

# Development of a classifier model based on gradient enhancement for diagnosis of cardiac arrhythmia from ECG data

Kinga Polak<sup>1</sup>, Tomasz Zajac<sup>2\*</sup>

<sup>1</sup> Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, kingapolak20@gmail.com

<sup>2</sup> mgr inż., Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, tzajac@ubb.edu.pl

\* Corresponding author, kingapolak20@gmail.com

**Abstract:** This paper presents the process of creating and optimizing classification models based on the Extreme Gradient Boosting algorithm for ECG data analysis, enabling the differentiation of healthy patients from patients with cardiac arrhythmia. The optimization was performed using Grid Search and Randomized Search methods to find the best combinations of hyperparameters that maximize the effectiveness of the models. Additionally, the cross-validation technique was used to obtain a more reliable assessment of the models' performance and to better understand their ability to generalize to new data.

**Keywords:** XGBoost, Grid Search, Randomized Search, Cross-validation, model evaluation metrics, KNN, SVM, classification, hyperparameter optimization

# Opracowanie modelu klasyfikatora opartego na wzmacnianiu gradientu w celu diagnozowania arytmii serca na podstawie danych EKG

Kinga Polak<sup>1\*</sup>, Tomasz Zajac<sup>2\*</sup>

<sup>1</sup> Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, kingapolak20@gmail.com

<sup>2</sup> mgr inż., Uniwersytet Bielsko-Bialski, Wydział Budowy Maszyn i Informatyki, tzajac@ubb.edu.pl

\* Corresponding author, kingapolak20@gmail.com

**Streszczenie:** W niniejszej pracy zaprezentowano proces tworzenia oraz optymalizacji modeli klasyfikacyjnych opartych na algorytmie Extreme Gradient Boosting do analizy danych EKG, umożliwiających rozróżnienie pacjentów zdrowych od pacjentów z arytmia serca. Optymalizację przeprowadzono przy użyciu metod Grid Search oraz Randomized Search, w celu znalezienia najlepszych kombinacji hiperparametrów, które maksymalizują skuteczność modeli. Dodatkowo zastosowano technikę cross-validation, aby uzyskać bardziej wiarygodną ocenę wydajności modeli i lepiej zrozumieć ich zdolność do generalizowania na nowych danych.

**Słowa kluczowe:** XGBoost, Grid Search, Randomized Search, cross-validation, metryki oceny modelu, KNN, SVM, klasyfikacja, optymalizacja hiperparametrów

## 1. Wstęp

Badanie elektrokardiograficzne (EKG) stanowi jedno z fundamentalnych narzędzi diagnostycznych w kardiologii, umożliwiając ocenę funkcji serca poprzez rejestrowanie i analizę jego aktywności elektrycznej. Wyniki EKG dostarczają

nieocenionych informacji niezbędnych do trafnej diagnozy oraz monitorowania pacjentów z chorobami układu sercowo-naczyniowego. Tradycyjnie analiza danych z badania EKG odbywa się ręcznie, co oznacza, że lekarz samodzielnie opracowuje raport na podstawie zarejestrowanych sygnałów, a następnie przystępuje do ich interpretacji i postawienia diagnozy. Proces ten jest jednak czasochłonny.

Współczesna technologia przynosi wiele innowacyjnych rozwiązań w różnych dziedzinach, w tym także w medycynie. W kardiologii coraz częściej wykorzystuje się algorytmy sztucznej inteligencji (ang. Artificial Intelligence, AI) do analizy sygnałów EKG.

Wykorzystanie sztucznej inteligencji w kardiologii przynosi wiele korzyści. Przede wszystkim AI znacząco redukuje czas potrzebny na analizę sygnałów EKG, co pozwala lekarzom diagnozować większą liczbę pacjentów. Ponadto, dzięki zaawansowanym i wytrenowanym algorytmom, AI może wykrywać anomalie, które mogłyby zostać przeoczone przez człowieka, co zwiększa precyzję diagnostyczną.

Prezentacja pokazuje proces opracowania modeli klasyfikacyjnych opartych na Extreme Gradient Boosting (XGBoost), które umożliwiają rozpoznawanie arytmii serca. W pracy przedstawiono model bez optymalizacji, model po optymalizacji z wykorzystaniem Grid Search oraz model po optymalizacji w ramach Randomized Search.

### Przykład istniejących rozwiązań

Istnieje wiele podobnych rozwiązań, a liczba modeli, które potrafią rozpoznawać różne zaburzenia na podstawie analizy EKG, stale rośnie. Większość z tych modeli opiera się na sztucznych sieciach neuronowych (ang. Convolutional Neural Networks, CNN), które doskonale sprawdzają się w przetwarzaniu danych w postaci obrazów. W Polsce firma Cardiomatics zajmuje się analizą wyników EKG przy użyciu głębokich sieci neuronowych (ang. Deep Neural Networks, DNN). Cardiomatics operuje w chmurze, pozyskując surowe dane EKG z holterów i innych urządzeń EKG, a następnie wspiera lekarzy i kliniki kardiologiczne w ich analizie. Firma posiada liczne dowody kliniczne potwierdzające skuteczność ich metod. [1]

Mimo że znacznie mniej jest rozwiązań opartych na algorytmie XGBoost, ich wyniki również są bardzo obiecujące. Jednym z przykładów istniejących rozwiązań jest model klasyfikatora dla uderzeń serca opracowany przez Association for the Advancement of Medical Instrumentation (AAMI), który mógłby być użyty w zastosowaniach klinicznych. W badaniu zastosowano hierarchiczną metodę klasyfikacji, która opiera się na XGBoost. W ramach metodologii badanie rozpoczęło się od wstępnego przetwarzania sygnału EKG, które obejmowało filtrowanie sygnału w celu eliminacji szumów oraz lokalizowanie poszczególnych uderzeń serca. Następnie zebrano dane na temat różnych cech z tych uderzeń, które zostały poddane selekcji przy użyciu rekurencyjnej eliminacji cech. Hierarchiczna metoda klasyfikacji składała się z dwóch głównych etapów. Na pierwszym etapie uderzenia serca były klasyfikowane jako należące do jednej z dwóch grup: "N-S" (normalne i nadkomorowe) lub "V-F" (komorowe i fuzyjne), przy użyciu klasyfikatora XGBoost. W drugim etapie, dla każdej z grup, stosowano dalszą klasyfikację z użyciem klasyfikatorów progowych i kolejnych klasyfikatorów XGBoost, co umożliwiło dokładniejsze rozróżnienie między poszczególnymi typami uderzeń serca. Wyniki eksperymentu, przeprowadzonego zgodnie ze standardem AAMI, wykazały wysoką skuteczność metody. Czułość dla poszczególnych klas uderzeń serca wynosiła odpowiednio 92,1% dla uderzeń normalnych, 91,7% dla nadkomorowych, 95,1% dla komorowych i 61,6% dla fuzyjnych. Dodatnia wartość predykcyjna dla tych klas była również bardzo wysoka, szczególnie 99,5% dla uderzeń normalnych, co potwierdza użyteczność metody w praktyce klinicznej. [2]

## 2. Opis metody

W celu klasyfikacji arytmii serca na podstawie sygnałów EKG zastosowano modele oparte na Extreme Gradient Boosting, zaawansowanym algorytmie wykorzystującym drzewa decyzyjne. Projekt został zrealizowany w Google Colaboratory, narzędziu umożliwiającym wykonywanie kodu Python bezpośrednio w przeglądarce internetowej.

### 2.1 Opis danych treningowych

Do analizy posłużono się bazą danych MIT-BIH Arrhythmia Database, która zawiera zbiór danych dotyczących osób z różnymi rodzajami arytmii serca oraz osób zdrowych. Zbiór ten został stworzony na podstawie cech wyodrębnionych z odprowadzeń II i V5.

Dane składają się z pięciu kategorii: N (Normal), S (Supraventricular ectopic beat), V (Ventricular ectopic beat), F (Fusion beat) oraz Q (Unknown beat). Kategorie S, V, F i Q wskazują na różne nieprawidłowości w rytmie serca. W sumie zbiór zawiera 17 cech dla każdego odprowadzenia i składa się z 100689 rekordów, z czego: 90083 to dane oznaczone jako

Normal, 7009 to Ventricular ectopic beat, 2779 to Supraventricular ectopic beat, 803 to Fusion beat, a 15 to Unknown beat.

Na początku proces przetwarzania danych polegał na zgrupowaniu wszystkich kategorii związanych z arytmia w jedną wspólną klasę „arrhythmia”. Następnie zbiory zostały zbalansowane, co oznacza, że z każdej klasy pobrano po 10000 próbek. W celu dalszej analizy, klasy zostały zakodowane na wartości 0 i 1, gdzie 0 oznaczało klasę normal, a 1 klasę arrhythmia.

Dalszym krokiem było przypisanie zmiennych do modelowania ( $x$ ) oraz do klasyfikacji ( $y$ ). Na koniec zbior danych został podzielony na zestaw treningowy i testowy, przy czym 70% danych przeznaczono do treningu, a pozostałe 30% do testowania modelu.

## 2.2 Architektura modelowania

Extreme Gradient Boosting to zaawansowany algorytm uczenia maszynowego, który opiera się na drzewach decyzyjnych wzmacnianych gradientem. Kluczową zasadą działania XGBoost jest stopniowe budowanie drzew decyzyjnych, gdzie każde kolejne drzewo jest trenowane na podstawie pozostałości błędów z poprzedniego drzewa. W trakcie tego procesu algorytm integruje wartości przewidywane przez wcześniej wyszkolone drzewa, aby poprawić ogólną dokładność modelu. [3][13]

Jedną z kluczowych zalet XGBoosta jest sposób, w jaki wykorzystuje funkcję celu w procesie uczenia. Funkcja ta składa się z dwóch głównych komponentów: funkcji straty oraz regularyzacji. Funkcja straty mierzy różnicę między przewidywanymi przez model wartościami a rzeczywistymi wartościami, pomagając modelowi poprawić swoje prognozy. Z kolei termin regularyzacji kontroluje złożoność modelu, zapobiegając nadmiernemu dopasowaniu. Dzięki regularyzacji nieistotne węzły decyzyjne są usuwane, co zmniejsza rozmiar drzewa i poprawia jego zdolność do generalizacji na nowych danych. [3]

Poniżej przedstawiono dane wejściowe oraz kolejne kroki działania algorytmu XGBoost.

- $\{(x_i, y_i)\}_{i=1}^N$  – Zestaw treningowy, gdzie  $x_i$  to cechy dla próbki  $i$ , a  $y_i$  to odpowiednia etykieta lub wartość.
- $L(y, F(x))$  – Funkcja straty mierzy, jak dobre są przewidywania modelu  $F(x)$  w porównaniu do prawdziwych wartości  $y$ .
- $M$  – Liczba słabych uczniów (np. liczba drzew decyzyjnych, które chcemy wytrenować).
- $\alpha$  – Współczynnik szybkości uczenia, który kontroluje, jak duże są kroki aktualizacji modelu (tzw. współczynnik uczenia).

W kroku pierwszym model bazowy zostaje zainicjalizowany stałą wartością.

$$\hat{f}_{(0)}(x) = \arg_{\theta} \min \sum_{i=1}^N L(y_i, \theta) \quad (1)$$

Następnie zainicjalizowana jest pętla dla każdego drzewa. Pętla zaczyna się od ustawienia zmiennej  $m$  na 1 i trwa aż do  $M$ , gdzie  $M$  to maksymalna liczba słabych uczniów, których chcemy wytrenować. Celem jest iteracyjne dodawanie nowych drzew do modelu, gdzie każde drzewo jest dostosowywane do reszty błędu modelu, który został wytrenowany w poprzednich iteracjach.

W każdym powtórzeniu pętli obliczany jest gradient i Hesjan.

$$\hat{g}_m(x_i) = \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f(x)=\hat{f}_{(m-1)}(x)} \quad (2)$$

Gradient dla każdej próbki  $i$  to pochodna funkcji straty względem aktualnej wartości przewidywania  $f(x_i)$ . Gradient wskazuje, w jakim kierunku i jak bardzo należy poprawić model, aby zminimalizować funkcję straty.

$$\hat{h}_m(x_i) = \left[ \frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2} \right]_{f(x)=\hat{f}_{(m-1)}(x)} \quad (3)$$

Hesjan to druga pochodna funkcji straty względem  $f(x_i)$ , która informuje o krzywiznie funkcji straty. Hesjan pomaga w bardziej precyzyjnej optymalizacji, pozwalając na lepsze dostosowanie kroku aktualizacji.

$$\hat{\phi}_m = \arg_{\phi \in \Phi} \min \sum_{i=1}^N \frac{1}{2} \hat{h}_m(x_i) \left[ \phi(x_i) - \frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} \right]^2 \quad (4)$$

Kolejno dopasowywany jest nowy uczeń bazowy. Dochodzi tu do optymalizacji słabego ucznia, który minimalizuje funkcję związaną z gradientami i Hesjanami. Czyli zostaje wytrenowane nowe drzewo  $f_m$ , które poprawi aktualny model minimalizując błędy.

$$\hat{f}_m(x) = \alpha \hat{\phi}_m(x) \quad (5)$$

Po znalezieniu najlepszego drzewa decyzyjnego jest ono skalowane przez współczynnik uczenia  $\alpha$ , aby kontrolować wielkość zmiany w modelu.

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) - \hat{f}_m \quad (6)$$

Następnie model ulega aktualizacji.

$$\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^M \hat{f}_m(x) \quad (7)$$

Ostateczny model to suma wszystkich stworzonych drzew, gdzie każdy słaby uczoney jest skalowany przez współczynnik uczenia  $\alpha$ .

Podczas tworzenia modelu XGBoost możemy regulować różne parametry, aby zoptymalizować jego działanie. Jednym z takich parametrów jest `max_depth`, który kontroluje maksymalną głębokość drzew decyzyjnych. Głębsze drzewa są bardziej złożone, jednak zbyt duża głębokość może prowadzić do przeuczenia modelu. W XGBoost model składa się z sekwencji drzew, gdzie każde kolejne drzewo poprawia błędy popełnione przez poprzednie. Liczbę drzew można kontrolować za pomocą parametru `n_estimators`. [4]

XGBoost wykorzystuje algorytm optymalizacji oparty na gradientach błędów, co oznacza, że na każdym etapie treningu model dąży do minimalizacji błędu prognoz. Parametr `learning_rate` reguluje, jak silnie każde kolejne drzewo wpływa na ostateczny wynik modelu. Niższe wartości tego parametru spowalniają proces uczenia, ale umożliwiają bardziej precyzyjne dopasowanie modelu do danych. [4]

Aby zapobiec przeuczeniu, stosuje się techniki regularyzacji za pomocą parametrów `reg_alpha` i `reg_lambda`. Parametr `reg_alpha` odpowiada za regularyzację L1 (tzw. Lasso), która nakłada kary na duże wartości wag, co prowadzi do zmniejszenia liczby aktywnych cech w modelu i pozwala wyeliminować te mniej istotne. Z kolei `reg_lambda` odpowiada za regularyzację L2 (tzw. Ridge), która również nakłada kary na duże wartości wag, jednak w inny sposób – karząc ich kwadraty. Dzięki temu wagi maleją, ale zazwyczaj nie są równe zero. [4]

Technika subsamplingu poprawia wydajność modelu i zmniejsza wariancję. Parametr `subsample` kontroluje, jaka część danych jest wykorzystywana do trenowania każdego drzewa, co pomaga chronić model przed nadmiernym dopasowaniem. Możemy także użyć parametru odpowiedzialnego za losowe próbkowanie cech, czyli `colsample_bytree`. Określa on, ile cech zostanie wybranych do budowy każdego drzewa. Dzięki temu proces ten staje się bardziej efektywny, a ryzyko przeuczenia maleje. [4]

Odpowiednia konfiguracja tych parametrów w modelu XGBoost umożliwia znalezienie balansu między złożonością modelu a jego zdolnością do uogólniania wyników, co jest niezwykle istotne dla efektywnej analizy danych.

### 2.3 Metryki oceniające model

Ocena modeli klasyfikacyjnych jest kluczowym etapem, który pozwala zrozumieć, jak skutecznie model przewiduje kategorie. Aby dokładnie ocenić wydajność modelu, wykorzystuje się różne metryki, z których każda dostarcza unikalnych informacji na temat jego działania. Dokładność (ang. accuracy) jest miarą, która określa odsetek wszystkich poprawnych klasyfikacji w stosunku do całkowitej liczby przypadków.

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN)$$

Gdzie:

- TP (ang. True Positives) to liczba przypadków, które zostały poprawnie sklasyfikowane jako pozytywne.
- TN (ang. True Negatives) to liczba przypadków, które zostały poprawnie sklasyfikowane jako negatywne.
- FP (ang. False Positives) to liczba przypadków, które zostały niepoprawnie sklasyfikowane jako pozytywne, mimo że są negatywne.
- FN (ang. False Negatives) to liczba przypadków, które zostały niepoprawnie sklasyfikowane jako negatywne, mimo że są pozytywne. [5]

Kolejną metryką oceniającą jest precyzja (ang. precision). Określa, jaki procent wyników, które model klasyfikacyjny wskazał jako pozytywne, jest rzeczywiście pozytywny. Innymi słowy, precyzja mierzy dokładność klasyfikatora w identyfikowaniu pozytywnych przypadków. Jest to istotne w kontekście sytuacji, w których koszt fałszywych alarmów jest wysoki. Na przykład, w medycynie, precyzja może być kluczowa przy diagnozowaniu chorób, gdzie fałszywie pozytywna diagnoza może prowadzić do niepotrzebnych i potencjalnie szkodliwych zabiegów. Precyzja jest obliczana jako stosunek liczby prawdziwie pozytywnych wyników do sumy prawdziwie pozytywnych i fałszywie pozytywnych wyników, co pozwala zrozumieć, jak niezawodny jest model w swoich pozytywnych przewidywaniach. [6]

$$\text{Precision} = TP / (TP + FP)$$

Czułość (ang. recall), ocenia zdolność modelu do prawidłowego wykrywania wszystkich przypadków pozytywnych. Inaczej ujmując, czułość określa, jaki odsetek rzeczywiście pozytywnych przypadków został poprawnie zidentyfikowany przez klasyfikator. Jest to szczególnie ważne w sytuacjach, gdzie kluczowe jest minimalizowanie liczby pominiętych pozytywnych przypadków.

W diagnostyce medycznej wysoka czułość oznacza, że model skutecznie identyfikuje chorobę u pacjentów, którzy faktycznie ją mają, co jest niezwykle istotne przy poważnych schorzeniach, gdzie brak diagnozy może mieć konsekwencje zdrowotne. Czułość oblicza się jako stosunek liczby prawdziwie pozytywnych przypadków do sumy prawdziwie pozytywnych i fałszywie negatywnych przypadków. [6]

$$\text{Recall} = TP / (TP + FN)$$

F1-Score to miara, która łączy precyzję i czułość w jedną wartość, będąc ich średnią harmoniczną. Jest szczególnie użyteczna w sytuacjach, gdzie ważne jest utrzymanie równowagi między precyzją a czułością, zwłaszcza gdy mamy do czynienia z nie zrównoważonymi klasami. W takich przypadkach model musi skutecznie identyfikować zarówno prawdziwie pozytywne przypadki, jak i minimalizować fałszywe alarmy. [7]

$$F = 2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$$

## 2.4 Proces modelowania

W badaniu stworzono modele XGBoosta, wykorzystując różne techniki optymalizacji. Na początek zbudowano podstawowy model, w którym nie ustalono konkretnych wartości hiperparametrów. Model ten został poddany dziesięciokrotnej walidacji krzyżowej (ang. cross-validation). Walidacja krzyżowa pozwala ocenić, jak model radzi sobie z nowymi danymi, dzieląc dane treningowe na określoną liczbę części (tzw. foldów). W każdej iteracji jedna z części służy jako zbiór testowy, a pozostałe jako zbiór treningowy. Proces ten powtarza się tak długo, aż każda część zostanie użyta jako zbiór testowy dokładnie raz. Dzięki temu model jest testowany na różnych podziałach danych, co umożliwia dokładniejszą ocenę jego wydajności niż w przypadku jednorazowego testowania na jednym zbiorze. [8]

Walidacja krzyżowa pozwala także lepiej ocenić zdolność modelu do generalizacji, czyli przewidywania wyników na nowych, nieznanych wcześniej danych. Analiza wyników uzyskanych z różnych podziałów danych pozwala zidentyfikować ewentualne problemy, takie jak nadmierne dopasowanie (ang. overfitting). W ten sposób można upewnić się, że model będzie w stanie skutecznie prognozować wyniki również na nowych danych. [8]

Do optymalizacji modelu wykorzystano techniki takie jak Grid Search oraz Randomized Search. Grid Search polega na przeszukiwaniu wszystkich możliwych kombinacji hiperparametrów w celu znalezienia optymalnych wartości dla danego modelu [9][12]. Dla każdej kombinacji model jest trenowany i testowany, co pozwala na dokładne zidentyfikowanie najlepszego zestawu hiperparametrów. W badaniu, w ramach Grid Search, użyto następujących hiperparametrów:

- n\_estimators: [100, 200, 300, 400],
- learning\_rate: [0.05, 0.1, 0.3],
- max\_depth: [3, 5, 7],
- reg\_alpha: [0, 0.1, 1],
- reg\_lambda: [1, 3, 5],
- colsample\_bytree: [0.5, 0.7, 0.9],
- subsample: [0.5, 0.7, 0.9].

Randomized Search działa nieco inaczej niż Grid Search. Zamiast testować i oceniać wszystkie możliwe kombinacje hiperparametrów, losowo wybiera zestawy hiperparametrów z określonego zakresu [9]. Takie podejście może być bardziej efektywne, zwłaszcza przy dużej liczbie możliwych kombinacji. W badaniu, podczas Randomized Search, wykorzystano następujące hiperparametry:

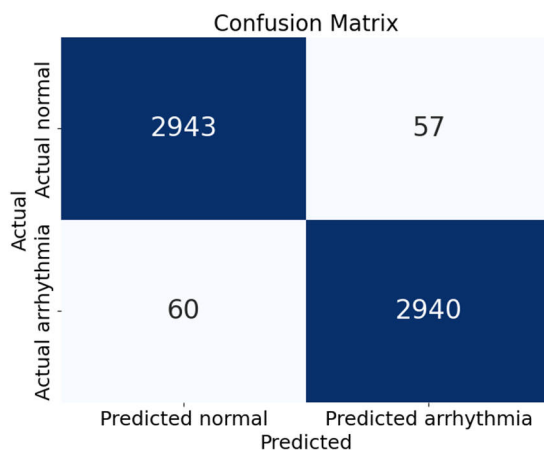
- `n_estimators`: [50, 100, 200, 300, 400, 500],
- `learning_rate`: [0.001, 0.01, 0.05, 0.1, 0.3, 0.5],
- `max_depth`: [2, 3, 4, 5, 6, 7, 8],
- `reg_alpha`: [0, 0.01, 0.1, 1, 2],
- `reg_lambda`: [1, 2, 3, 5, 10],
- `colsample_bytree`: [0.3, 0.5, 0.7, 0.9],
- `subsample`: [0.3, 0.5, 0.7, 0.9].

Dodatkowo, wartość parametru `n_iter` została ustawiona na 30, co oznacza, że model został przetestowany na trzydziestu losowo wybranych zestawach hiperparametrów. Aby zagwarantować, że wyniki będą powtarzalne, w modelu opracowanym przy pomocy Randomized Search wprowadzono parametr `random_state` z wartością 123. Umożliwia to każdej iteracji treningu korzystanie z tego samego źródła losowości, co przyczynia się do uzyskania jednolitych rezultatów. W procesach optymalizacji z wykorzystaniem Grid Search oraz Randomized Search zastosowano także pięciokrotną walidację krzyżową. Takie podejście przyspiesza proces optymalizacji, jednocześnie zapewniając rzetelność uzyskiwanych wyników.

Dla porównania wyników opracowano modele K-najbliższych sąsiadów (ang. K-Nearest Neighbors, KNN) oraz Maszyny Wektorów Nośnych (ang. Support Vector Machine, SVM). Algorytm KNN klasyfikuje dane na podstawie przynależności ich najbliższych sąsiadów do poszczególnych klas, podczas gdy SVM wyznacza najlepszą hiperpłaszczyznę oddzielającą różne klasy danych [10] [11].

### 3. Rezultaty

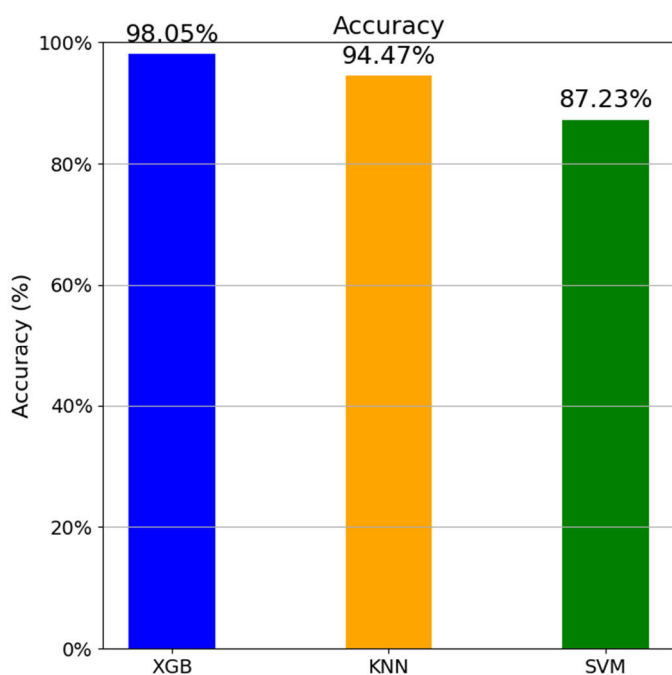
W eksperymencie model XGBoost, bez optymalizacji hiperparametrów, po przeprowadzeniu walidacji krzyżowej uzyskał średnią dokładność wynoszącą 97,98% na zbiorze walidacyjnym. Wyniki na niezależnym zbiorze testowym wykazały jeszcze wyższą dokładność na poziomie 98,05%. Na poniższej macierzy pomyłek (Rysunek 1) możemy zobaczyć, że spośród 3000 prawdziwie zdrowych osób model poprawnie sklasyfikował 2943 przypadki, popełniając błąd w jedynie 57 przypadkach. W grupie 3000 osób chorych model prawidłowo zaklasyfikował 2940 osób, błędnie przypisując zdrowie w 60 przypadkach.



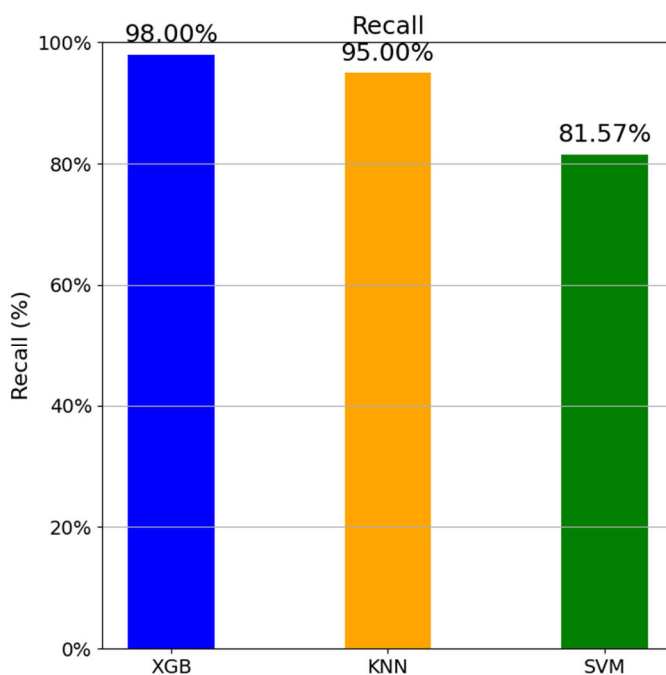
**Rysunek 1.** Macierz pomyłek dla modelu XGBoosta

Pozostałe metryki również potwierdziły wysoką skuteczność modelu XGBoosta. Wartości te przedstawiają się następująco: recall wyniósł 98,00%, precision 98,10%, a F1 score 98,05%.

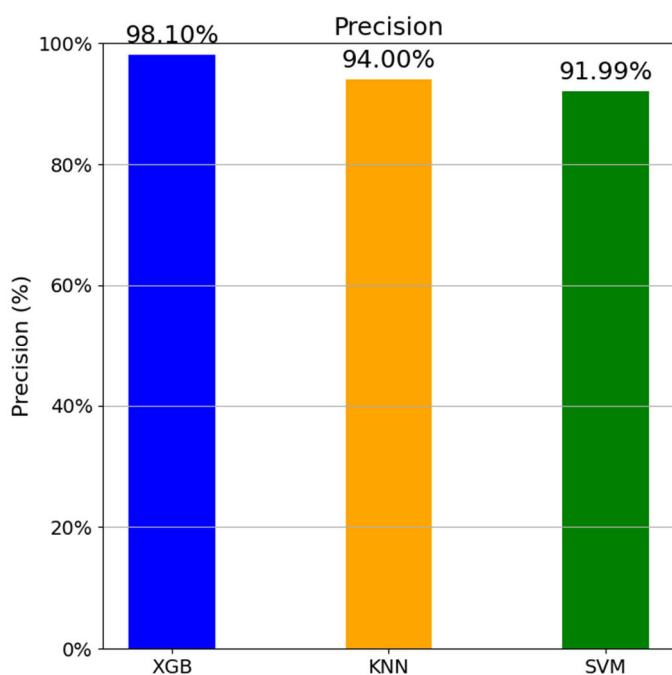
Wyniki dla modelu KNN były niższe w porównaniu do XGBoost. Model KNN uzyskał accuracy na poziomie 94,47%, recall wyniósł 95,00%, precision 94,00%, a F1 score osiągnął wartość 94,50%. Z kolei model SVM osiągnął jeszcze niższe wyniki, z dokładnością wynoszącą 87,23%, recall na poziomie 81,57%, precision 91,99% oraz F1 score równy 86,47%. Wartości accuracy dla wszystkich trzech modeli możemy zobaczyć na (Rysunek 2), wartości recall na (Rysunek 3), wartości precision na (Rysunek 4), a wartości dla F1 score na (Rysunek 5).



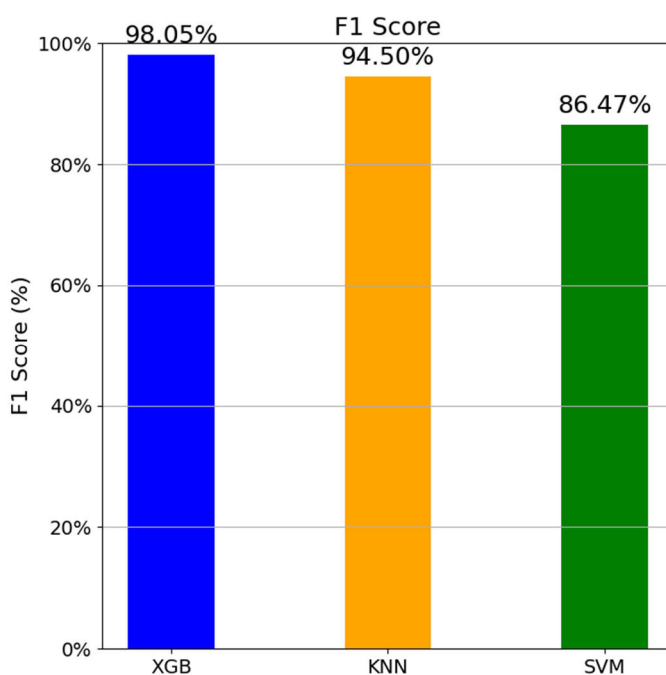
Rysunek 2. Wykres przedstawiający wyniki accuracy



Rysunek 3. Wykres przedstawiający wyniki recall



Rysunek 4. Wykres przedstawiający wyniki precision



Rysunek 5. Wykres przedstawiający wyniki F1-score

Na podstawie tych wyników można stwierdzić, że model XGBoost przewyższył inne algorytmy, osiągając najwyższe wartości we wszystkich kluczowych metrykach.

Model XGBoost, po zastosowaniu Grid Search, osiągnął lepsze wyniki w porównaniu do wersji bez optymalizacji hiperparametrów. Najlepsze znalezione hiperparametry to:

- `colsample_bytree`: 0.7,
- `learning_rate`: 0.3,
- `max_depth`: 7,
- `n_estimators`: 300,
- `reg_alpha`: 0.1,

- reg\_lambda: 3,
- subsample: 0.7.

Dzięki tym parametrom model uzyskał najwyższą dokładność na zbiorze walidacyjnym wynoszącą 97,99%.

Dalsze testowanie modelu na niezależnym zbiorze testowym wykazało, że jego dokładność wyniosła 98,32%, co świadczy o jego wysokiej skuteczności oraz umiejętności generalizacji na nowych danych.

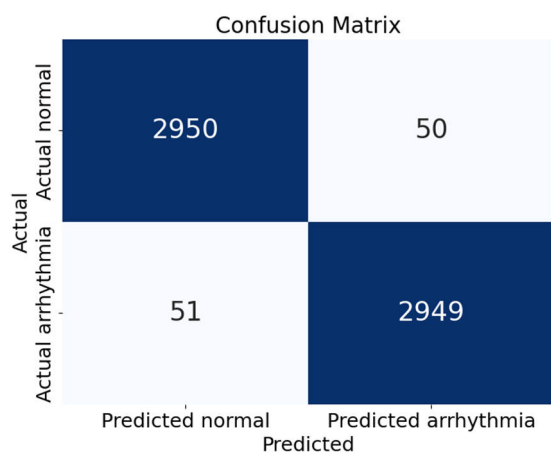
Inne metryki oceny również wykazały wysoką skuteczność. Wartość precyzji wyniosła 98,33%, czułość osiągnęła 98,30%, natomiast F1-Score wyniósł 98,32%.

Analiza macierzy pomyłek dla najlepszego modelu XGBoost, uzyskanego po zastosowaniu techniki Grid Search, umożliwiła dokładniejszą ocenę skuteczności klasyfikacji obu klas (Rysunek 6). Zauważono poprawę w liczbie prawidłowo sklasyfikowanych przypadków chorych. Spośród 3000 osób chorych model błędnie zaklasyfikował jedynie 51 przypadek. Również liczba osób prawidłowo zaklasyfikowanych jako zdrowe wzrosła. Na 3000 zdrowych osób model popełnił błąd w zaledwie 50 przypadkach. Model po zastosowaniu Randomized Search osiągnął także imponujące wyniki. W procesie optymalizacji modelu zidentyfikowano zestaw najlepszych hiperparametrów, które pozwoliły na uzyskanie równie wysokiej dokładności. Optymalne wartości hiperparametrów to:

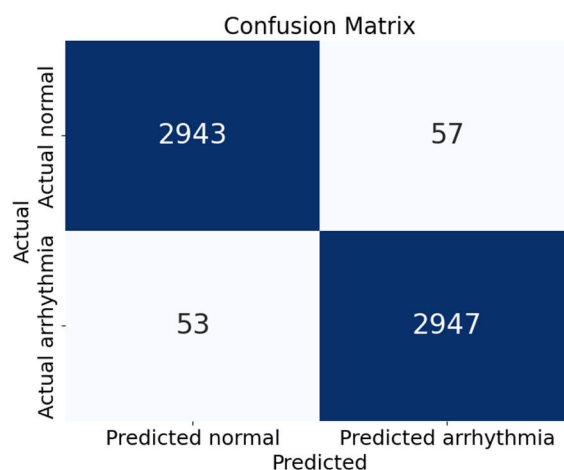
- subsample: 0.9,
- reg\_lambda: 2,
- reg\_alpha: 0,
- n\_estimators: 500,
- max\_depth: 7,
- learning\_rate: 0.3,
- colsample\_bytree: 0.5.

Najlepsza dokładność modelu na zbiorze walidacyjnym wyniosła 97,89%.

Po przetestowaniu modelu na niezależnym zbiorze testowym jego wydajność pozostała na bardzo wysokim poziomie. Dokładność wyniosła 98,17%, co sugeruje, że model dobrze generalizuje na nowych danych i nie jest nadmiernie dopasowany do zbioru treningowego. Dodatkowo precyzja osiągnęła wartość 98,10%, czułość wyniosła 98,23%, a F1-score osiągnął 98,17%. Na macierzy pomyłek można zaobserwować, że model błędnie zaklasyfikował 53 osoby jako zdrowe oraz 57 osób jako osoby z arytmią (Rysunek 7).



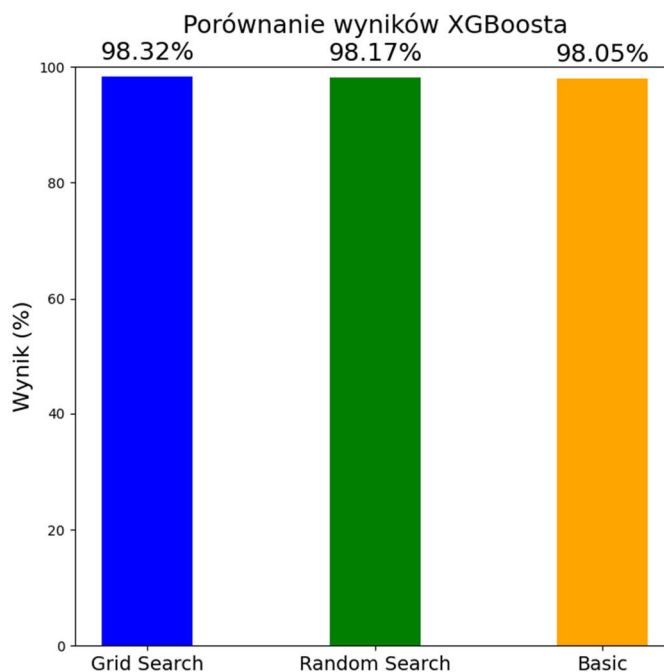
Rysunek 6. Macierz pomyłek dla modelu XGBoosta



Rysunek 7. Macierz pomyłek dla modelu XGBoosta

Poniżej możemy zobaczyć wykres z poszczególnymi wynikami accuracy na danych testowych dla modelu XGBoosta bez optymalizacji parametrów, po zastosowaniu Grid Search i zastosowaniu Randomized Search (Rysunek 8).





Rysunek 8. Wykres porównujący wyniki dokładności modeli XGBoosta

#### 4. Możliwości rozwoju modelu

Przedstawione modele mają potencjał do dalszego rozwoju. W ramach Grid Search i Randomized Search istnieje możliwość zwiększenia liczby hiperparametrów oraz rozszerzenia ich zakresu. Taki krok może prowadzić do znalezienia skuteczniejszej kombinacji hiperparametrów, co z kolei może przyczynić się do lepszego generalizowania modelu na nowych danych. Dodatkowo warto rozważyć zastosowanie alternatywnych metod optymalizacji, takich jak optymalizacja bayesowska, która jest bardziej zaawansowana i może efektywniej wyszukiwać najlepsze hiperparametry.

#### 5. Podsumowanie

Podsumowując, modele XGBoosta, zarówno w wersji podstawowej, jak i po zastosowaniu technik optymalizacji Grid Search oraz Randomized Search, wykazały się niezwykłą skutecznością w klasyfikacji arytmii serca. Dzięki wysokim wskaźnikom dokładności oraz zdolności do generalizacji na nowych zbiorach danych modele te mają potencjał do praktycznego wykorzystania w dziedzinie medycyny, wspierając diagnostykę arytmii. Dodatkowo ich szybkość operacyjna oraz stabilność w różnych scenariuszach testowych czynią je cennym narzędziem w codziennej pracy klinicznej. W porównaniu z innymi algorytmami, takimi jak KNN i SVM, modele XGBoost osiągnęły lepsze wyniki, co podkreśla ich unikalne walory. Ich zastosowanie może wspomagać lekarzy w podejmowaniu bardziej trafnych decyzji dotyczących leczenia pacjentów z arytmia, co ostatecznie może przyczynić się do podniesienia jakości opieki zdrowotnej. W przyszłości warto rozważyć dalsze badania nad optymalizacją tych modeli oraz ich integracją z innymi systemami diagnostycznymi, co może zwiększyć ich efektywność i zastosowanie w praktyce medycznej.

#### LITERATURA

1. Serwis internetowy: <https://cardiomatics.com/pl/> (dostęp: 15.10.2024)
2. Shi Haotian, Wang Haoren, Huang Yixiang, Zhao Liqun, Qin Chengjin, Liu Chengliang. A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification. *Computer Methods and Programs in Biomedicine*, Volume 171, Pages 1-10, April 2019
3. Serwis internetowy: <https://datascience.eu/pl/programowanie-komputerowe/xgboost/> (dostęp: 15.10.2024)
4. Serwis internetowy: <https://medium.com/@rithpansanga/the-main-parameters-in-xgboost-and-their-effects-on-model-performance-4f9833cac7c/> (dostęp: 15.10.2024)
5. Wojciech Jaworski, Instytut Informatyki Uniwersytet Warszawski. *Miary jakości*. Pages 1-3

6. Serwis internetowy: <https://statystyczny.pl/macierz-bledow-raport-dokladnosc-czulosc-precyzja/> (dostęp: 15.10.2024)
7. Serwis internetowy: <https://www.deepchecks.com/f1-score-accuracy-roc-auc-and-pr-auc-metrics-for-models/> (dostęp: 15.10.2024)
8. Serwis internetowy: <https://datascience.eu/pl/uczenie-maszynowe/k-krotna-walidacja-krzyzowa/> (dostęp: 15.10.2024)
9. Siddharth Ghosh. Hyper Parameter Tuning Techniques – Grid Search, Bayesian & Halving – Wonders of ML Realm
10. Serwis internetowy: [https://ichi.pro/pl/wszystko-o-czym-nalezy-wiedziec-support-vector-machines-svm-27379677884233#google\\_vignette](https://ichi.pro/pl/wszystko-o-czym-nalezy-wiedziec-support-vector-machines-svm-27379677884233#google_vignette) (dostęp: 15.10.2024)
11. Serwis internetowy: <https://www.geeksforgeeks.org/k-nearest-neighbours/> (dostęp: 15.10.2024)
12. A.Prawin, G. Saranya. Grid Search based Optimum Feature Selection by Tuning hyperparameters for Heart Disease Diagnosis in Machine learning. *The Open Biomedical Engineering Journal*, Volume 17, 2023
13. Jad Botros, Farah Mourad-Chehade, David Laplanche. Explainable multimodal data fusion framework for heart failure detection: Integrating CNN and XGBoost. *Biomedical Signal Processing and Control*, Volume 100, Part C, February 2025