfikacji wieloklasowej. Wadą zaś to, że może być kosztowna obliczeniowo, ponieważ wymaga obliczeń wykładniczych dla każdej z klas.

Funkcje aktywacji stanowią kluczowy element sieci neuronowych. Poprzez wprowadzenie nieliniowości dają możliwość analizowania skomplikowanych wzorców oraz zależności pomiędzy danymi. Odpowiedni wybór funkcji aktywacji wpływa na wydajność i skuteczność modelu.

4.1.3 Warstwy w sieci neuronowej

Warstwy w sieciach neuronowych są głównym elementami strukturalnymi sieci. To one organizują neurony w zbiory. To decyduje o tym jak pomiędzy sobą są połączone neurony, a co za tym idzie określają w jaki sposób przekazywane i przetwarzane są dane w sieci. Każda głęboka sieć neuronowa złożona jest z co najmniej trzech warstw:

Warstwa wejściowa (input layer)

Warstwa wejściowa jest swego rodzaju łącznikiem pomiędzy danymi a pozostałymi warstwami w sieci. Liczba neuronów w tej warstwie jest równa liczbie cech w danych wejściowych.

Warstwy ukryte (hidden layer)

Warstwy ukryte to warstwy umieszczone pomiędzy warstwą wejściową a wyjściową. Liczba warstw ukrytych może być różna, w zależności od danych i złożoności problemu. Neurony w tych warstwach przekształcają dane z poprzedniej warstwy wykorzystując wagi, sumy ważone

oraz wybraną funkcję aktywacji, a następnie przekazują zmodyfikowane dane do neuronów w następnej warstwie. Liczba neuronów w poszczególnych warstwach ukrytych może być różna. Głębokie sieci neuronowe to takie które posiadają w swojej strukturze więcej niż jedną warstwę ukrytą.

Warstwa wyjściowa (Output layer)

Warstwa wyjściowa to warstwa która przekazuje finalny wynik. W zależności od rozwiązywanego problemu liczba neuronów w tej warstwie może się różnić. W przypadku klasyfikacji, liczba neuronów wyjściowych będzie równa liczbie klas do których model będzie przyporządkowywał dane wejściowe. Wyjątkiem jest problem klasyfikacji binarnej, wtedy sieć będzie miała jeden neuron z sigmoidalna funkcją aktywacji. Podobnie przy problemie klasyfikacji binarnej, warstwa wyjściowa może posiadać tylko jeden neuron, ale bez aktywatora.

4.2 Hiperparametry

Hiperparametry to parametry ustalane w trakcie tworzenia modelu sieci neuronowej. Od ich wartości zależy jak będzie wyglądała struktura sieci, jak dobrze będzie się model uczył, a to ma bezpośredni wpływ na jakość predykcji finalnego modelu. Dobieranie odpowiednich hiperparametrów następuje tuż przed trenowaniem sieci neuronowej. W odróżnieniu do wartości wag poszczególnych neuronów, hiperparametry nie zmieniają się dynamicznie podczas treningu. Oto kilka przykładów hiperparametrów występujących w sieciach neuronowych:

Liczba warstw i neuronów

Liczbę warstw oraz liczbę neuronów w każdej z nich określa się w trakcie tworzenia modelu sieci. Od wartości tych hiperparametrów zależy struktura finalnego modelu. W zależności od problemu który ma rozwiązywać sieć neuronowa oraz od danych liczba warstw oraz neuronów może się różnić.

Batch size

Jest to hiperparametr bezpośrednio wpływający na jakość uczenia. Definiuje on liczbę próbek danych w trakcie jednego kroku uczenia modelu. W zależności od wybranej wartości hiperprarametru, trening modelu może trwać dłużej lub krócej, stabilniej lub mniej stabilnie.

Learning rate

Learning rate to hiperparametr odpowiadający za wielkość kroków w kierunku minimalizacji funkcji straty podczas treningu sieci neuronowej. Dobranie odpowiedniej wartości jest kluczowe, ponieważ zbyt mały współczynnik nauki spowoduje, że model będzie się uczył wolno. W przeciwnym wypadku, kiedy wartość learning rate będzie zbyt duża, uczenie modelu będzie bardzo niestabilne.

Funkcje aktywacji

Kolejnym bardzo ważnym typem hiperparametrów są funkcje aktywacji. Głównym ich zadaniem jest wprowadzenie nieliniowości do modelu. Funkcje aktywacji, wraz z przykładami zostały szerzej opisane w rozdziale „Funkcje aktywacji".

Liczba epok (epochs)

Epoką w kontekście sieci neuronowych nazywa się, jeden cykl przetworzenia całego zestawu danych treningowych przez model. W zależności od liczby epok, model może niewystarczająco dopasować swoje parametry i jakość finalnego wyniku nie będzie zadowalająca. W przeciwnym wypadku, kiedy liczba epok będzie zbyt duża, może dojść do overfittingu. Problem overfittingu występuje wtedy, gry model zbyt dobrze dopasował swoje parametry do danych treningowych i wyniki które osiąga podczas analizowania danych treningowych są bardzo dobre, lecz jakość predykcji lub klasyfikacji dla nowych danych jest dużo gorsza. Wówczas umiejętność generalizacji modelu jest bardzo niska.

Optymalizatory

Optymalizatory są to algorytmy wykorzystywane do aktualizacji wag w modelu. Ich zadaniem jest manipulacja wagami modelu w taki sposób, aby zminimalizować funkcje kosztu. Proces ten odbywa się na podstawie gradientów obliczanych przy pomocy algorytmu propagacji

wstecznej. Optymalizator jest odpowiedzialny za dobranie odpowiedniego learning rate dla aktualizacji wartości wag. Przykładami optymalizatorów są:

Stochastic Gradient Descent (SGD)

Jest to jeden z najprostszych optymalizatorów. Analizuje on wagi na podstawie gradientu obliczanego dla pojedynczych próbek. Jego zaletą jest prostota i łatwość w implementacji, problemem może być fakt, że przez wykorzystanie gradientu zbliżając się do optymalnych wartości wag minimalizujących funkcję straty, staje się wolny. Dodatkowo jest podatny na zatrzymanie się w minimum lokalnym, które nie jest globalnym minimum do którego powinien dążyć.

Momentum

Optymalizator rozszerzający ideę SGD o dodanie momentum w trakcie aktualizacji wag. Pozwala to na uniknięcie problemu minimów lokalnych oraz dynamiczniejszym przejściu przez bardziej płaskie obszary. Dodatkowo wykorzystuje poprzednie gradienty do aktualizacji wag co może wpłynąć pozytywnie na prędkość uczenia.

Adagrad (Adaptive gradient)

Jest to optymalizator charakteryzujący się tym, że zmienia współczynnik uczenia dla każdego parametru w sposób adaptacyjny, bazując na wcześniejszych gradientach. Indywidualne podejście do każdego z parametrów może przyczynić się do lepszych wyników uczenia, szczególnie w przypadku rzadziej występujących danych.

RMSprop

Optymalizator bazujący na optymalizatorze Adagrad, dodatkowo wykorzystujący wygładzenie wykładnicze, aby zachować ścisły zakres współczynnika uczenia. Zaletą takiego podejścia jest eliminacja szybko zmieniających się wartości współczynników uczenia, które mogą wystąpić w Adagrad.

Adam (Adaptive Moment Estimation) [12]

Optymalizator bazujący na RMSprop i Momentum. Do dostosowania odpowiedniej wartości współczynnika uczenia wykorzystuje zarówno średnią gradientów jak i ich wariancje. Jest to

jeden z najczęściej wykorzystywanych optymalizatorów. Ceniony za dużą wydajność i adaptacyjność do różnych danych.

Regularyzacja

Regularyzacja jest zbiorem technik wykorzystywanych w trakcie treningu modelu sieci neuronowej, pozwalających na ograniczenie zjawiska overfittingu. Najczęściej wykorzystywanymi technikami regularyzacji są:

Dropout

Dropout polega na losowym wyłączeniu pewnej części neuronów w warstwie w trakcie każdej iteracji treningowej. W zależności od ustawionego parametru dropout rate dla wybranej warstwy, odpowiednia liczba neuronów zostaje zdezaktywowana. Kluczowy jest fakt, że przy każdej iteracji inny zestaw neuronów zostaje wyłączony. Po ukończonym trenowaniu modelu, zostają wszystkie neurony włączone. Oprócz zmniejszenia prawdopodobieństwa overfittingu, technika ta poprawia generalizację modelu.

L1 i L2

Regularyzacja L1 i L2 zapobiegają przeuczeniu poprzez dodanie do wartości funkcji kosztu kar. To powoduje, że model dąży do uproszczenia, a co za tym idzie redukcji złożoności. Techniki te realizują podobne założenia, lecz w różny sposób.

Regularyzacja L1 (Lasso Regresion)

$$J(\theta) = MSE + \lambda \sum_{i=1}^{n} |\theta_i|$$

Gdzie:

$J(\theta)$ – funkcja kosztu,

MSE – średni błąd kwadratowy (Mean Squared Error)

λ – parametr regularyzacji odpowiadający za siłę regularyzacji

θ_i – współczynniki modelu

Model będzie dążył do minimalizacji wartości wag, w niektórych przypadkach zostaną one wyzerowane. Ta technika często wykorzystywana jest w sytuacji kiedy niezbędne jest zmniejszenie liczby cech.

Regularyzacja L2 (Ridge Regression)

$$J(\theta) = MSE + \lambda \sum_{i=1}^{n} {\theta_i}^2$$

$J(\theta)$ – funkcja kosztu,

MSE – średni błąd kwadratowy (Mean Squared Error)

λ – parametr regularyzacji odpowiadający za siłę regularyzacji

θ_i – współczynniki modelu

Wykorzystując Ridge regression wartości wag stają się coraz mniejsze, lecz nigdy nie osiągną zera. Pozwala to na uproszczenie modelu, lecz struktura pozostaje taka sama, jedynie wartości wag stają się mniejsze.

Zważywszy na dużą liczbę hiperparametrów i potrzebę ich optymalizacji, ręczny ich dobór staje się bardzo czasochłonny, a czasami wręcz niemożliwy do zrealizowania. W poszukiwaniu optymalnych wartości hiperparametrów wykorzystywane są różne techniki automatyzujące ten proces, które pozwalają na skrócenie czasu poszukiwań. Metodami najczęściej wykorzystywanymi w przypadku sieci neuronowych są:

Grid search

Grid search jest najprostszą techniką polegającą na zdefiniowaniu zakresów wartości dla każdego z hiperparametrów, a następnie trenowaniu modelu dla każdej możliwej kombinacji tych wartości. Niewątpliwie zaletą tej metody jest prostota oraz to że optymalne wartości hiperparametrów z podanych przedziałów zostaną zawsze znalezione. Problemem staje się sytuacja w której liczba hiperparametrów oraz ich zakresy są duże. Liczba kombinacji rośnie wykładniczo, co sprawia że proces ten staje się bardzo czasochłonny.

Random search

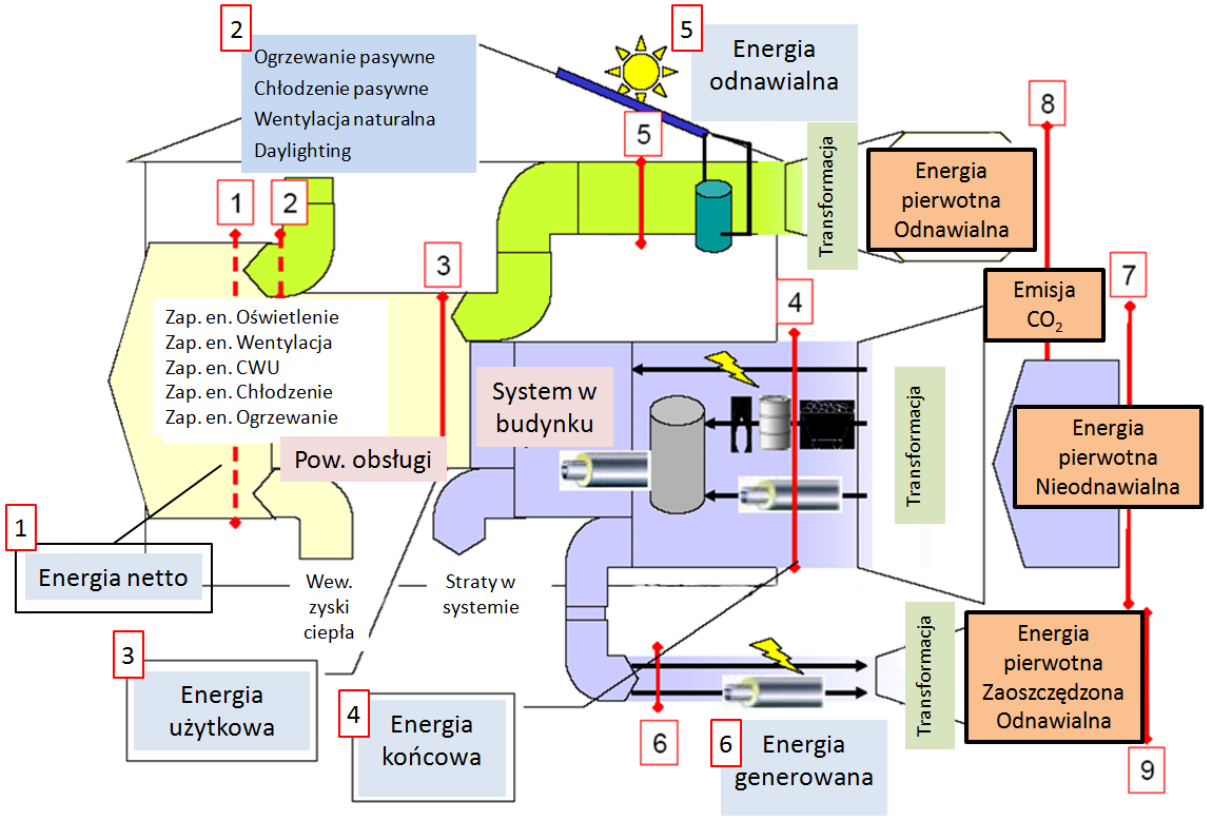
Metoda jest podobna do grid search, tak samo wymaga określenia zakresów wartości szukanych hiperparametrów, ale odróżnią ją to że testowane są losowe ich losowe kombinacje. Technika ta jest bardziej efektywna czasowo, ale nie gwarantuje odnalezienie optymalnych wartości.

Optymalizacja Bayesowska

Optymalizacja Bayesowska jest bardziej skomplikowaną metodą wyszukiwania optymalnych hiperparametrów od wcześniej wymienionych. Wykorzystuje probabilistyczne modele zastępcze np. procesy Gaussa, do przewidywania które kombinacje hiperparametrów są bardziej prawdopodobne do bycia optymalnymi.

5 Opis problemu

Nowoczesne budynki biurowe zapewniają wysokie parametry komfortu użytkownikom dzięki zastosowaniu szeregu systemów i instalacji budynkowych. Do najważniejszych z nich można zaliczyć systemy ogrzewania, wentylacji i klimatyzacji (HVAC), elektryczne, oświetlenie, przygotowanie ciepłej wody, niskonapięciowe, teletechniczne, wodociągowe, kanalizacyjne itp. Nad ich właściwą pracą czuwają systemy automatyki i sterowania (BMS i BEMS). Zagadnienie przepływu energii w budynkach przedstawiono na poniższym rysunku.



Rysunek 4 - Przepływ energii w budynku

Przedstawione powyżej zapotrzebowanie na energię wynikającą z funkcjonowania użytkownika oznaczono energią netto. Część z niniejszego strumienia energii może pochodzić z pasywnych technik (wykorzystanie światła dziennego - daylight, ogrzewanie i chłodzenie pasywne, wentylacja naturalna) lub wewnętrznych zysków ciepła. Po uwzględnieniu niniejszych elementów uzyskuje się tzw. energię użytkową. Może ona być pokrywana ze źródeł odnawianych zainstalowanych w budynku lub wspomnianych powyżej systemów budynkowych. Do pracy niniejszych niezbędne staje się pozyskanie energii elektrycznej i ciepła z sieci zewnętrznych. Jej ilość oznaczana jest pojęciem energii końcowej. Ponieważ energia ta

może być pozyskiwana poprzez zakup na rynku energii (o zmiennej cenie) cennym staje się możliwość predykcji zapotrzebowania na energię w określonym przedziale czasowym. Najczęściej przewidywania dotyczą poszczególnych godzin najbliższej doby, zapotrzebowania w cyklu tygodniowym oraz bilansowo w cyklu rocznym.

Opisane powyżej systemy stanowią złożony, spójny system, w którym wykorzystuje się prognozowane dane pogodowe pozwalając na optymalne funkcjonowanie systemów budynkowych. Wiedząc bowiem, że np. w piątkowy wieczór (kiedy użytkownicy opuszczą biuro) nastąpi obniżenie temperatury można wcześniej wyłączyć system klimatyzacji. Podobnie, jeśli widomo, że w kolejnym dniu budynek nie będzie użytkowany możliwe jest zastosowanie obniżonej temperatury (obniżenie nocne) w systemie ogrzewania powodując kolejne ograniczenie zużywanego ciepła. Co więcej systemy mogą wykorzystywać akumulację ciepła w strukturze budynku lub magazynach (energii elektrycznej, ciepła, chłodu). Sterowanie takimi systemami wymaga precyzyjnych informacji na temat przewidywanego profilu zapotrzebowania. Jeśli do tego dodać zmienną produkcję ze źródeł odnawialnych (np. fotowoltaika) cennym staje się możliwość przewidywanego zapotrzebowania na energię i ciepło w budynku.

W pracy skupiono się zatem na ocenie możliwości wykorzystania sieci neuronowych do przewidywania przyszłego zapotrzebowania na energię dla 3 wybranych zagadnień – systemów ogrzewania (heating), klimatyzacji (cooling) oraz całkowitego zapotrzebowania na energię dla całego budynku. Zagadnienia te, charakteryzujące się odmiennymi profilami są jednymi z najważniejszych w kontekście inżynierskiej informacji o rzeczywistym profilu zużycia energii w budynku.

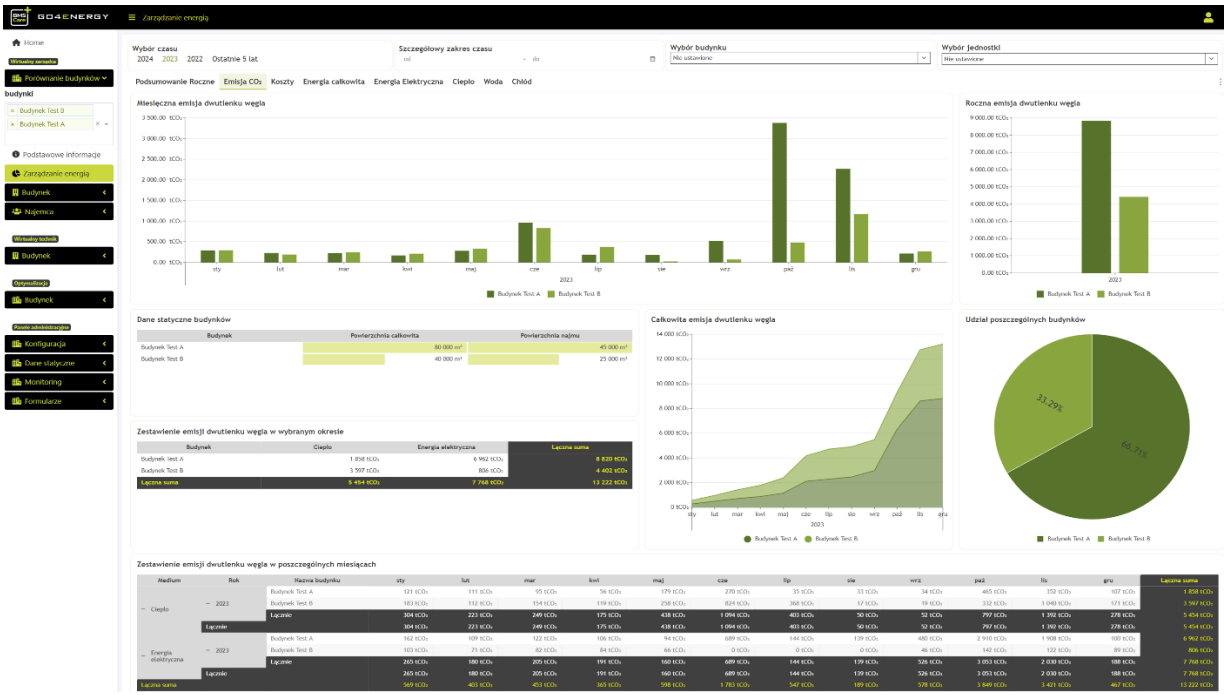
Oczekiwanym celem działania systemu opartego o sieci neuronowe jest wykrywanie odmiennego – nieoptymalnego sposobu działania systemów budynkowych. Znając oczekiwane – prognozowane zużycie energii można je porównać z chwilowym (godzinowym) zużyciem. W przypadku występowania znaczącej różnicy pomiędzy odczytem, a przewidywaniem zostanie wykryta anomalia, która po zasygnalizowaniu technikom pomoże zdiagnozować wykryty problem. Zagadnienie wielkości różnicy sygnalizującej anomalię zależy od budynku, jego złożoności, profilu eksploatacji itp. i powinno być identyfikowane indywidualnie dla każdego obiektu.

6 Implementacja

Po dogłębnej analizie stanu wiedzy, z którego wynika, że około 41%[3] algorytmów wykorzystywanych do predykcji w budownictwie, to sieci neuronowe, do rozwiązania problemu zawartego w celu pracy, zostały stworzone modele sieci neuronowej.

6.1 Wykorzystane dane

Do właściwego wykorzystania opisywanych zagadnień niezbędne staje się pozyskanie danych o historycznym, rocznym rzeczywistym zapotrzebowaniu budynku na energię i ciepło. Pozyskanie niniejszych danych jest możliwe dzięki systemom zbierającym i porządkującym informację o funkcjonowaniu budynku np. BMSCare.



Rysunek 5 - Informacja o zużyciu energii i emisji w budynku [BMSCare]

Wykorzystanie danych rzeczywistych pochodzących z istniejących budynków obarczone jest szeregiem ograniczeń technicznych. Po pierwsze niezbędnym staje się uzyskanie pełnej, spójnej i wiarygodnej informacji o zużyciu energii. Oznacza to trudne do realizacji ciągłe działanie wszystkich czujników przez wszystkie 8760 godzin w ciągu całego roku. Dodatkowo, dane muszą być pozbawione zakłóceń i błędów tak, aby móc stanowić wiarygodną bazę do nauczenia sieci. Dodatkowym problemem jest fakt, że dane te nadal traktowane są jako dane poufne, a zatem pozyskanie ich może stanowić problem w przypadku pracy inżynierskiej.

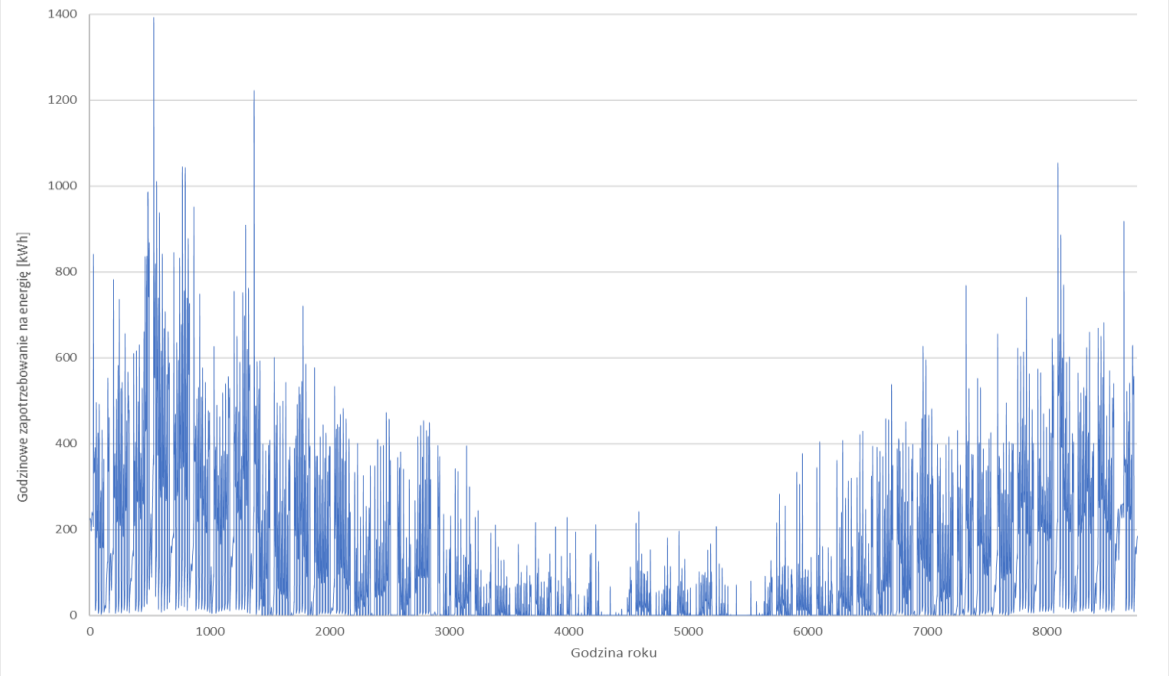
Dlatego na potrzeby niniejszej pracy wykorzystano, zgodnie z opisywaną we wstępie Dyrektywą [1] dane pochodzące z cyfrowego bliźniaka rzeczywistego budynku. Niniejszy bliźniak został przygotowany w oparciu o system HAP firmy Carrier i zawierał dane rzeczywistego budynku biurowego znajdującego się w Polsce.

Budynek jest obiektem biurowym o powierzchni 40 153 m^2. Znajduje się w nim 409 przestrzeni w których utrzymywane są parametry komfortu. Służy do tego wykorzystanie 27 systemów ogrzewania, wentylacji i klimatyzacji (HVAC). Dane, z których korzystano do analizy, trenowania modeli, a finalnie do ich testowania pochodziły z programu HAP dla inteligentnych budynków. Skalibrowana, rzeczywista roczna charakterystyka energetyczna budynku została przedstawiona w poniższej tabeli.

Zapotrzebowanie na energię	**Site Energy (kWh)**
Wentylatory – System wentylacji	899 720
Chłodzenie (Cooling) – system klimatyzacji	343 098
Ogrzewanie (Heating) – system ogrzewania	991 452
Pompy	192 589
Odzyski ciepła	4 969
Razem systemy HVAC	**2 431 828**
Oświetlenie – system oświetlenia	946 213
Urządzenia elektryczne – system elektryczny	1 607 412
Pozostałe zapotrzebowanie – system elektryczny	543 963
Pozostałe zapotrzebowanie na energię w budynku	98 389
Razem pozostałe systemy	**3 195 977**
Razem budynek	**5 627 805**

Na potrzeby pracy wybrano 3 wiodące zagadnienia dotyczące różnego funkcjonowania systemów budynkowych. Każde z nich charakteryzuje się innym charakterem i zależy od odmiennych warunków.

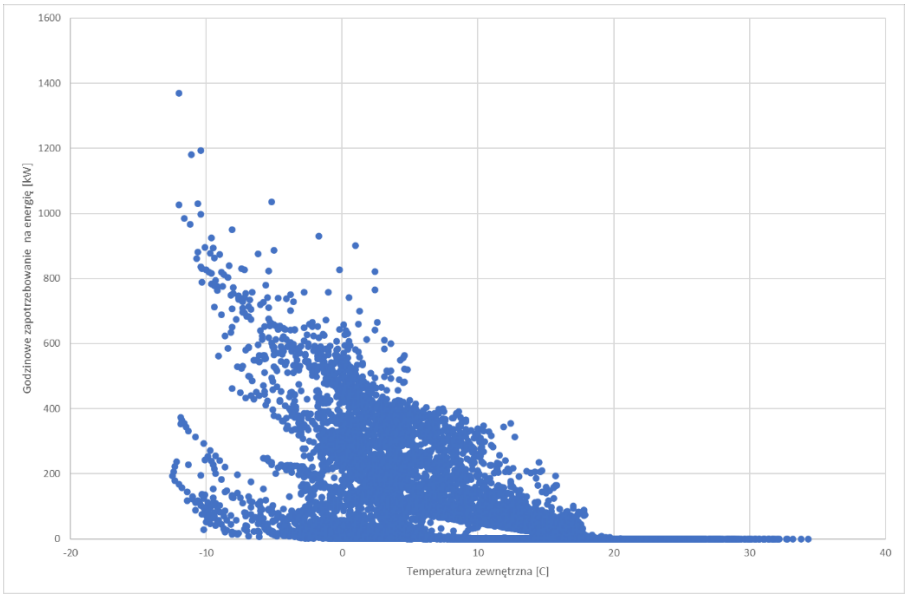
System ogrzewania



Rysunek 6 - Godzinowe zapotrzebowanie na ciepło w ciągu roku [kWh]

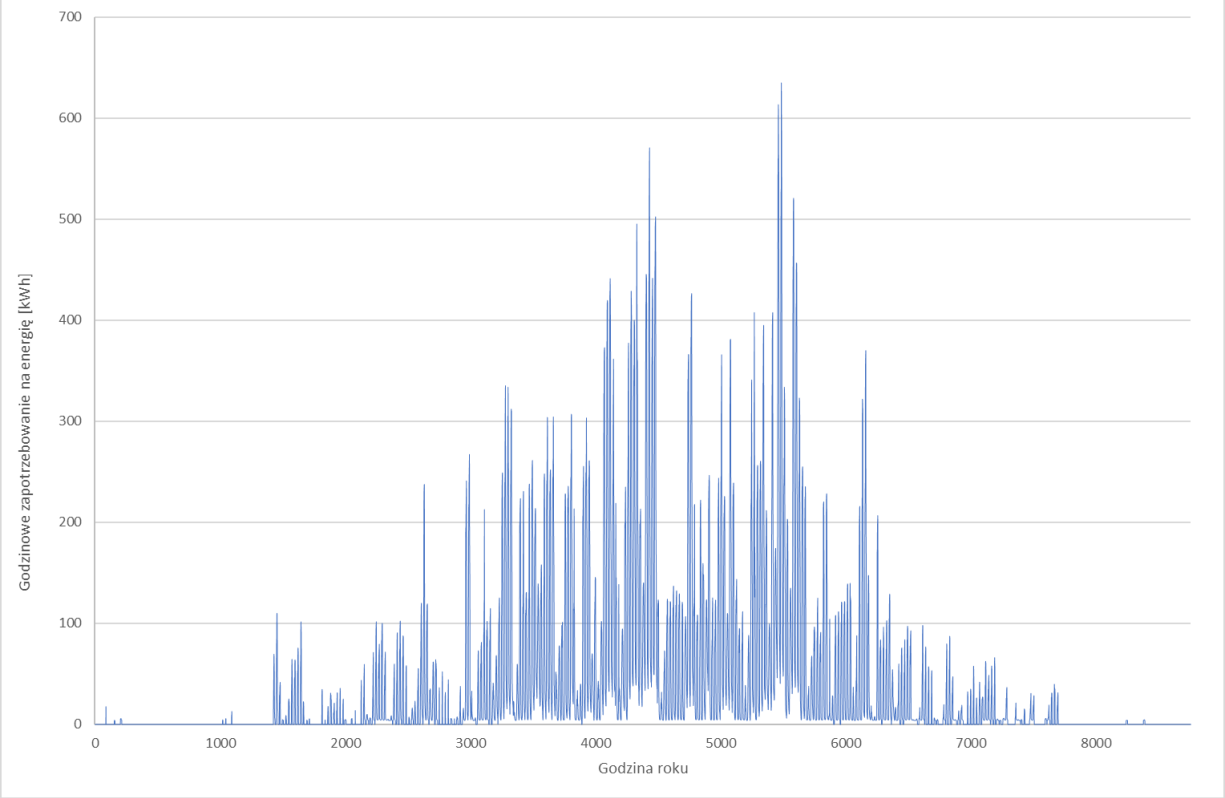
Powyższy wykres przedstawia godzinowe zapotrzebowanie na ciepło dla całego roku.

System ogrzewania w dużej mierze zależy od temperatury zewnętrznej co przedstawia poniższy wykres godzinowego zapotrzebowania na ciepło w budynku w zależności od temperatury zewnętrznej.



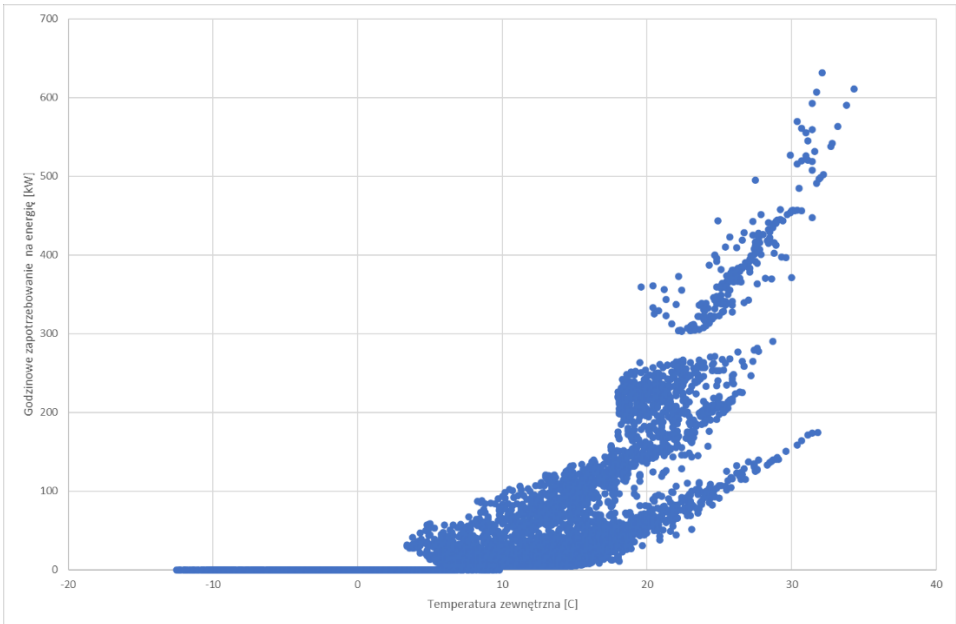
Rysunek 7 - Godzinowe zapotrzebowanie na ciepło w zależności od temperatury zewnętrznej [kWh]

System klimatyzacji - chłodzenia



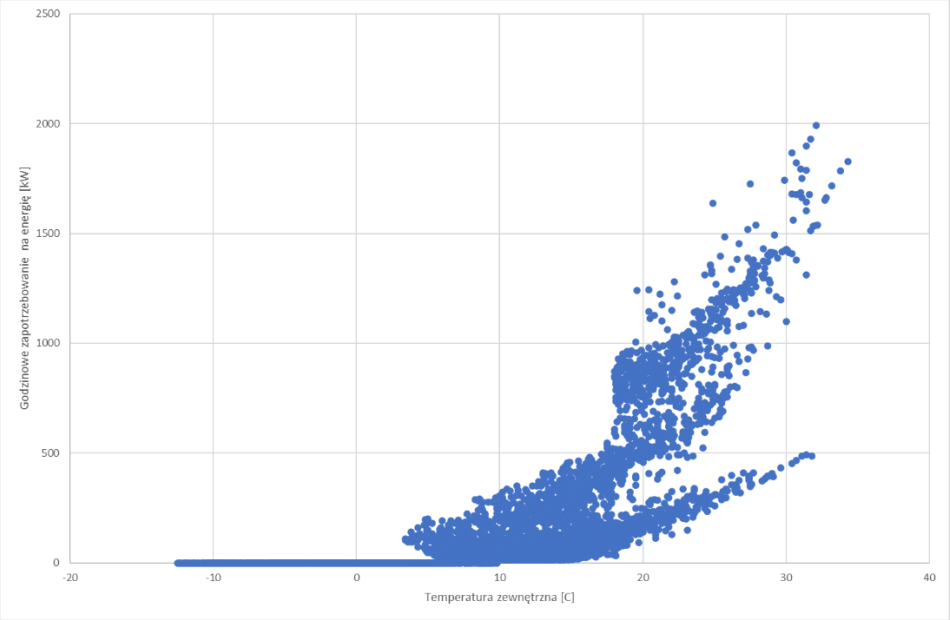
Rysunek 8 - Godzinowe zapotrzebowanie na „chłód" w ciągu roku [kWh]

System chłodzenia w dużej mierze zależy od temperatury zewnętrznej co przedstawia poniższy wykres godzinowego zużycia energii elektrycznej na potrzeby produkcji chłodu w budynku.



Rysunek 9 - Godzinowe zużycie energii elektrycznej na potrzeby chłodzenia w zależności od temperatury zewnętrznej [kWh]

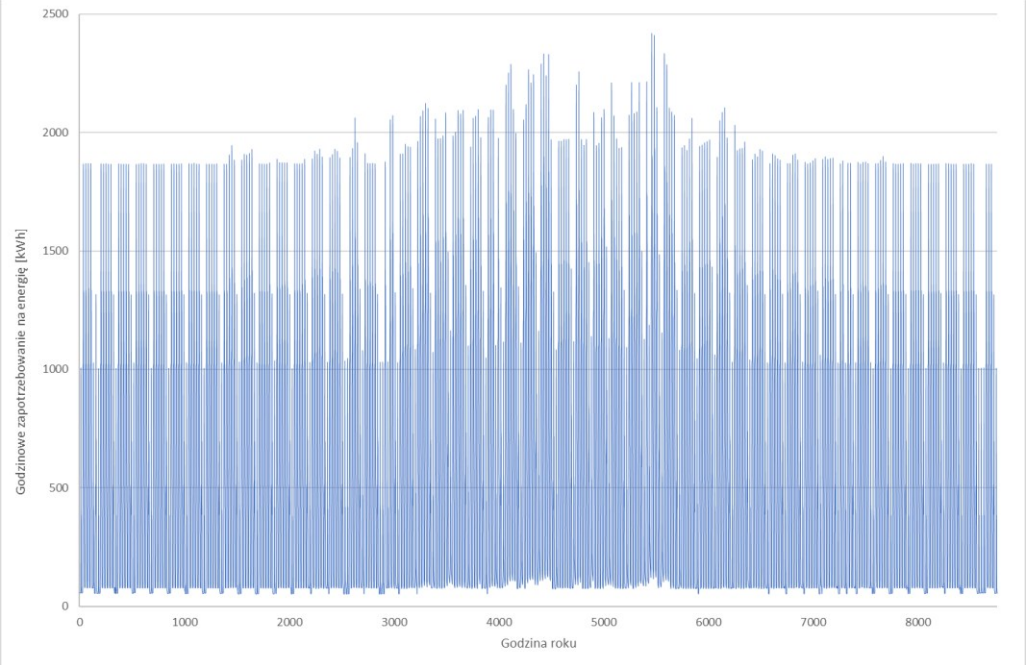
Ponieważ z pobranej energii elektrycznej produkowany jest „chłód” jego analiza wydaje się kluczowa w kontekście predykcji zapotrzebowania na „chłód”.



Rysunek 10 - Godzinowe zapotrzebowanie na chłodzenie w zależności od temperatury zewnętrznej [kWh]

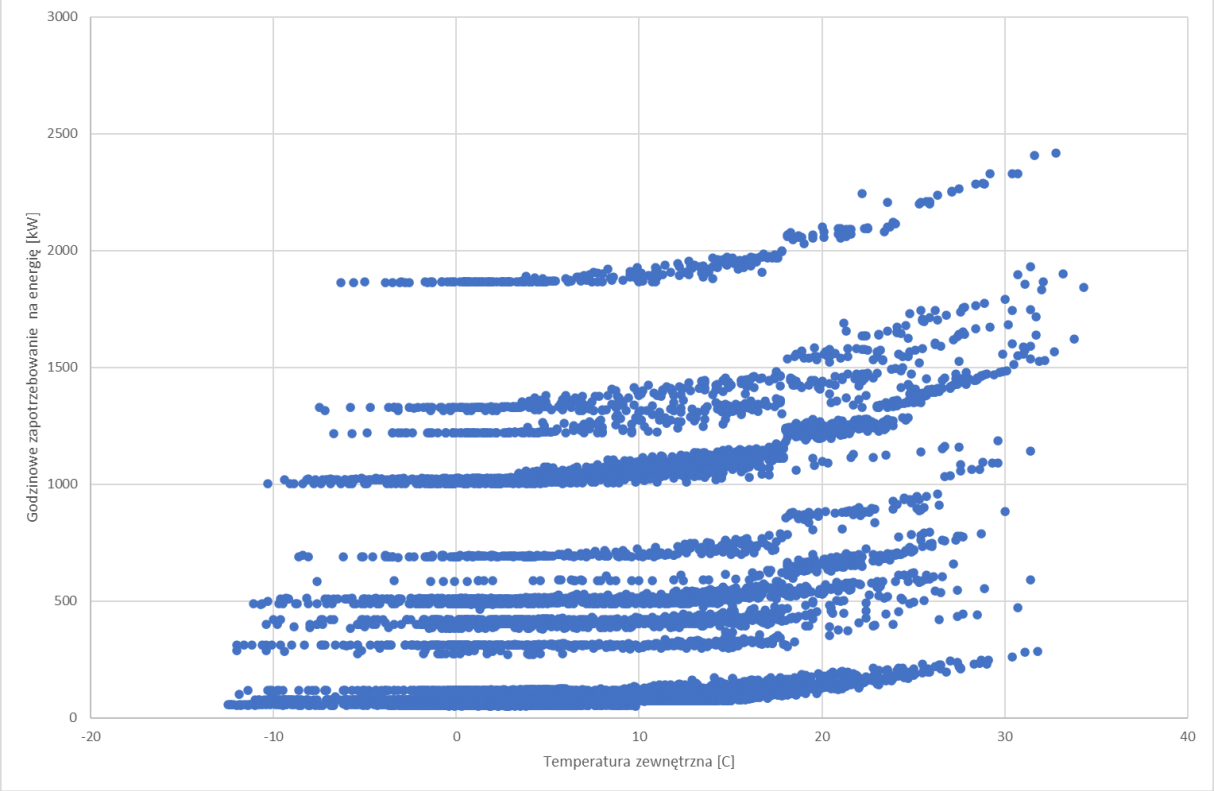
Całkowite zużycie energii elektrycznej w budynku

Trzecim, elementem analizy było całkowite zużycie energii elektrycznej w budynku. Poniżej przedstawiono godzinowe zużycie energii elektrycznej w budynku dla całego roku.



Rysunek 11 - Godzinowe zużycie energii elektrycznej w budynku w ciągu roku [kWh]

W tym przypadku zużycie w mniejszym stopniu (w stosunku od systemów ogrzewania i chłodzenia) zależało od temperatury zewnętrznej, choć wyraźnie widać wpływ klimatyzacji dla temperatury zewnętrznej powyżej 20°C. Warstwowy charakter zużycia energii elektrycznej dla całego budynku wynika z przyjętych i zaobserwowanych profili użytkowania budynku. Dla 10 typowych profili zajętości budynku zaobserwowano związane z nimi różne zużycie energii elektrycznej dla całego budynku.

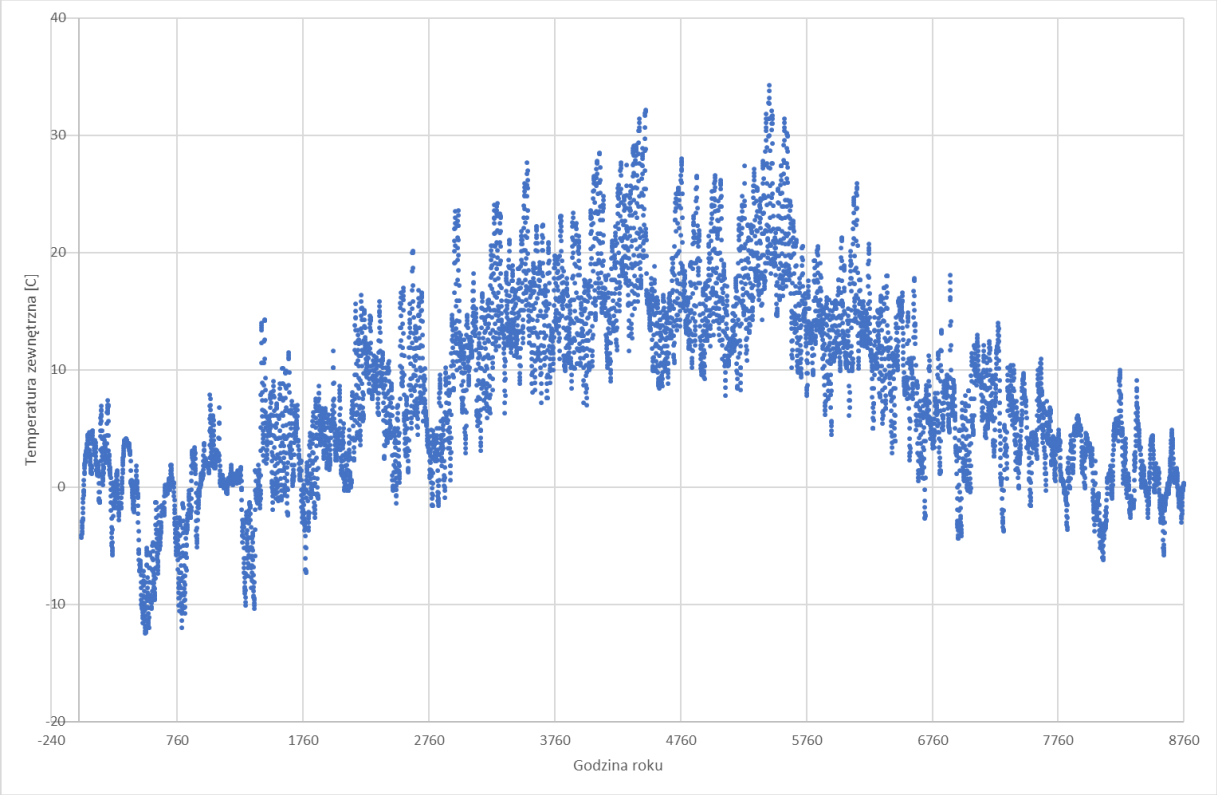


Rysunek 12 - Godzinowe zapotrzebowanie na energię elektryczną budynku w zależności od temperatury zewnętrznej [kWh]

Jak wskazano na powyższych wykresach omawiane 3 zagadnienia znacząco różniły się w kontekście ich korelacji z warunkami zewnętrznymi. Różnią się także specyfiką funkcjonalną stanowiąc ciekawe, odmienne zagadnienie na potrzeby predykcji wartości zapotrzebowania i zużycia energii w budynkach.

Analizę danych przyjętych do dalszych prac należy zakończyć prezentacją danych meteorologicznych. Dane te zostały wybrane na podstawie reprezentatywnego dla analiz energetycznych roku meteorologicznego dla Warszawy stworzonego przez ASHRAE na podstawie wieloletnich danych pogodowych pozyskanych z IMGW [4]. Dane przyjęte do analizy zawierały informacje o temperaturze zewnętrznej powietrza, wilgotności, natężeniu promieniowania słonecznego, kierunku i prędkości wiatru itp. Najważniejszym parametrem dla

niniejszych analiz stała się temperatura powietrza zewnętrznego, której rozkład w cyklu rocznym przedstawiono na poniższym wykresie.



Rysunek 13 - Rozkład temperatury zewnętrznej

6.2 Preprocessing danych

Preprocessing danych jest to proces polegający na przygotowaniu danych, w taki sposób, aby siec neuronowa była w stanie wykorzystać pełen potencjał informacji zawartych w zbiorze. Preprocessing jest jednym z kluczowych elementów tworzenia dobrze działającego modelu uczenia maszynowego, ponieważ od jakości danych wejściowych, w dużym stopniu zależy finalny wynik predykcji. Najczęściej polega on na analizie jakości danych, uzupełnieniu brakujących danych, ujednoliceniu typów danych, czyszczeniu danych z wartości niechcianych, transformacji danych, a finalnie na skalowaniu i normalizacji.

6.2.1 Analiza danych

Preprocessing zaczęto od dokładnej analizy danych zbiorów z których korzystano. Dane były zapisane w dwóch plikach: „Raw_data_2023” oraz „Raw_data_2024”, zawierały one informacje o budynku wzbogacone o dane pogodowe. Każdy z plików zawierał 8760 rekordów

opisanych 19 cechami. Rekord opisywał jakie jest zapotrzebowanie energetyczne wybranych systemów w danej godzinie. Każdy z plików zawierał informacje o zapotrzebowaniu w skali roku.

					Budynek				Szczegół										
Month	Day	Week day	Hour	Dry-Bulb Temp (°C)	Heating Load (kW)	Chiller Output (kW)	Chiller Input (kW)	Total Building Electric (kW)	Precool Coil L	Preheat Coil	Terminal Co	Terminal He	Ventilation I	Exhaust Fan	Terminal Fa	Vent. Reclai	Lighting (kW	Electric Equi
Jan	1	Niedziela	0	-4,3	222,6	0	0	55,2	0	0	0	222,5	0	0	3,1	0	15,3	27,6
Jan	1	Niedziela	1	-4,1	224,1	0	0	55,2	0	0	0	224,1	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	2	-4	225,4	0	0	55,3	0	0	0	225,4	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	3	-3,8	225,5	0	0	55,3	0	0	0	225,6	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	4	-3,5	225	0	0	55,3	0	0	0	225,1	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	5	-3,3	224,7	0	0	55,2	0	0	0	224,8	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	6	-3	223	0	0	55,2	0	0	0	222,9	0	0	3,2	0	15,3	27,6
Jan	1	Niedziela	7	-2,9	223,1	0	0	55,2	0	0	0	223,2	0	0	3,2	0	15,3	27,6

Rysunek 14 Dane w pliku "Raw_data_2023"

Analiza jakości danych

Analizę jakości danych zaczęto od sprawdzenia czy w pliku znajdują się brakujące dane.

```
data_frame.isnull().sum()
```

```
Month                          0
Day                            0
Week day                       0
Hour                           0
Dry-Bulb Temp (°C)             0
Heating Load (kW)              0
Chiller Output (kW)            0
Chiller Input (kW)             0
Total Building Electric [kW]   0
Precool Coil Load (kW)         0
Preheat Coil Load (kW)         0
Terminal Cooling Coil Load (kW)  0
Terminal Heating Coil Load (kW)  0
Ventilation Fan (kW)           0
Exhaust Fan (kW)               0
Terminal Fan (kW)              0
Vent. Reclaim Device (kW)      0
Lighting (kW)                  0
Electric Equipment (kW)        0
dtype: int64
```

Rysunek 15 Brakujące wartości w pliku „Raw_data_2023”

Jak widać zbiór danych nie zawierał, żadnych brakujących danych. Dzięki czemu nie było potrzeby uzupełniania braków.

Wybór parametrów użytych do trenowania modelu

Parametry które uznano, że niosą istotne informacje do poruszanego w pracy problemu:

- Miesiąc („Month”)
- Dzień tygodnia („Week day”)
- Godzina („Hour”)
- Temperatura zewnętrzna („Dry-Bulb Temp (°C)”)
- Zapotrzebowanie budynku do systemów ogrzewania („Heating Load (kW)”)
- Zapotrzebowanie budynku do systemów chłodzenia („Chiller Output (kW)”)
- Całkowite zapotrzebowanie budynku („Total Building Electric [kW]”)

Wartości z kolumn: miesiąc, dzień tygodnia, godzina oraz temperatura zewnętrzna były cechami na podstawie których model się uczył, a finalnie na ich podstawie dokonywał predykcji. W zależności od modelu atrybutem przewidywanym były wartość z kolumn: „Heating Load (kW)”, „Chiller Output (kW)” oraz „Total Building Electric [kW]”.

Zamiana typów parametrów kategorycznych

Do poprawnego funkcjonowania modelu, kluczowa jest zamiana typów danych w kolumnach zawierających dane kategoryczne na dane liczbowe. Modele uczenia maszynowego operują na danych liczbowych. Dane kategoryczne nie posiadają interpretacji liczbowej, dlatego zamiana na wartości liczbowe pozwoli modelowi lepiej je zrozumieć. Istnieją trzy najczęściej wykorzystywane metody konwersji typów danych:

Label Encoding

Jest to metoda polegająca na zamianie kategorii na unikalną wartość liczbową. Zaletą takiego podejścia jest jego prostota, wadą zaś może być niechciane utworzenie zależności pomiędzy różnymi kategoriami.

One-hot encoding

Jest to metoda polegająca na zamianie każdej z kategorii na dodatkową zmienną binarną. Zaletą jest wyeliminowanie problemu sztucznego wprowadzenia porządku. Poprzez generowanie dodatkowych cech dla każdej kategorii. Metoda ta zaś, nie będzie optymalna dla przypadków danych w których występuje duża liczba różnych kategorii, ponieważ znacząco zwiększa liczbę cech.

Binary encoding

Metoda polegająca na zamianie kategorii na wartość liczbową, a następnie na jej binarny zapis. Jest ona połączeniem dwóch wyżej wymienionych metod, dzięki czemu unika błędu związanego ze sztucznym wprowadzeniem porządku pomiędzy kategoriami, ale nie prowadzi do zwiększonego zużycia pamięci.

Ze względu na niewielką liczbę kategorii, wartości w cechach „Month" oraz „Week day" zostały zamienione na dane liczbowe przy wykorzystaniu metody One-hot encoding. Było to kluczowe do poprawnego analizowania danych przez model sieci neuronowej.

```
weekday_encoded = pd.get_dummies(weekday_df, columns = ['Week day'])
weekday_encoded
```

	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
0	False	False	False	True	False	False	False
1	False	False	False	True	False	False	False
2	False	False	False	True	False	False	False
3	False	False	False	True	False	False	False
4	False	False	False	True	False	False	False
...	...	...	...	...	...	...	...
8755	False	False	False	True	False	False	False
8756	False	False	False	True	False	False	False
8757	False	False	False	True	False	False	False
8758	False	False	False	True	False	False	False
8759	False	False	False	True	False	False	False

8760 rows × 7 columns

Rysunek 16: Zamiana danych "Week day"

Skalowanie danych

Skalowanie danych, nazywane również standaryzacją, jest kolejnym kluczowym elementem preprocessingu danych. Polega na przekształceniu wartości cech do pewnego zakresu lub miały określone wartości statystyczne. Najczęstszymi metodami skalowania są:

Standaryzacja (Z-score normalization) – jest to metoda zamieniająca wartości liczbowe w taki sposób, że ich średnia jest równa 0 a odchylenie standardowe równe 1.

$$x' = \frac{x - \mu}{\sigma}$$

Gdzie:

x to oryginalna wartość

μ to średnia wartość w zbiorze

σ to odchylenie standardowe

x' to przeskalowana wartość

Normalizacja do przedziału (Min-Max scaling) – jest to metoda przekształcająca wartości cech do zadanego przedziału. Najczęściej wykorzystywanymi przedziałami są [0,1] lub [-1,1]

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Gdzie:

x to oryginalna wartość

x_{min} to minimalna wartość w zbiorze

x_{max} to maksymalna wartość w zbiorze

x' to przeskalowana wartość

Dane zostały przeskalowane przy wykorzystaniu metody standardScaler z biblioteki Scikitlearn. Metoda ta implementuje wyżej opisaną standaryzację.

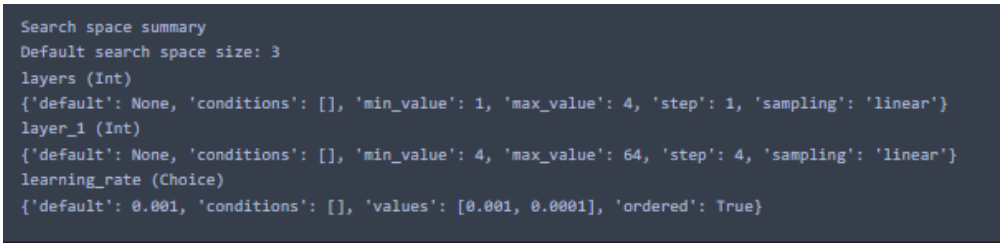
6.2.2 Podział danych

Kolejnym etapem preprocessingu danych jest odpowiedni podział danych na zbiory. Zazwyczaj dane dzieli się na dwa zbiory treningowy i testowy. W zależności od problemu i metodyki, dane mogą być podzielone na większą liczbę zbiorów np. zbiór walidacyjny. Na podstawie danych zawartych w zbiorze treningowym model uczy się ,dostosowując swoje parametry do jak najlepszego przewidywania lub kategoryzowania. Zbiór walidacyjny jest wykorzystywany do oceny jak dobrze radzi sobie model w trakcie uczenia. Zbiór testowy nie bierze udziału w trakcie nauki. Dane z tego zbioru służą do weryfikacji. Na ich podstawie można ocenić jak dobrze radzi sobie model z danymi których wcześniej nie widział. Ważne żeby dane w obydwu zbiorach były jak najbardziej różnorodne, dlatego najczęściej podział robi się w sposób losowy. W zależności od wykorzystywanego modelu oraz ilości danych, procentowy podział danych może być różny, ale zazwyczaj 80% to dane treningowe a 20% testowe lub 75% dane treningowe a 25% testowe. W sytuacji niewielkiego zbioru danych, można zastosować podział 90% do 10%, zwiększy to pulę na której model będzie się uczył, lecz wówczas wyniki testów mogą być mniej dokładne.

Zważając na charakter problemu, ważnym elementem podczas nauki modelu było to, aby trening został przeprowadzony na danych zawierających rekordy z całego roku. Dane miały charakter temporalny, strumieniowy oraz były uporządkowane w czasie co powodowało, że ich losowy podział nie jest odpowiednim podejściem. Zestawem treningowym stały się więc dane z 2023 roku, a zbiorem testowym dane z 2024 roku.

6.3 Zaimplementowane modele

Do osiągnięcia celu zostały wytrenowane trzy modele sieci neuronowej. Każdy z nich miał za zadanie przewidywać inny atrybut (zapotrzebowanie na ciepło, zapotrzebowanie na „chłód”, ogólne zapotrzebowanie energetyczne budynku). Najlepsze hiperparametry każdego z modeli zostały obliczone z wykorzystaniem, wcześniej opisanej, metody random search. Obszar przestrzeni wyszukiwań dla każdego z modeli wyglądał tak:



Rysunek 17: Obszar przeszukiwań przestrzeni hiperparametrów

Warstwa wejściowa modelu odpowiedzialnego za wyszukiwanie optymalnych hiperparametrów przyjmowała 21 cech utworzonych na podstawie wcześniej wykonanego prepocessingu. Liczba warstw ukrytych była dobierana przy pomocy tunera. Wartości liczby warstw były liczbą całkowitą z zakresu [1,4], liczba neuronów w każdej z warstw również była parametrem szukanym z zakresu [4,64] z krokiem 4. Learning rate 0.001 lub 0.0001. Optymalizator Adam oraz funkcja aktywacji „Relu", zostały wybrane na podstawie wcześniejszych eksperymentów.

Dla każdego z zagadnień zostały utworzone dwa tunery. Pierwszy optymalizował hiperparametry modelu kiedy danymi wejściowymi były dane przeskalowane, drugi zaś dla danych nie przeskalowanych. Tuner optymalizował wartości hiperparametrów dążąc do minimalizacji wartości MAE. MAE (ang. Mean Absolute Error) jest to średni błąd bezwzględny, wykorzystywany w problemach regresji. Jest to miara różnicy bezwzględnej między wartościami oczekiwanymi, a przewidywanymi przez model.

Wzór MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Gdzie:

n to liczba próbek w zbiorze danych

y_i to wartość rzeczywista i-tej próbki

$\hat{y}_i$ to wartość przewidziana przez model i-tej próbki

Cennym wydaje się przeanalizowanie błędu względnego niemniej

6.3.1 Struktura otrzymanych modeli

Dla wszystkich modeli tuner uznał wartość hiperparametru Learning rate = 0.001 za najlepszą.

- Cooling(zapotrzebowanie energetyczne budynku na systemy chłodzenia):

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 32)	704
dense_10 (Dense)	(None, 44)	1,452
dense_11 (Dense)	(None, 8)	360
dense_12 (Dense)	(None, 1)	9

Rysunek 18: Struktura modelu cooling dla danych skalowanych

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 32)	704
dense_5 (Dense)	(None, 32)	1,056
dense_6 (Dense)	(None, 52)	1,716
dense_7 (Dense)	(None, 24)	1,272
dense_8 (Dense)	(None, 1)	25

Rysunek 19: Struktura modelu cooling dla danych bez skalowania

- Heating (zapotrzebowanie energetyczne budynku na systemy ogrzewania):

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 32)	704
dense_3 (Dense)	(None, 28)	924
dense_4 (Dense)	(None, 4)	116
dense_5 (Dense)	(None, 1)	5

Rysunek 20: Struktura modelu heating dla danych skalowanych

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 32)	704
dense_3 (Dense)	(None, 28)	924
dense_4 (Dense)	(None, 4)	116
dense_5 (Dense)	(None, 1)	5

Rysunek 21: Struktura modelu heating dla danych bez skalowania

- Total building (całkowite zapotrzebowanie energetyczne budynku)

Layer (type)	Output Shape	Param #
dense_10 (Dense)	(None, 32)	704
dense_11 (Dense)	(None, 48)	1,584
dense_12 (Dense)	(None, 32)	1,568
dense_13 (Dense)	(None, 40)	1,320
dense_14 (Dense)	(None, 1)	41

Rysunek 22: Struktura modelu total building dla danych skalowanych

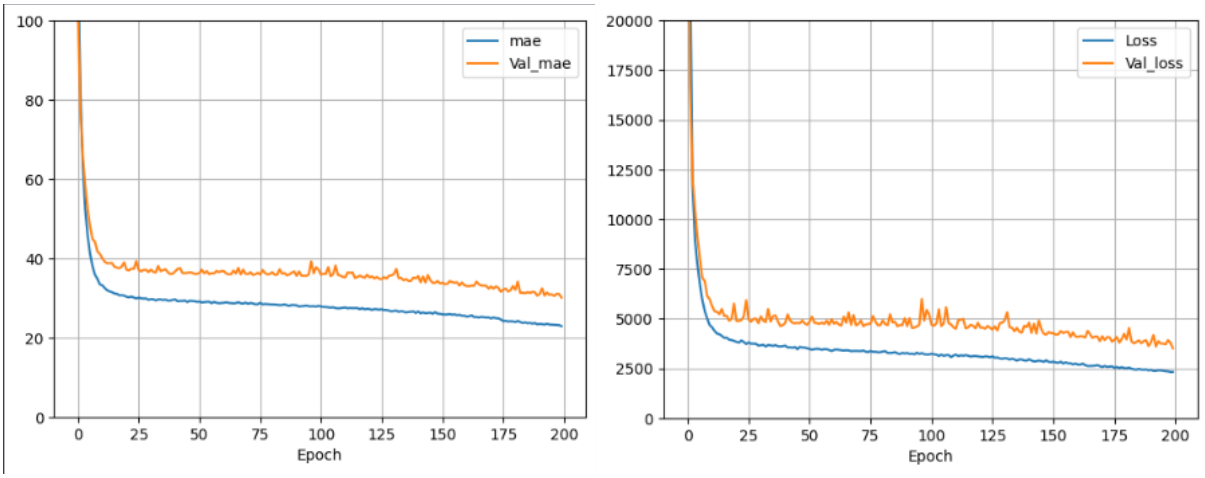
Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 32)	704
dense_6 (Dense)	(None, 60)	1,980
dense_7 (Dense)	(None, 4)	244
dense_8 (Dense)	(None, 60)	300
dense_9 (Dense)	(None, 1)	61

Rysunek 23: Struktura modelu total building dla danych bez skalowania

Na podstawie obliczonych najlepszych wartości hiperparametrów zostały utworzone modele. Finalne modele zostały wytrenowane na podstawie danych zawierających rekordy przedstawiających stan parametrów budynku w skali roku.

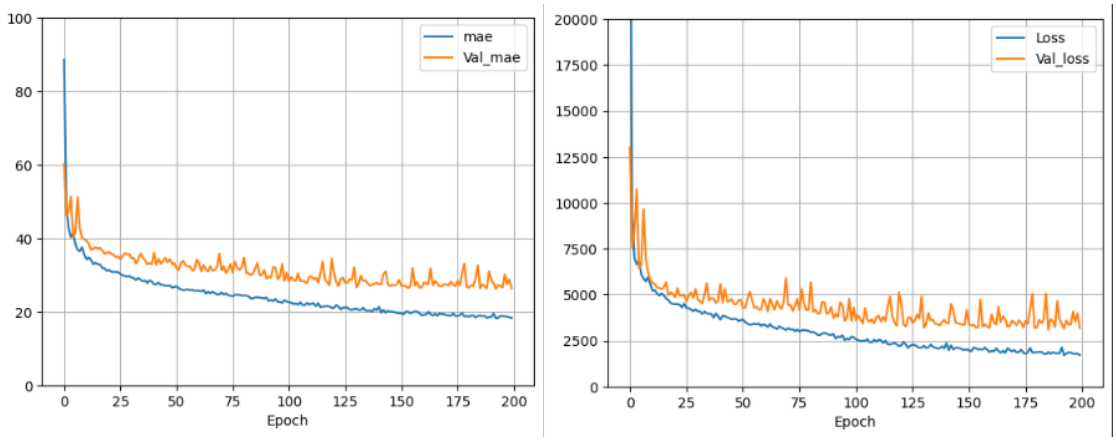
6.3.2 Analiza procesu trenowania modeli

Analiza wyników trenowania modeli cooling



Rysunek 24: Wykresy uczenia modelu cooling na danych skalowanych

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 30,085 kWh.

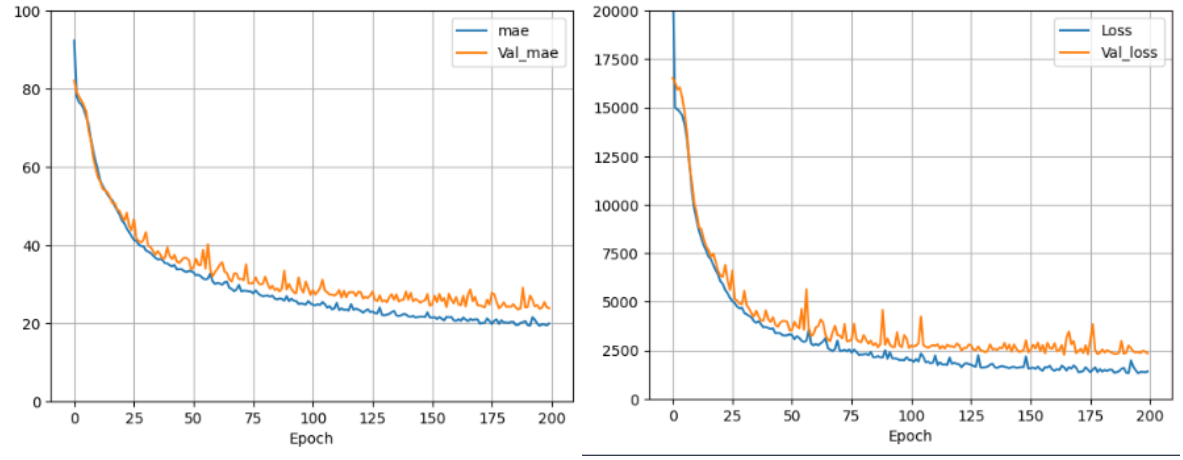


Rysunek 25: Wykresy uczenia modelu cooling na danych bez skalowania

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 26,414 kWh.

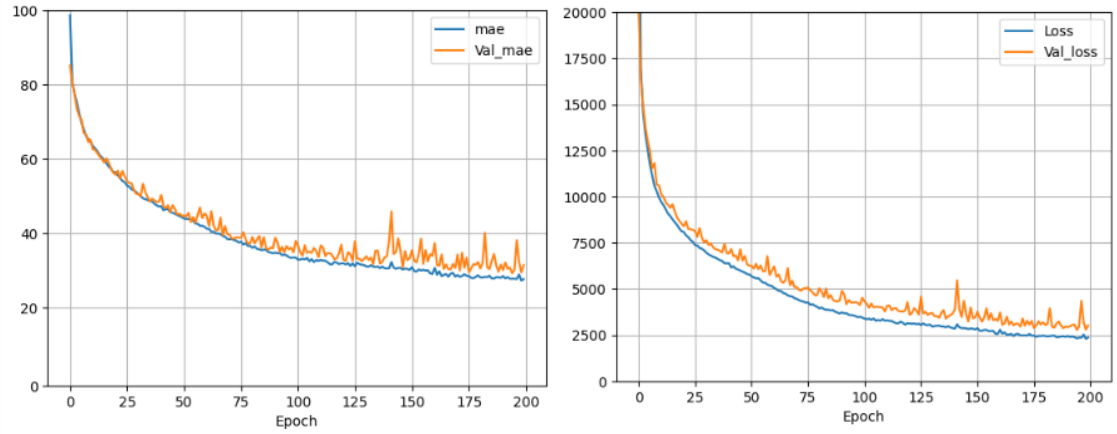
Analizując powyższe wykresy widać, że obydwa modele, niezależnie od skalowania danych bardzo szybko osiągnęły podobną wartość błędu MAE. Trening modelu wykorzystującego dane przeskalowane przebiegał dużo spokojniej, wahania wartości MAE były dużo mniejsze. Uwzględniając mniejszą wartość finalnego średniego błędu kwadratowego w aplikacji został wykorzystany model operujący na danych nieprzeskalowanych.

Analiza wyników trenowania modeli heating



Rysunek 26: Wykresy uczenia modelu heating na danych skalowanych

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 25.14 kWh.

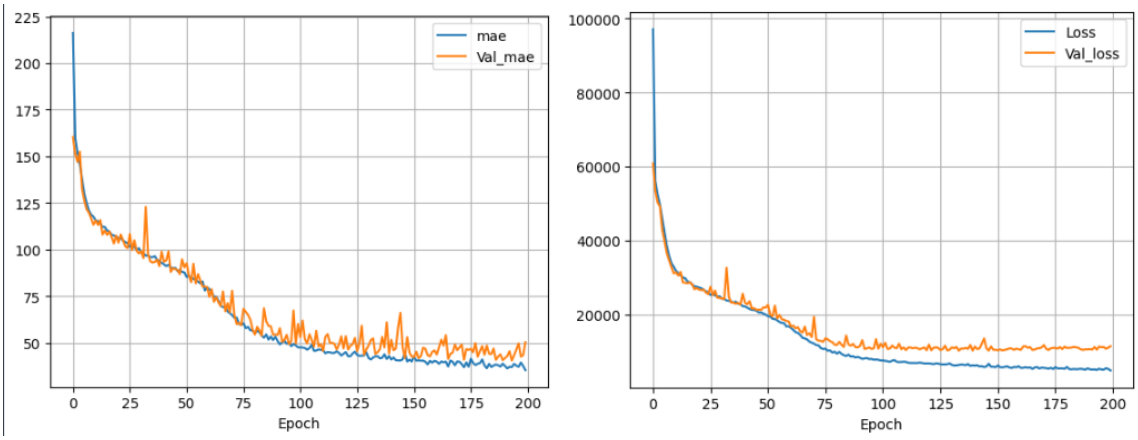


Rysunek 27: Wykresy uczenia modelu heating na danych bez skalowania

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 36.85 kWh.

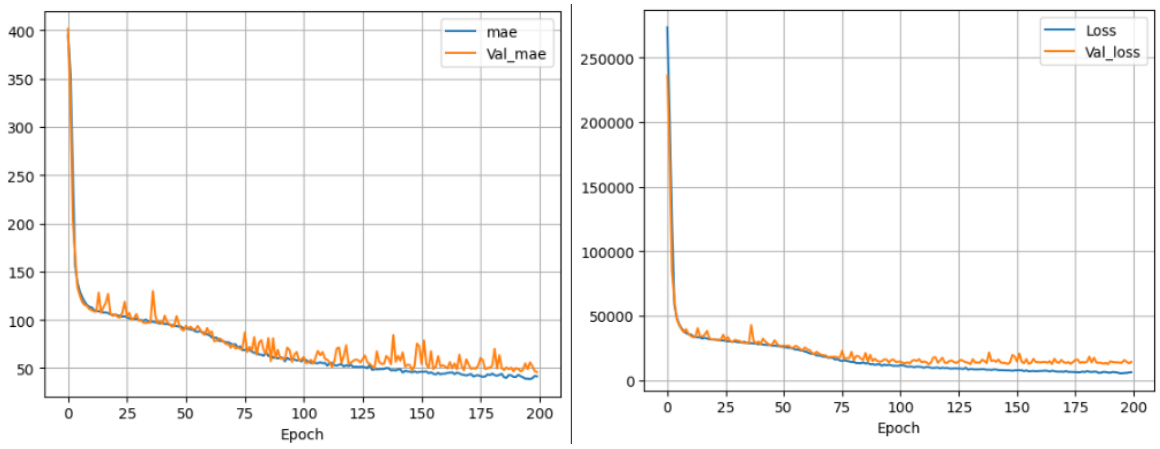
Analizując powyższe wykresy widać, że proces nauki obydwu modeli, niezależnie od skalowania danych był podobny, lecz wartość funkcji straty przy modelu wykorzystującym dane skalowane dużo szybciej malała. Model wykorzystujący dane skalowane osiągnął mniejszą finalną wartość MAE. Dodatkowo predykcje wykonywane na zestawie walidacyjnym w przypadku modelu korzystającym z danych skalowanych osiągały bardziej stabilne, opatrzone mniejszym błędem wartości. Ze względu na lepsze wyniki, w aplikacji, został wykorzystany model optymalizowany i trenowany na danych skalowanych.

Analiza wyników trenowania modelu total building



Rysunek 28: Wykresy uczenia modelu total building na danych skalowanych

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 50.44 kWh.



Rysunek 29: Wykresy uczenia modelu total building na danych bez skalowania

Model sieci po treningu z wartościami hiperparametrów: epochs = 200 oraz batch size =16, learning rate = 0,001, osiągną finalną wartość MAE równą 45.96 kWh.

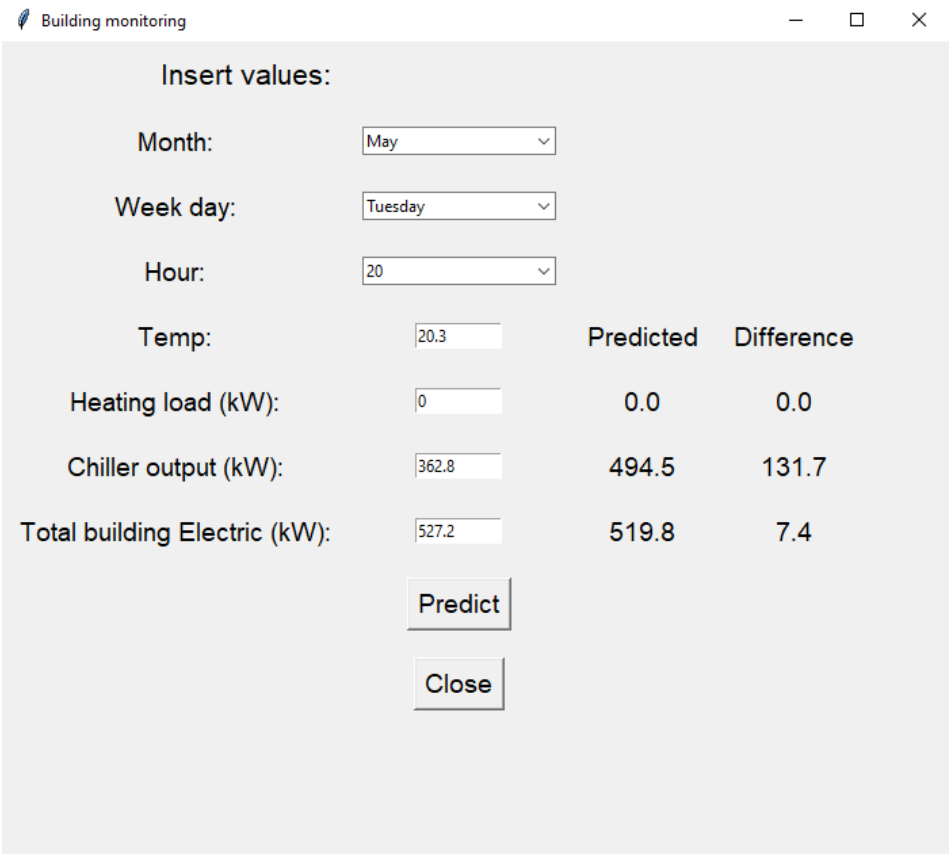
Powyższe wykresy przedstawiają krzywe funkcji MAE i funkcji strat dla danych nieprzeskalowanych i skalowanych. Niezależnie od skalowania danych treningowych proces uczenia przy obydwu modelach wygląda podobnie, różnice pojawiają się przy głębszej analizie wykresów funkcji strat na zbiorach walidacyjnych. Do 75 epoki wartości funkcji strat dla zbiorów treningowych i walidacyjnych w obydwu modelach maleją, w modelu wykorzystującym dane nieprzeskalowane krzywe osiągają stałe wartości do końca treningu, w

przypadku modelu wykorzystującego dane skalowane, wartość funkcji strat na zbiorze treningowym maleje, a na zbiorze walidacyjnym zaczyna rosnąć, co może sugerować problem z overfittingiem (przeuczeniem). Ze względu na mniejszy błąd predykcji do aplikacji został wykorzystany model zoptymalizowany i wytrenowany na danych nieprzeskalowanych.

Finalnie wytrenowane modele zostały zapisane do plików z rozszerzeniem .keras. Dzięki temu aplikację prezentującą możliwości modeli można zasilić gotowymi wytrenowanymi modelami. Pełny proces preprocessingu został danych, dobierania hiperparametrów modeli, tworzenia modeli oraz treningu, został przedstawiony w osobnych jupyter notebookach (cooling.ipynb, heating.ipynb, total_building.ipynb).

6.4 Aplikacja

Do zaprezentowania możliwości predykcji wytrenowanych modeli została utworzona prosta aplikacja okienkowa.



Opis działania

Aplikacja pozwala użytkownikowi podać wartości obecnego całkowitego zapotrzebowania budynku na energię, zapotrzebowania systemów ogrzewania i chłodzenia. Dodatkowo użytkownik podaje dane pogodowe: temperatura zewnętrzna budynku oraz dane dotyczące daty: miesiąc, godzina i dzień tygodnia. Na podstawie danych wejściowych aplikacja wykonuje predykcję wykorzystując wcześniej wytrenowane modele. Finalnie pokazuje wartości przewidziane oraz różnicę bezwzględną pomiędzy nimi a rzeczywistym stanem zapotrzebowania.

6.5 Narzędzia i biblioteki

Preprocessing danych, utworzenie modeli, wyszukanie optymalnych hiperparametrów oraz trenowanie modeli zostały wykonane środowisku Jupyternotebook w wersji 6.4.5. Aplikacja została zaimplementowana całkowicie przy użyciu języka programowania Python. Środowiskiem implementacji został Pycharm Professional 2022.1.1. Aplikacja korzysta z interpretera Python 3.10.

Biblioteki które wykorzystano:

Tensorflow (wersja 2.17.0)

Tensorflow jest to jedna z najczęściej wykorzystywanych bibliotek wykorzystywanych w dziedzinie sztucznej inteligencji. Umożliwia implementacje, trenowanie i testowanie algorytmów uczenia maszynowego, w szczególności uczenia głębokiego.

Matplotlib (wersja 3.9.2)

Matplotlib jest biblioteką pozwalającą na tworzenie wykresów oraz wizualizacji danych w języku Python.

Tk (wersja 8.6.14)

Tk jest to zestaw narzędzi pozwalający tworzyć graficzne interfejsy użytkownika (GUI). Często wykorzystywany do tworzenia aplikacji okienkowych w języku Python.

NumPy (wersja 1.26.4)

NumPy to często wykorzystywana biblioteka w języku Python. Przede wszystkim dodaje narzędzia wspierające pracę z danymi numerycznymi przechowywanymi w wielowymiarowych macierzach, tablicach czy też wektorach. Pozwala na efektywną manipulację danymi i wykonywanie skomplikowanych operacji matematycznych na dużych zbiorach.

Scikit-learn (wersja 1.5.1)

Scikit-learn to potężna biblioteka w języku Python pozwalająca na implementację elementów machine learning w programie. Bazuje na bibliotekach takich jak Numpy, matplotlib czy SciPy. Pozwala na tworzenie, testowanie oraz wdrażanie modeli uczenia maszynowego, przetwarzania danych czy też strojenia hiper parametrów. Jej głównymi zaletami są wszechstronność, wydajność i prostota w użyciu.

Pandas (wersja 2.2.2)

Pandas to popularna biblioteka w języku Python służąca do analizy i manipulacji danymi tabelarycznymi. Bazuje na bibliotece NumPy. Najczęściej wykorzystywanym jej elementem jest struktura danych Dataframe, w budowie i funkcjonowaniu przypominająca arkusz kalkulacyjny. Biblioteka szczególnie przydatna do przechowywania i przygotowywania danych wykorzystywanych w uczeniu maszynowym.

Keras (wersja 3.5.0)

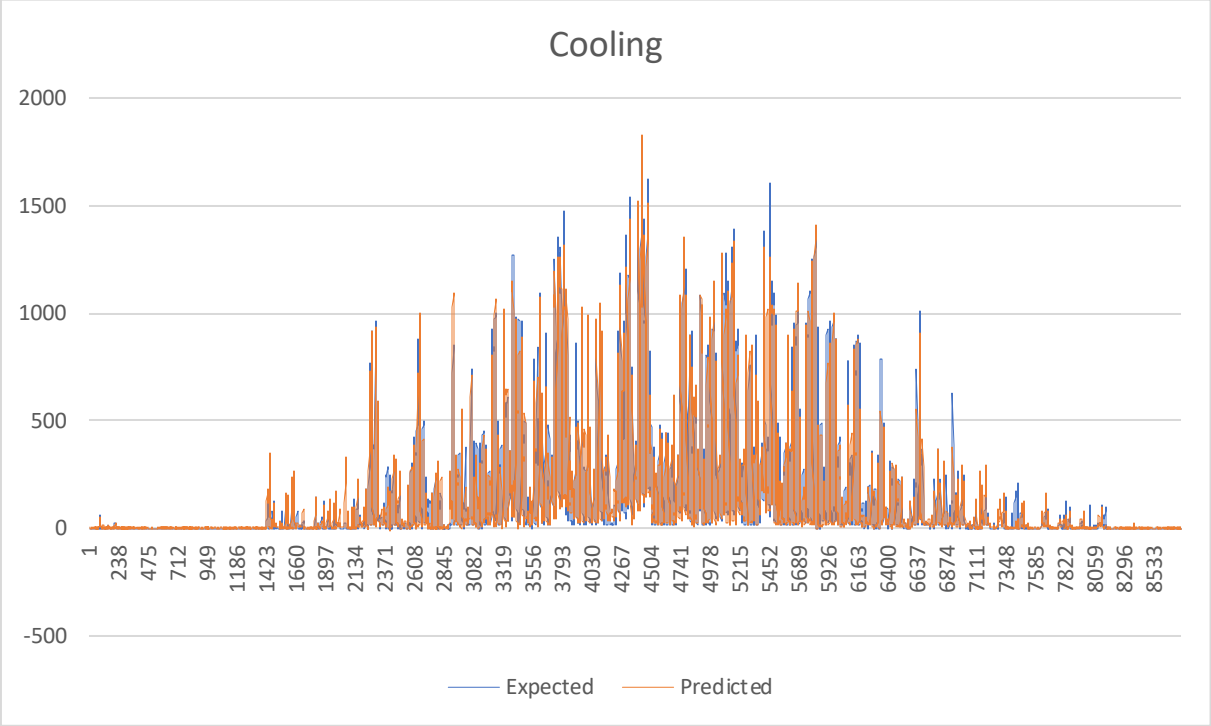
Keras to potężna biblioteka w języku Python pozwalająca na tworzenie, trenowanie, optymalizacja modeli głębokiego uczenia. Ceniona za prostotę, elastyczność i intuicyjność.

7 Analiza wyników

Do analizy wyników zostały wykorzystane predykcje wykonane na podstawie danych z 2024 roku.

7.1 Cooling

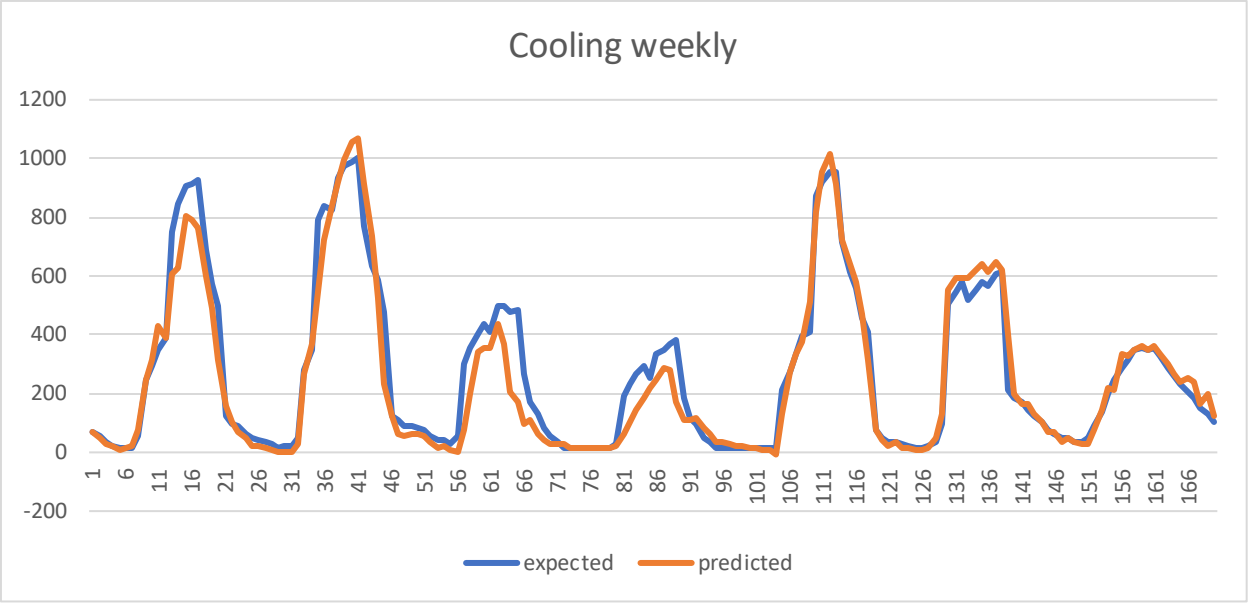
Analizę wyników modelu wytrenowanego do przewidywania zapotrzebowania budynku na chłód warto zacząć porównania wyników w zakresie całorocznym.



Rysunek 24: Wynik predykcji całorocznej dla modelu cooling

Wykres ilustruje wartości zapotrzebowania na energię do chłodzenia (zaznaczona niebieską linią) oraz wartości przewidzianych przez model (pomarańczowa linia). Jak widać model wręcz idealnie przewiduje wartości zużycia w początkowych i końcowych miesiącach roku. W środkowych miesiącach, kiedy zapotrzebowanie jest większe, pojawiają się większe wahania, lecz ogólny zarys zmian model przewiduje w dobry sposób.

Kolejnym aspektem na który warto zwrócić uwagę jest wartość bezwzględna różnicy z sum wartości rzeczywistych oraz przewidzianych. Łączna suma wartości rzeczywistych wynosi 1164525 kW, przewidzianych zaś jest równa 1137495 kw. Co oznacza, że model w skali roku pomylił się o 27030 kW, co stanowi około 2,32% łącznego rzeczywistego zapotrzebowania rocznego.

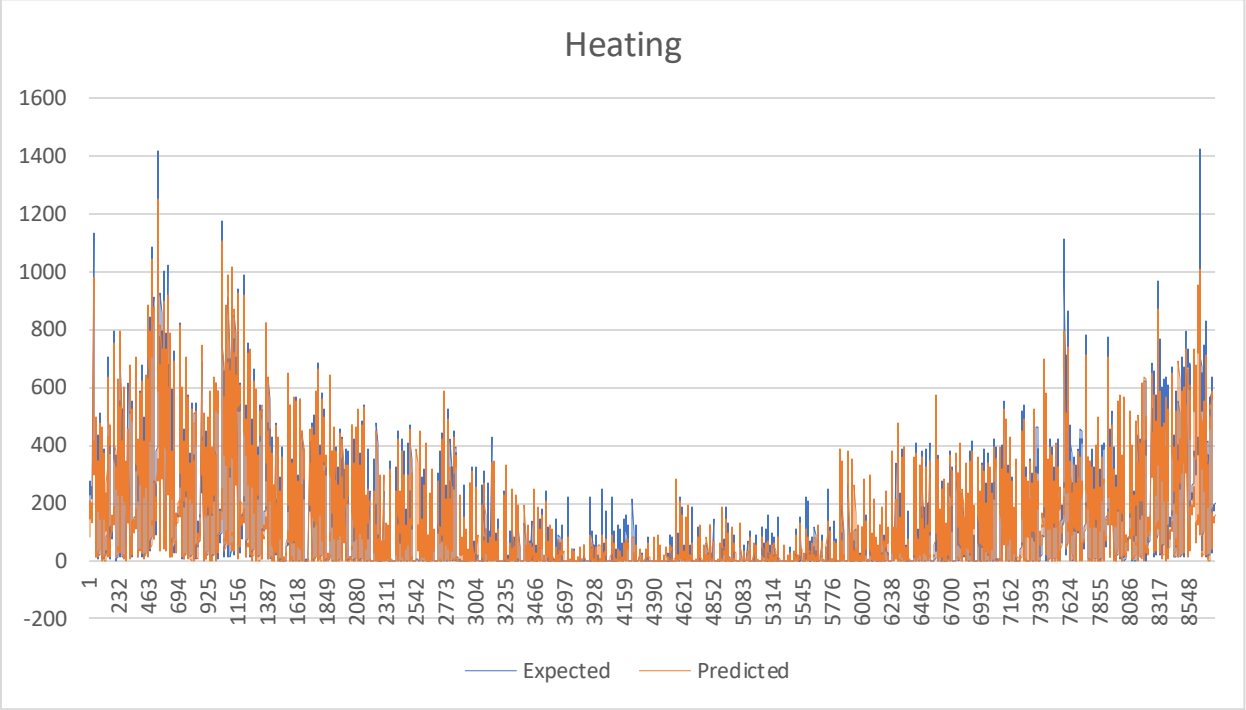


Rysunek 25: Wykres predykcji tygodniowej dla modelu cooling

Ostatnim wykresem który warto jest przeanalizować, jest wykres przedstawiający wartości przewidziane i oczekiwane w zakresie tygodniowym. Zakresem tego wykresu jest tydzień od 15.05.2024r. (poniedziałek) do 21.05.2024r. (piątek). Jak widać model bardzo dobrze zauważył charakter poszczególnych dni w tygodniu i w zadowalający sposób przewidział wartości zapotrzebowania w poszczególnych godzinach.

7.2 Heating

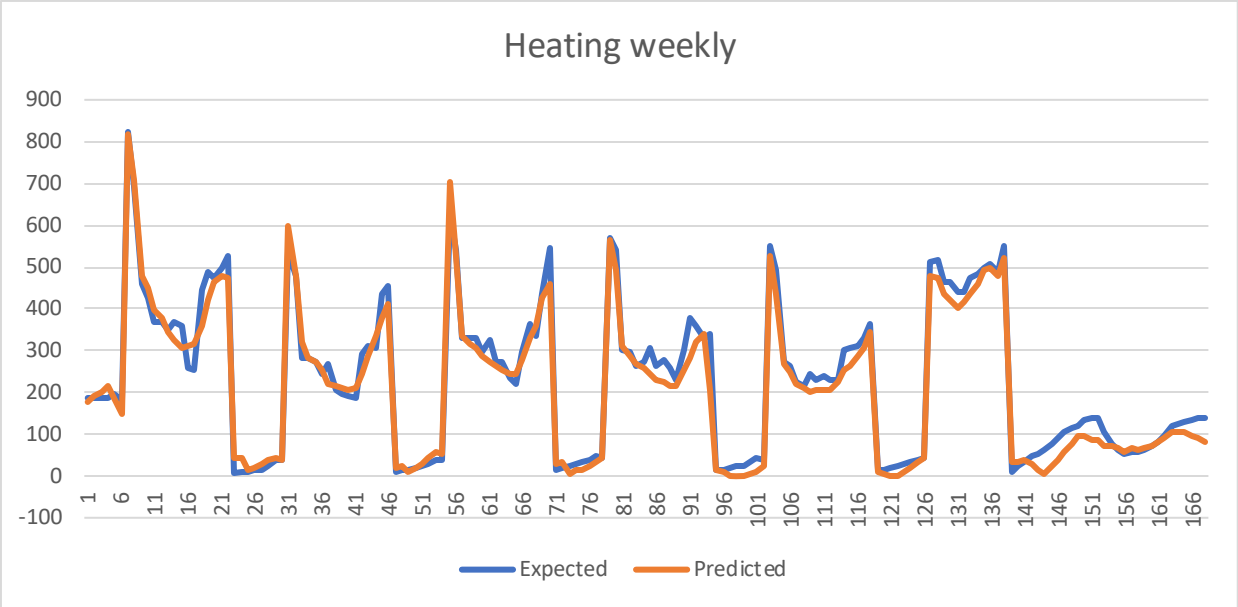
Analizę wyników modelu heating warto rozpocząć, tak samo jak w przypadku modelu cooling, od wykresu przedstawiającego zapotrzebowanie na ciepło rzeczywiste i przewidziane w skali roku.



Rysunek 26: Wynik predykcji całorocznej dla modelu heating

Na powyższym wykresie zaprezentowano wartości oczekiwane niebieską linią oraz wartości przewidziane pomarańczową. Jak widać model dość dobrze przewidywał zapotrzebowanie na energię do ogrzewania.

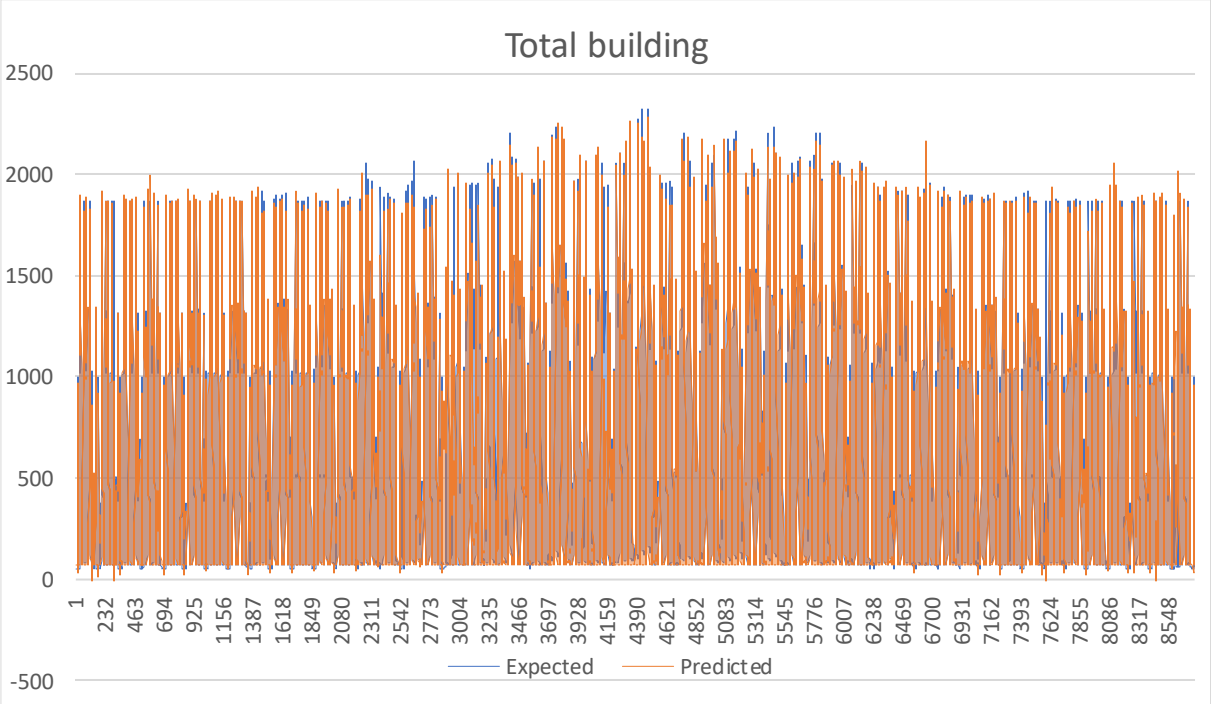
Tak samo jak w przypadku chłodzenia, warto zwrócić uwagę na wartość sumy bezwzględnych błędów w skali roku. Suma wartości oczekiwanych w skali roku wynosiła 994759, suma wartości predykcji modelu wynosiła 940386. Oznacza to że średni błąd bezwzględny w skali roku wynosi 18,29%. Jest to wynik gorszy niż w przypadku modelu przewidującego zapotrzebowanie energię potrzebną do chłodzenia, ale dalej jest zadowalający.



Rysunek 27: Wykres predykcji tygodniowej dla modelu heating

Powyższy wykres przedstawia rzeczywiste zapotrzebowanie energetyczne na ogrzewanie oraz przewidziane. Zakres danych obejmuje dni od 30.01.2024r do 6.02.2024r. Jak widać model dobrze przewiduje wartości zużycia w poszczególnych dniach.

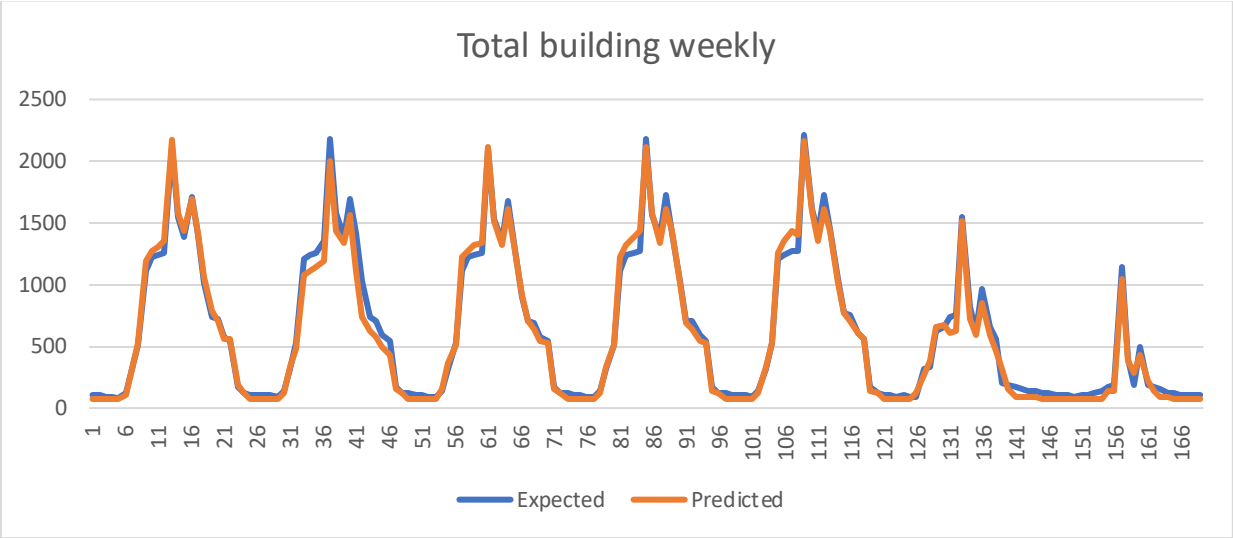
7.3 Total building



Rysunek 28: Wynik predykcji całorocznej dla modelu total building

Powyższy wykres przedstawia całkowite zapotrzebowanie budynku na energię elektryczną w skali roku (oznaczony niebieskimi liniami). Dodatkowo zostały nałożone wartości przewidziane przez model (pomarańczowe linie). Analizując model dość dobrze przewidywał wartości zużycia.

Tak samo jak w poprzednich przypadkach, warto zwrócić uwagę na wartość sumy bezwzględnych błędów w skali roku. Suma wartości oczekiwanych w skali roku wynosiła 4538015, suma wartości predykcji modelu wynosiła 4532788. Oznacza to że średni błąd bezwzględny w skali roku wynosi 0,11%. Jest to wynik bardzo dobry, pozwalający na wykorzystanie modelu do predykcji całkowitego zużycia energii na następne lata, przy wykorzystaniu danych z prognoz pogodowych.



Rysunek 29:Wykres predykcji dla modelu total building w zakresie tygodniowym

Na powyższym wykresie przedstawiono wartości rzeczywiste i przewidywane przez model całkowitego zużycia energii elektrycznej budynku w zakresie tygodniowym. Jak widać jakość predykcji modelu jest na wysokim poziomie. Model dobrze odwzorowuje charakter poszczególnych dni tygodnia.

Analiza podsumowująca

Analizę zakończono na przeanalizowaniu wykresów błędów bezwzględnych z podziałem na miesiące.

Miesiąc	Średnia z różnicy cooling	Średnia z różnicy heating	Średnia z różnicy total building
Styczeń	1,1	42,5	42,0
Luty	0,9	29,5	22,5
Marzec	10,9	18,5	23,3
Kwiecień	25,8	20,0	44,4
Maj	52,2	13,9	84,4
Czerwiec	54,2	9,5	55,8
Lipiec	42,1	7,1	35,0
Sierpień	59,0	9,1	58,0
Wrzesień	36,1	11,9	28,9
Październik	19,0	20,5	22,9
Listopad	10,7	46,7	62,4
Grudzień	3,1	61,6	70,1
Średnia całkowita	**26,41410123**	**24,21467912**	**45,9685031**

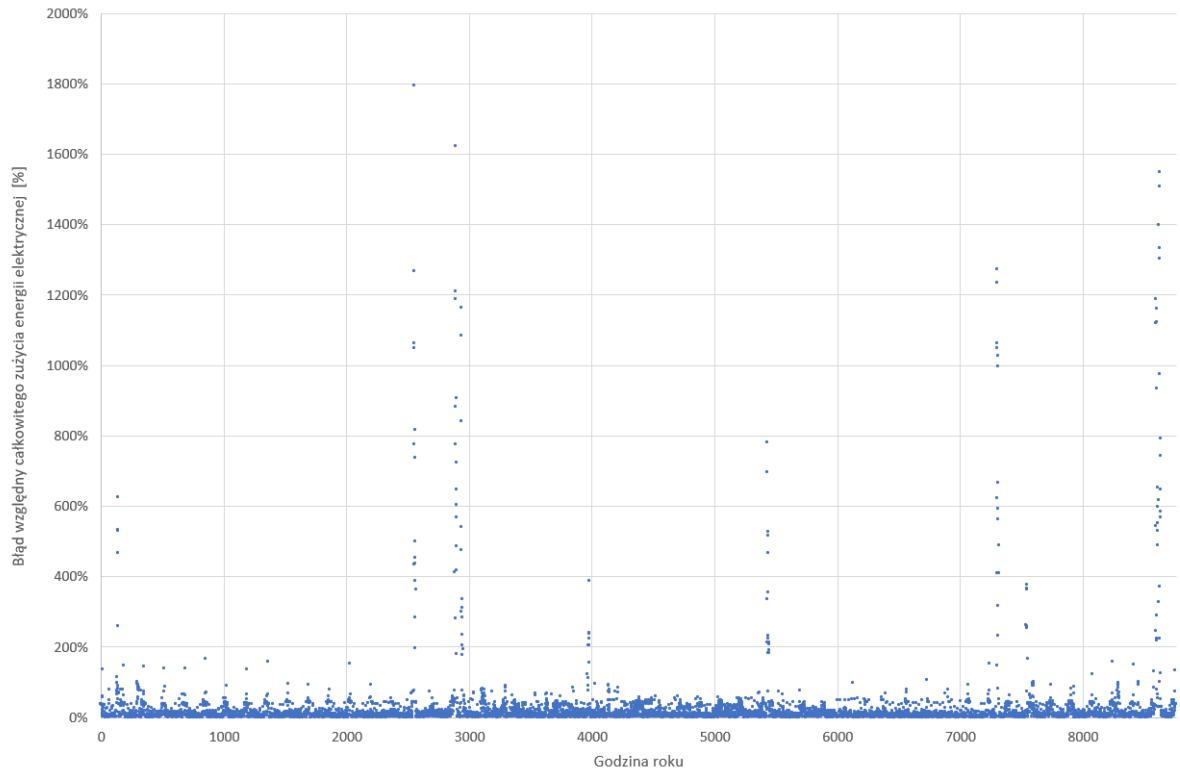
Rysunek 30: Zestawienie błędów modeli

Na powyższym zestawieniu widać jak rozkładał się średni błąd pomiędzy wartością rzeczywistą, a wartością przewidzianą. Analizując wykresy błędów modeli cooling i heating można zauważyć, że modele bardzo dobrze przewidują sytuacje, kiedy systemy im odpowiadające powinny być wyłączone tj. dla heatingu drugi i trzeci kwartał roku, a dla coolingu pierwszy i czwarty. W pozostałych kwartałach wartość błędu rośnie, ale wówczas zapotrzebowanie również rośnie, więc błąd procentowy nie jest duży.

8 Dyskusja

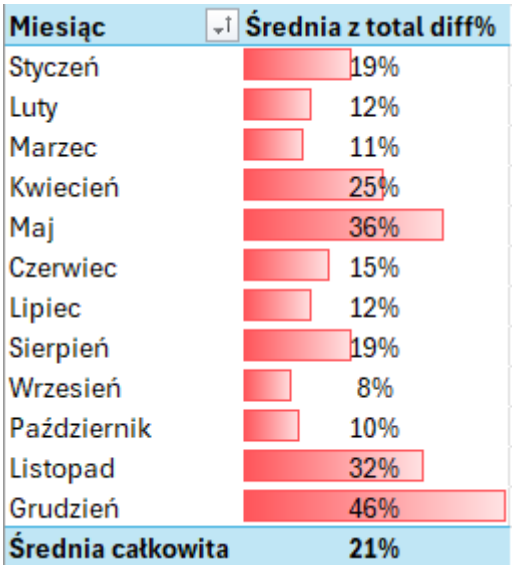
Pomimo dobrych rezultatów predykcji poszczególnych modeli, implementacja sieci neuronowych w istniejących budynkach niesie szereg potencjalnych problemów. W odróżnieniu do wygenerowanych danych wykorzystanych w powyższej pracy, dane odczytywane z systemów w istniejących budynkach mogą być dużo gorszej jakości. Przy tak dużych systemach występuje prawdopodobieństwo pojawienia się szumów w danych, braku kalibracji poszczególnych czujników, a także awarii. Wszystkie te cechy wpływają na problem uzyskania wiarygodnego, pełnego i spójnego zbioru danych z całego roku, który ma wpływ na jakość wytrenowanych modeli.

Tradycyjnie modele oceniane są na podstawie błędu względnego, niemniej przy jego określaniu należy dokonać szczegółowej analizy. W omawianym przypadku średni błąd względny przy wyznaczaniu całkowitego zużycia energii elektrycznej wynosił 21%. Szczegółowa analiza wskazała na znacząco odbiegające wartości w charakterystycznym przypadku. Poniższy wykres przedstawia wartość błędu względnego w poszczególnych godzinach roku:



Rysunek 31: Błąd względny całkowitego zużycia energii w ciągu roku

Na wykresie wskazano na pojedyncze dni których błąd osiągnął wartości 1800%. Analiza wykazała, że były to dni świąteczne, a zatem budynek funkcjonował w trybie weekendowym. Model nie mając tej informacji oszacował wynik jak dla typowego dnia roboczego. Potwierdzeniem znaczenia dni świątecznych może być poniższe zestawienie na którym wyraźnie widać, że typową wartością błędu względnego w miesiącach bez dni świątecznych jest około 8-12%. Miesiące w których występowały święta (styczeń, maj, listopad, grudzień) posiadały zdecydowanie większą wartość błędu względnego.



Miesiąc	Średnia z total diff%
Styczeń	19%
Luty	12%
Marzec	11%
Kwiecień	25%
Maj	36%
Czerwiec	15%
Lipiec	12%
Sierpień	19%
Wrzesień	8%
Październik	10%
Listopad	32%
Grudzień	46%
Średnia całkowita	**21%**

Rysunek 32: Procentowy błąd względny z podziałem na miesiące

Dodatkowym aspektem który nie został wzięty pod uwagę, jest zmienny profil eksploatacji. Jednym z jej przykładów jest zmienność sposobu wykorzystywania powierzchni budynku przez użytkowników, który ma bezpośredni wpływ na zapotrzebowanie energetyczne budynku, a jego dynamiczna zmiana spowoduje pogorszenie się jakości predykcji. Rozwiązywany w powyższej pracy problem dotyczy budynków biurowych, budynki o innych charakterach mogą być gorzej opomiarowane, dlatego przed implementacją modeli sieci neuronowych należy sprawdzić dostępność, jakość i wiarygodność danych. Ze względu na wymagane dane do trenowania sieci neuronowych, zastosowanie modeli jest możliwe dopiero po co najmniej roku funkcjonowania budynku. Warunkiem koniecznym do możliwości wykorzystania modeli sieci neuronowych, jest zainstalowany w budynku system BMS, ale biorąc pod uwagę trendy, jest to dobry kierunek na najbliższą przyszłość.

9 Podsumowanie i wnioski

Celem projektu była ocena możliwości zastosowania modeli sieci neuronowych do predykcji zużycia energii w systemie ogrzewania, systemie klimatyzacji oraz całkowitego zużycia energii w inteligentnych budynkach. Z powodzeniem udało się utworzyć i wytrenować trzy modele sieci neuronowych, które pomimo małej liczby cech, tworzą zadowalające predykcje. Każdy z modeli został wytrenowany i zoptymalizowany do predykcji zużycia energii przez jeden z wyżej wymienionych systemów. Model przewidujący zużycie energii potrzebnej do ogrzewania osiągnął średni błąd względny na poziomie 14%, do chłodzenia 16% a całkowitego zużycia energii w budynku 21%. Wyniki te bez dogłębnej analizy trudno uznać za zadowalające. Dokładniejsza analiza wykazała, że błędy wynikają głównie z powodu ruchomych dni świątecznych, w których błąd względny sięgał 1800%, co znacząco wpływa na sumaryczny średni błąd względny. Dodatkowo zaobserwowano, że w skali roku błąd predykcji modeli jest bardzo niski. Model przewidujący zapotrzebowanie energetyczne całego budynku w skali roku osiągnął błąd na poziomie 0,11%, co z powodzeniem pozwala na wykorzystanie go do analizy zużycia energii w budynku w zakresie całorocznym. Wykorzystane techniki automatyzacji w poszukiwaniu optymalnych hiperparametrów sieci neuronowych pozwalają na uniwersalne wykorzystanie w różnych budynkach inteligentnych bez konieczności wprowadzania wielu zmian w kodzie.

Głównym problemem napotkanym w trakcie implementacji był odpowiedni sposób zamiany danych kategorycznych na liczbowe. Początkowo zastosowano metodę Label encoding, co spowodowało sztuczne utworzenie porządku pomiędzy wartościami cechy Month oraz Week day. Predykcje tworzone przez modele obarczone były bardzo dużym błędem. Problem udało się rozwiązać wykorzystując podejście One-hot encoding. Podczas implementacji nie zaobserwowano problemów związanych z wydajnością obliczeniową i pamięciową modeli. Potencjalnie problemy mogą pojawić się w sytuacji implementacji modeli w systemach złożonych z większej liczby czujników, a co za tym idzie model będzie przyjmował większą liczbę cech. Wyszukiwanie hiperparametrów wraz z trenowaniem każdego z modeli na danych godzinowych zajmował około 25 minut, czas ten jest silnie uzależniony od odstępów czasowych pomiędzy kolejnymi rekordami. Zakładając, że modele trenowane są na zbiorach zawierających dane z całego roku, zmiana przedziałów na półgodzinne, dziesięciominutowe czy też minutowe może znacząco wpłynąć na wydajność zaproponowanego rozwiązania.

Na podstawie uzyskanych wyników, można stwierdzić, że wykorzystanie sieci neuronowych do analizy i optymalizacji funkcjonowania budynków inteligentnych jest uzasadnione i stanowi dobry kierunek w zwiększeniu efektywności energetycznej. Dodatkowo, zaproponowane podejście może przyczynić się do zmniejszenia kosztów operacyjnych związanych z eksploatacją budynków, a także do obniżenia emisji dwutlenku węgla, co jest istotne w kontekście globalnych dążeń do zrównoważonego rozwoju. Wyniki uzyskane w ramach tego projektu mogą również stanowić solidną podstawę do dalszych badań, mających na celu integrację bardziej zaawansowanych technik sztucznej inteligencji oraz ich zastosowanie w innych aspektach zarządzania budynkami inteligentnymi. Potencjalne rozwinięcia obejmują rozszerzenie modeli o dodatkowe cechy, co mogłoby jeszcze bardziej poprawić precyzję predykcji, szczególnie dni wyjątkowych takich jak święta, których obchody nie są ściśle powiązane z datą, oraz pozwolić na ich szersze zastosowanie w innych typach budynków.

10 Bibliografia

[1] Dyrektywa Parlamentu Europejskiego i Rady (UE) 2024/1275 z dnia 24 kwietnia 2024 r. w sprawie charakterystyki energetycznej budynków, Dziennik Urzędowy Unii Europejskiej z 8.5.2024.

[2] Zeyu Wang, Ravi S. Srinivasan A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renewable and Sustainable Energy Reviews 2017; 75; 796-808

[3] Foucquier A, Robert S, Suard F, Stéphan L, Jay A. State of the art in building modelling and energy performances prediction: a review. Renew Sustain Energy Rev 2013;23:272–88.

[4] HAP v6.0: Enhanced and Expanded Weather Data - https://www.shareddocs.com/hvac/docs/1001/Public/03/EXHANGE_7_3.pdf

[5] Zhao H-x, Magoules F. A review on the prediction of building energy consumption. Renew Sustain Energy Rev 2012:3586–92.

[6] Neto AH, Fiorelli FAS. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. Energy Build 2008;40:2169–76.

[7] National Renewable Energy Laboratory, "EnergyPlus," National Renewable Energy Laboratory, Online Available: https://energyplus.net/. [Accessed 15.07.15]

[8] Turhan C, Kazanasmaz T, Uygun IE, Ekmen KE, Akkurt GG. Comparative study of a building energy performance software (KEP-IYTE-ESS) and ANN-based building heat load estimation. Energy Build 2014;85:115–25.

[9] KEP-SDM . Dwelling energy performance-standard assessment procedure. Izmir: Chambers of Mechanical Engineers; 2008.

[10] Zeyu Wang, Ravi S. Srinivasan , “A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models”

[11] Ghezlane Halhoul Merabet, et al “Intelligent building control systems for thermal comfort and energy-efficiency: A systematic review of artificial intelligence-assisted techniques”

[12] Diederik P. Kingma, Jimmy Lei Ba „ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION” https://arxiv.org/pdf/1412.6980

11 Spis rysunków

Rysunek 1: Wykres obszarów analizy w pracach badawczych 13
Rysunek 2: Wykres obszarów analizowanych przy pomocy algorytmów AI 14
Rysunek 3: Wykres przedziałów czasowych wykorzystywanych do analizy 14
Rysunek 4 - Przepływ energii w budynku 26
Rysunek 5 - Informacja o zużyciu energii i emisji w budynku [BMSCare] 28
Rysunek 6 - Godzinowe zapotrzebowanie na ciepło w ciągu roku [kWh] 30
Rysunek 7 - Godzinowe zapotrzebowanie na ciepło w zależności od temperatury zewnętrznej [kWh] 30
Rysunek 8 - Godzinowe zapotrzebowanie na „chłód" w ciągu roku [kWh] 31
Rysunek 9 - Godzinowe zużycie energii elektrycznej na potrzeby chłodzenia w zależności od temperatury zewnętrznej [kWh] 31
Rysunek 10 - Godzinowe zapotrzebowanie na chłodzenie w zależności od temperatury zewnętrznej [kWh] 32
Rysunek 11 - Godzinowe zużycie energii elektrycznej w budynku w ciągu roku [kWh] 32
Rysunek 12 - Godzinowe zapotrzebowanie na energię elektryczną budynku w zależności od temperatury zewnętrznej [kWh] 33
Rysunek 13 - Rozkład temperatury zewnętrznej 34
Rysunek 14 Dane w pliku "Raw_data_2023" 35
Rysunek 15 Brakujące wartości w pliku „Raw_data_2023" 35
Rysunek 16: Zamiana danych "Week day" 37
Rysunek 17: Obszar przeszukiwań przestrzeni hiperparametrów 40
Rysunek 18: Struktura modelu cooling dla danych skalowanych 41
Rysunek 19: Struktura modelu cooling dla danych bez skalowania 41
Rysunek 20: Struktura modelu heating dla danych skalowanych 41
Rysunek 21: Struktura modelu heating dla danych bez skalowania 42
Rysunek 22: Struktura modelu total building dla danych skalowanych 42
Rysunek 23: Struktura modelu total building dla danych bez skalowania 42
Rysunek 24: Wynik predykcji całorocznej dla modelu cooling 50
Rysunek 25: Wykres predykcji tygodniowej dla modelu cooling 51
Rysunek 26: Wynik predykcji całorocznej dla modelu heating 52
Rysunek 27: Wykres predykcji tygodniowej dla modelu heating 53

Rysunek 28: Wynik predykcji całorocznej dla modelu total building 53
Rysunek 29:Wykres predykcji dla modelu total building w zakresie tygodniowym 54
Rysunek 30: Zestawienie błędów modeli 55
Rysunek 31: Błąd względny całkowitego zużycia energii w ciągu roku 56
Rysunek 32: Procentowy błąd względny z podziałem na miesiące 57

Oprogramowanie zostało umieszczone w repozytorium na platformie GitHub: https://github.com/BlazejBartkiewicz/PracaInzynierska