

Streszczenie pracy magisterskiej.....	2
Summary	2
Wstęp	3
1. Pojęcia związane z pracą.....	3
1.1 Pojęcie domu inteligentnego.....	3
1.1.1 Historia domów inteligentnych.....	5
1.1.2 Cechy inteligentnego budynku	5
1.2 Uczenie maszynowe	5
1.2.1 Definicja uczenia maszynowe	6
1.2.2 Historia uczenia maszynowego	6
1.2.3 Rodzaje algorytmów uczenia maszynowego.....	6
2. Część projektowa.	14
2.1 Przedmiot opracowania.....	14
2.2 Charakterystyka obiektu	14
2.3 Założenia projektowe.....	14
2.4 Opis zastosowanych rozwiązań.	15
2.4.1 Pobieranie pogody	15
2.4.2 Wykorzystanie Sztucznej Sieci Neuronowej (SSN) w pracy	15
2.4.3 Ważona średnia krocząca	16
2.4.4 Obliczenie obciążenia cieplnego dla budynku.	16
2.4.5 Obliczanie czasu nagrzania domu	20
2.5.3 Połączenie z urządzeniem grzewczym	20
3 Opis badań.....	23
3.1 Dane testowe dla procesu wyliczania godziny powrotu do domu.....	23
3.2 Sieć neuronowa.....	27
3.3 Liniowa regresja	27
3.4 Ważona średnia krocząca.....	27
4 Wyniki badań	28
5. Wnioski	30
6. Literatura	31
7. Spis tabel	32
8. Spis rysunków	32

Streszczenie pracy magisterskiej

Głównym celem pracy było stworzenie systemu sterowania ogrzewaniem dla użytkownika budynku inteligentnego. System samouczący wyznacza godziny powrotu lokatora do domu oraz dostosowuje czas uruchomienia kotła do przewidywanej godziny powrotu mieszkańca do domu. Aplikację napisano w języku programowania C#. Nacisk położono na porównanie trzech algorytmów samouczących: sztucznej sieci neuronowej, regresji liniowej oraz ważonej średniej kroczącej. Opracowanie podzielono na trzy części. W pierwszej wyjaśniono podstawowe pojęcia o domu inteligentnym, uczeniu maszynowym i stosowanych algorytmach w oparciu o literaturę, a także krótki rys historyczny. Opisano wady i zalety domów inteligentnych. Kolejny rozdział poświęcono opisowi części projektowej (opis techniczny, przedmiot opracowania, charakterystykę obiektu, założenia projektowe oraz opis zastosowanych rozwiązań). W kolejnych rozdziałach przedstawiono przeprowadzone badania i ich wyniki. Badania zostały przeprowadzone przez okres 2 tygodni w okresie grzewczym. Podczas badań porównano wartości predykowane z algorytmów z rzeczywistymi wartościami powrotu mieszkańca do domu. Wyniki badań przedstawiono w tabelach. Przy obliczeniu zapotrzebowania ciepłego dla budynku dostosowano się do procedur i wzorów wykazanych w polskiej normie PN-EN 12831:2006 „Instalacje ogrzewcze w budynkach. Metoda obliczania projektowego obciążenia ciepłego”. Norma szczegółowo opisuje wzory i wskazania przy obliczaniu i przyjmowaniu współczynników korekcyjnych. Założenia projektowe dla systemu sterowania ogrzewaniem to: sterowanie ogrzewaniem za pomocą wbudowanego sterowania kotła grzewczego, wbudowana automatyka kotła realizująca funkcję przeciwwamrożeniową, zaprojektowana aplikacja predykująca godziny powrotu właściciela do domu oraz obliczanie czasu nagrzewania i uruchamiania kotła o odpowiedniej godzinie.

Summary

The main objective of the work was to create a system of heating control for the user of an intelligent building. Self-training system predicts hours return tenant home and adjust the run time of the boiler to the estimated time of return. The application is written in the programming language C#. Emphasis on the comparison of three self-learning algorithms: artificial neural network, linear regression and weighted moving average. The development is divided into three parts. The first explains basic concepts about smart home, machine learning, applied algorithms based on literature and a brief history. Describes the advantages and disadvantages of smart homes. The next chapter is devoted to the description of the design part (technical description, subject of development, the characteristics of the object, design assumptions and a description of the solutions). In the following chapters are studies carried out and their results. Studies have been carried out for a period of 2 weeks in the heating period. During the studies compares the values of the predictions with actual values of the return home. Results of the studies are presented in tables. The calculation of the heat demand for the building has been adapted to procedures and patterns shown in the Polish standard PN-EN 12831:2006 „Instalacje ogrzewcze w budynkach. Metoda obliczania projektowego obciążenia ciepłego”. The standard describes in details the patterns and indications in the calculations and the adoption of the correction factors. Design assumptions for heating control: control of heating via boiler control, built-in automatic boiler that provides the antifreeze function, designed the application predicates hours to return the owner of the home and the calculation of heating time and run the boiler at the right time.

Wstęp

Świat i jego technologie, choć poprawiły nam jakość życia, to jednak ciągły pęd za pracą i pieniądzem zmusiły naukę, do stworzenia systemów, które ułatwią życie, a przede wszystkim sprostają naszym coraz śmielszym oczekiwaniom. Jeśli kiedyś otwieranie bram automatycznie było czymś nadzwyczajnym, dziś staje się standardem w większości domów. Kiedy domy inteligentne zaczynały się pojawiać, zainstalowanie takich systemów było niezwykle kosztowne. W chwili obecnej firmy oferują wykonawcom i inwestorom przystępniejsze ceny, co powoduje, że znaczna liczba właścicieli domków jednorodzinnych coraz częściej decyduje się na zastosowanie rozwiązań, które nie tylko podniosą komfort życia, ale także zapewnią bezpieczeństwo oraz znacznie obniżą koszty utrzymania. Dlatego też podjąłem próbę stworzenia systemu, który pozwoli sterować urządzeniami odpowiadającymi za utrzymanie właściwej temperatury pomieszczeń i dostosować ją do potrzeb użytkowników domu. Niniejszą pracę podzieliłem na 4 rozdziały. W pierwszym rozdziale odniosłem się do źródeł, gdzie w oparciu o naukowe definicje wyjaśniam najważniejsze pojęcia związane z pracą np. pojęcie domu inteligentnego, cechy takiego domu, uczenie maszynowe. Przedstawiłem także podział algorytmów uczenia maszynowego ze względu na sposób działania. W tej części przedstawiłem także krótki rys historyczny domów inteligentnych i uczenia maszynowego, jak również cechy budynku inteligentnego. Podjąłem także temat zagrożeń i wad domów inteligentnych. Opisałem także rodzaje algorytmów uczenia maszynowego.

W rozdziale drugim opisałem system, który napisałem w języku programowania C#. Napisany program ma przewidywać godzinę powrotu właściciela do domu i dostosowywać do niego prace urządzeń grzewczych. Całość zebrałem w opisie technicznym, a w nim m. innymi: szczegóły dotyczące obiektu symulacji, założenia programowe, opis zastosowanych rozwiązań, pobieranie danych pogodowych, obliczanie czasu nagrzania domu, jak również połączenie z urządzeniami grzewczymi i komunikację z nimi. Przedstawiłem także zastosowane biblioteki oraz wybrane algorytmy.

W kolejnym rozdziale przedstawiłem organizację i przebieg badań. Rozdział zawiera również wyniki, a całość podsumowałem wnioskami wynikającymi z przeprowadzonej analizy.

1. Pojęcia związane z pracą

1.1 Pojęcie domu inteligentnego

„Inteligentny budynek to zespół jego możliwości technicznych pozwalających na podstawie stanów różnych czynników wewnętrznych i zewnętrznych odpowiednio reagować, w celu zapewnienia jego użytkownikom komfortowych warunków życia i wysokiego poziomu bezpieczeństwa” [6.12]

Intelligent Building Institute mówi, że inteligentny budynek jest budynkiem, który integruje różne systemy, aby skutecznie, w sposób skoordynowany zarządzać zasobami w celu zapewnienia jak najlepszego funkcjonowania jego użytkowników, maksymalizować oszczędności w zakresie inwestycji i kosztów operacyjnych oraz umożliwiać maksymalną elastyczność. [6.11]

System zarządzania budynkiem (ang. *BMS – Building Management Systems*) znajduje zastosowanie w budynkach mieszkalnych, biurach, zakładach przemysłowych. Można zainstalować tam wiele systemów i połączyć je w całość. Oczywiście spełnia także funkcje monitorowania i kontroli. Optymalizacja i dostosowywanie do indywidualnych potrzeb

użytkowników, jak również raportowanie działania urządzeń i całego systemu, to zadania budynku inteligentnego.

Krótko przedstawię kilka z systemów, które mają zastosowanie w domach jednorodzinnych:

- Sterowanie oświetleniem wewnętrznym i zewnętrznym uwzględniając obecność mieszkańców, reagowanie na ruch w pomieszczeniach, jak również natężenie światła w wybranych miejscach.
- Sterowanie ogrzewaniem wybranych pomieszczeń, które pozwala podgrzać lub schłodzić powietrze tylko w wybranym pomieszczeniu. Zadaniem systemu jest reakcja na ruch (wejście lub wyjście) osoby znajdującej się w pomieszczeniu.
- Sterowanie wentylacją, klimatyzacją i filtracją, pamiętając o zawartości dwutlenku węgla i wilgotności (parametry jakości powietrza).
- Symulacja obecności użytkowników w domu np. w celach bezpieczeństwa podczas urlopu właściciela. Polega ona np. na włączaniu i wyłączaniu oświetlenia w wybranych pomieszczeniach. Można też zastosować moduł, dzięki któremu system odtwarza dźwięki nagrane przez domowników w czasie ich nieobecności w domu. Można tu wykorzystać inteligentny głośnik, jak również połączone z Internetem urządzenia (telefon, TV).
- System alarmowy i monitoring. Działa za pomocą zainstalowanych czujników np. detektor ruchu. System uruchamia się, gdy zostanie stłuczona szyba, otworzy się drzwi lub np. przekroczy określoną linię. Automatycznie zostanie uruchomiony system sygnałów dźwiękowych, powiadomiona zostanie policja lub służby ochrony. Każde opuszczenie domu, zapewnia nam spokój, gdyż można zaprogramować sterowanie tak, by dom sprawiał wrażenie zamieszkałego. Każdy incydent powoduje powiadomienie. Zatem na odległość można monitorować sytuację w domu.
- System przeciwpożarowy ostrzega mieszkańców przed pożarem i zatruciem CO₂. Czujniki dymu i temperatury uruchamiają np. tryskacze lub zraszacze. Ponadto odcinają dopływ gazu oraz komunikują mieszkańca o sposobie ewakuacji.
- System kontroli dostępu, to system stosowany przede wszystkim w biurach, zakładach pracy czy instytucjach. Jest to przydzielanie dostępu wyznaczonym osobom do konkretnych pomieszczeń. Naruszenie tego przydziału spowoduje, że drzwi nie zostaną otwarte. Dodatkowo sieć kamer rejestruje, kto przekracza lub próbuje bezskutecznie przekroczyć zabezpieczone drzwi.
- System pogodowy jest wyposażony w różne czujniki pogodowe. Jeśli warunki atmosferyczne pogorszą się (obniży się znacznie temperatura, zacznie padać deszcz, pojawi się burza) system zamyka okna lub przechodzi na oddzielne zasilanie w przypadku wyłączeń sieci z powodu wyładowań atmosferycznych. Dużą zaletą takiego systemu jest to, że czynniki zewnętrzne (atmosferyczne) niezależnie od pogody i od pory roku nie mają wpływu na warunki w domu. Temperatura jest zawsze optymalna uwzględniając wymagania wszystkich domowników. Oczywiście systemy umożliwiają także ustawienie temperatury wody i podgrzewanie jej w określonych godzinach. Niesie to za sobą efektywne zarządzanie kosztami ogrzewania.

Dzięki tym elementom, zarządzaniu wszystkimi połączonymi ze sobą urządzeniami w domu, można sterować obiektem z bliska lub z daleka. Sterowanie to jest bardzo proste i przyjazne w obsłudze. Zapewnia wygodę życia i bezpieczeństwo.

Jednak oprócz wygody życia, obniżenia kosztów eksploatacji pojawiają się także zagrożenia. Wszystko związane jest z dostępem do Internetu zarówno kablowym, jak i bezprzewodowym. Dzięki tym możliwościom, możemy kierować systemami znajdując się daleko od domu. W tym momencie dajemy jednak szansę potencjalnym złodziejom, którzy korzystają z usług hakerów. Na dzień dzisiejszy w Polsce nie zgłoszono jeszcze takiego

przypadku, jednak w innych krajach takie zdarzenia nie są czymś nieznanym. Jeśli urządzenia nie mają specjalnych zabezpieczeń, można bardzo łatwo przejąć nad nimi kontrolę. Oczywiście firmy oferujące swoje produkty zapewniają pełen zakres zabezpieczeń, nie możemy jednak zapomnieć i pominąć możliwości hakerów. Mimo, że zdajemy sobie sprawę z zagrożeń wynikających z dostępu do Internetu i połączenia z nim urządzeń, wygodne życie i podniesiony standard przewyższają nasze lęki i obawy.

1.1.1 Historia domów inteligentnych

Po raz pierwszy budynek inteligentny a zarazem pomysł inteligentnego budynku pojawił się w latach siedemdziesiątych w branży przemysłowej Stanów Zjednoczonych. Zdobyte doświadczenia systemów kontroli zastosowanych w przemyśle i innych dziedzinach pozwoliły stworzyć systemy i podsystemy, które stopniowo zostały przeniesione do sektora prywatnego jako Intelligent Building Systems. Oczywiście początkowo nie było, to tak rozumiane jak w chwili obecnej. Jednak chodziło o to, by zastosować i wykorzystać systemy informatyczne tak, by wspierały jak najlepiej wszystkie działania i procesy w danym pomieszczeniu.

1.1.2 Cechy inteligentnego budynku

Inteligentny budynek to zespół różnych systemów, które zmieniają nasze pojmowanie przy prowadzeniu i zarządzaniu domem. Kiedy moi rodzice podłączali instalację gazową zapoznawali się z instrukcją działania pieca. Już wtedy było sukcesem, gdy po zamianie pieca węglowego na gazowy można było ustawić godzinę włączania pieca lub ogrzewania wody. Kolejnym systemem, który pojawił się było zraszanie trawy. Ale ponownie trzeba było uczyć się ustawień.

Natomiast założeniem inteligentnego domu jest zaprogramować jednocześnie wszystkie urządzenia zgodnie z potrzebami i wymaganiami użytkownika. Urządzenia mają za zadanie np. zapamiętać ulubione ustawienia z czasem parzenia kawy włącznie. Aby uzyskać pełną satysfakcję z użytkowania domu inteligentnego, musi cechować go:

- łatwość w obsłudze,
- możliwość dowolnej zmiany lokalizacji urządzeń sterujących , jak również możliwości łatwego dokonywania zmian i modyfikowania w zależności od potrzeb,
- estetyka wnętrza i całego budynku,
- samodzielna i automatyczna praca i eksploatacja zainstalowanych systemów,
- dbałość o bezpieczeństwo życia i dóbr materialnych,
- niezawodność działania wszystkich urządzeń,
- dbałość o dostosowanie optymalnych dla człowieka warunków (wilgotność, temperatura, oświetlenie),
- ciągły dostęp do Internetu,
- niskie koszty utrzymania domu.

1.2 Uczenie maszynowe

Zadaniem uczenia maszynowego jest uczenie się, jak rozpoznawać nowe pojęcia za pomocą porównań, odkrywanie nowych powiązań w danych oraz przekształcanie odkrytej wiedzy na język jasny i czytelny dla człowieka. W uczeniu maszynowym chodzi o umiejętność skutecznego i wydajnego a przede wszystkim szybkiego wyszukiwania funkcji, która znajdzie pożądane i oczekiwane rozwiązanie.

1.2.1 Definicja uczenia maszynowe

„Mówimy, że maszyna uczy się zadania T w oparciu o doświadczenie E i miarę jakości P, jeśli wraz z przyrostem doświadczenia E poprawia się jakość wykonywanego zadania T mierzona przez miarę P.” [6.7]

„Uczenie maszynowe (ang. machine learning) to zdolność maszyny (systemu) do rozszerzania wiedzy i udoskonalania swoich zachowań na podstawie zdobywanych doświadczeń” [6.6]

1.2.2 Historia uczenia maszynowego

W latach 1952-1962 Arthur Samuel pracując w firmie IBM stworzył i rozwijał program szkolący zawodników grających w szachy. Ten właśnie projekt można uznać za pierwszy przykład uczenia maszynowego. Celem projektu było na podstawie zdobytej wiedzy, automatyczne uczenie się oraz doskonalenie umiejętności maszyny.

Jednak za właściwy punkt zwrotny w tej dziedzinie świat nauki uznał wykreowanie systemu Dendral, który zadebiutował w Stanach Zjednoczonych na Uniwersytecie w Stanford. Został on stworzony przez kilkoosobowy zespół naukowców. Zastosowano w nim algorytm, który miał systematycznie generować możliwe struktury cząsteczek związków organicznych. Efekt pracy był tak nieoczekiwany i epokowy (dokonany przez maszynę) że doczekał się publikacji. W kolejnych latach pojawiły się kolejne systemy:

- 1978 – stworzony przez Bruce G. Buchmana profesora informatyki na Uniwersytecie w Pittsburghu, system META_DENDRAL. Bruce G. Buchman zajmował się systemami opartymi na wiedzy ekspertów medycznych, systemów i biologii obliczeniowej.
- 1997- program Automated Mathematician, który miał za zadanie odkrywanie nowych reguł matematycznych na podstawie hipotez.
- 1981- Eurisko program komputerowy, uznawany za następcę systemu AM. Miał on zastosowanie w grze „Podróżnicy”.

Uczenie maszynowe staje się bardzo popularne w początkach lat 90. Gerald Tesauro pracował nad teoretycznym i praktycznym stosowaniem uczenia. Pracował również nad zastosowaniem uczenia maszynowego przy redukcji wymiarowości, wyszukiwaniu i rozpoznawaniu wirusów komputerowych itp. Jego program TD-Gammon mógł toczyć pojedynki szachowe z mistrzami świata. Na uwagę zasługuje fakt, że algorytm z tego programu wykorzystano w neuronauce.

Pod koniec lat 90-tych uczenie maszynowe wykorzystano w sieci Internet, a co za tym idzie w programach do wyszukiwania informacji w Internecie (Google, Bing, Yahoo). W chwili obecnej uczenie maszynowe stosowane jest praktycznie w każdej dziedzinie życia codziennego.

1.2.3 Rodzaje algorytmów uczenia maszynowego

Algorytmy uczenia maszynowego możemy pogrupować wg dwóch kryteriów:

- Ze względu na styl uczenia.
- Ze względu na sposób działania.

Szczegółowy podział przedstawia poniższy rysunek:



Rysunek 1.1 Podział stylów uczenia maszynowego ze względu na sposób działania [6.2]

Podział algorytmów uczenia maszynowego ze względu na styl uczenia.

Biorąc pod uwagę styl uczenia algorytmy uczenia maszynowego dzielimy na:

- algorytmy nadzorowane,
- algorytmy bez nadzoru,
- algorytmy z częściowym nadzorem.

Algorytmy nadzorowane

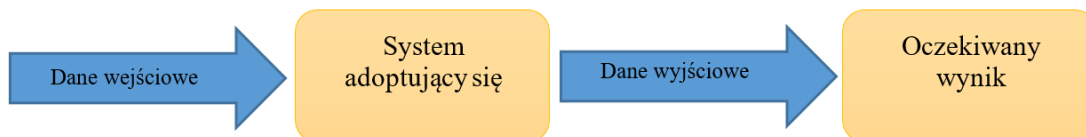
Każdy algorytm, który przewiduje nadzorowanie przez człowieka przy kreowaniu funkcji odtwarzającej wejście do schematu na jego wyjście nazywamy algorytmem nadzorowanym.

W tym przypadku nadzór polega na wykreowaniu pakietu danych uczących (par). Para składa się z wejściowego obiektu szkolącego oraz do tego spodziewaną kontrakcję np. wartość liczbowa. W efekcie system powinien nauczyć się prognozować jaka jest prawidłowa odpowiedź na różne przypadki, nawet takie, których się nie spodziewał. Takie uczenie często stosowane jest w formowaniu procesów technicznych.

Każdy system oparty o algorytmy nadzorowane musi otrzymać dane wejściowe oraz informacje na nie odpowiadające na wyjściu (pary danych).

Najlepiej zobrazuje to poniższy przykład. Wektor X zawiera informacje dotyczące wejścia do systemu oraz wektor Y , w którym zawarte są informacje oczekiwane na wyjściu. System ma się przyswoić sobie funkcję odwzorowującą, by podane informacje z wektora x wytwarzały wartości wektora Y .

W wyniku uczenia z nadzorem otrzymujemy podział na znane klasy. Celem takiego uczenia jest generalizacja.



Rysunek 1.2 Schemat uczenia nadzorowanego.

Podsumowując do realizacji algorytmu nadzorowanego niezbędne są elementy:

- dane wejściowe,
- system adaptujący się,
- dane wyjściowe,
- pożądane odpowiedzi.

Algorytmy bez nadzoru

Algorytm bez nadzoru, to system który uczy się za pomocą tylko podanych danych wejściowych (wektor X), bez zdefiniowania oczekiwanych wyników. Zatem nie nadzorujemy jej proces z zewnątrz. Natomiast spodziewamy się, że system znajdzie ciekawe rozwiązania (wzorze).

Celem tego systemu jest przetworzenie wprowadzonych do systemu tylko informacji wejściowych i wyniku uczenia się nienadzorowanego oczekuje się wykrywanie nieznanych klas, odkrywania ciekawych rozwiązań, czy też uczenie się samorzutne i spontaniczne. Bardzo ważna jest współzależność postępowania systemu ze zmianą struktur. Charakterystycznym przykładem tego uczenia się jest wyznaczenie prawidłowości we wprowadzonych danych wejściowych i określanie kategorii i grupowanie ich w klasy. Stąd pochodzi nazwa klasteryzacja lub grupowanie pojęciowe (conceptual clustering). Algorytmy te uważane są niebagatelny zespół algorytmów uczenia sieci neuronowych.

Algorytmy z częściowym nadzorem

Algorytm z częściowym nadzorem (z angielskiego semi-supervised learning) stosowany jest w przypadku, gdy nie zdefiniowano części danych uczących. Obecnie ten dział rozwija się bardzo energicznie. Jest to spowodowane wielorakim i wszechstronnym zastosowaniem. Jest to związane z dużym kosztem uzyskania danych etykietowanych. W tym przypadku mamy na myśli nie tylko aspekt ekonomiczny, ale i czasowy, ponieważ najczęściej etykietowanie danych odbywa się ręcznie. Inny aspekt przemawiający za wyborem tego algorytmu jest fakt łatwego i tańszego pozyskiwania danych nieetykietowanych. Takie algorytmy wykorzystują niesprecyzowane dane, by w efekcie końcowym generalizować informacje do nowego wzoru (nowej próbki).

Przykładem algorytmu stosowany w uczeniu z częściowym nadzorem jest Model Co-Training. Zaemonstrował go Avrim'a Blum'a oraz Tom'a Mitchell'a w 1998 roku w pracy pod tytułem Combining Labeled and Unlabeled Data with Co-]. Podkreślić należy uznano ją za najlepszą pracę ostatniej dekad. Był to sygnał, że doceniono wagę tej metody i jaki miała wpływ. Dobrze obrazuje to także jak ważna jest to metoda oraz jak bardzo wpłynęła na całą obszar uczenia maszynowego.

Podział algorytmów uczenia maszynowego ze względu na sposób działania.

Uwzględniając metodę funkcjonowania można wymienić kilka algorytmów:

- algorytmy regresyjne,
- drzewa decyzyjne,

- algorytmy Bayesowe,
- algorytmy grupowania,
- sztuczne sieci neuronowe.

Algorytmy regresyjne

Regresja jest metodą, która pozwala na wywnioskować wartości spodziewane. Przykładem wykorzystania jest np. przewidywanie wyniku z losowego doświadczenia. Posiadając wiele zmiennych, którym nie można przypisać miary liczbowej (np. zjawiska o charakterze losowym) można także zastosować algorytmy regresyjne. Można sprecyzować klucz tak, aby możliwe było ich badanie w intrygującym nas problemie. Dobrym przykładem na powyższą definicję jest np. cena domu, oczywiście wchodzą tu różne elementy (ciągłe predyktory i predyktory kategorialne):

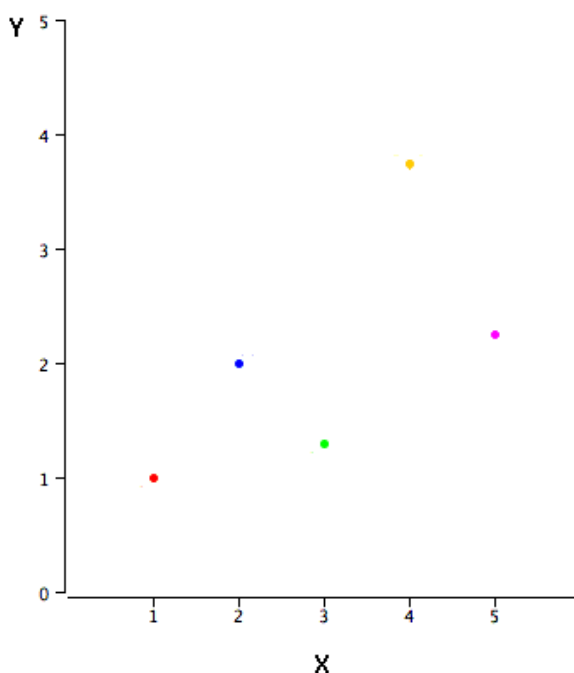
- Powierzchnia lokalu (użytkowa i całkowita)
- Architektura
- Wiek
- lokalizacja

Mając kilka predyktorów, stosujemy regresję wieloraką. Zatem tworzymy równanie liniowe, które da nam poszukiwaną cenę domu. Użytkownik może swobodnie tworzyć równanie składające się z parametrów, które wylicza program.

Regresja liniowa

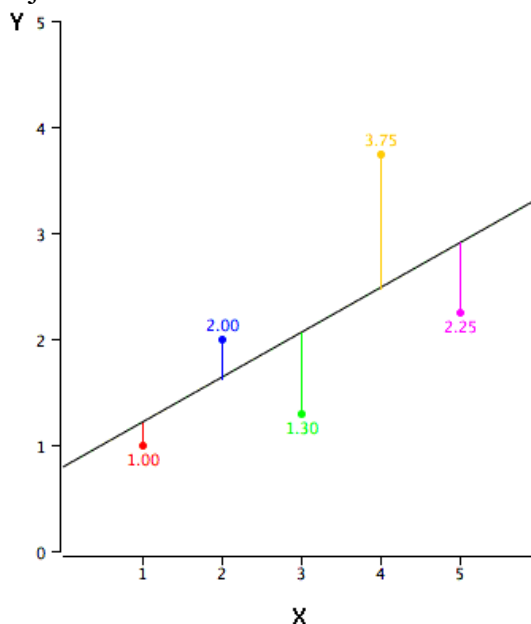
Model regresyjny jest najbardziej powszechną metodą analizy danych statystycznych. Zadaniem tej metody jest ekstrapolacja i przypuszczanie wyników dla zmiennej na podstawie innych zmiennych. W rzeczywistości chodzi tu o określenie, jaką wartość otrzymamy, gdy będziemy znali wartość innej zmiennej. Aby wynik był satysfakcjonujący musimy, wykorzystując analizę regresji stworzyć właściwy model który wyszuka wartości jednej zmiennej w oparciu o inne zmienne.

Działanie regresji liniowej zostało przedstawione na rysunkach nr 3 i 4



Rysunek 1.3 Zbiór danych uczących dla regresji liniowej [6.5]

Regresja liniowa polega na znalezieniu najlepiej pasującej prostej linii przez punkty. Taka linia jest nazywana linią regresji. Czarna linia na rysunku nr 4 jest linią regresji. Składa się ona z przewidywanych wyników Y dla możliwych wartości X. Pionowe kolorowe linie reprezentują wartości błędów dla predykcji.



Rysunek 1.4 Wyliczona linia regresji [6.5]

Drzewa decyzyjne

Typowy algorytm C&RT (Classification and Regression Trees) pojawił się na rynku w 1998 r. i został wprowadzony przez Leo Briemana. Jego celem jest stworzenie algorytmu budowy drzew, który znajduje zbiór logicznych warunków i uporządkowania zgodnie z zaklasyfikowaniem obiektów. Drzewo to jest budowane podczas podziału rekurencyjnego. Oznacza to, że wielokrotnie przeszukuje w przestrzeni zbiorów zadań, by w efekcie końcowym dokonać podziału na maksymalnie dwa podzbiory danych różniących się zmienną zależną.

Drzewa decyzyjne są stosowane przy algorytmie regresji lub klasyfikacji. Jest to nadzorowana metoda uczenia, gdzie nie podajemy parametrów. Metoda ta ma za zadanie stworzyć taki model, który oszacuje wartości zmiennej, za pomocą prostych wzorów.

Warunki wykorzystywania drzew decyzyjnych:

- Mamy dwa momenty otrzymania informacji. Pierwszy - moment dokonywania oceny oraz drugi - moment podejmowania decyzji. Optymalny powinien być moment podejmowania decyzji.
- Efektywność stosowanych metod jest uzależniona od kosztów i możliwości pozyskiwanych informacji. Aby obniżyć koszty wybiera się korzystanie z danych wcześniejszych.
- Zastępczo można zastosować prawdopodobieństwo subiektywne.
- Można podać dodatkową daną, która zwiększy szanse na otrzymanie pożądanego wyniku.

Elementami składowymi każdego drzewa decyzyjnego są: węzeł korzenia, węzły wewnętrzne, oraz węzły wyjściowe czyli „liście”.

Każdy węzeł jest opisany poprzez podzbiór próby uczącej. Poza liśćmi każdy z węzłów dzieli ten zbiór na dwa podzbiory lewy i prawy. Podział jest scharakteryzowany przez regułę bazującą na wyselekcjonowanej zmiennej. Załóżmy że zmienna X jest ciągła - podział wygląda następująco: $X \leq s$ lub $X > s$ dla ustalonego s. Przy sklasyfikowanej zmiennej X podział

wygląda następująco: X należy do S lub X nie należy do S . S oznacza niepusty podzbiór liczb z X . Jeśli okaże się że nie możemy już różnicować danych, to nie kontynuujemy już podziału. Sam moment przzerwania procesu podziału jest ważny w przypadku stosowania drzew regresyjnych oraz klasyfikacyjnych do uzyskania rzetelnych danych. Jeśli nie uda nam się „trafić” w odpowiednim czasie może dojść do „nadmiernego dopasowania”. Zatem należy trafić w punkt czasowy.

Po zakończeniu tworzenia drzewa, wskazane jest oszacować wartość predykcji, wykorzystując je do nowych badań, nie użyte w budowie modelu. Wykorzystując to można dokonać selekcje czyli drzewo prostrze niż pierwowzór tzw. „przycinanie drzewa”. Należy pamiętać, iż mają one porównywalną precyzję predykcijną w odniesieniu do nowych wyników.

Algorytmy Bayesowe

Thomas Bayes brytyjski matematyk i duchowny zapoczątkował dział statystyki, mimo iż jego twierdzenie zostało opublikowane dopiero pośmiertnie. Naiwny klasyfikator bayesowski (metoda uczenia maszynowego oparta na twierdzeniu Bayes’a) jest stosowany przy zagadnieniach sortowania i klasyfikacji. Ma on za zadanie zaklasyfikowanie każdego epizodu do właściwego i zakończonego zbioru klas decyzyjnych. Klasyfikator ten jest dokładny nawet dla dużej ilości danych.

Poniżej przedstawiam twierdzenie Bayes’a

Twierdzenie (reguła Bayesa, wzór Bayesa)

Jeśli B_1, B_2, \dots, B_k tworzą podział przestrzeni S oraz $P(B_i) > 0, i = 1, 2, \dots, k$, to dla dowolnego zdarzenia A o dodatnim prawdopodobieństwie oraz dowolnego zdarzenia B_m spośród zdarzeń $B_1, B_2, \dots, B_k, 1 \leq m \leq k$, zachodzi wzór:

$$P(B_m|A) = \frac{P(A|B_m)P(B_m)}{P(A)} = \frac{P(A|B_m)P(B_m)}{\sum_{j=1}^k P(A|B_j)P(B_j)}$$

Rysunek 1.5 Twierdzenia Bayesa [6.2]

Sieć bayesowska to acykliczny (nie zawierający cykli) graf skierowany, w którym:

- węzły reprezentują zmienne losowe (np. temperaturę jakiegoś źródła, stan pacjenta, cechę obiektu itp.)
- łuki (skierowane) reprezentują zależność typu „zmienna X ma bezpośredni wpływ na zmienną Y ”,
- każdy węzeł X ma stowarzyszoną z nim tablice prawdopodobieństw warunkowych określających wpływ wywierany na X przez jego poprzedników (rodziców) w grafie,
- Zmienne reprezentowane przez węzły przyjmują wartości dyskretne (np. TAK, NIE)

Zatem, aby stworzyć sieć bayesowską musimy zdefiniować zmienne i połączenia między nimi, określić prawdopodobieństwa warunkowe, czyli założenia. Kolejnym etapem jest wprowadzenie danych do sieci i jej aktualizacja a na koniec wyznaczenie następstw.

Algorytmy grupowania

Algorytm grupowania jest to metoda uczenia się bez nadzoru polegająca na podziale zbioru na podzbiory, kategorie, grupy (z angielskiego clustering). Algorytmy grupowania są podzielone na trzy grupy. [6.3] Należą do nich algorytmy poszukiwania ogólnego ekstremum funkcji kryterialnej (polega on na wyznaczeniu wszelkich możliwych segregacji elementów przestrzeni na zakładaną ilość grup), następnie algorytmy hierarchicznego podziału i grupowania oraz algorytmy iteracyjne.

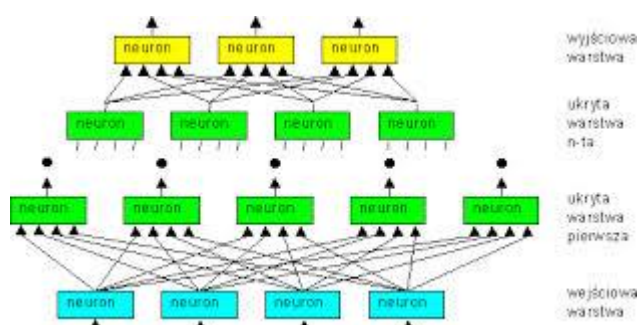
Należy zwrócić uwagę na fakt iż dane powinny być porównywalne do innych elementów w tej samej kategorii. W procesie tworzenia algorytmów grupowania wykorzystujemy dane nieetykietowane, natomiast w algorytmach klasyfikowania dane etykietowane. W pierwszym przypadku podczas grupowania nadaje się etykiety danych, jednak odbywa się to tylko i wyłącznie z wiedzy posiadanej i wynikającej z danych. Algorytmy te mają za zadanie wyszukać wzorce, następnie wyeksponować wspólne właściwości, wyszukać podobne elementy i na końcu zebrać je w odpowiednie grupy znaczeniowe. Algorytmu ma zastosowanie w wielu dziedzinach np. przyrodzie (gatunki zwierząt i roślin), informatyce i statystyce (data mining- zgłębianie danych),

Sieci neuronowe

Prototypem sieci neuronowej jest ludzki mózg, a właściwie 3 mm kora mózgowa, w której znajdują się komórki nerwowe. Ryszard Tadeusiewicz w swojej pracy napisał „Sieć neuronowa jest bardzo uproszczonym modelem mózgu. Składa się ona z dużej liczby (od kilkuset do kilkudziesięciu tysięcy) elementów przetwarzających informacje. Elementy te nazywane są neuronami, chociaż w stosunku do rzeczywistych komórek nerwowych ich funkcje są bardzo uproszczone, by nie powiedzieć- sprymitywizowane. Neurony są powiązane w sieć za pomocą połączeń o parametrach (tak zwanych wagach) modyfikowanych w trakcie tak zwanego procesu uczenia. Topologia połączeń oraz ich parametry stanowią program działania sieci, zaś sygnały pojawiające się na jej wyjściach w odpowiedzi na określne sygnały wejściowe są rozwiązaniami stawianych jej zadań [6.9].

Sztucznymi sieciami neuronowymi (SSN) nazywamy symulatory Neural nets (networks) lub AI (Artificial Intelligence - sztuczna inteligencja), modeli obliczeniowych, które przetwarzają dane. Proces ten odbywa się pseudorównolegle, a celem jest naśladowanie działania biologicznych struktur mózgowych, za pomocą wielu wzajemnie połączonych neuronów. Ich zakres obejmuje wiele dziedzin: fizykę, biologię i matematykę, informatykę oraz medycynę itp. Można je zastosować także przy rozpoznawaniu wzorców (znaków, liter, kształtów, sygnałów mowy), w ekonomii do prognozowania i oceny ryzyka ekonomicznego, a także zmian cen rynkowych lub prognozowania sprzedaży czy też zdolności kredytowej.

Zaletą SSN jest umiejętność uogólniania wiedzy dla nowych informacji, które wcześniej nie były przedstawiane.



Rysunek 1.6 Schemat jednokierunkowej wielowarstwowej sieci neuronowej [6.10].

Większość stworzonych sieci ma konstrukcję warstwową, jednak do procesu uczenia eksponowane są warstwy wejściowe, warstwy ukryte oraz warstwy wyjściowe

Podstawową cechą różniącą SSN od programów realizujących algorytmiczne przetwarzanie informacji jest zdolność generalizacji czyli uogólniania wiedzy dla nowych danych nieznanych wcześniej, czyli nie prezentowanych w trakcie nauki. Określa się to także jako zdolność SSN do aproksymacji wartości funkcji wielu zmiennych w przeciwieństwie do interpolacji możliwej do otrzymania przy przetwarzaniu algorytmicznym. Można to ująć

jeszcze inaczej np. systemy ekspertowe z reguły wymagają zgromadzenia i bieżącego dostępu do całej wiedzy na temat zagadnień, o których będą rozstrzygały. *SSN* wymagają natomiast jednorazowego nauczania, przy czym wykazują one tolerancję na nieciągłości, przypadkowe zaburzenia lub wręcz braki w zbiorze uczącym. Pozwala to na zastosowanie ich tam, gdzie nie da się rozwiązać danego problemu w żaden inny, efektywny sposób. Uczenie *SSN* odbywa się za pomocą przedstawionych informacji, w których model uczący formułuje istotność zespołów. To one mają zapamiętane informacje, które pozwolą rozwikłać dane zagadnienie. Początki *SSN* zauważono w latach 50 ubiegłego stulecia, jednak intensywny rozwój i zastosowanie ich do złożonych aplikacji odnotowano w połowie lat 80. W chwili obecnej sztuczne sieci neuronowe wykorzystuje się do rozwiązywania bardzo skomplikowanych problemów, nie tylko informatycznych czy matematycznych.

Przykłady zastosowań: w medycynie (badania psychiatryczne, analiza badań medycznych), w ekonomii (prognozy giełdowe prognozowanie sprzedaży, prognozy cen, typowanie w grach, wyścigach konnych, analiza i kojarzenie danych), gospodarce i przemyśle (poszukiwanie bogactw naturalnych, planowanie remontów maszyn, sterowanie procesów przemysłowych) czy nawet dobór pracowników. Najczęstsze wykorzystanie: predykcja, klasyfikacja i rozpoznawanie podmiotów gospodarczych, kojarzenie danych, analiza danych, filtracja sygnałów czy optymalizacja. Nadmienić trzeba, że ta dziedzina w dalszym ciągu się rozwija i znajdzie zastosowanie w jeszcze szerszym zakresie.

2. Część projektowa.

2.1 Przedmiot opracowania

Projekt samouczącego się systemu sterowania ogrzewaniem w domu jednorodzinnym. Zaprojektowano system który przewiduje godziny powrotu właściciela do domu oraz zapewnia odpowiednio wczesne uruchamianie ogrzewania w zależności od temperatury zewnętrznej oraz czasu nagrzewania domu. Celem systemu jest zapewnienie komfortu grzewczego użytkownikom domu.

Projekt opracowano w oparciu o:

- Temat pracy
- Środowisko programistyczne Visual Studio 2015
- Język programowania C#
- Bibliotekę Encog
- Literaturę, artykuły, strony internetowe ujęte w rozdziale 6

2.2 Charakterystyka obiektu

Jako obiekt badań przyjęto dom jednorodzinny drewniany zlokalizowany we Wrocławiu, na terenie miejskim, niezacienionym. Dom jest jednokondygnacyjny, ocieplony wełną mineralną do uzyskania współczynników przenikania ciepła na poziomie współczynników zgodnych z Rozporządzeniem Ministra Infrastruktury z dnia 12 kwietnia 2002r. „W sprawie warunków technicznych jakim powinny odpowiadać budynki i ich usytuowanie” wraz z późniejszymi zmianami.

2.3 Założenia projektowe.

Założenia projektowe dla budynku:

- Powierzchnia domu: 100m^2
- Powierzchnia ścian zewnętrznych: 98m^2
- Powierzchnia przegród szklanych (okien): 22m^2
- Powierzchnia dachu: 100m^2
- Powierzchnia podłogi na gruncie 100m^2
- Ściany zewnętrzne drewniane, modrzewiowe +ocieplenie wełną mineralną
- Preferowana temperatura wewnętrzna 21°C
- Budynek jest ogrzewany piecem gazowym kondensacyjnym o mocy 15 kW
- Rodzaj ogrzewania grzejnikowe

Dla uproszczenia obliczeń przyjęto, iż cały obrys budynku stanowi jedną powierzchnie o preferowanej temperaturze wewnętrznej 21°C . Dla powyższych założeń wyznacza straty ciepła dla budynku w zależności od temperatury zewnętrznej oraz czasu nagrzewania budynku.

Założenia projektowe dla systemu sterowania ogrzewaniem:

- Sterowanie ogrzewaniem odbywa się za pomocą wbudowanego sterowania kotła grzewczego.
- Wbudowana automatyka kotła realizuje funkcję przeciwwzamrozeniową.
- Zaprojektowana aplikacja predykuje godziny powrotu właściciela do domu oraz oblicza czas nagrzewania i uruchamia kocioł o odpowiedniej godzinie.
- Do przewidywania godziny powrotu lokatorów do domu zastosowano algorytmy: Sztuczną Sieć Neuronową (Multi-Layer Perceptron), regresję liniową oraz ważoną średnią kroczącą.

- Aktualną pogodę tj. temperaturę aplikacja pobiera z serwisu OpenWeatherService.
- Założono, że sterowanie ogrzewaniem będzie odbywało się za pomocą biblioteki Modbus.

2.4 Opis zastosowanych rozwiązań.

2.4.1 Pobieranie pogody

Pobieranie aktualnej temperatury odbywa się przy pomocy interfejsu Weather API udostępnionego przez serwis OpenWeatherMap. Program pobiera aktualne warunki pogodowe dla Wrocławia w formacie Xml.

```
string we Burl = "http://api.openweathermap.org/data/2.5/forecast/city?id=3081368&APPID=c7cd5e  
a156fa1a32bde78b54d0beaae1&mode=xml&units=metric&cnt=1";
```

Wartość temperatury pobieram z odpowiedniego węzła („temperature”)

```
string szTemp = doc.DocumentElement.SelectSingleNode("forecast").SelectSingleNode("time").Select  
SingleNode("temperature").Attributes["value"].Value;
```

2.4.2 Wykorzystanie Sztucznej Sieci Neuronowej (SSN) w pracy

Sieć neuronowa stara się przewidzieć przyszłe wartości opierając się na teraźniejszych oraz przeszłych danych. Do przewidywania wartości używamy przejściowej sieci neuronowej. Przejściowa sieć neuronowa jest zwykle siecią jednokierunkową lub rekurencyjną.

Przewidująca sieć neuronowa korzysta z danych wejściowych, by przyjąć informacje o aktualnych danych i używa ich do prognozowania przyszłych danych. Stosowane są zatem dwa okna:

- okna przeszłości
- okna przyszłości

Oba okna muszą mieć określoną wielkość. Wielkość okna przeszłości określa, jaka ilość danych jest potrzebna do wykonania predykcji. Wielkość okna przyszłości odpowiada ilości danych, jakie mają zostać przewidziane. Przed przesłaniem do sieci neuronowej dane powinny być znormalizowane. Normalizacja polega na mapowaniu danych do wartości w przedziale 0-1.

Wykorzystany algorytm

Do zbudowania sieci neuronowej skorzystałem z opensourcowej biblioteki Encog, która jest dostępna dla języków m.in. C# i Java.

By wykorzystać przejściową sieć neuronową musimy zainicjalizować obiekt klasy TemporalMLDataSet – jest to klasa, w której będą przechowywane dane użyte do trenowania sieci. Wartości potrzebne do zainicjalizowania obiektu to wielkość okna przeszłości oraz wielkość okna przyszłości. Następnie musimy opisać, jakie dane są podawane do sieci neuronowej. Ze względu na wcześniejszą normalizację danych, do sieci dostarczamy dane surowe (Type.Raw). Oznacza to, iż dane w sieci nie zostaną ponownie znormalizowane. W kolejnym kroku stworzymy zbiór punktów, które sieć wykorzysta do nauki. Podstawowym zagadnieniem jest inicjalizacja samej sieci.

Sieć neuronową dzielimy na warstwy. W naszym algorytmie skorzystamy z 3 warstw:

- Warstwy wejściowej o wielkości 10 (wielkość okna wejściowego)
- Warstwy ukrytej o wielkości 10
- Warstwy wyjściowej o wielkości 1 (wielkość okna wyjściowego)

Ilość neuronów w warstwie ukrytej została ustalona metodą prób i błędów, gdyż nie jest znany algorytm pozwalający na obliczenie najlepszej ilości neuronów. Rozpocząłem od wartości minimalnej i zwiększałem ilość neuronów do uzyskania zadowalających rezultatów. Dla każdej warstwy należało wybrać odpowiednią funkcję aktywacji (użyto tangensa hiperbolicznego).

$$f(s) = \tanh(s) = \frac{1 - \exp(-\beta s)}{1 + \exp(-\beta s)}, \quad (2.1)$$

Gdy aplikacja dysponuje już siecią i odpowiednimi danymi treningowymi, należy wybrać odpowiedni algorytm trenujący. W zbudowanej sieci korzystam z algorytmu Levenberga-Marquadta. Jest to popularna metoda, która odznacza się dużą szybkością działania. Jej minusem jest duże zapotrzebowanie na pamięć.

Następnym krokiem jest trenowanie sieci. Sieć jest trenowana do momentu, gdy średnia różnica pomiędzy obliczoną odpowiedzią oraz poprawną odpowiedzią (średni błąd) nie przekracza 0.00001.

Gdy sieć skończy się uczyć możemy rozpocząć wykonanie przewidywania.

2.4.3 Ważona średnia krocząca

Ważona średnia krocząca wyliczana jest poprzez zsumowanie wartości pomnożonych przez wagi nadane przedziałom, a następnie podzielenie ich przez sumę wag.

$$LWMA_{n,0} = \frac{1 \cdot x_{-n+1} + 2 \cdot x_{-n+2} + \dots + (n-1)x_{-1} + n \cdot x_0}{1 + 2 + \dots + (n-1) + n}, \quad (2.2)$$

2.4.4 Obliczenie obciążenia cieplnego dla budynku.

Projektowe obciążenie cieplne dla budynku to inaczej wielkość (moc) źródła ciepła, która jest potrzebna do uzyskania i utrzymania projektowej temperatury wewnątrz budynku czyli tzw. komfortu cieplnego.

Obciążenie cieplne budynku obliczane jest dla projektowej temperatury zewnętrznej oraz średniej rocznej temperatury zewnętrznej. Obliczeniowa temperatura zewnętrzna oraz średnia roczna temperatura zewnętrzna przyjmowana jest według normy [6.1] w zależności od lokalizacji budynku. Polska jest podzielona na pięć stref klimatycznych dla okresu zimnego zgodnie z rysunkiem przedstawionym poniżej.



Rysunek 2.1 Podział terytorium Polski na strefy klimatyczne [6.1]

Temperatury dla poszczególnych stref przedstawia tabela:

Strefa klimatyczna	Projektowa temperatura zewnętrzna, °C	Średnia roczna temperatura wewnętrzna, °C
I	-16	7,7
II	-18	7,9
III	-20	7,6
IV	-22	6,9
V	-24	5,5

Rysunek 2.2 Projektowa temperatura zewnętrzna i średnia roczna temperatura wewnętrzna [6.1]

Obciążenie cieplne dla budynku oblicza się poprzez wyznaczanie projektowych obciążeń cieplnych dla poszczególnych pomieszczeń. Szczególnie istotna jest znajomość typów oraz przeznaczenia pomieszczeń w celu wyznaczenia poszczególnych projektowych temperatur wewnętrznych. Temperatury dla poszczególnych typów pomieszczeń określa norma [6.1].

Założenia dla obiektu symulacji:

- Lokalizacja: Wrocław
- projektowa temperatura zewnętrzna dla okresu zimnego: -18stC
- średnia roczna temperatura zewnętrzna: 7,9stC
- Projektowa temperatura wewnętrzna: 21stC

Dla uproszczenia obliczeń przyjęto, iż całość budynku stanowi jedną przestrzeń (jedno pomieszczenie) o jednakowej projektowej temperaturze wewnętrznej.

Całość obliczeń obciążenia cieplnego dla budynku sporządzono zgodnie z normą [6.1].

Przegrody zewnętrzne

Przedmiotem symulacji jest dom o powierzchni 100m³ o ścianach drewnianych, modrzewiowych. Poniżej przedstawiono zestawienie przegród budowlanych dla obiektu.

Tabela 2.1 Zestawienie warstw przegród - ściana zewnętrzna drewniana.

l.p.	warstwa	grubość d	wsp. przew. ciepła λ	opór R
	ściana zewnętrzna modrzewiowa (SZ)	m	$\frac{W}{mK}$	$\frac{m^2K}{W}$
1	deska oblicówka	0,02	0,2	0,100
2	wełna min.	0,15	0,042	3,571
3	Płyta OSB	0,01	0,14	0,071
4	wełna min.	0,15	0,042	0,357
5	Deska elewacyjna modrzewiowa	0,04	0,2	0,200
				0,04 0,13 4,30
				0,23

Tabela 2.2 Zestawienie warstw przegród - podłoga na gruncie.

l.p.	warstwa	grubość d	wsp. przew. ciepła λ	opór R
	Podłoga na gruncie (PG)	m	$\frac{W}{mK}$	$\frac{m^2K}{W}$
1	Posadzka (parkiet lub płytki ceramiczne)	0,035	1,05	0,033
2	Wylewka betonowa	0,05	1,7	0,029
3	Styropian (20)	0,1	0,04	2,500
4	Płyta żelbetowa	0,25	1,7	0,147
5	Podsypka żwirowa	0,25	2,0	0,125
				0,04 0,13 2,835
				0,35

Tabela 2.3 Zestawienie warstw przegród – stropodach.

l.p.	warstwa	grubość d	wsp. przew. ciepła λ	opór R
	stropodach niewentylowany (Std)	m	$\frac{W}{mK}$	$\frac{m^2K}{W}$
1	płyta żelbetowa	0,14	1,70	0,082
2	płyty z wełny mineralnej (szczelne)	0,15	0,042	3,57
3	pustka powietrzna	0,009		0,143
4	deski sosnowe w poprzek włókien	0,02	0,16	0,125
5	3x papa na lepiku	0,008	0,18	0,044

	0,04
	0,10
	4,11
	0,24

Tabela 2.4 Zestawienie warstw przegród - drzwi zewnętrzne.

Drzwi zewnętrzne (DZ)
2,60

Tabela 2.5 Zestawienie warstw przegród - okno zewnętrzne.

Okno (O)
1,60

Tabela 2.6 Zapotrzebowanie na ciepło dla budynku.

Dom jednorodzinny Kubatura: 300 m ³ Powierzchnia: 100m ²			Przegroda					U	ΔU_{tb}	Wspól. korekcyjne					H_i
			Symbol przegr	Temp za przegrodą °C	Długość m	Wysokość / szerokość m	Powierzchnia m ²			b_u	f_{g1}	f_{g2}	f_{ij}	G_w	
1	2	3	4	5	6	7	8	9	10	11					12
Temp. wewnętrzna			SZ	-18	40	3	98	0,23	0,05	-	-	-		-	22,54
θ_{int}	21	°C	O	-18	11	2	22	1,6	0,4	-	-	-		-	44
Wentylacja/infiltracja			DZ	-18	1	2	2	2,6	0,5	-	-	-		-	6,2
n_{mi}	0,5		Std	-18	10	10	100	0,24	0,2	-	-	-		-	44
V_{inf}	120		PG	-18	10	10	100	0,35	0,2	-	-	-		-	27,5
V_{mi}	150														
V_i	150	m ³													
Współczynnik strat ciepła na wentylację:										51					
Projektowa wentylacyjna strata ciepła: $\phi_{v,i} = H_{v,i} \cdot (\theta_{int,i} - \theta_e), W$										1989					
Projektowa strata ciepła przez przenik.: $\phi_{T,i} = (H_{T,ie} + H_{T,iue} + H_{T,lq} + H_{T,ij}) \cdot (\theta_{int,i} - \theta_e), W$										5625,36					
Całkowita projektowa strata ciepła przez przenikanie i wentylację: $\phi_i = \phi_{T,i} + \phi_{v,i}, W$										7614,36					

Przy obliczeniu zapotrzebowania cieplnego dla budynku dostosowałem się do procedur i wzorów wykazanych w polskiej normie PN-EN 12831:2006 „Instalacje ogrzewcze w budynkach – Metoda obliczania projektowego obciążenia cieplnego”. Norma szczegółowo opisuje wzory i wskazania przy obliczaniu i przyjmowaniu współczynników korekcyjnych. Metoda obliczeniowa zawarta w normie oparta jest na założeniach:

- Rozkład temperatury powietrza w pomieszczeniu jest równomierny, a więc wysokość pomieszczeń jest nie większa niż 5m.
- Budynek jest dobrze zaizolowany.
- Zakłada się stałą temperaturę wewnętrzną.
- Właściwości elementów konstrukcyjnych budynku są stałe w funkcji temperatury.

Norma określa, iż przy wyznaczaniu strat ciepła przez przenikanie powinno się stosować wymiary zewnętrzne budynku tj. mierzone po zewnętrznej stronie. Określając wymiary poziome należy przyjąć połowę grubości ściany wewnętrznej i całą grubość ściany zewnętrznej. Wysokość ścian należy mierzyć pomiędzy powierzchniami podłóg pomieszczeń. Projektowe obciążenie cieplne dla budynku wynosi 7,6 kW. Jest to obciążenie cieplne dla obliczeniowej temperatury zewnętrznej wynoszącej -18stC.

Program symulujący pobiera aktualną pogodę (temperaturę zewnętrzną) z serwisu internetowego, a następnie oblicza aktualne zapotrzebowanie cieplne dla budynku.

2.4.5 Obliczanie czasu nagrzania domu

Ilość ciepła potrzebna do ogrzania masy powietrza w domu od aktualnej temperatury zewnętrznej do żądanej temperatury wewnętrznej:

$$Q = m * cwp * \Delta t, \quad (2.3)$$

Gdzie:

- m - gęstość powietrza * kubatura budynku,
- cwp – ciepło właściwe powietrza,
- Δt – różnica temperatur.

Obliczono ilość energii jaką dostarcza kocioł o mocy 15 kW:

$$E = n * P * t, \quad (2.4)$$

Gdzie:

- n- sprawność kotła,
- P-moc kotła,
- t-czas przez jaki pracuje.

Kocioł musi pracować tak długo aż pokryje zapotrzebowanie do ogrzania masy powietrza w domu, W związku z tym możemy porównać dwa powyższe równania i wyznaczyć czas t w sekundach.

$$t = \frac{Q}{n * P}, \quad (2.5)$$

Ze względu na straty ciepła, potrzebna jest kompensacja tych strat więc przeliczamy straty ciepła na energię:

$$J = W * s, \quad (2.6)$$

Do mocy potrzebnej do ogrzania masy powietrza należy dołożyć energię potrzebną do skompensowania strat ciepła. Mając zsumowane zapotrzebowanie na energię ponownie obliczamy czas pracy kotła.

2.5.3 Połączenie z urządzeniem grzewczym

Wykorzystanie Protokołu Modbus

Modbus jest protokołem komunikacyjnym, który w roku 1979 stworzyła firma Modicon (aktualnie właścicielem firmy jest Schneider Electric). Jest on wykorzystywany do połączenia ze sterownikami PLC. Dzięki prostocie i niezawodności stał się on standardem w automatyce i pomimo upływu czasu, jest jednym z najczęściej stosowanych protokołów w urządzeniach elektronicznych.

Główne powody wykorzystania protokołu Modbus:

- Został opracowany dla zastosowań w automatyce.

- Protokół jest darmowy.
- Komunikaty są uodpornione na przekłamania.
- Polecenia nie zostaną wykonane bez potwierdzenia.
- Błędy są odpowiednio sygnalizowane.
- Jest to standard stosowany przez przeważającą ilość twórców sterowników przemysłowych.
- Wdrożenie jak i utrzymanie są bezproblemowe.

Dzięki Modbusowi jesteśmy w stanie umożliwić komunikację nawet 248 urządzeniom znajdującym się w tej samej sieci. Najczęściej korzystamy z tego protokołu do połączenia komputera jako Master (urządzenie nadrzędne), a urządzeniami Slave (urządzenie otrzymujące polecenia) w systemach SCADA (Supervisory Control And Data Acquisition).

Rozwój i aktualizacje protokołu odbywają się dzięki Organizacji Modbus. Do organizacji należą niezależni użytkownicy i producenci urządzeń. Istnieje kilka wersji protokołu np. do komunikacji szeregowej jak i sieci Ethernet. Dużą popularność w automatyce przemysłowej ma interfejs RS485.

Modbus RTU

Najczęściej wykorzystywany w połączeniu szeregowym. Bajty w ramce przesyłane są binarnie jako znaki 8-bitowe. Ramka jest zabezpieczana sumą kontrolną CRC, która umożliwia spostrzeżenie przekłamań. Znaki w ramce należy przysyłać w sposób nieprzerwany, a także nie powinno być przerw między znakami.

Modbus ASCII

Protokół do komunikacji szeregowej. Bajty wysyłane są szesnastkowo (po dwa znaki ASCII). W tym przypadku suma kontrolna LRC zabezpiecza dane. Ramki rozpoczyna się od dwukropka a kończy znakiem nowej linii.

Modbus TCP/IP lub Modbus TCP

Protokół do komunikacji w sieciach TCP/IP. W ramach nie ma sumy kontrolnej gdyż niższe warstwy protokołu TCP/IP są wykorzystywane do kontroli błędów.

Modbus over TCP/IP lub Modbus over TCP lub Modbus RTU/IP

Wersja różniąca się od Modbus TCP tym, że w ramce znajduje się suma kontrolna tak jak w Modbus RTU

Dane i funkcje we wszystkich wersjach są takie same a różnica jest w opakowaniu ramek.

Komunikacja i urządzenia

Urządzenia łączące się z pomocą protokołu Modbus dostają niepowtarzany adres. Dla sieci szeregowych urządzenie będące nadrzędnym ma możliwość zainicjowania transmisji, a w sieci Ethernet wszystkie urządzenia są w stanie przesłać polecenie rozkaz. Zazwyczaj jest to jednak urządzenie nadrzędne.

Wykorzystanie modbusa

Modbusa wykorzystujemy do odczytania aktualnego stanu pieca (czy jest włączony lub wyłączony). W mojej aplikacji zastosowałem NModbus4. Jest to implementacja protokołu Modbus dla C#.

Jako symulator sterownika pieca, zostało wykorzystane oprogramowanie Modbus PLC Simulator Home, który symuluje działanie sterownika PLC. By połączyć się z odpowiednim urządzeniem musimy wybrać odpowiednie wyjście:

```
SerialPort port = new SerialPort("COM2");
```

Następnie ustalamy szybkość transmisji , ilość danych bitowych itp.

```
port.BaudRate = 9600;
```

```
port.DataBits = 8;
```

```
port.Parity = Parity.None;
```

```
port.StopBits = StopBits.One;
```

Dane te muszą być tożsame z danymi w naszym sterowniku.
Kolejnym krokiem jest otworezenie portu.

```
port.Open();
```

Gdy mamy już otwarty port musimy ustawić czy nasza aplikacja jest typu Master czy Slave.

W naszym przypadku jest to Master gdyż zadaniem aplikacji jest zmiana danych w Sterowniku.

```
IModbusSerialMaster master = ModbusSerialMaster.CreateRtu(port);
```

Zapis danych do sterownika następuje poprzez metodę WriteMultipleRegisters.

```
master.WriteMultipleRegisters(slaveId, startAddress, registers);
```

3 Opis badań

Aby zapewnić komfort cieplny użytkownikom budynku zaprojektowano system samouczący wyznaczający godziny powrotu lokatorów do domu. Zaprojektowano i porównano trzy algorytmy samouczące. Wszystkie algorytmy potrzebują danych testowych do przewidywania wyniku.

Badania zostały przeprowadzone przez okres 2 tygodni od dnia 15.02.2016 do dnia 28.02.2016. Podczas badań porównujemy wartości predykowane z algorytmów z rzeczywistymi wartościami powrotu do domu. Algorytm oblicza przewidywany powrót do domu po wciśnięciu w aplikacji przycisku „Wychodzę”. Następnie pobierana jest aktualna temperatura i program wykorzystuje te dane do obliczenia czasu nagrzewania budynku. Dane te porównujemy z danymi, które pozyskamy gdy osoba wróci do domu i przycisnie w aplikacji przycisk „Wróciłem”. Docelowo aplikacja powinna działać na czujnikach obecności. Badania zostały przeprowadzone dla 27 letniego mężczyzny.

3.1 Dane testowe dla procesu wyliczania godziny powrotu do domu

Dane do przeprowadzenia badań pochodzą z okresu od 07.05.2015 do 28.02.2016. Kolumna druga Tabeli nr 1 zawiera daty pomiarów, kolumna trzecia zawiera zmierzone godziny powrotu. Algorytmy przewidujące zostały zasilone takimi samymi danymi by można wiarygodnie porównać wyniki badań. Dane przy wykonaniu predykcji są wyselekcjonowane dla danego dnia tygodnia. Oznacza to, iż przewidywanie powrotu mieszkańca do domu np. w poniedziałek zostanie obliczone na podstawie danych ze wszystkich dostępnych poniedziałków w danych testowych.

Tabela 3.1 Wyniki pomiarów godzin powrotu do domu.

Lp.	Data	Godzina powrotu do domu	Lp.	Data	Godzina powrotu do domu
1	2	3	1	2	3
1	07.05.2015	18 00	148	01.10.2015	17 00
2	08.05.2015	14 00	149	02.10.2015	17 15
3	09.05.2015	17 00	150	03.10.2015	16 00
4	10.05.2015	18 00	151	04.10.2015	17 25
5	11.05.2015	17 00	152	05.10.2015	16 55
6	12.05.2015	16 30	153	06.10.2015	16 55
7	13.05.2015	14 00	154	07.10.2015	17 26
8	14.05.2015	17 27	155	08.10.2015	17 27
9	15.05.2015	18 00	156	09.10.2015	18 00
10	16.05.2015	23 30	157	10.10.2015	23 30
11	17.05.2015	22 00	158	11.10.2015	22 00
12	18.05.2015	17 15	159	12.10.2015	17 15
13	19.05.2015	14 00	160	13.10.2015	14 00
14	20.05.2015	17 25	161	14.10.2015	17 25
15	21.05.2015	17 30	162	15.10.2015	17 26
16	22.05.2015	17 00	163	16.10.2015	17 27
17	23.05.2015	17 45	164	17.10.2015	18 00
18	24.05.2015	16 22	165	18.10.2015	17 00

Lp.	Data	Godzina powrotu do domu	Lp.	Data	Godzina powrotu do domu
1	2	3	1	2	3
19	25.05.2015	17 00	166	19.10.2015	17 15
20	26.05.2015	16 55	167	20.10.2015	17 30
21	27.05.2015	16 55	168	21.10.2015	17 26
22	28.05.2015	17 30	169	22.10.2015	17 27
23	29.05.2015	16 30	170	23.10.2015	18 00
24	30.05.2015	17 45	171	24.10.2015	17 30
25	31.05.2015	18 00	172	25.10.2015	17 00
26	01.06.2015	16 00	173	26.10.2015	17 15
27	02.06.2015	17 00	174	27.10.2015	16 00
28	03.06.2015	17 15	175	28.10.2015	17 25
29	04.06.2015	17 00	176	29.10.2015	17 26
30	05.06.2015	17 25	177	30.10.2015	17 27
31	06.06.2015	17 26	178	31.10.2015	18 00
32	07.06.2015	16 55	179	01.11.2015	16 55
33	08.06.2015	18 45	180	02.11.2015	17 27
34	09.06.2015	17 30	181	03.11.2015	16 23
35	10.06.2015	17 00	182	04.11.2015	17 53
36	11.06.2015	17 15	183	05.11.2015	17 00
37	12.06.2015	16 00	184	06.11.2015	16 30
38	13.06.2015	17 25	185	09.11.2015	18 00
39	14.06.2015	16 55	186	10.11.2015	16 30
40	15.06.2015	16 55	187	11.11.2015	18 00
41	16.06.2015	17 26	188	12.11.2015	17 05
42	17.06.2015	17 27	189	13.11.2015	22 00
43	18.06.2015	18 00	190	14.11.2015	18 00
44	19.06.2015	23 30	191	15.11.2015	18 11
45	20.06.2015	22 00	192	16.11.2015	17 30
46	21.06.2015	17 15	193	17.11.2015	16 30
47	22.06.2015	14 00	194	18.11.2015	16 30
48	23.06.2015	17 25	195	19.11.2015	18 00
49	24.06.2015	17 26	196	20.11.2015	16 00
50	25.06.2015	17 27	197	21.11.2015	16 55
51	26.06.2015	18 00	198	22.11.2015	18 45
52	27.06.2015	17 00	199	23.11.2015	17 30
53	28.06.2015	17 15	200	24.11.2015	17 00
54	29.06.2015	17 30	201	25.11.2015	17 45
55	30.06.2015	17 26	202	26.11.2015	16 22
56	01.07.2015	17 27	203	27.11.2015	17 00
57	02.07.2015	18 00	204	28.11.2015	16 55
58	03.07.2015	17 30	205	29.11.2015	16 55

Lp.	Data	Godzina powrotu do domu	Lp.	Data	Godzina powrotu do domu
1	2	3	1	2	3
59	04.07.2015	17 00	206	30.11.2015	17 30
60	05.07.2015	17 15	207	01.12.2015	16 30
61	06.07.2015	16 00	208	02.12.2015	17 45
62	07.07.2015	17 25	209	03.12.2015	18 00
63	08.07.2015	17 26	210	04.12.2015	16 00
64	09.07.2015	17 27	211	07.12.2015	17 00
65	10.07.2015	18 00	212	08.12.2015	17 15
66	11.07.2015	16 55	213	09.12.2015	17 00
67	12.07.2015	18 00	214	10.12.2015	17 25
68	13.07.2015	17 15	215	11.12.2015	17 26
69	14.07.2015	16 00	216	12.12.2015	16 55
70	15.07.2015	17 25	217	13.12.2015	18 45
71	16.07.2015	17 26	218	14.12.2015	17 30
72	17.07.2015	17 27	219	15.12.2015	17 00
73	18.07.2015	23 00	220	16.12.2015	17 15
74	19.07.2015	12 00	221	17.12.2015	16 00
75	20.07.2015	17 04	222	18.12.2015	17 25
76	21.07.2015	17 15	223	19.12.2015	16 55
77	22.07.2015	18 00	224	20.12.2015	16 55
78	23.07.2015	17 25	225	21.12.2015	17 26
79	24.07.2015	17 25	226	22.12.2015	17 27
80	25.07.2015	24 00	227	23.12.2015	18 00
81	26.07.2015	12 00	228	24.12.2015	23 30
82	27.07.2015	17 40	229	25.12.2015	22 00
83	28.07.2015	17 00	230	26.12.2015	17 15
84	29.07.2015	18 00	231	27.12.2015	14 00
85	30.07.2015	18 00	232	28.12.2015	17 25
86	31.07.2015	16 30	233	29.12.2015	17 26
87	01.08.2015	13 00	234	30.12.2015	17 27
88	02.08.2015	12 00	235	31.12.2015	18 00
89	03.08.2015	17 36	236	02.01.2016	17 00
90	04.08.2015	17 02	237	03.01.2016	17 15
91	05.08.2015	18 00	238	04.01.2016	17 30
92	06.08.2015	17 10	239	05.01.2016	17 26
93	07.08.2015	16 25	240	06.01.2016	17 27
94	08.08.2015	15 00	241	07.01.2016	18 00
95	09.08.2015	13 00	242	08.01.2016	17 30
96	10.08.2015	17 30	243	09.01.2016	17 00
97	11.08.2015	16 25	244	10.01.2016	17 15
98	12.08.2015	18 00	245	11.01.2016	16 00

Lp.	Data	Godzina powrotu do domu	Lp.	Data	Godzina powrotu do domu
1	2	3	1	2	3
99	13.08.2015	16 25	246	12.01.2016	17 25
100	14.08.2015	17 30	247	13.01.2016	17 26
101	15.08.2015	15 00	248	14.01.2016	17 27
102	16.08.2015	15 00	249	15.01.2016	18 00
103	17.08.2015	17 40	250	16.01.2016	16 55
104	18.08.2015	17 00	251	17.01.2016	18 00
105	19.08.2015	17 55	252	18.01.2016	17 15
106	20.08.2015	16 30	253	19.01.2016	16 00
107	21.08.2015	17 23	254	20.01.2016	17 25
108	22.08.2015	16 00	255	21.01.2016	17 26
109	23.08.2015	16 00	256	22.01.2016	17 27
110	24.08.2015	16 23	257	23.01.2016	23 00
111	25.08.2015	17 53	258	24.01.2016	12 00
112	26.08.2015	17 00	259	25.01.2016	17 04
113	27.08.2015	16 30	260	26.01.2016	17 15
114	28.08.2015	18 00	261	27.01.2016	18 00
115	29.08.2015	16 30	262	28.01.2016	17 25
116	30.08.2015	18 00	263	29.01.2016	17 25
117	31.08.2015	17 05	264	30.01.2016	24 00
118	01.09.2015	22 00	265	31.01.2016	12 00
119	02.09.2015	18 00	266	01.02.2016	17 40
120	03.09.2015	18 11	267	02.02.2016	17 00
121	04.09.2015	17 30	268	03.02.2016	18 00
122	05.09.2015	16 30	269	04.02.2016	18 00
123	06.09.2015	16 30	270	05.02.2016	16 30
124	07.09.2015	18 00	271	06.02.2016	13 00
125	08.09.2015	16 00	272	07.02.2016	12 00
126	09.09.2015	16 55	273	08.02.2016	17 36
127	10.09.2015	18 45	274	09.02.2016	17 02
128	11.09.2015	17 30	275	10.02.2016	18 00
129	12.09.2015	17 00	276	11.02.2016	17 10
130	13.09.2015	17 45	277	12.02.2016	16 25
131	14.09.2015	16 22	278	13.02.2016	15 00
132	15.09.2015	17 00	279	14.02.2016	13 00
133	16.09.2015	16 55	280	15.02.2016	17 30
134	17.09.2015	16 55	281	16.02.2016	16 25
135	18.09.2015	17 30	282	17.02.2016	18 00
136	19.09.2015	16 30	283	18.02.2016	16 25
137	20.09.2015	17 45	284	19.02.2016	17 30
138	21.09.2015	18 00	285	20.02.2016	15 00

Lp.	Data	Godzina powrotu do domu	Lp.	Data	Godzina powrotu do domu
1	2	3	1	2	3
139	22.09.2015	16 00	286	21.02.2016	15 00
140	23.09.2015	17 00	287	22.02.2016	17 40
141	24.09.2015	17 15	288	23.02.2016	17 00
142	25.09.2015	17 00	289	24.02.2016	17 55
143	26.09.2015	17 25	290	25.02.2016	16 30
144	27.09.2015	17 26	291	26.02.2016	17 23
145	28.09.2015	16 55	292	27.02.2016	16 00
146	29.09.2015	18 45	293	28.02.2016	16 00
147	30.09.2015	17 30			

3.2 Sieć neuronowa

Sieć neuronowa potrzebuje co najmniej 40 danych by wykonać przewidywanie. Jest pewne, że na początku okresu gromadzenia danych nie będziemy posiadać wymaganej ilości pomiarów. Do czasu gdy będziemy posiadać potrzebną ilość danych system będzie wykorzystywał do predykcji inny algorytm uczenia. Algorytm jako okno wejściowe potrzebuje wielokrotności 10 danych. Oznacza to iż przy ilości np. 46 danych testowych, program pobierze wyłącznie 40 ostatnich pomiarów.

3.3 Liniowa regresja

Algorytm liniowej regresji zasilamy wszystkimi danymi jakie posiadamy.

3.4 Ważona średnia krocząca

Algorytm ważonej średniej kroczącej jest zasilany wszystkimi danymi które posiadamy.

4 Wyniki badań

W tabeli przedstawiono wyniki badań dla okresu 2 tygodni.

Kolumna ANN – przedstawia wyliczone wyniki dla sztucznej sieci neuronowej

Kolumna RL – przedstawia wyliczone wyniki dla regresji liniowej

Kolumna WMA – przedstawia wyliczone wyniki dla ważonej średniej kroczącej

Kolumna Rzeczywista godzina powrotu – przedstawia dane zebrane poprzez wciśnięcie przycisku „Wróciłem” w aplikacji.

Tabela 4.1 Wyniki predykcji dla algorytmów.

Data	ANN	RL	WMA	Rzeczywista godzina powrotu
15.02.2016	17:51	16:52	16:40	17:30
16.02.2016	18:11	16:55	16:52	17:57
17.02.2016	16:07	17:22	17:21	18:00
18.02.2016	16:58	17:10	17:28	16:25
19.02.2016	17:41	17:39	18:00	17:30
20.02.2016	16:05	18:23	18:50	15:00
21.02.2016	14:31	17:20	17:40	15:00
22.02.2016	16:24	16:51	16:40	17:40
23.02.2016	16:54	16:55	16:52	17:00
24.02.2016	16:52	17:20	17:21	17:55
25.02.2016	17:35	17:13	17:28	16:30
26.02.2016	17:26	17:43	18:00	17:23
27.02.2016	16:47	18:32	18:50	16:00
28.02.2016	17:15	17:29	17:48	16:00

Poniższa tabela przedstawia różnice dla poszczególnych algorytmów pomiędzy przewidzianym czasem powrotu do domu, a rzeczywistym czasem powrotu.

Tabela 4.2 Różnice pomiędzy czasem predykowanym a czasem powrotu.

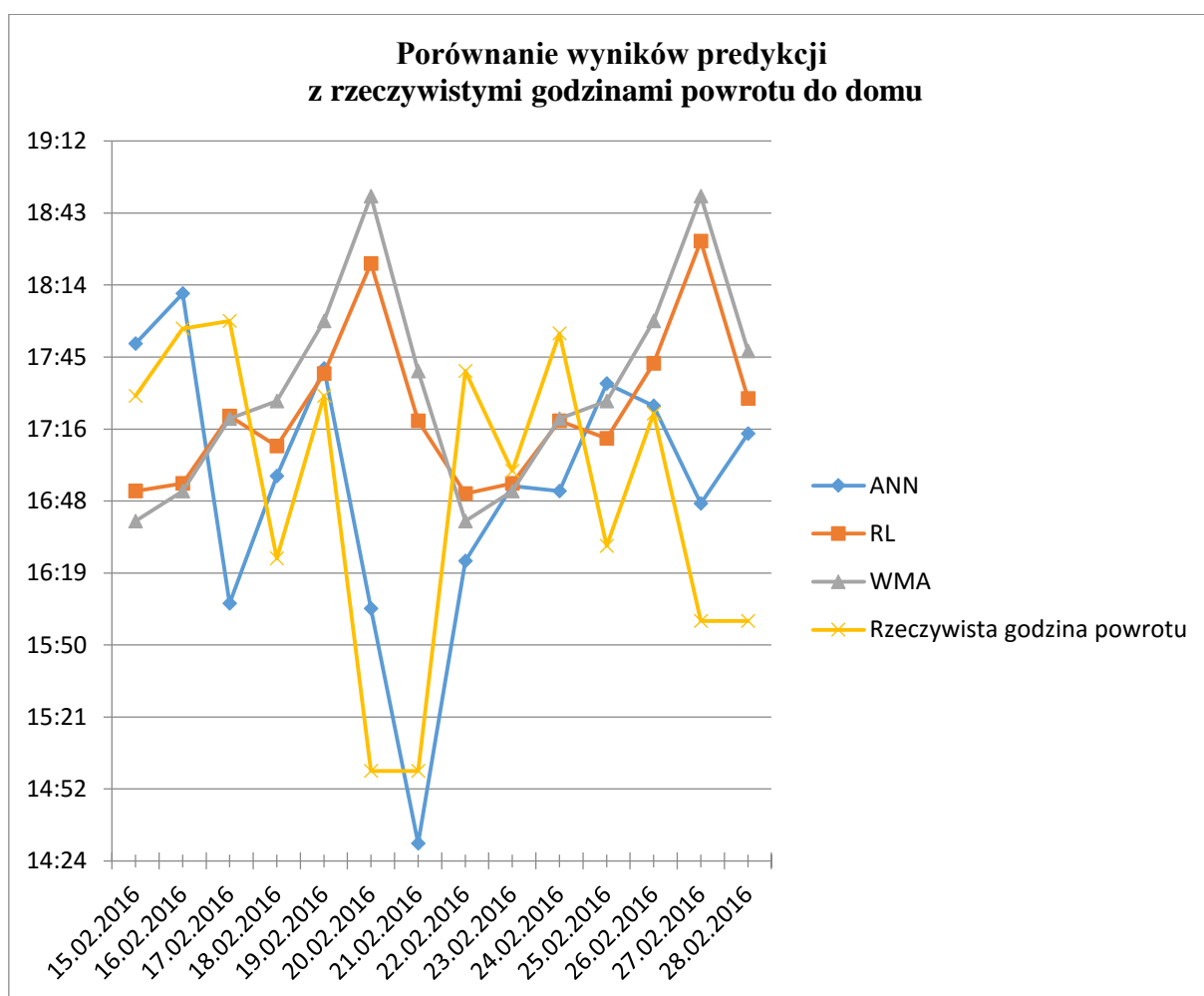
Lp.	Data	ANN	RL	WMA
1	15.02.2016	+00:21	-00:38	-00:50
2	16.02.2016	+00:14	-01:02	-01:05
3	17.02.2016	-01:53	-00:38	-00:39
4	18.02.2016	+00:33	+00:45	+01:03
5	19.02.2016	+00:11	+00:09	+00:30
6	20.02.2016	+01:05	+03:23	+03:50
7	21.02.2016	-00:29	+02:20	+02:40
8	22.02.2016	-01:16	-00:49	-01:00
9	23.02.2016	-00:06	-00:05	-00:08
10	24.02.2016	-01:03	-00:35	-00:34
11	25.02.2016	+01:05	+00:43	+00:58
12	26.02.2016	+00:03	+00:20	+00:37

13	27.02.2016	+00:47	+02:32	+02:50
14	28.02.2016	+01:15	+01:29	+01:48

Tabela 4 zestawia wyliczone średnie różnice pomiędzy przewidzianymi przez algorytmy powrotami użytkownika do domu a rzeczywistymi godzinami powrotu.

Tabela 4.3 Średnie różnice pomiędzy przewidywanym czasem a czasem powrotu.

ANN	RL	WMA
00:44	01:06	01:19



Rysunek 4.1 Wykres porównania wyników predykcji dla poszczególnych algorytmów.

Na wykresie zostały przedstawione wyniki pomiarów. Na osi X mamy przedstawione godziny powrotów. Na osi Y mamy dni dla których zostały przeprowadzone badania.

5. Wnioski

Na potrzeby pracy stworzyłem aplikację sterującą ogrzewaniem która przewiduje godzinę powrotu do domu. Godzina powrotu do domu użytkownika zależy od wielu czynników np. natężenia ruchu na drogach, sytuacji w pracy, potrzeb konsumpcyjnych, a także zdarzeń nieprzewidzianych. Dlatego też nie jest możliwe idealne wyznaczenie godziny powrotu do domu na podstawie danych historycznych. Na podstawie przeprowadzonych badań mogę wywnioskować, iż z algorytmów wykorzystanych najlepiej sprawdziła się sztuczna sieć neuronowa (SSN). Przewidywania tego algorytmu najczęściej były bliskie rzeczywistej godzinie powrotu mieszkańca do domu. W związku z tym zastosowanie tego algorytmu w systemie sterowania ogrzewaniem jest uzasadnione. Zastosowanie sieci neuronowej zbudowanej jak w programie symulacyjnym ma też minusy. Jednym z nich jest ilość danych potrzebnych do wykonania predykcji (co najmniej 40). Do czasu zgromadzenia wymaganej ilości danych system musi korzystać z innego algorytmu, który przy mniejszej ilości danych uczących daje wymierne wyniki. Badanie wykazało iż drugim w kolejności jeśli chodzi o trafność predykcji był algorytm ważonej średniej kroczącej. Z tego powodu do czasu osiągnięcia wymaganej ilości danych uczących system korzysta z tego algorytmu. Tak zbudowany system daje duże możliwości rozszerzania o nowe funkcjonalności np. sterowanie oświetleniem.

6. Literatura

- [6.1] PN-EN 12831:2006. Instalacje ogrzewcze w budynkach – Metoda obliczania projektowego obciążenia cieplnego
- [6.2] Brownlee Jason, “A tour of machine learning algorithms”, 2013
<http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms>
- [6.3] Cholewa W., „Metoda diagnozowania maszyn z zastosowaniem zbiorów rozmytych Zeszyty naukowe – Mechanika”, Wydawnictwo Politechniki Śląskiej, Gliwice 1983
- [6.4] Ferenstein Elżbieta, Mielniczuk Jan, „SAD - STATYSTYCZNA ANALIZA DANYCH“, <http://edu.pjwstk.edu.pl/wyklady/sad/scb/index27.html>
- [6.5] Lane David, „Introduction to Linear Regression”,
<http://onlinestatbook.com/2/regression/intro.html>
- [6.6] Meisner Adam, „Elementy uczenia maszynowego”, 2016,
<http://users.man.poznan.pl/ameis/SiW-MaszynUcz.pdf>
- [6.7] Mitchell T. M., McGraw Hill, „Machine Learning”, 1997.
- [6.8] Nowak-Brzezińska Agnieszka, „Sieci Bayesowskie”, <http://docplayer.pl/1749426-Sieci-bayesowskie-agnieszka-nowak-brzezinska.html>
- [6.9] Tadeusiewicz R., „Sieci neuronowe”, Akademicka Oficyna Wydawnicza 1993 s. 13
- [6.10] Tadeusiewicz R., „Odkrywanie właściwości sieci neuronowych”, Kraków 2007 s. 41
- [6.11] Ting-pat So Albert, Wai Lok Chan, “Intelligent Building Systems” , Springer US
- [6.12] Włodarczyk Jacek, Podosek Zbigniew, ”Systemy Teletechniczne Budynków Inteligentnych”

7. Spis tabel

Tabela 2.1 Zestawienie warstw przegród - ściana zewnętrzna drewniana.....	18
Tabela 2.2 Zestawienie warstw przegród - podłoga na gruncie.....	18
Tabela 2.3 Zestawienie warstw przegród – stropodach.....	18
Tabela 2.4 Zestawienie warstw przegród - drzwi zewnętrzne.	19
Tabela 2.5 Zestawienie warstw przegród - okno zewnętrzne.....	19
Tabela 2.6 Zapotrzebowanie na ciepło dla budynku.....	19
Tabela 3.1 Wyniki pomiarów godzin powrotu do domu.....	23
Tabela 4.1 Wyniki predykcji dla algorytmów.	28
Tabela 4.2 Różnice pomiędzy czasem predykowanym a czasem powrotu.....	28
Tabela 4.3 Średnie różnice pomiędzy przewidywanym czasem a czasem powrotu.....	29

8. Spis rysunków

Rysunek 1.1 Podział stylów uczenia maszynowego ze względu na sposób działania [6.2].....	7
Rysunek 1.2 Schemat uczenia nadzorowanego.....	8
Rysunek 1.3 Zbiór danych uczących dla regresji liniowej [6.5]	9
Rysunek 1.4 Wyliczona linia regresji [6.5]	10
Rysunek 1.5 Twierdzenia Bayesa [6.2]	11
Rysunek 1.6 Schemat jednokierunkowej wielowarstwowej sieci neuronowej [6.10].....	12
Rysunek 2.1 Podział terytorium Polski na strefy klimatyczne [6.1]	17
Rysunek 2.2 Projektowa temperatura zewnętrzna i średnia roczna temperatura zewnętrzna [6.1].....	17
Rysunek 4.1 Wykres porównania wyników predykcji dla poszczególnych algorytmów.	29