Analiza próbek szmerów oddechowych dla potrzeb opracowania aplikacji rozpoznawania choroby płuc

**Marianna Tybura, Zuzanna Zając, Miłosz Wyrębkiewicz, Krzysztof Kaniuka, Igor Tylak**

**dr. inż. Paweł Drzymała**

Politechnika Łódzka, WEEIA, Informatyka, Łódź, Polska

STRESZCZENIE

Celem projektu było opracowanie systemu asystującego przy analizie lekarskiej pod kątem identyfikacji choroby płucnej, a także koordynującego współpracę lekarza z badanym pacjentem.

W skład wyżej wymienionego rozwiązania wchodził model klasyfikujący, baza danych SQL Server, a także aplikacja ASP .NET [14].

Część modelową stanowiło oprogramowanie przetwarzające otrzymane pliki dźwiękowe do postaci spektrogramów oraz wytrenowana sieć neuronowa [3], która na podstawie spektrogramów [2][6], klasyfikowała stan zdrowia pacjenta pod kątem potencjalnych schorzeń płucnych.

Baza danych zbierała informacje niezbędne do funkcjonowania systemu. Przechowywała informacje o kontach lekarzy, przypisanych do nich pacjentach (w tym kontach pacjentów) oraz sesjach lekarzy z pacjentami. Nagrania zamieszczane przez pacjentów również przechowywano w bazie, dzięki czemu lekarze i pacjenci mieli do nich dostęp z perspektywy aplikacji.

Aplikacja stanowiła podstawowy interfejs użytkowy dla administratorów aplikacji, lekarzy i pacjentów. To za jej pomocą możliwe było zamieszczanie nagrań, tworzenie kont [13], uzyskiwanie dostępu do nagrań i poszczególnych sesji. Za pomocą aplikacji, lekarze mogli również umieszczać zalecenia dla pacjentów w obrębie następujących spotkań.

KEYWORDS

Audio Data Processing, Bronchiectasis, Bronchiolitis, COPD, Choroba płuc, Database, Health, LRTI, Machine Learning, Neural network, Ostre zapalenie oskrzelików, POChP, Pneumonia, Przetwarzanie dźwięku, Pulmonologia, Pulmonology, Respiratory, Respiratory diseases, SI, Sieć neuronowa, URTI, Zapalenie dolnych dróg oddechowych, Zapalenie górnych dróg oddechowych, Zapalenie płuc, Zdrowie, aplikacja, client-server, układ oddechowy

1. WSTĘP
   1. Obszar / tło działań

Wraz z rozwojem metod analizy danych w oparciu o algorytmy sztucznej inteligencji na przestrzeni ostatnich kilku lat dostrzeżono potencjał implementacji tego typu rozwiązań w zastosowaniach medycznych [16]. Na przestrzeni stuleci choroby płuc diagnozowane były przez lekarzy pulmonologów, którzy na bazie swojego wieloletniego doświadczenia byli w stanie określić na jaką chorobę cierpiał pacjent w oparciu głównie o tzw. szmery płuc.

Istnieją jednak liczne przykłady pokazujące, że technologia potrafi być precyzyjniejsza i dokładniejsza w kwestii analizy dźwięków, ponieważ analizuje częstotliwości których ludzkie ucho nie jest w stanie wychwycić. Gwarantuje również powtarzalność takiej diagnozy czego nie możemy być pewni w przypadku tradycyjnej diagnozy lekarskiej.

Szczególny wzrost zainteresowania tematem szybkiej, przystępnej i możliwie najcelniejszej diagnozy pod kątem potencjalnych chorób płuc nastąpił w roku 2020 wraz z wybuchem pandemii COVID - 19. Kluczowym okazała się wtedy zdolność do sprawnego odróżnienia wirusa od innych dolegliwości płucnych przy jednoczesnym zapewnieniu izolacji pacjenta i lekarza [1]. Dostrzeżono wtedy zysk jaki mogą nieść ze sobą aplikacje asystujące lekarzom w stawianiu diagnozy.

Diagnostyka chorób płuc w oparciu o modele uczenia maszynowego i algorytmy sztucznej inteligencji cieszy się szczególną popularnością w miejscach, gdzie dostęp do pomocy medycznej jest ograniczony. Szczególnie warto zwrócić tutaj uwagę na kraje afrykańskie a także odizolowane regiony Australii cechującej się nierównomiernością zaludnienia kontynentu.

Według National Library Of Medicine niemal 545 milionów ludzi cierpi na chroniczne schorzenia związane z drogami oddechowymi co stanowi 7,4% populacji. Nie bez znaczenia pozostają zatem dedykowane aplikacje mobilne pozwalające użytkownikom cierpiącym na schorzenia płuc bezpłatnie kontrolować czy też zadbać lepiej o swój stan zdrowia.

* 1. Cel projektu

Chociaż rynek obfituje w aplikacje mniej lub bardziej związane z tematyką chorób płuc czy też ogólnie respiracji, zdecydowanie brakuje rozwiązań opartych o modele uczenia maszynowego czy też wykorzystujące sztuczną inteligencję. Jest to temat stosunkowo świeży, a odpowiednie wyuczenie sieci neuronowej wymaga czasu i zasobów. Z tego powodu, rozwiązania te są trudno dostępne na rynku.

Co więcej, większość przystępnych użytkownikom aplikacji niekoniecznie daje możliwość zestawienia wyników zwróconych przez program z lekarskim doświadczeniem w celu ich weryfikacji.

Nasze rozwiązanie oferuje aplikację, której trzonem jest wyuczony algorytm sztucznej inteligencji. Jest to bowiem najskuteczniejsza jak dotychczas metoda automatyzacji procesu diagnozy choroby płuc.

Co więcej aplikacja dedykowana jest zarówno dla pacjenta i dla lekarza, ponieważ zależy nam na tym, aby sesje leczenia choroby odbywały się pod okiem profesjonalisty w ścisłej współpracy pomiędzy dwiema stronami.

1. STAN WIEDZY
   1. Konstruktywna analiza istniejących rozwiązań

Wśród dostępnych na rynku rozwiązań związanych z chorobami płuc, dostrzec można pewne archetypy aplikacji.

Pierwszym z nich będą tutaj aplikacje stanowiące **kompendia wiedzy na temat danej choroby**. Mają one ułatwić zrozumienie jak należy postępować w przypadku wystąpienia objawów, a także z czego one wynikają i jakich czynności należy się wystrzegać. Przykładami mogą być tutaj [Breathe Better: Beat Asthma!](https://apps.apple.com/us/app/breathe-better-beat-asthma/id1187328412) czy też My Asthma App.

Na rynku znaleźć możemy również **aplikacje – organizery**, których podstawową funkcją będzie ułatwienie użytkownikowi życia z chorobą na co dzień. Przypominają one o zażywaniu leków, informują alergików o miejscach szczególnego nasilenia alergenu, dostarczają map okolicznych przychodni pulmonologicznych oraz zachęcają do podejmowania ćwiczeń niwelujących objawy choroby. Za przykład może posłużyć tutaj aplikacja [Breathefree: Lung Health App](https://play.google.com/store/apps/details?id=breathefree.lung.health.asthma.breathing&hl=en_US&pli=1https://play.google.com/store/apps/details?id=breathefree.lung.health.asthma.breathing&hl=en_US&pli=1).

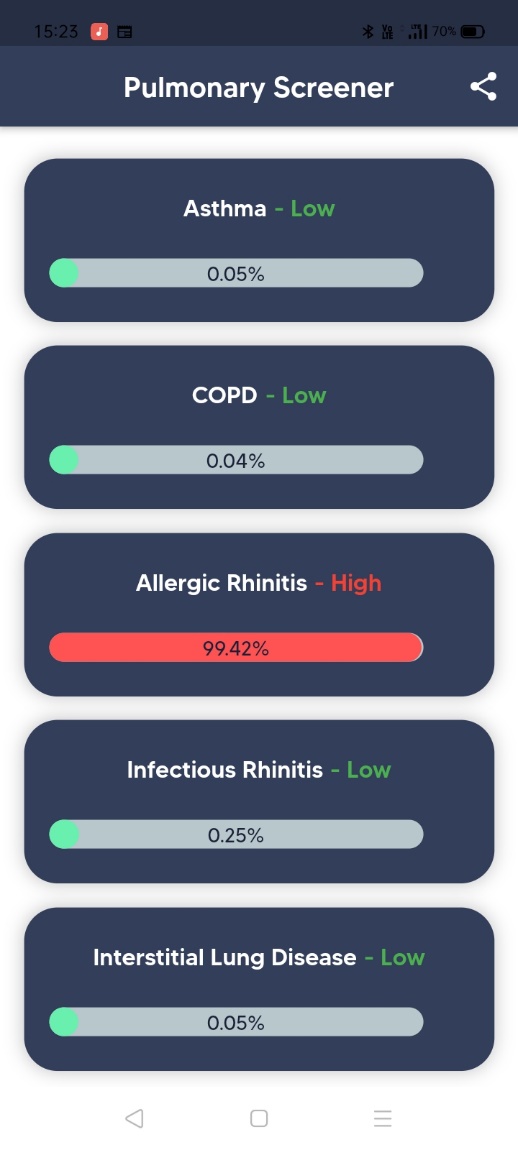
Nieco mniej powiązane z tematem, jednakże nie mniej liczne okazują się być **aplikacje do spirometrii** czyli do wykonywania badania czynnościowego układu oddechowego sprawdzającego ogólną prawidłowość oddychania pacjenta. Aplikacje uwzględniają takie aspekty jak objętość płuc (wskaźnik FEV1), pojemność płuc (wskaźnik FVC), a także przepływ powietrza przez drobne oskrzela (tzw. MEF). Do innych zadań aplikacji należy również rysowanie spirogramów z wykonywanych pomiarów czy też gromadzenie historii pomiarów pacjenta.

Aplikacje w większości zakładają, że jesteśmy w posiadaniu urządzenia do wykonywania pomiarów spirometrycznych, które jest łączone ze smartfonem w celu szczegółowej analizy wyników i ich archiwizacji. Jako przedstawiciel aplikacji z tejże kategorii może posłużyć aplikacja [Spirogram](https://play.google.com/store/apps/details?id=nl.curavista.apps.spirogram&hl=pl&gl=US) od [Curavista](https://curavista.health/), która prócz wyżej wymienionych funkcji, zakłada również konsultację wyników przeprowadzonych badań z lekarzem na podstawie założonego na stronie producenta konta. Nie jest to oczywiście usługa darmowa, a wymagająca płatności w postaci subskrypcji.

Nieco inne podejście prezentuje aplikacja [**NuvoAir Home.**](https://play.google.com/store/apps/details?id=com.nuvoair.aria&hl=pl&gl=US) Również umożliwia stosunkowo łatwe udostępnianie wyników przeprowadzonych badań, jednakże darzy użytkownika większym zaufaniem powierzając mu kontrolę nad uzyskanymi rezultatami. Aplikacja zachęca do wykonywania badań regularnie i przypomina o wykonaniu pomiaru, jeżeli uzna, że dane z poprzedniego badania są przedawnione.

Kolejną grupą aplikacji są rozwiązania próbujące dokonać diagnozy na jaką chorobę płuc może cierpieć użytkownik. Słowo próbujące jest tutaj jednak uzasadnione, ponieważ nie przeprowadzają one żadnej profesjonalnej diagnozy. Werdykt opierają natomiast o serię pytań zadawanych użytkownikowi. Na podstawie wybranych odpowiedzi (w większości będą to pytania tak i nie), dokonywana jest analiza w jakim stopniu istnieje prawdopodobieństwo zachorowania na chorobę.

Za przykład może posłużyć aplikacja [Pulmonary Screener](https://play.google.com/store/apps/details?id=com.mobiletechnologylab.pulmonaryscreening&hl=en_US):



Rys. 1 widok wyników aplikacji Pulmonary Screener jako rezultat odpowiedzi na serię pytań

Ostatnia, najbardziej utożsamiająca się z naszym rozwiązaniem grupa aplikacji to produkty, które faktycznie opierają swoje działanie o modele uczenia maszynowego oraz analizę szmerów oddechowych pacjenta. Pierwsza z nich, [ResAppDx](https://www.centivis.com/news/2021/12/9/turning-smartphones-into-tools-for-respiratory-disease-management-resapp-health-pm6lk-7mh7l), charakteryzuje się dość dużą skutecznością w rozpoznawaniu chorób takich jak astma, zapalenie płuc, ostre zapalenie oskrzelików, błonica czy POChP. Do wykorzystania aplikacji nie jest wymagane połączenie z Internetem, a badanie wykonywane jest na podstawie nagrania kaszlu pacjenta. Nagranie wykonywane jest przy użyciu telefonu.

W czasie pandemii COVID – 19 aplikacja okazała się również dość skuteczna przy rozpoznawaniu obecności wirusa. Niestety nie jest ona udostępniona publicznie stąd dostęp dla przeciętnego użytkownika jest praktycznie niemożliwy.

Kolejna aplikacja, czyli [Tambua App](https://tambua-health-225131a6743-b4876b01f58d3.webflow.io/company) udostępniana jest przede wszystkim na rynku afrykańskim stąd też zostawia po sobie niewiele informacji w Internecie w sferze globalnej. Pomysł na jej powstanie wiązał się z dość ubogim dostępem do medycyny w krajach afrykańskich, a także problem z zapewnieniem lekarzom odpowiedniego sprzętu do prowadzenia klinik pulmonologicznych. Tambua App kierowana była przede wszystkim do lekarzy w tamtych rejonach i miała pełnić rolę asystenta, pomagającego w postawieniu diagnozy. Jej działanie oparto o model uczenia maszynowego oraz sieć neuronową zbudowaną przy pomocy TensorFlow[3] stąd można spodziewać się autentycznych wyników.

* 1. Uzasadnienie i formy rozwiązania

W wyniku powyższej analizy konkurencji, dostrzegamy, że na rynku brakuje przystępnych użytkownikom rozwiązań profesjonalnie klasyfikujących choroby pulmonologiczne. Takie rozwiązania, dzięki włączeniu w proces sztucznej inteligencji, gwarantują wysoką skuteczność diagnozy [8][2][1][9]. Niemniej jednak, uważamy, że nie zwalania to użytkownika z obowiązku wykonywania regularnych, lekarskich konsultacji.

W tym celu nasz produkt, oprócz skutecznego, komputerowego asystenta-diagnosty, oferuje również wsparcie lekarza, który w oparciu o nagranie pacjenta, będzie mógł postawić własną diagnozę i modyfikować kartę pacjenta w celu dostosowania odpowiedniej metody leczenia choroby. Proponowana przez nas forma leczenia to sesje spotkań lekarzy z pacjentami.

W tym celu zalecamy pacjentom zarejestrowanym wypożyczenie w przychodni stetoskopu cyfrowego. Ma to na celu poprawę jakości nagrań i tym samym zwiększenie szans dokonania poprawnej klasyfikacji zarówno przez model jak i przez lekarza. Istnieje możliwość wykonania nagrania przy użyciu własnych urządzeń wyposażonych w mikrofon jednakże klasyfikacja choroby w tym przypadku może być utrudniona.

Pacjenci zarejestrowani w ramach leczenia zobowiązani są wysyłać w terminach ustalonych przez lekarza nagrania za pośrednictwem aplikacji. Dzięki nim, na podstawie szmerów oddechowych pacjenta, algorytm będzie mógł w sposób automatyczny zaklasyfikować chorobę.

Po stronie lekarza, będzie natomiast zatwierdzenie lub odrzucenie wyniku postawionego przez automat. Będzie mógł on również dać pacjentowi informację zwrotną na temat dalszego toku leczenia choroby za pomocą przygotowanego przez nas systemu kart zdrowia.

1. OPIS ROZWIĄZANIA
   1. Opis techniczny rozwiązania

Nasze rozwiązanie składa się w całości ze ściśle współpracujących ze sobą modułów: modułu modelu klasyfikującego, modułu bazy danych oraz aplikacji LungMed

**Model klasyfikacyjny [1]** napisany został w języku Python z użyciem biblioteki Keras[3]. Na podstawie dostarczonych spektrogramów dźwięków, wyuczono sieć neuronową [1][2][8][9] tak aby rozpoznawała następujące choroby:

* Przewlekłą obturacyjną chorobę płuc (COPD)
* Zapalenie oskrzelików (Bronchiolitis)
* Rozstrzenie oskrzeli (Bronchiectasis)
* Astmę (Asthma)
* Zakażenie górnych dróg oddechowych (URTI)
* Zapalenie płuc (Pneumonia)
* Zakażenie dolnych dróg oddechowych (LRTI)
* COVID – 19
* Symptomy COVID – 19 (Symptomatic)
* Niewydolność Serca (Heart Failure)

Do przetwarzani dźwięków na spektrogramy wykorzystano bibliotekę Librosa.

Model wytrenowano w oparciu o 3 zbiory danych:

* KAUH
* ICBHI
* COVID-19

Początkowo zakładano wykorzystanie jednie zestawu [ICBHI](https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database). W procesie przygotowywania i przetwarzania danych okazało się jednak, że jest on bardzo niesymetryczny, a COPD stanowi ponad połowę chorób w zbiorze.

Obraz zawierający tekst, zrzut ekranu, diagram, krąg

Opis wygenerowany automatycznie

Rys. 2 Rozkład chorób w zbiorze ICBHI przed wykonaniem augmentacji

Z tego powodu, do zestawu treningowego dodano również dane ze zbioru KAUH oraz COVID-19 zwiększając tym samym zakres chorób które model może rozpoznać. Wykonano również augmentację danych [4][5] dzięki czemu udało się jeszcze bardziej powiększyć zbiór treningowy o sztuczne dane oraz uodpornić go na szum i nagrywanie w niesprzyjających warunkach.

Obraz zawierający tekst, zrzut ekranu, diagram, krąg

Opis wygenerowany automatycznie

Rys. 3 Rozkład chorób w zbiorze po dołączeniu zbiorów i augmentacji

Kolejnym krokiem było ustalenie typu spektrogramów wykorzystywanych końcowo w modelu sieci neuronowej. Zapoznano się ze specyfiką spektrogramów MEL oraz MFCC i dokonano porównania efektywności algorytmu dla każdego z typów [6][7]. W kontekście naszego rozwiązania, ostatecznie lepiej sprawdziły się spektrogramy MEL.

Wymagania funkcjonalne dla modelu zakładały jego wydajność i poprawność diagnozy na poziomie 70%. Jednocześnie, oczekiwano, że algorytm będzie stosunkowo odporny na zakłócenia otoczenia, w którym pacjenci będą wykonywali swoje nagrania. Ostatecznie udało nam się osiągnąć skuteczność na poziomie 90%.

Obraz zawierający tekst, niebo, zrzut ekranu, budynek

Opis wygenerowany automatycznie

**Aplikacja LungMed** napisana w .NET MVC [11] z wykorzystaniem Frameworka Identity [13] stanowi warstwę widoku i kontrolera dla naszego systemu. Framework Identity pozwolił nam na stworzenie odpowiednich ról użytkowników oraz zapewnienie autoryzacji i autentykacji w naszej aplikacji za sprawą systemu logowania [12][13][14].

Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

Opis wygenerowany automatycznie

Rys. 4 Role w aplikacji dla klientów zalogowanych

Lekarze za jej pomocą mają wgląd w sesje spotkań ze swoimi pacjentami. Mogą również przeglądać przesłane przez nich pliki dźwiękowe (przeanalizowane uprzednio przez model), weryfikować diagnozę postawioną przez model oraz dawać pacjentom feedback w postaci adnotacji do spotkania za sprawą karty pacjenta.

Obraz zawierający tekst, numer, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Rys. 5 Widok listy lekarzy

Pacjenci mają możliwość wglądu w historię swojej choroby, a także przesyłania nagrań w ramach sesji z lekarzami. Nagrania pacjentów niezarejestrowanych będą analizowane jedynie przed model klasyfikacyjny. Nie będą również zapisywane w bazie danych.

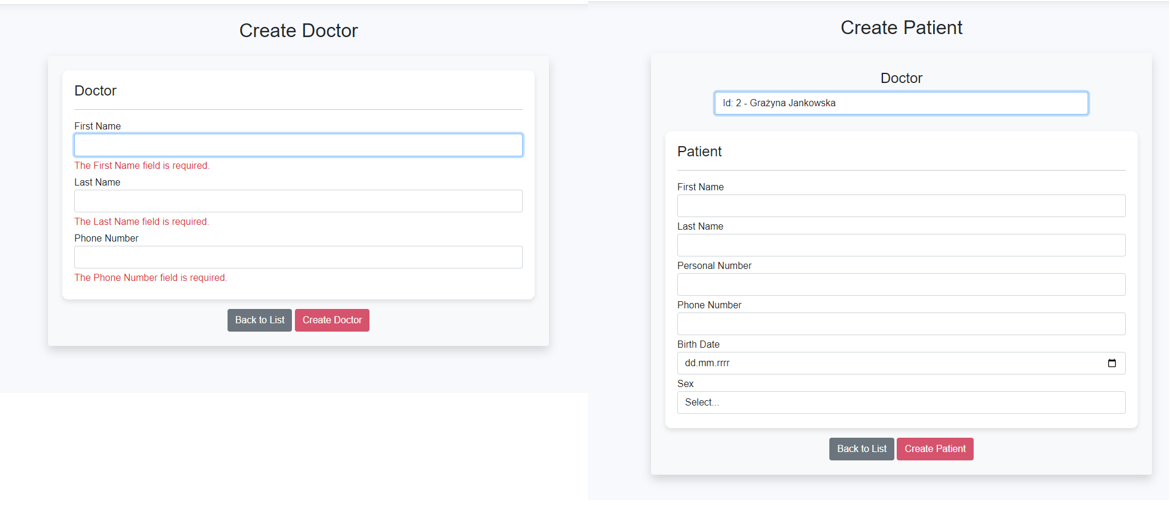
Obraz zawierający tekst, numer, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Rys. 6 Widok listy pacjentów

Zadaniem administratorów jest tworzenie kont dla lekarzy i pacjentów oraz przypisywanie lekarzy do pacjentów, których leczą.

Pacjent zarejestrowany, a także lekarz muszą się zalogować do aplikacji, aby móc z niej skorzystać. Wymagany jest e-mail i hasło. Hasło przechowywane jest w bazie danych w postaci hasza w celu ochrony danych logowania użytkowników.



Rys. 7 Tworzenie kont dla użytkowników

**Baza danych** napisana w SQL Server stanowi przede wszystkim strukturę organizacyjną dla aplikacji. Zajmuje się również składowaniem nagrań przesyłanych przez pacjentów.

Zawarto w niej dane do kont i ról przypisanych do tych kont. Ponadto, rejestrowane są sesje pacjentów z lekarzami, wraz ze związanymi z nimi nagraniami i zaleceniami zdrowotnymi. Dzieje się to za sprawą sytemu kart zdrowia pacjentów, stanowiących historię choroby pacjenta.

Obraz zawierający tekst, diagram, Równolegle, Plan

Opis wygenerowany automatycznie

Rys. 8 Struktura bazy danych Aplikacji LungMed

**Integracja** aplikacji LungMed z modelem predykcyjnym została zrealizowana poprzez wywołanie skryptu Python z perspektywy aplikacji napisanej w języku C#. Umożliwiła to klasa ProcessStartInfo.

Wraz z załączeniem pliku przez pacjenta, wykonywana jest jego tymczasowa kopia.

Następnie wywołany przez aplikację skrypt odpowiada za przetworzenie pliku audio do postaci spektrogramów MEL, a później klasyfikację choroby przy użyciu wytrenowanego modelu predykcyjnego.

Na koniec wynik jest zwracany do aplikacji, a plik wraz z wynikiem trafia na serwer bazodanowy jeżeli pacjent wysyłający nagranie był użytkownikiem zarejestrowanym.

Lokalna kopia nagrania jest kasowana po przetworzeniu przez model i ewentualnym umieszczeniu w bazie danych.

Jeżeli pacjent jest zalogowany, lekarz prowadzący ma możliwość zaakceptowania lub odrzucenia diagnozy postawionej przez model.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Rys. 9 Diagnoza modelu predykcyjnego na podstawie nagrania   
udostępnionego przez zalogowanego użytkownika

* 1. Zarządzanie projektem i pracą zespołową

W pierwszych dniach pracy sporządzony został ogólny harmonogram zadań nad projektem w postaci [wykresu Gantta](https://tulodz-my.sharepoint.com/:x:/g/personal/240697_edu_p_lodz_pl/EUlsK77YRwxCmfHAyseEOxQBqDaybPd7KBSftxhLtcdpeQ?e=c3y5NY). Jego celem było wytyczenie kierunku dla projektu, a także ułatwienie trzymania się początkowo przyjętych założeń.

W odstępie cotygodniowym (z kilkoma wyjątkami) realizowane były spotkania zespołu projektowego z promotorem w celu podsumowania pracy z zeszłego tygodnia i ustalenia podziału pracy na następny tydzień. Przyznanie zadań poszczególnym członkom zespołu odbywało się za pomocą systemu [minutek](https://tulodz-my.sharepoint.com/:x:/g/personal/240697_edu_p_lodz_pl/EUlsK77YRwxCmfHAyseEOxQBqDaybPd7KBSftxhLtcdpeQ?e=c3y5NY) sporządzanych w Excelu.

Takie rozwiązanie, za sprawą systemu współdzielenia dokumentów, umożliwiło łatwy dostęp do minutek dla wszystkich członków zespołu, a także wygodniejszy wgląd w harmonogram, który sporządzono w tym samym skoroszycie co minutki.

Rozważono również aspekt wpływu projektu na środowisko. Przede wszystkim, zakładając zaangażowanie ze strony sektora medycznego, jest do dosyć istotny krok w kierunku uczynienia polskiego lecznictwa bardziej zdalnym. Co za tym idzie, to:

* Redukcja spalin wynikająca z braku konieczności dojazdu pacjentów do najbliższej przychodni pulmonologicznej na wizytę.
* Redukcja kosztów środowiskowych wynikających z prowadzenia stacjonarnej przychodni pulmonologicznej.
* Zmniejszenie zużycia papieru wynikające z przeniesienia części administracji na zdalną bazę danych.

1. TESTOWANIE APLIKACJI

Testowanie przygotowanego przez nas systemu miało miejsce równolegle wraz z jego rozbudowywaniem.

W przypadku modelu, początkowym problemem była nierównomierność zbioru treningowego co powodowało niezbyt trafną klasyfikację.

Po zastosowaniu augmentacji i rozszerzeniu zbioru treningowego, udało się zachować równomierność występowania poszczególnych chorób w zbiorze uzyskując skuteczność na poziomie około 90%.

W ramach testowania aplikacji, poddano próbie funkcjonalności poszczególnych ról użytkowników w celu sprawdzenia poprawności ich działania w różnych warunkach. Na równi z dołączaniem do aplikacji nowych funkcjonalności, sprawdzano również poprawność funkcjonowania tych starych na bieżąco naprawiając wszelkie napotkane usterki.

Dość problematyczna okazała się również komunikacja między modelem i aplikacją. Po wstępnym zaimplementowaniu, wymagała ona kilu dni napraw, aby ostatecznie zapewnić skuteczność dla wszelkich przewidzianych przez nas scenariuszy testowych.

Problem stanowiło między innymi odpowiednie dostosowanie ścieżek do plików w dostarczenia użytkownikom niezawodnego systemu zdolnego do pracy w różnych warunkach.

1. WNIOSKI I PERSPEKTYWY ROZOWJU

Celem projektu kompetencyjnego było stworzenie w Python Keras modelu klasyfikacji chorób oddechowych na podstawie próbek audio szmerów płuc oraz przygotowanie aplikacji .NET umożliwiającej użytkownikom komfortowe wykorzystanie tego modelu.

W celu przetworzenia dźwięku w wytrenowanym przez nas modelu sztucznej inteligencji, konieczne było przekształcenie plików audio do postaci spektrogramów MEL, na bazie których dokonywano dalszej klasyfikacji. Do przeprowadzenia tej konwersji, wykorzystano bibliotekę Librosa.

Aplikacja zakładała możliwość organizowania zdalnych sesji lekarzy z ich pacjentami. Wyróżniono w niej następujące role:

* Administratora
* Lekarza
* Pacjenta
* Użytkownika niezalogowanego

Role w aplikacji zrealizowano przy użyciu frameworka .NET Identity.

Zadaniem Administratora było tworzenie i zarządzanie kontami.

Konto lekarza umożliwiało tworzenie kont dla pacjentów których dany lekarz leczył. Pozwalało również na weryfikowanie oceny modelu predykcyjnego oraz wypisywanie kart zdrowia, w których informowano o dalszym przebiegu terapii i przepisanych lekach.

Rola pacjenta pozwalała na udostępnianie za pomocą aplikacji nagrań, które analizowane były kolejno przez model predykcyjny, a później lekarza prowadzącego. Pacjenci mieli również wgląd w historię swoich nagrań oraz kart zdrowia.

Użytkownik niezarejestrowany miał możliwość udostępnienia nagrania w celu analizy przez model predykcyjny. Wynik modelu nie był jednak weryfikowany przez lekarza ani zapisywany w bazie danych.

Stworzona w ramach systemu baza danych przechowywała informacje na temat kont użytkowników, nagrań przesyłanych przez pacjentów oraz kart zdrowia generowanych przez lekarzy.

Aby zintegrować model z aplikacją, w aplikacji wywoływano skrypt w języku Python odpowiedzialny za przetworzenie dźwięku i klasyfikację potencjalnych chorób. Miało to miejsce w momencie udostępniania nagrania przez pacjenta lub też użytkownika niezalogowanego.

Następnie wynik modelu zwracano do aplikacji LungMed skąd nagranie trafiało do bazy danych opatrzone wynikiem analizy modelu gdzie mogło zostać zweryfikowane przez lekarza prowadzącego. Lokalna kopia nagrania wykonana na serwerze aplikacji była kasowana zaraz po zapisaniu nagrania w bazie danych.

W ramach organizacji systemu pracy, wykorzystano program Excel w celu sporządzenia wykresu Gantta i prowadzenia cotygodniowych minutek.

Projekt zawierał elementy przetwarzania dźwięku, projektowania i trenowania sieci neuronowej zapewniając jednocześnie intuicyjny i wygodny interfejs w postaci aplikacji .NET. Możliwa jest również jego dalsza rozbudowa, choćby poprzez dodanie możliwości wyboru modelu klasyfikacji choroby, zastosowanie kilku różnych modeli w celu zwiększenia dokładności klasyfikacji czy zaimplementowanie analizy porównawczej dla trudno-klasyfikowalnych nagrań.

Zarówno .NET jak i Python przechodzą obecnie fazę dynamicznego rozwoju co również ułatwia modyfikację czy też rozbudowę zaproponowanego przez nas rozwiązania.

ZAŁĄCZNIKI

* Link do repozytorium [GitHub](https://github.com/mylosz01/lung_diseases_detection)

BIBLIOGRAFIA

1. **Sound Dr: Reliable Sound Dataset and Baseline Artificial Intelligence System for Respiratory Illnesses**​  
   *Truong V. Hoang, Quang H. Nguyen, Cuong Q. Nguyen, Phong X. Nguyen, Hoang D. Nguyen*​
2. **Deep learning-based lung sound analysis for intelligent stethoscope**​  
   *DongMin Huang ,‑ Jia Huang , Kun Qiao , Nan‑Shan Zhong, Hong‑Zhou Lu,  Wen‑Jin Wang*​
3. **Keras Deep Learning Cookbook*​***  
   *Rajdeep Dua,*Manpreet Singh Ghotra​
4. **Generating Synthetic Audio Data for Attention-Based Speech Recognition Systems**​  
   *Nick Rossenbach, Albert Zeyer, Ralf Schlüter, Hermann Ney*​
5. **Audio Data Augmentation: Techniques and Methods**​  
   *José miguel Herrera*​
6. **Automatic Classification of Bird Sounds: Using MFCC and Mel Spectrogram Features with Deep Learning**​  
   *Silvestre Carvalho and Elsa Ferreira Gomes*​
7. **Comparative Study of Mfcc and Mel Spectrogram for Raga Classification Using CNN**​  
   *Dipti Joshi, Jyoti Pareek, Pushkar Ambatkar*
8. **Single-Line to Ground-Fault Detection for Unit Generator-Transformer Based on Wavelet Transform and Neural Networks​***Ahmad Rizal Sultan, Mohd Wazir Mustafa, Makmur Saini​*
9. **INTERNAL FAULT DETECTION IN THREE PHASE TRANSFORMER USING MACHINE LEARNING METHODS​***Ravi Shankar Chauhan​*
10. **Generalized Regression Neural Network and Wavelet Transform for Transformer Protection​***Jonnavarapu Narapareddy, Harish Balaga​*
11. [**Get started with ASP.NET Core MVC**](https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio)
12. [**Overview of ASP.NET Core authentication**](https://learn.microsoft.com/en-us/aspnet/core/security/authentication/?view=aspnetcore-8.0)
13. [**Introduction to Identity on ASP.NET Core**](https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-8.0&tabs=visual-studio)
14. **Pro ASP.NET Core Identity: Under the Hood with Authentication and Authorization in ASP.NET Core 5 and 6 Applications**  
    *Adam Freeman*
15. **Application of semi-supervised deep learning to lung sound analysis​***Daniel Chamberlain, Rahul Kodgule, Daniela Ganelin, Vivek Miglani, Richard Ribón Fletcher​*
16. **Diagnosing Chronic Obstructive Pulmonary Disease Based on Breathing Sound Using Machine Learning​***Thanh Han-Trong​*
17. **A comparative analysis of various respiratory sound denoising methods​***Gwo-Ching Chang*

WKŁAD W PRACĘ NAD PROJEKTEM

**Marianna Tybura**, Studentka 5 semestru Informatyki na WEEIA

* Projekt i realizacja aplikacji LungMed
* Stworzenie bazy danych dla aplikacji
* System logowania dla użytkowników
* System udostępniania plików

**Zuzanna Zając**, Studentka 5 semestru Informatyki na WEEIA

* Projekt i realizacja aplikacji LungMed
* Stworzenie bazy danych dla aplikacji
* System logowania dla użytkowników
* System udostępniania plików

**Miłosz Wyrębkiewicz**, Student 5 semestru Informatyki na WEEIA

* Stworzenie modelu predykcyjnego
* Integracja modelu predykcyjnego z aplikacją LungMed

**Krzysztof Kaniuka**, Student 5 semestru Informatyki na WEEIA

* Integracja modelu predykcyjnego z aplikacją LungMed
* Protokołowanie i raportowanie postępów pracy
* Rozeznanie literaturowe
* Rozeznanie rynkowe

**Igor Tylak**, Student 5 semestru Informatyki na WEEIA

* Wsparcie przy tworzeniu modelu predykcyjnego
* Trenowanie i testowanie modelu predykcyjnego