

Sieci Neuronowe – laboratorium

Ćwiczenie nr. 4

Kajetan Pynka, 254495

Spis treści

Wstęp teoretyczny.....	3
Porównanie komórek RNN i LSTM.....	4
Rozmiar wymiaru wyjściowego warstwy embeddingu.....	4
Rozmiar wymiaru wyjściowego warstwy embeddingu – 30.....	4
Rozmiar wymiaru wyjściowego warstwy embeddingu – 50.....	7
Rozmiar wymiaru wyjściowego warstwy embeddingu – 80.....	8
Rozmiar wymiaru wyjściowego warstwy embeddingu – 120.....	9
Rozmiar wymiaru wyjściowego warstwy embeddingu – 150.....	10
Wnioski.....	11
Liczba jednostek warstwy rekurencyjnej	12
Liczba jednostek warstwy rekurencyjnej – 30	12
Liczba jednostek warstwy rekurencyjnej – 50	14
Liczba jednostek warstwy rekurencyjnej – 80	15
Liczba jednostek warstwy rekurencyjnej – 120	16
Liczba jednostek warstwy rekurencyjnej – 150	17
Wnioski.....	18
Funkcja aktywacji.....	19
Funkcja aktywacji – tangens hiperboliczny	19
Funkcja aktywacji – sigmoid	21
Funkcja aktywacji – ReLU	22
Wnioski.....	23
Efektywność paddingu	24
Rozmiar paddingu / długość maski.....	24
Rozmiar paddingu / długość maski – 50.....	24
Rozmiar paddingu / długość maski – 80.....	26
Rozmiar paddingu / długość maski – 100.....	27
Rozmiar paddingu / długość maski – 130.....	28
Rozmiar paddingu / długość maski - 150.....	29
Wnioski.....	30
Wnioski.....	31

Wstęp teoretyczny

W ramach ćwiczenia zaimplementowałem modele sieci rekurencyjnej, raz wykorzystujące zwykłe komórki RNN, a w przypadku drugiego komórki LSTM (long short-term memory). Struktura sieci i cały proces nauki został zrealizowany przy użyciu języka Python 3.10 oraz biblioteki 'keras' będącej składową większej biblioteki 'tensorflow'. Oba modele jako funkcję straty zakładały binarną entropię krzyżową oraz używały optymalizatora RMSprop. W ramach ćwiczeń wykonałem następujące eksperymenty: porównanie komórek RNN i LSTM (na podstawie wartości: rozmiaru embeddingu, liczby wymiarów oraz użytej funkcji aktywacji), ocena użyteczności i efektywności paddingu (ze względu na rozmiar maski). Z danych wyciągnięte zostało najpopularniejsze 15000 słów (wdg wewnętrznej częstotliwości występowania danego słowa).

Dla każdego eksperymentu przyjęte zostały następujące wartości hiperparametrów (tam, gdzie dany parametr nie był oczywiście przedmiotem badania):

- Rozmiar paczki = 128
- Liczba epok = 10
- Funkcja aktywacji = ReLU
- Rozmiar wymiaru wejściowego warstwy embeddingu = 15000
- Rozmiar wymiaru wyjściowego warstwy embeddingu = 64
- Rozmiar paddingu / długość maski = 100
- Liczba jednostek warstwy rekurencyjnej = 10

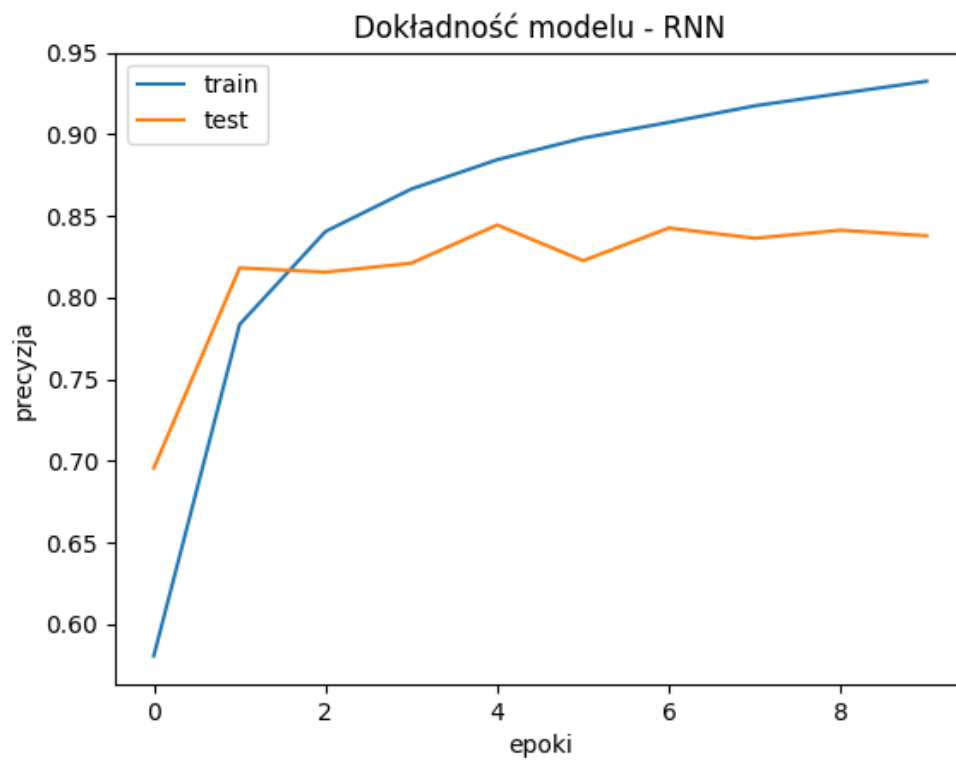
Dla każdego eksperymentu sporządzone zostały wykresy:

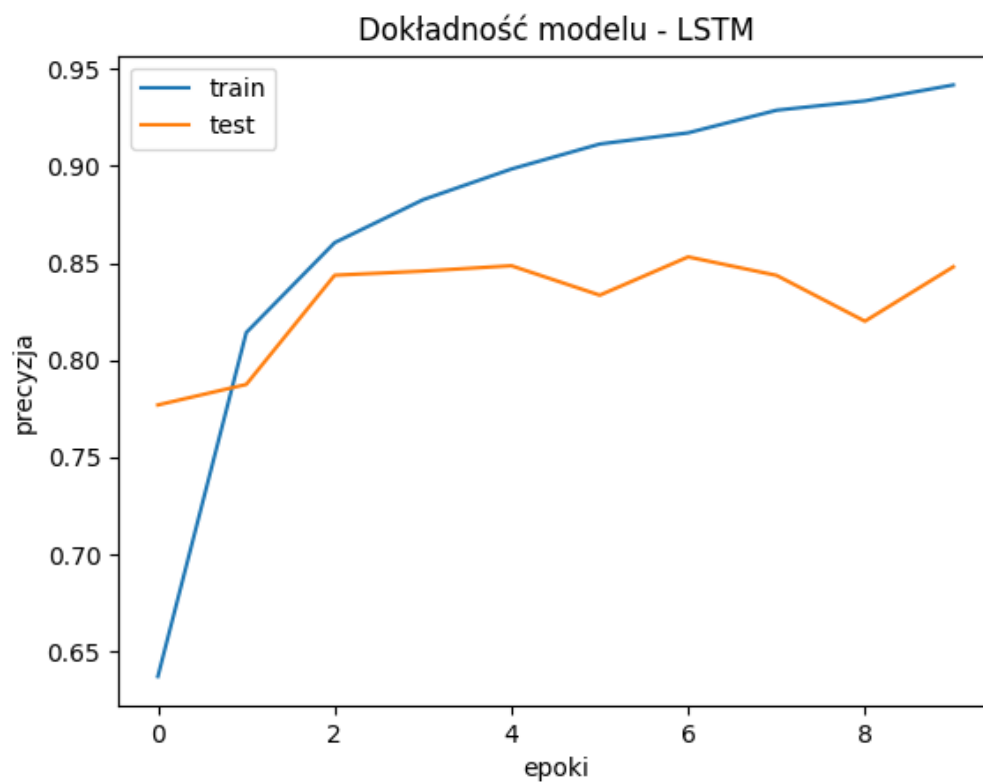
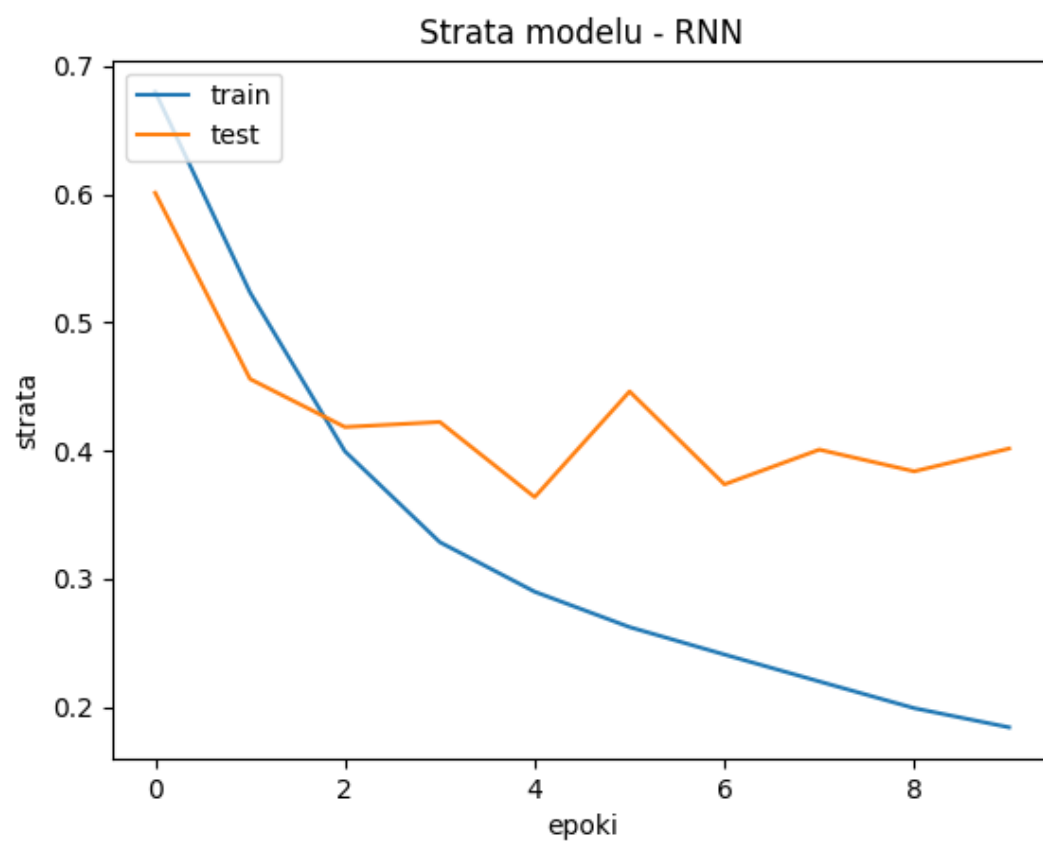
- precyzji/skuteczności modelu (w skali 0-1) na zbiorze treningowym oraz testowym w zależności od liczby epok dla komórek RNN i LSTM
- wartość funkcji straty (binarnej entropii krzyżowej) na zbiorze treningowym oraz testowym w zależności od liczby epok dla komórek RNN i LSTM
- tablicy pomyłek – zawsze po lewej stronie tablica pomyłek dla komórek RNN a po lewej stronie dla komórek LSTM

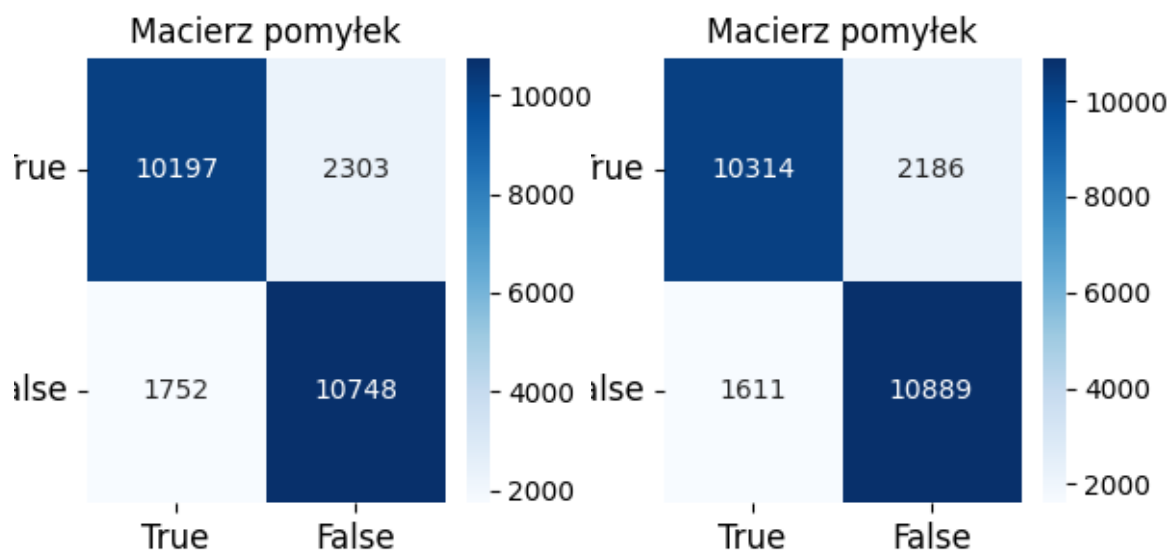
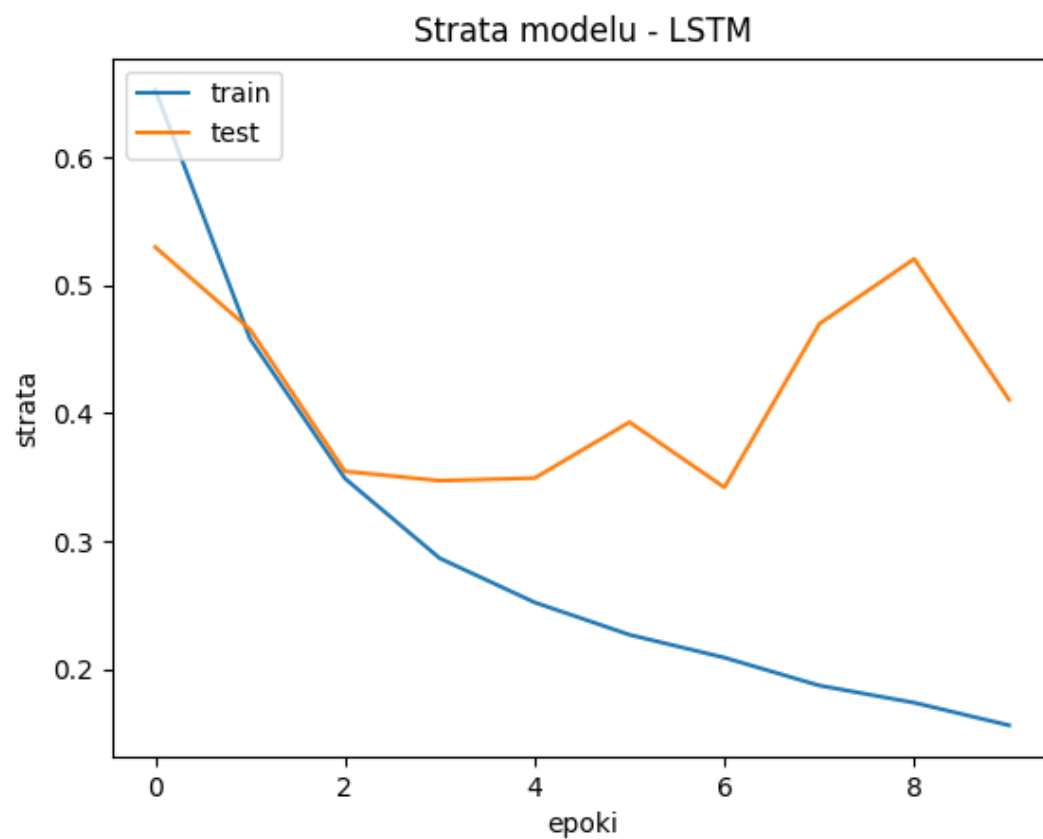
Porównanie komórek RNN i LSTM

Rozmiar wymiaru wyjściowego warstwy embeddingu

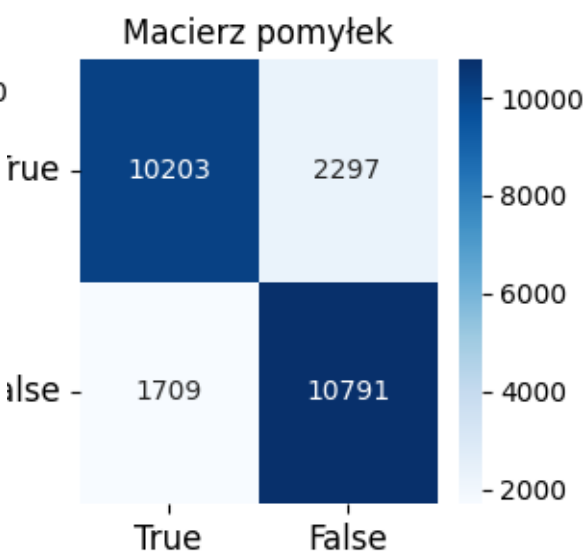
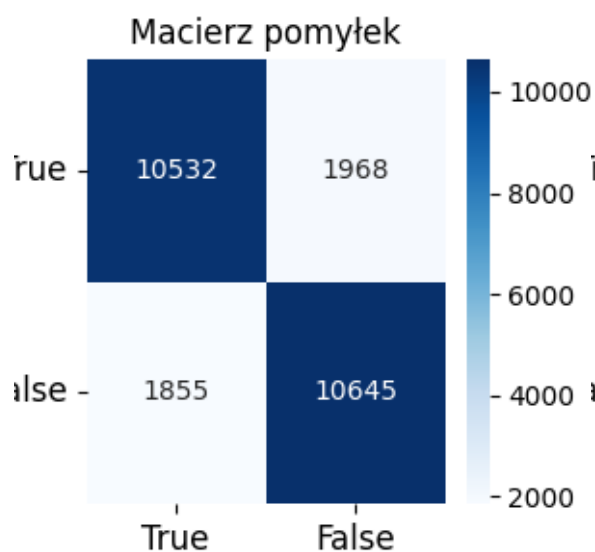
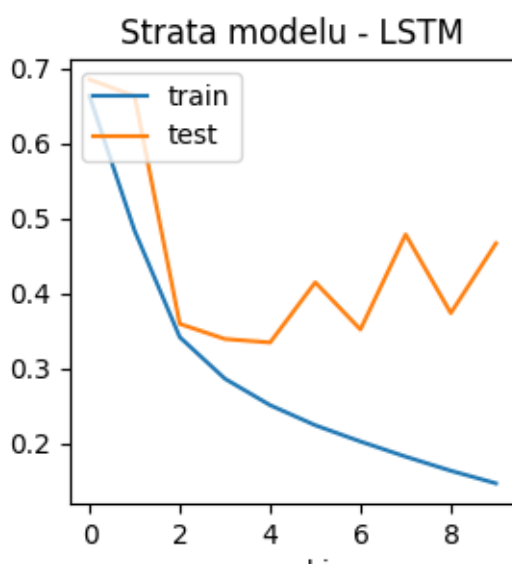
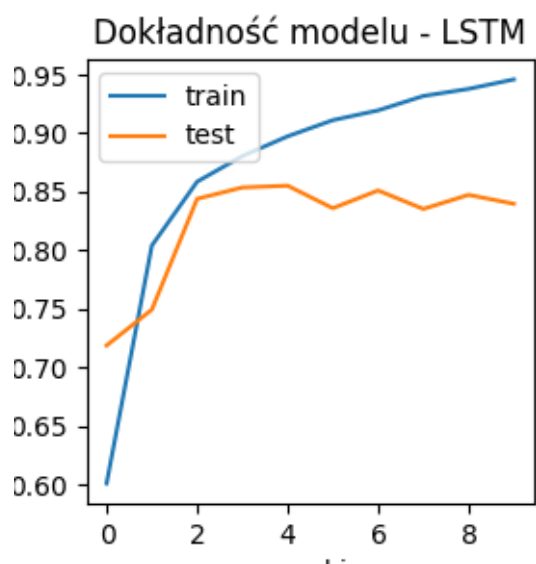
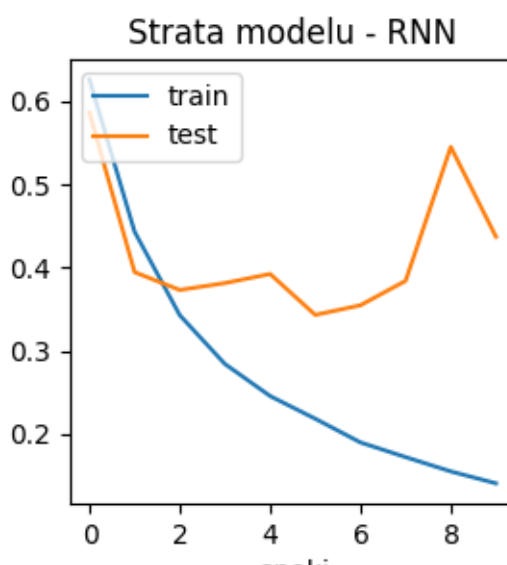
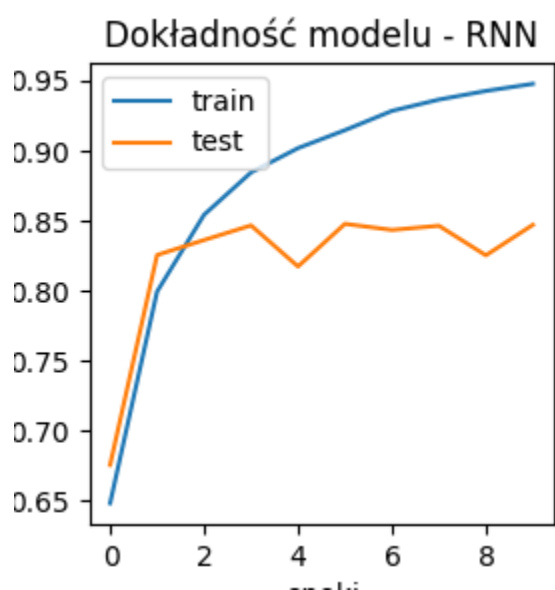
Rozmiar wymiaru wyjściowego warstwy embeddingu – 30



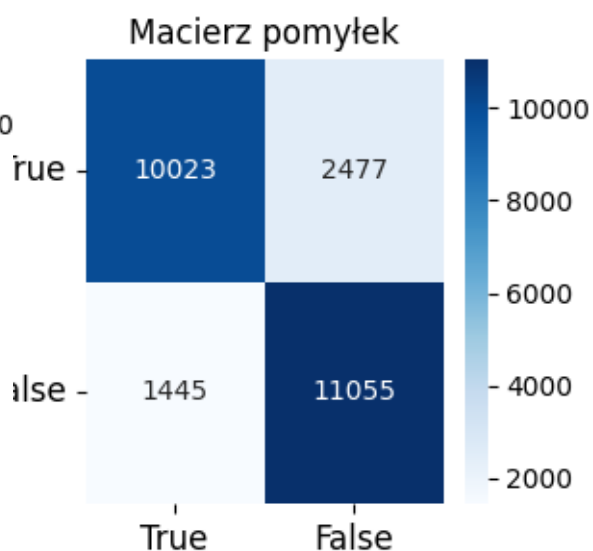
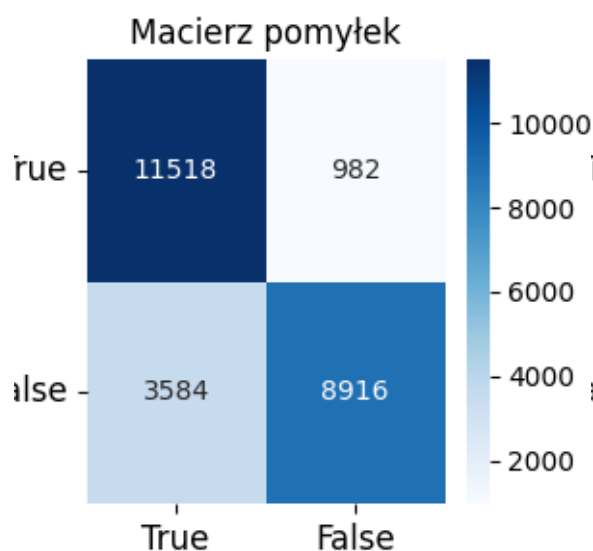
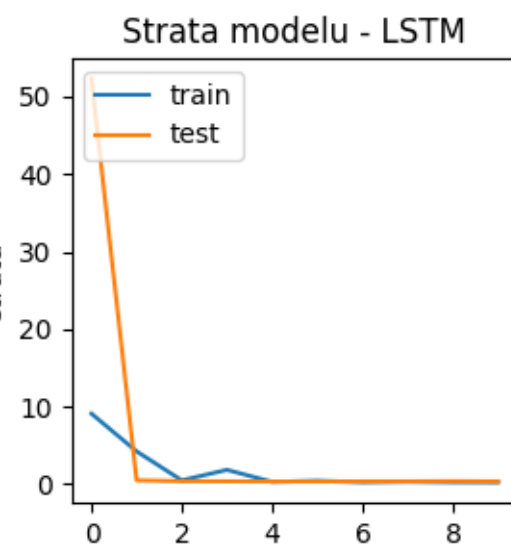
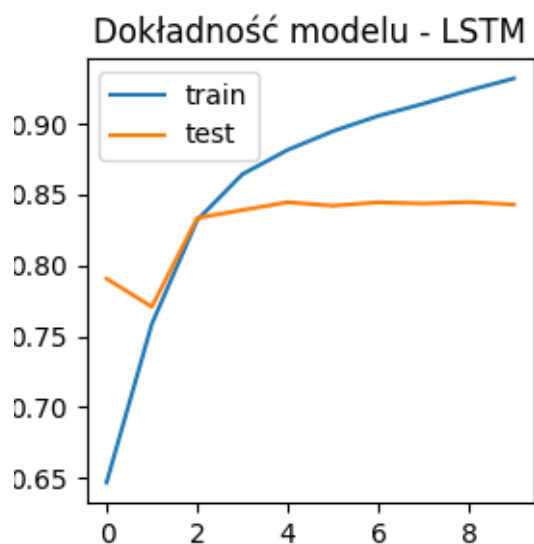
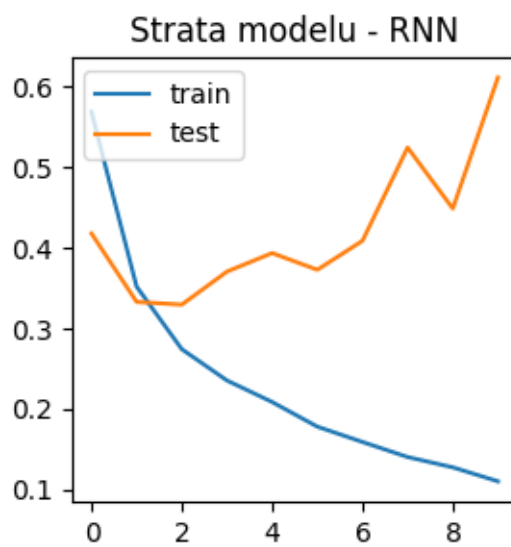
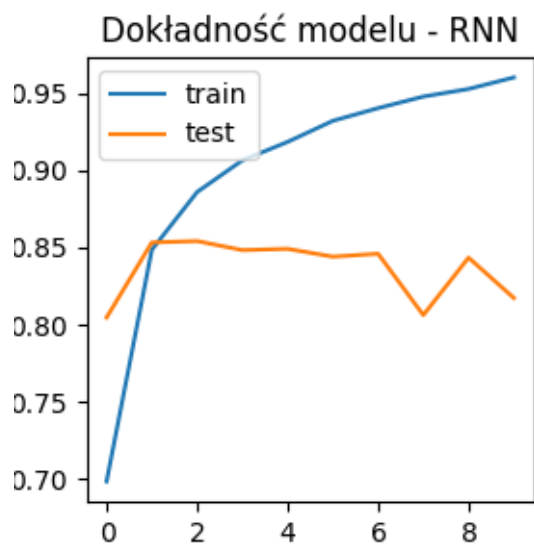




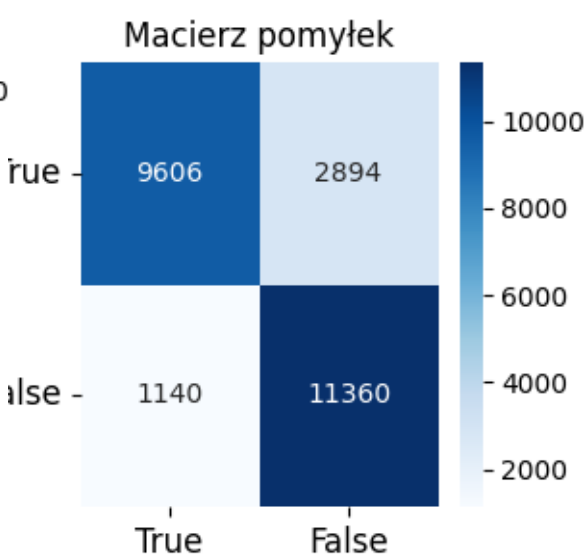
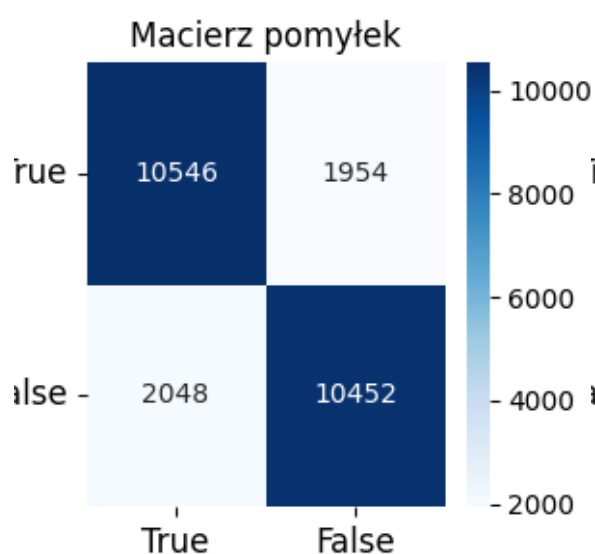
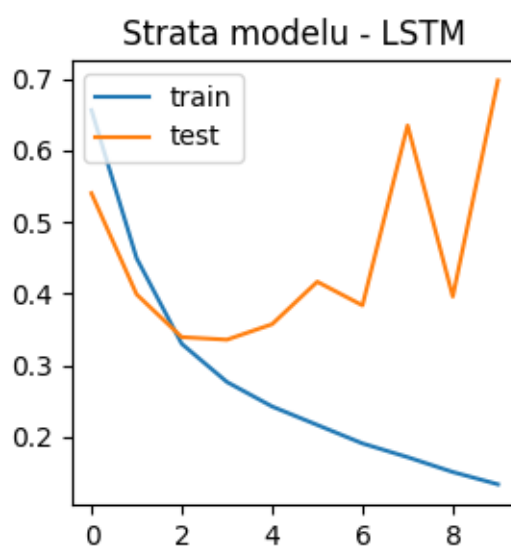
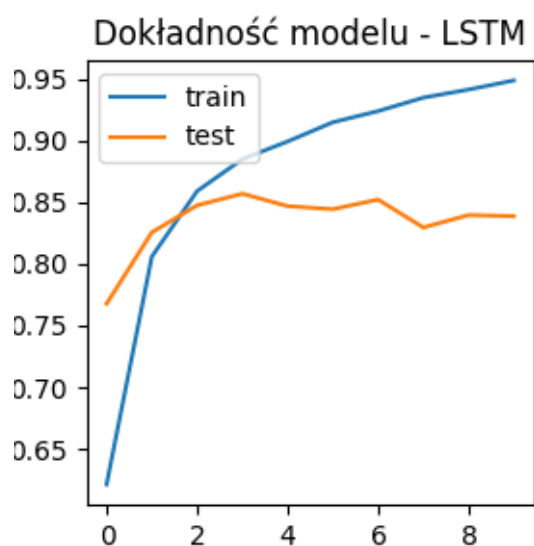
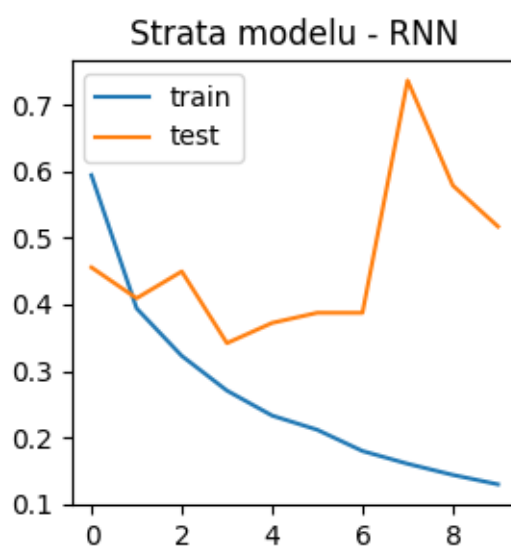
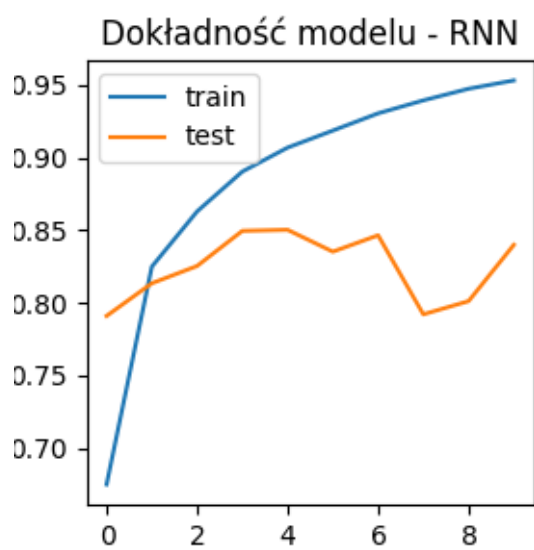
Rozmiar wymiaru wyjściowego warstwy embeddingu – 50



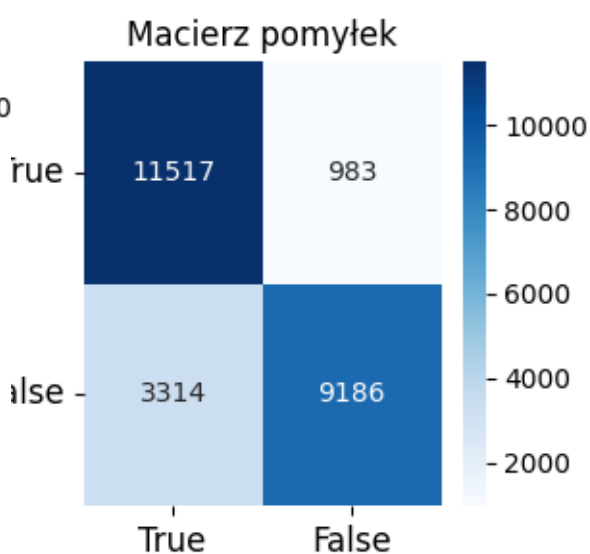
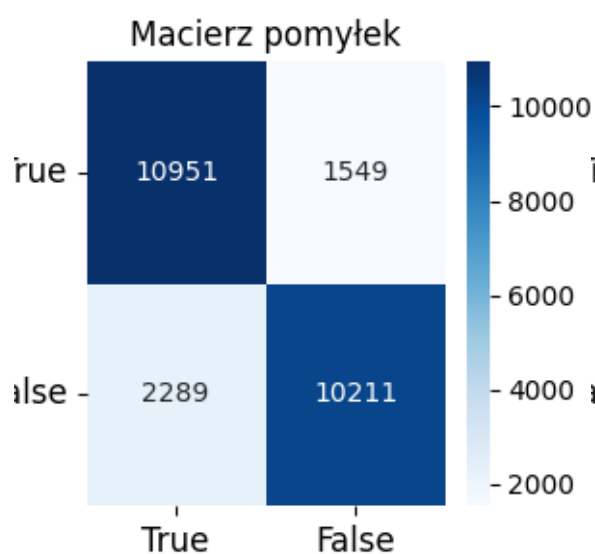
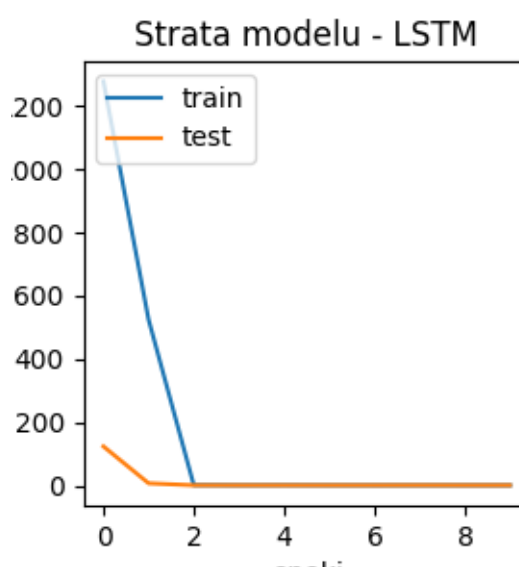
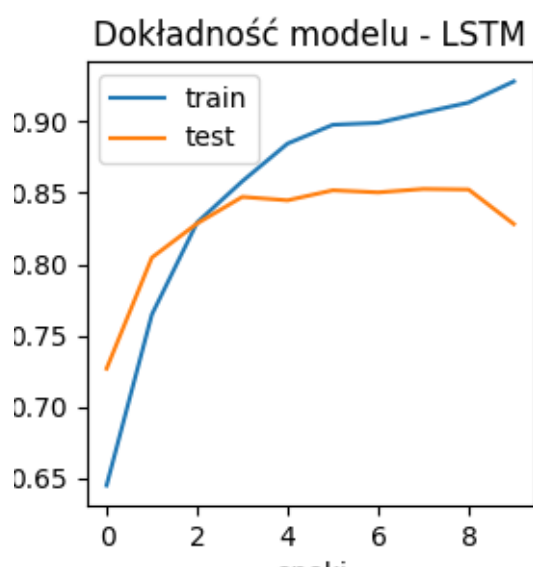
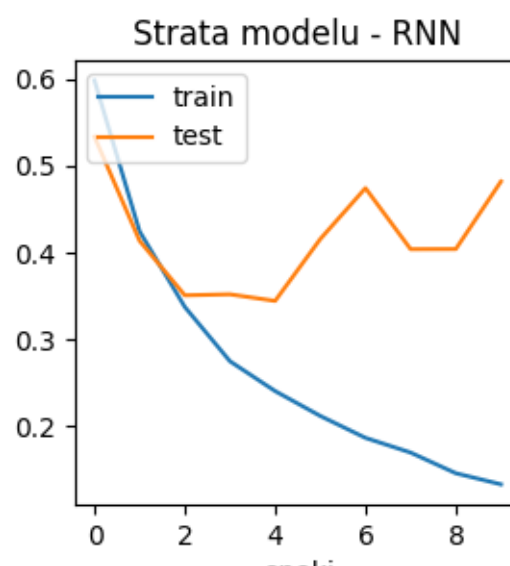
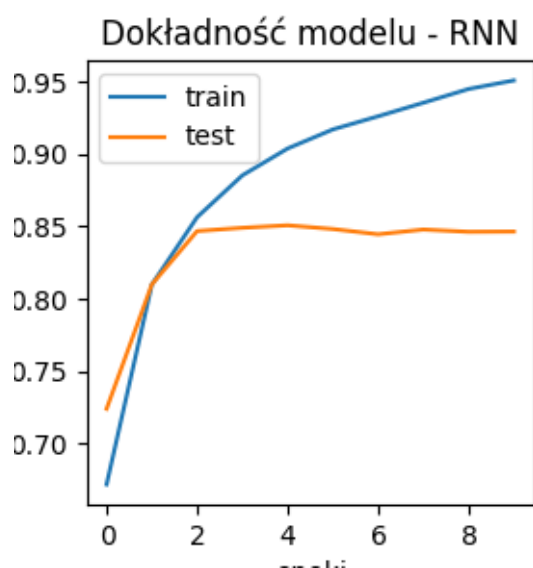
Rozmiar wymiaru wyjściowego warstwy embeddingu – 80



Rozmiar wymiaru wyjściowego warstwy embeddingu – 120



Rozmiar wymiaru wyjściowego warstwy embeddingu – 150

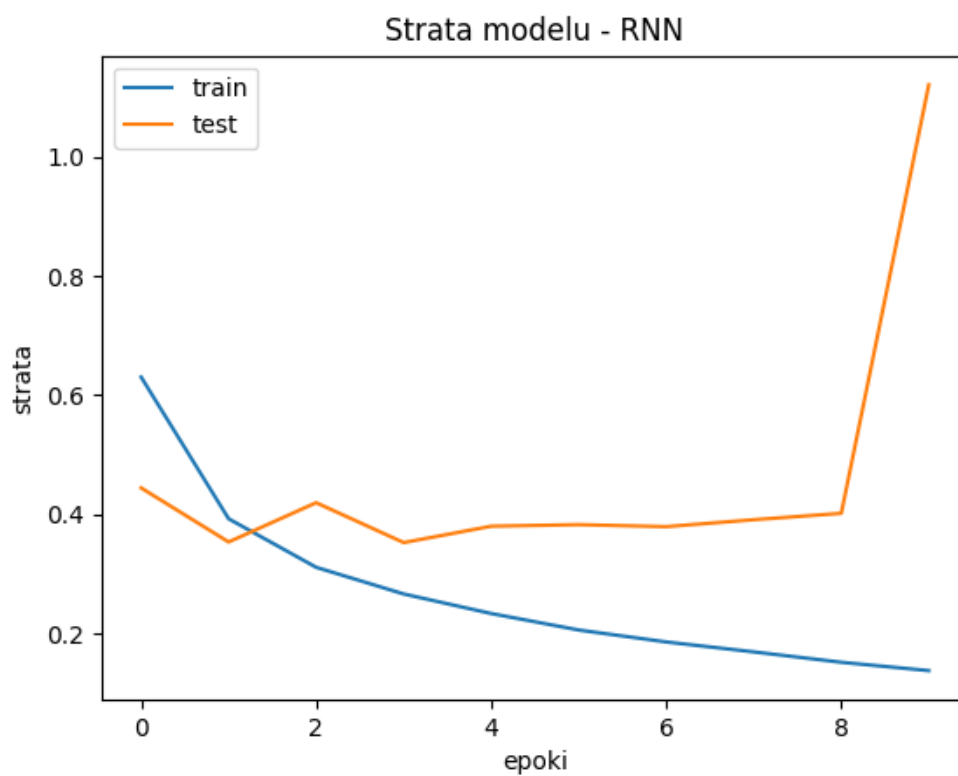
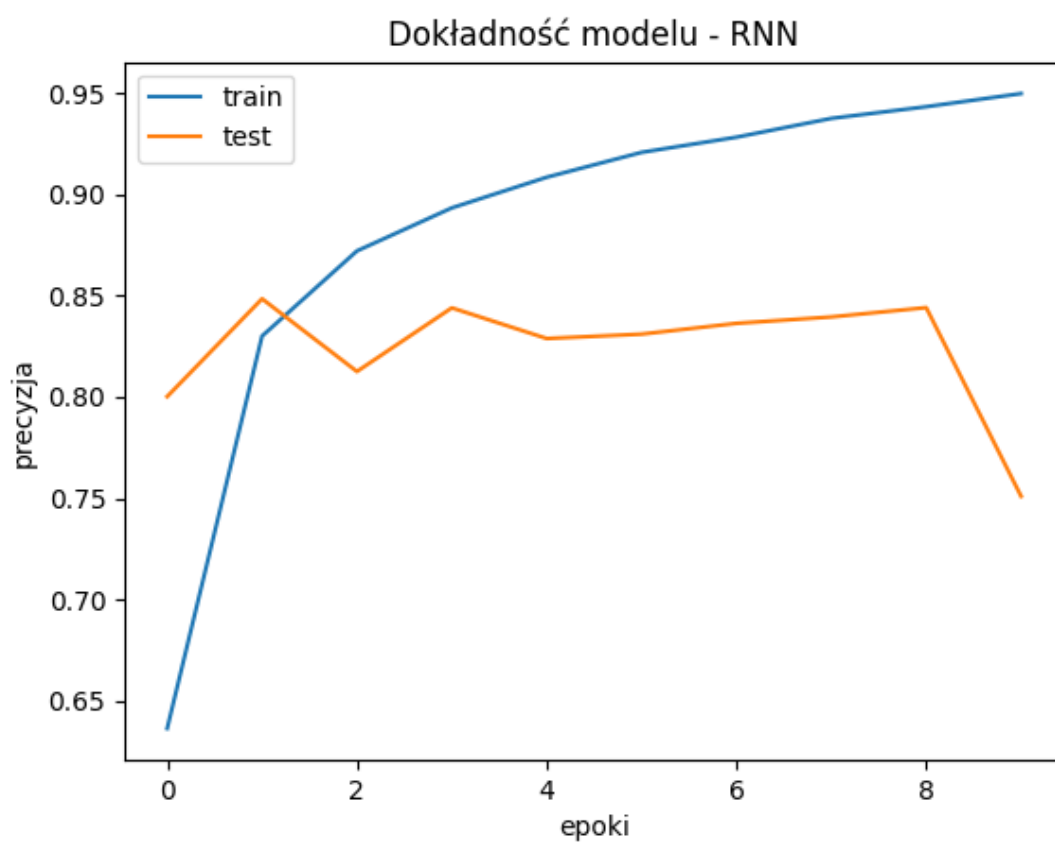


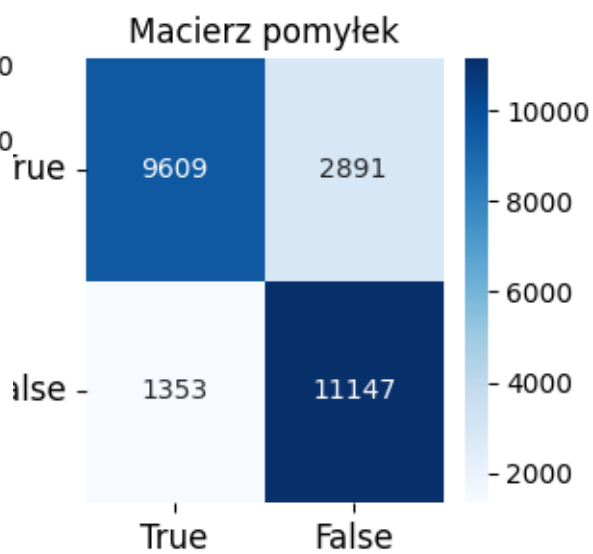
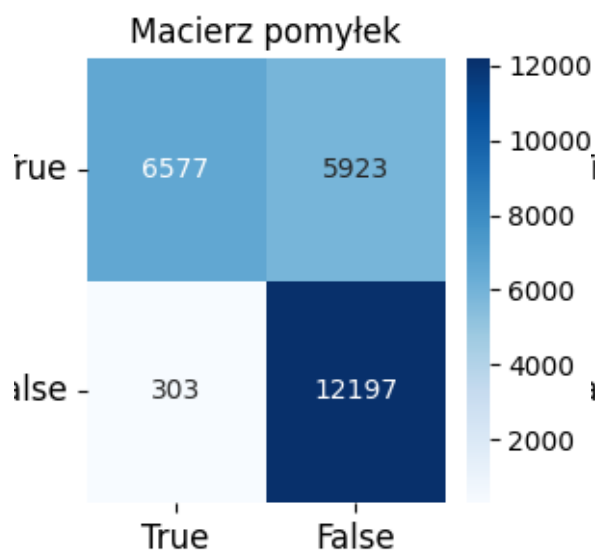
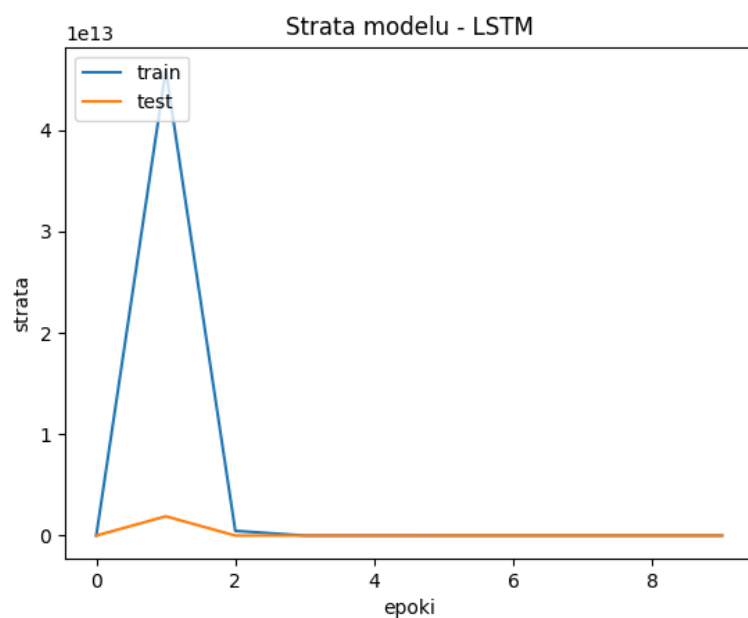
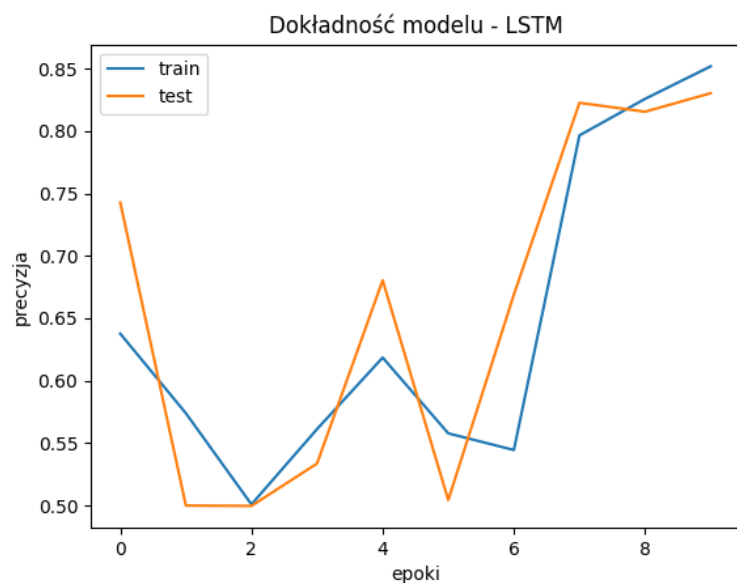
Wnioski

Niezależnie od rozmiaru wymiaru wyjściowego warstwy embeddingu, oba modele (dla komórek RNN oraz LSTM) wydają się osiągać podobne rezultaty (dokładność na zbiorze testowym oscyluje w okolicach 85%). Tempo uczenia na zbiorze treningowym jest zasadniczo dokładnie takie samo.

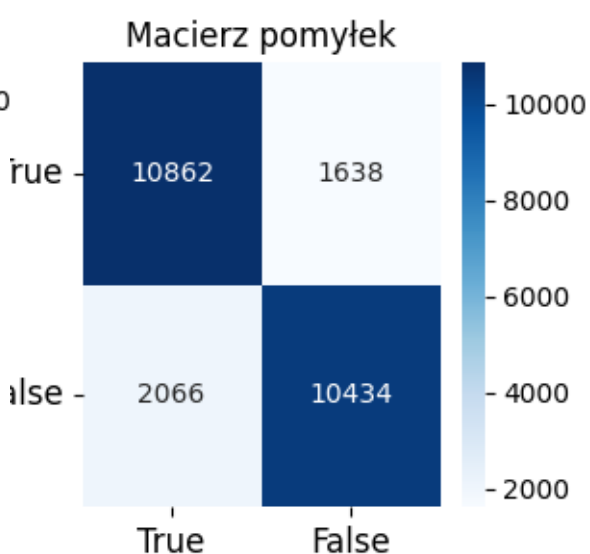
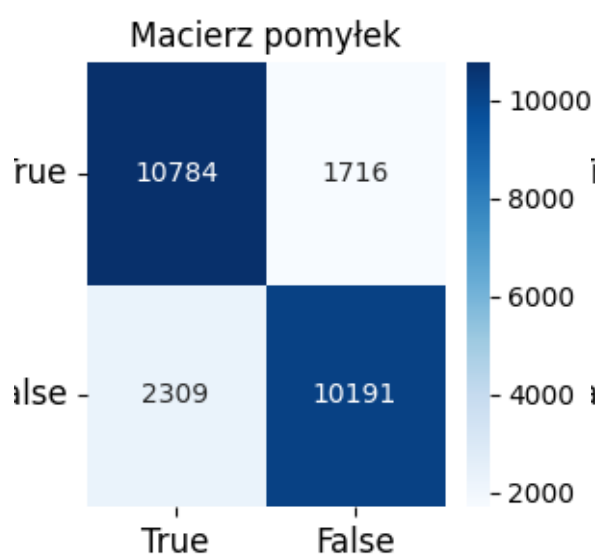
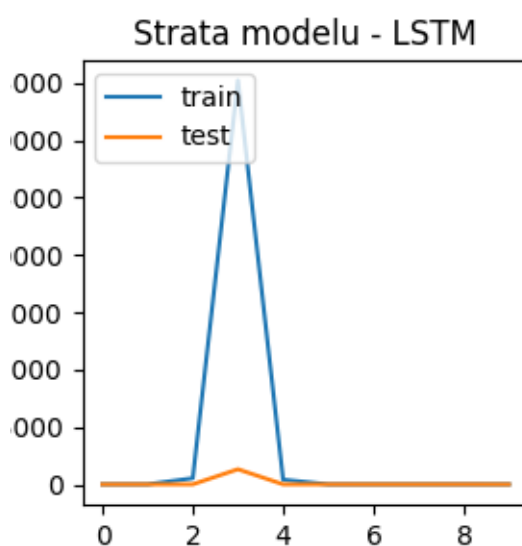
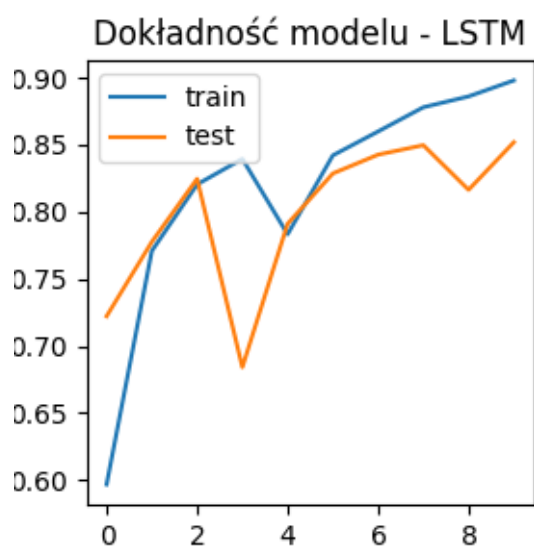
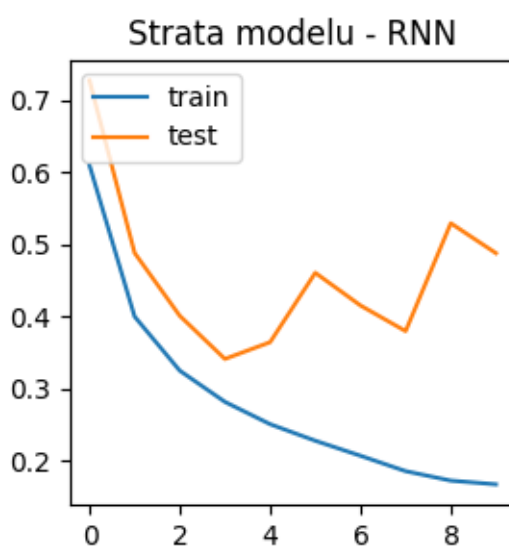
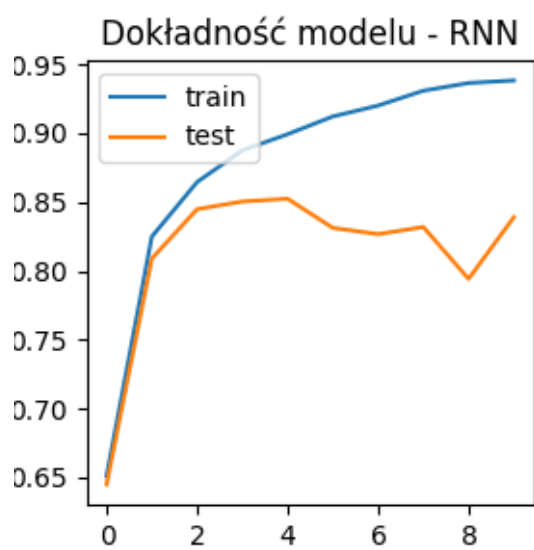
Liczba jednostek warstwy rekurencyjnej

Liczba jednostek warstwy rekurencyjnej – 30

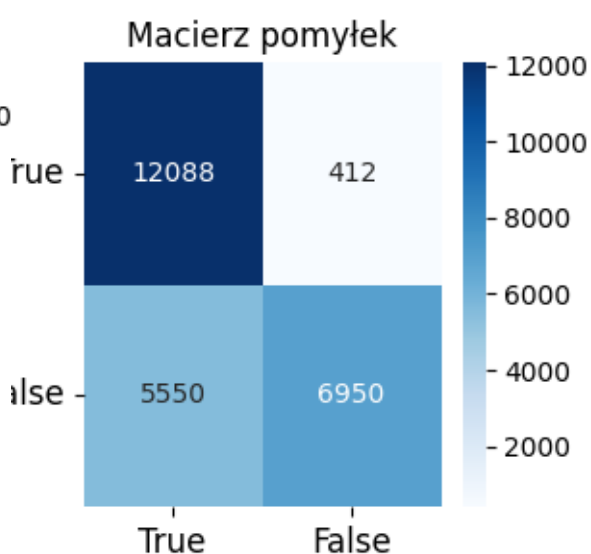
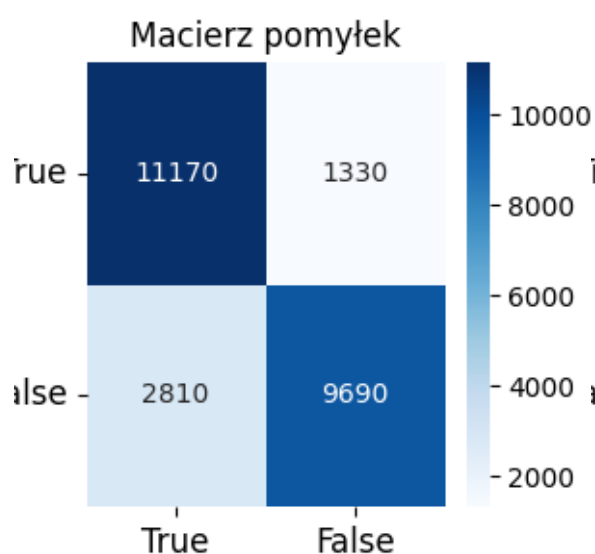
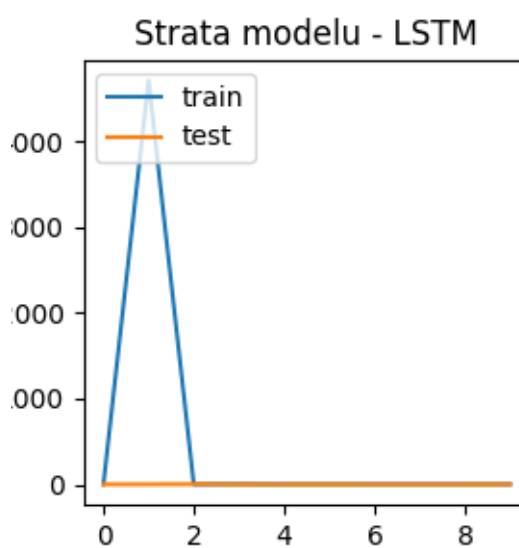
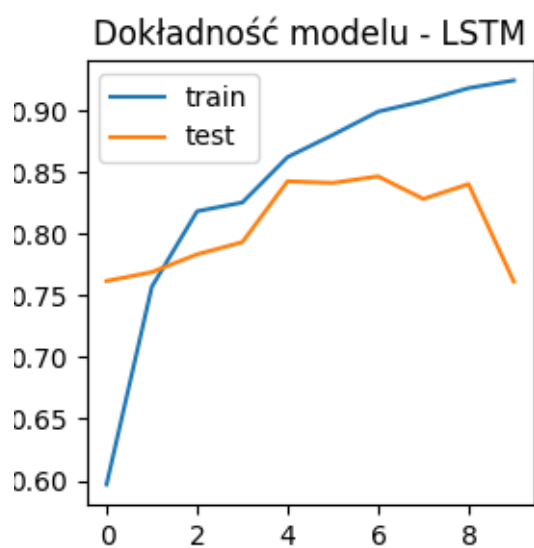
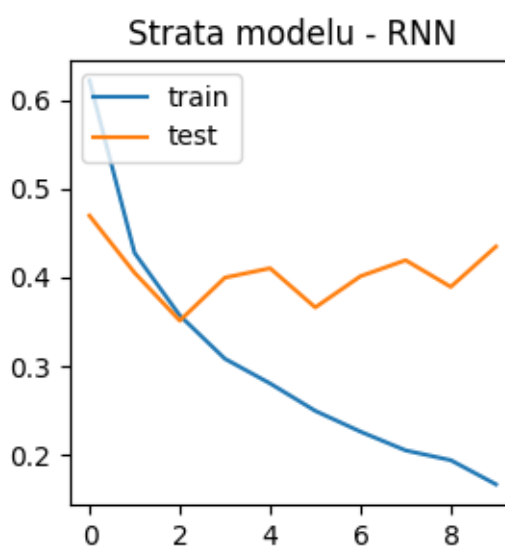
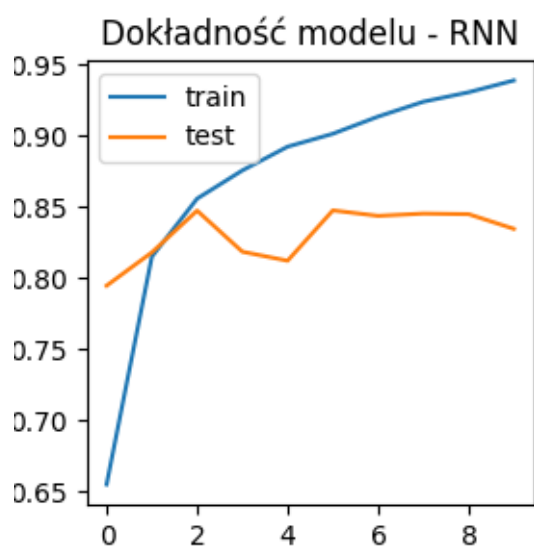




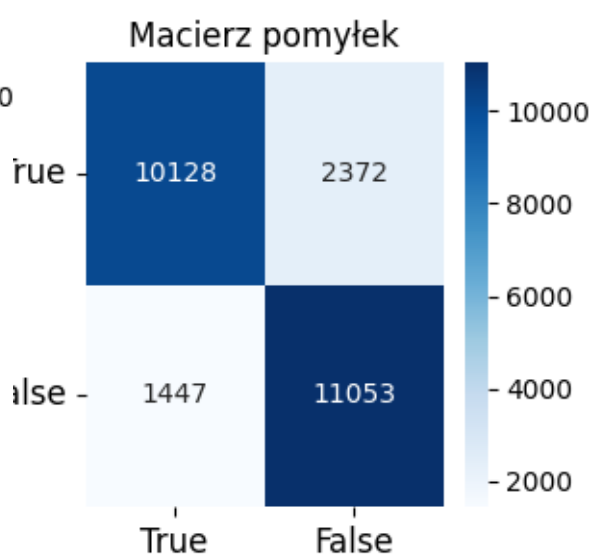
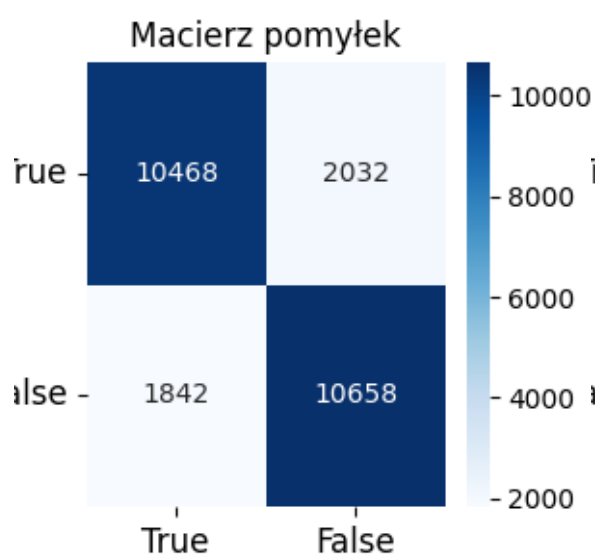
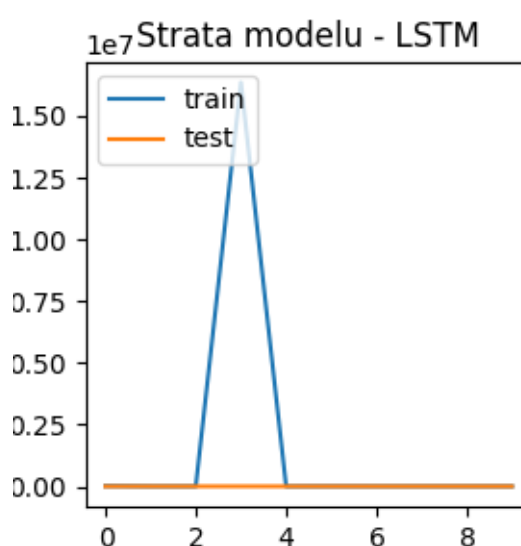
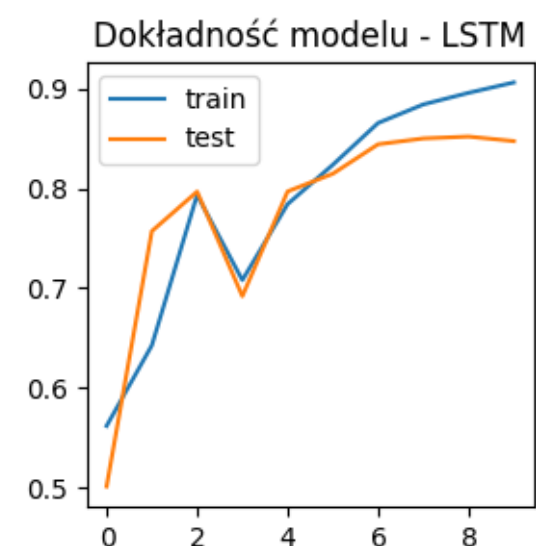
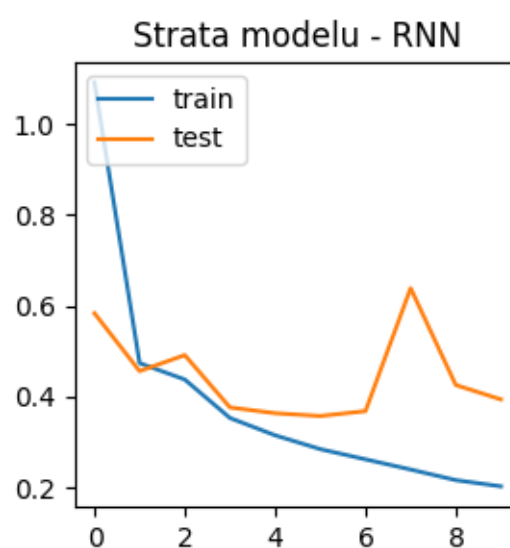
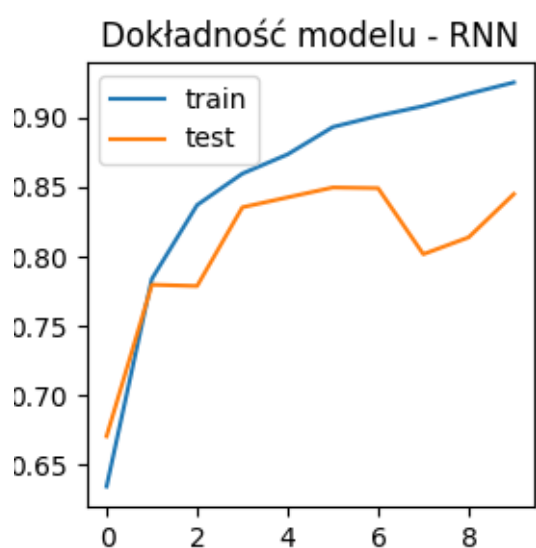
Liczba jednostek warstwy rekurencyjnej – 50



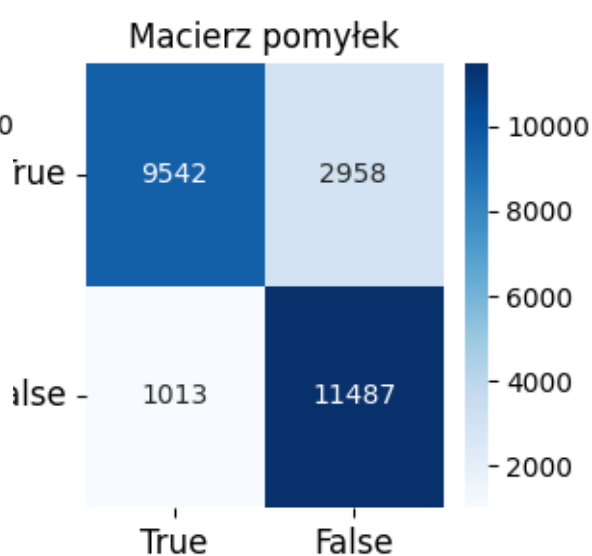
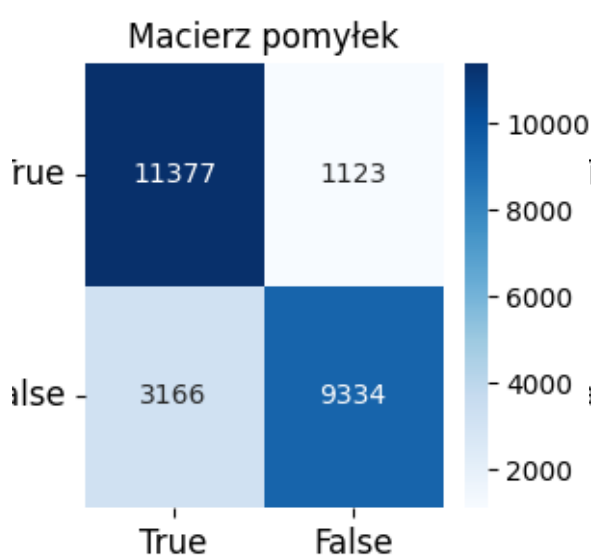
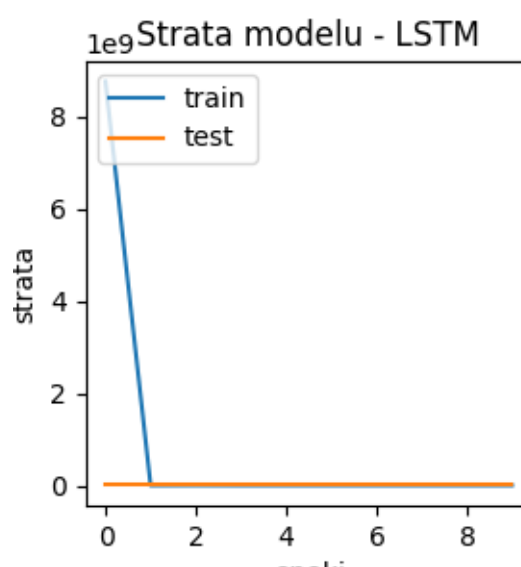
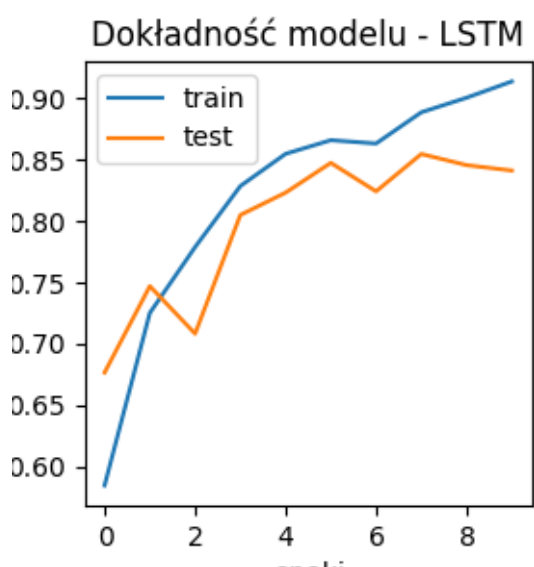
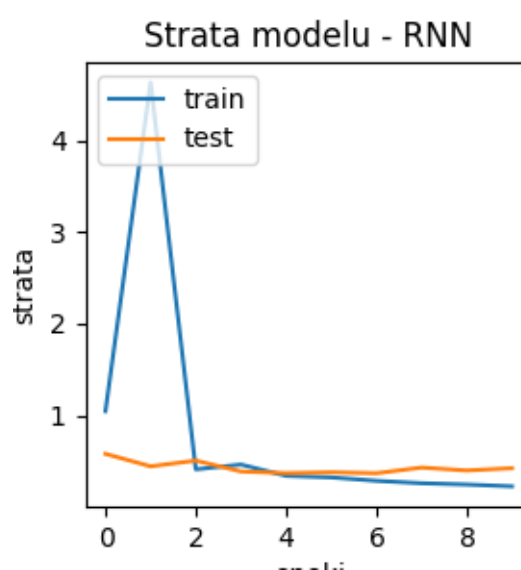
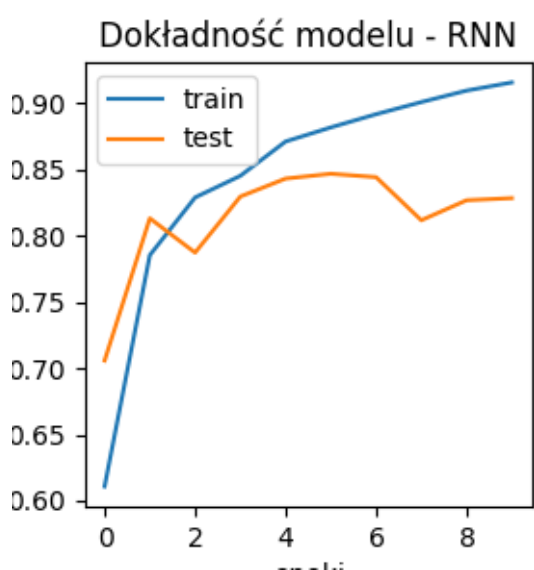
Liczba jednostek warstwy rekurencyjnej – 80



Liczba jednostek warstwy rekurencyjnej – 120



Liczba jednostek warstwy rekurencyjnej – 150

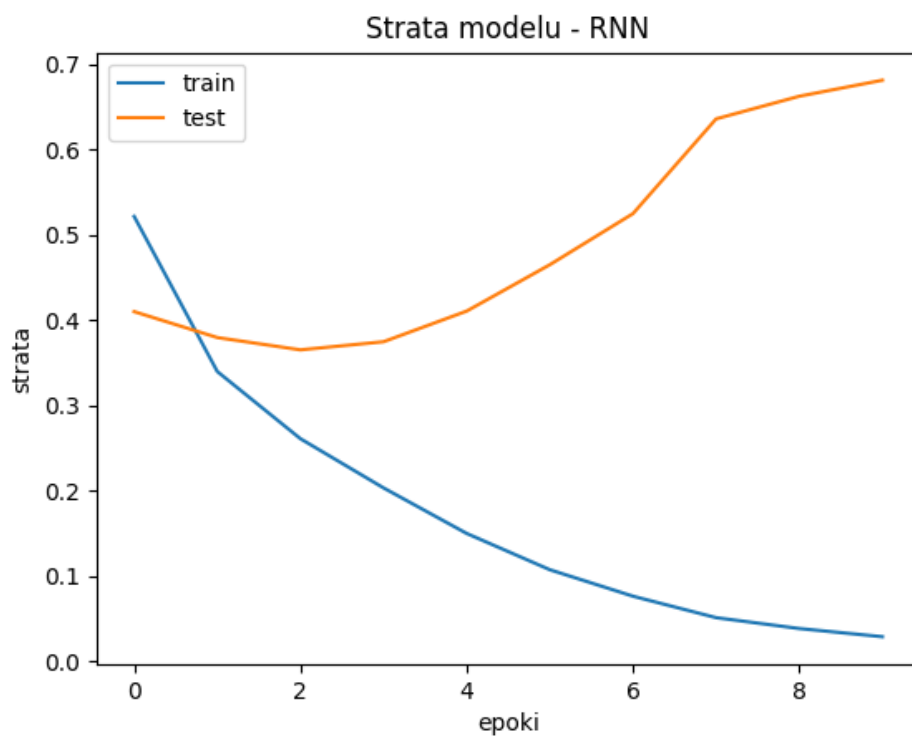
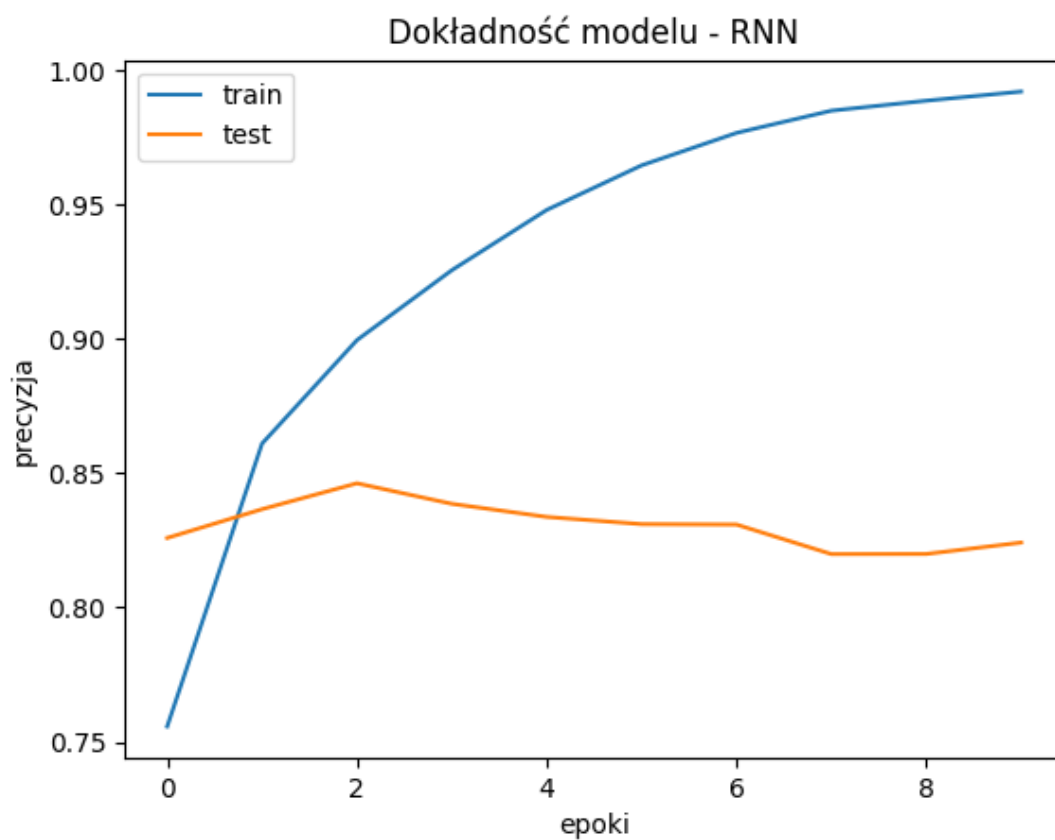


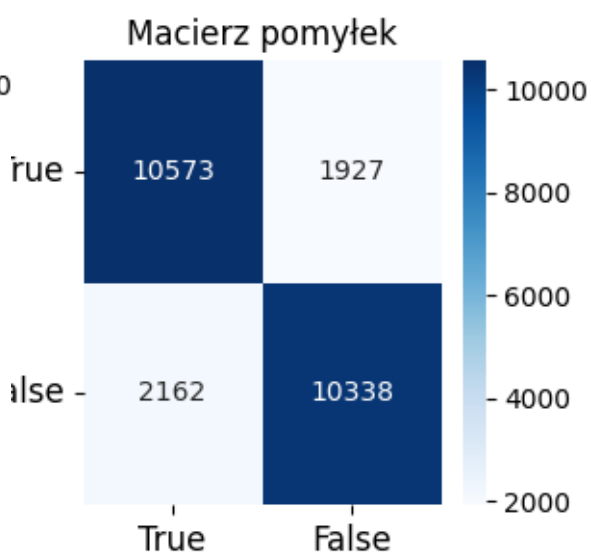
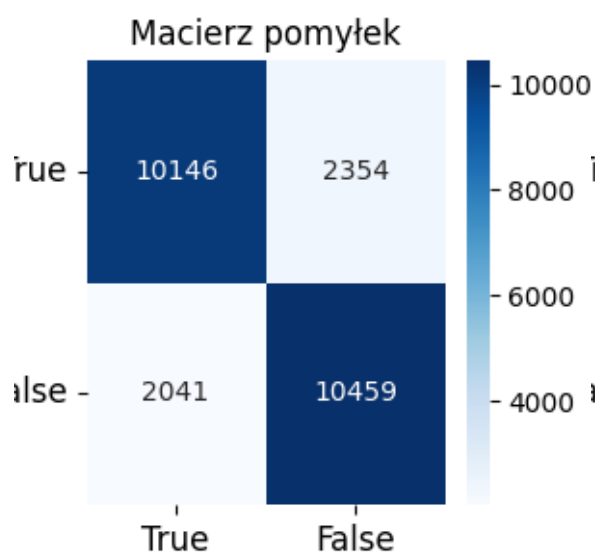
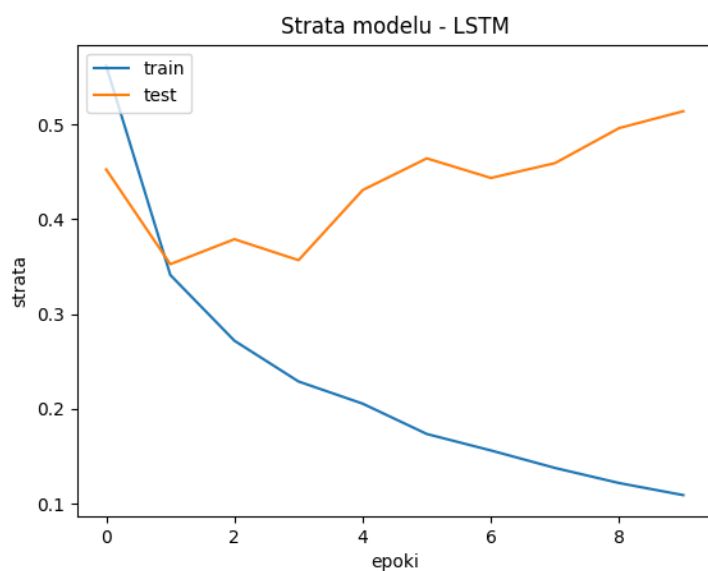
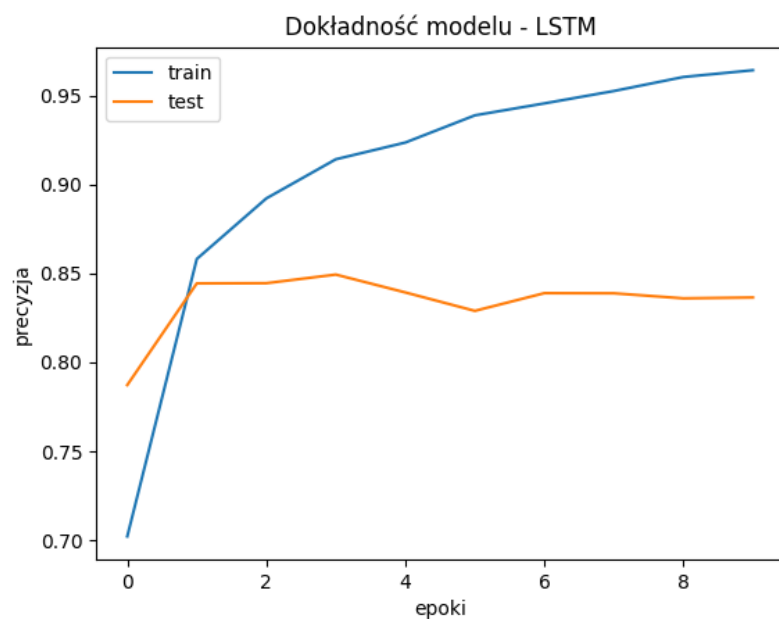
Wnioski

Im więcej jednostek warstwy rekurencyjnej posiadały oba modele tym mniej widać zjawisko przeuczenia. Ostateczna dokładność modelu, zarówno dla RNN jak i LSTM, jest dalej na poziomie 85% natomiast widać jak dużo 'bliżej' jest wykres dotyczący zbioru testowego w stosunku do wykresu zbioru treningowego. Dodatkowo, im mniej jednostek warstwy rekurencyjnej tym bardziej 'skokowe' wydają się modele i potrafią tracić na dokładności w perspektywie następnych epok. Dopiero dla rozmiarów 120 i 150 widać mniej więcej zrównoważony wzrost dokładności.

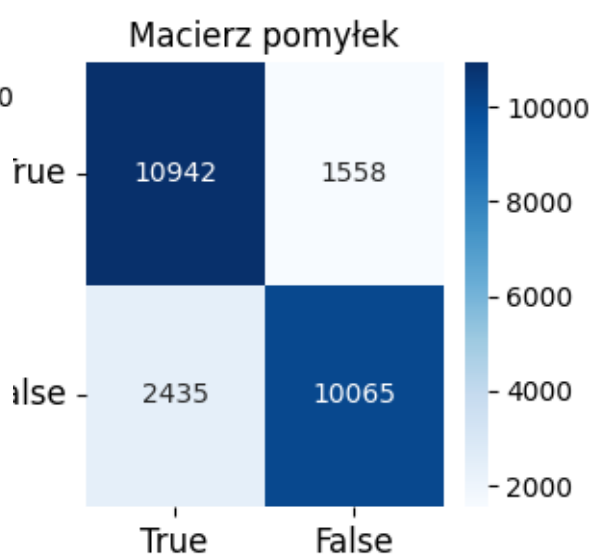
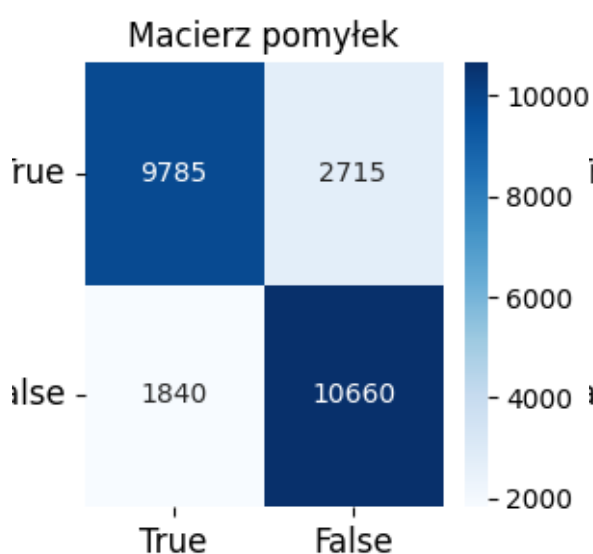
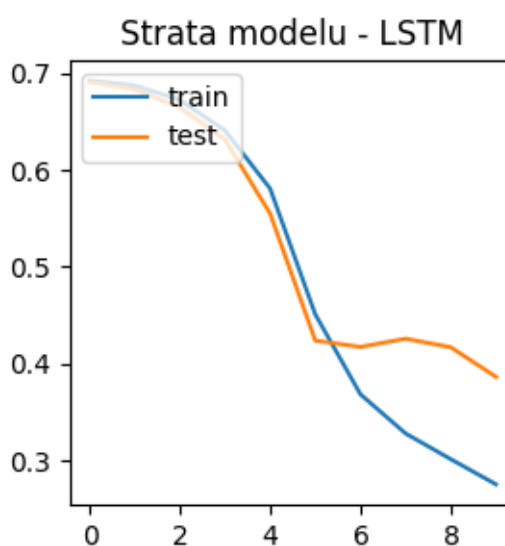
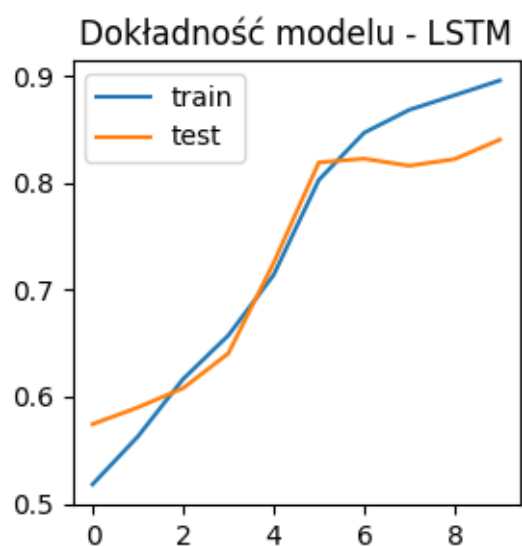
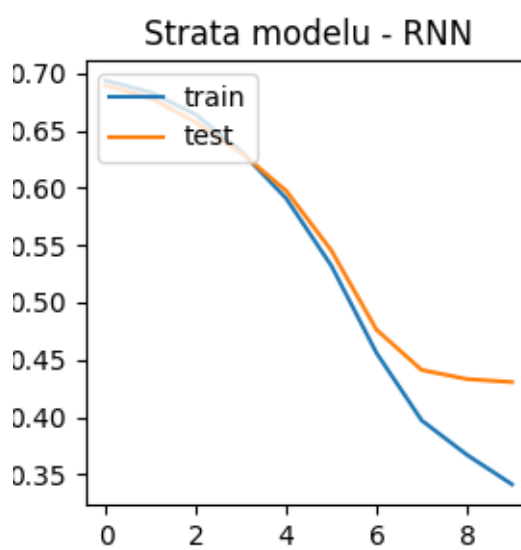
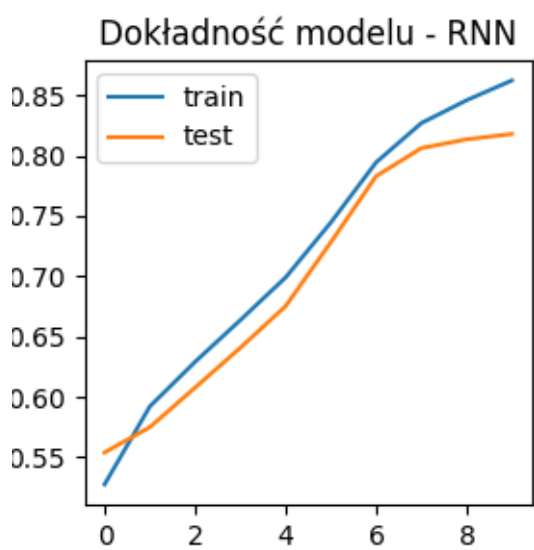
Funkcja aktywacji

Funkcja aktywacji – tangens hiperboliczny

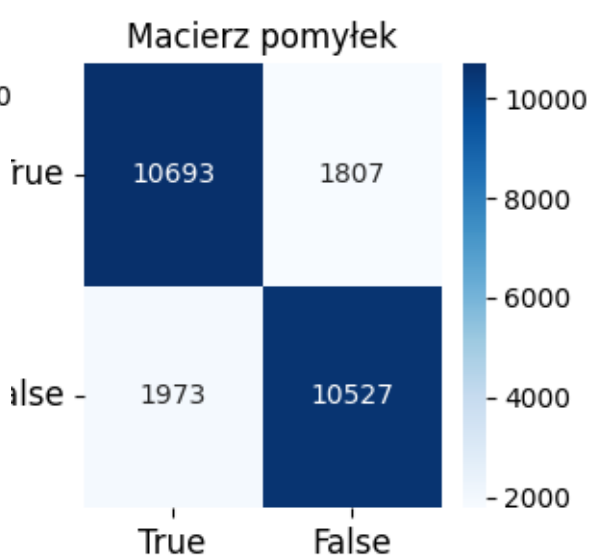
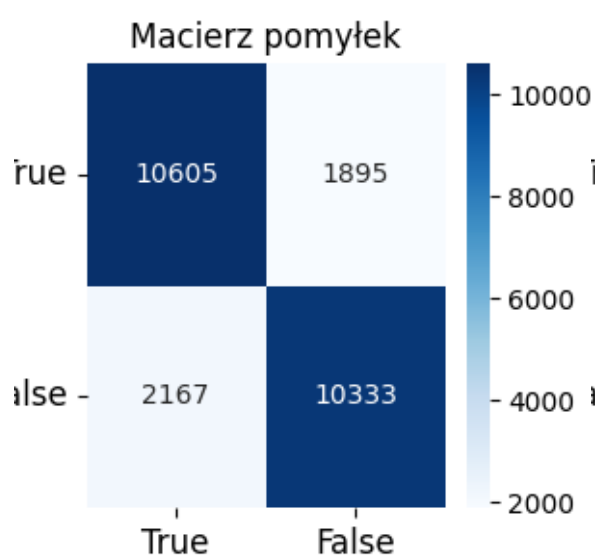
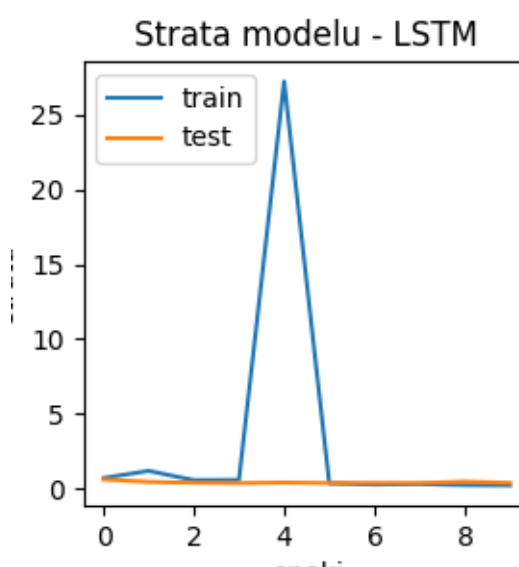
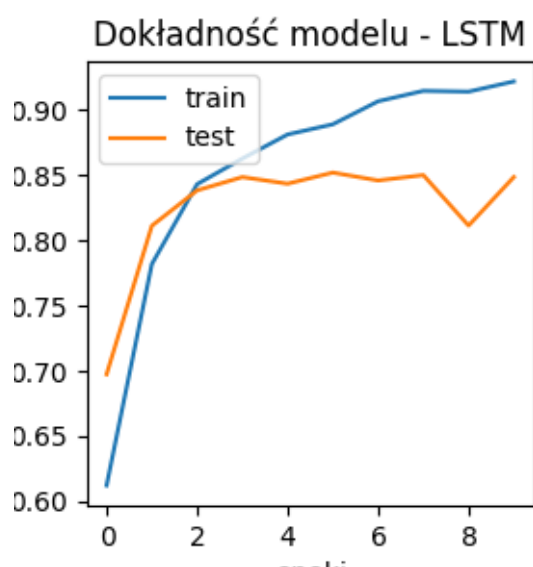
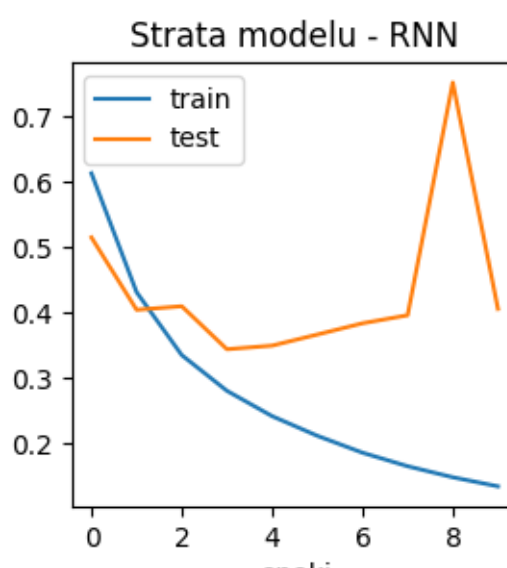
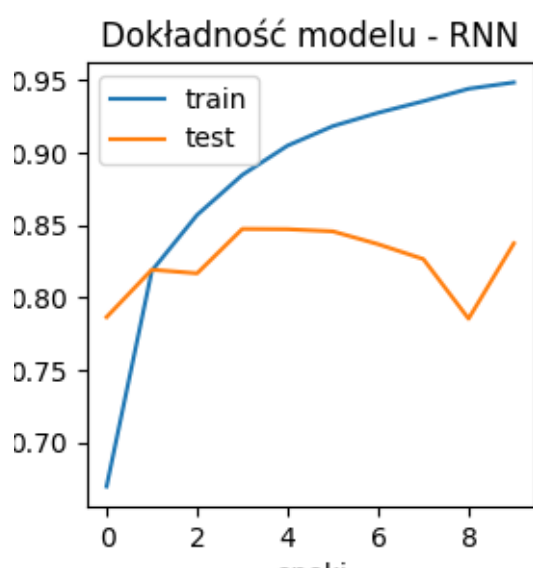




Funkcja aktywacji – sigmoid



Funkcja aktywacji – ReLU



Wnioski

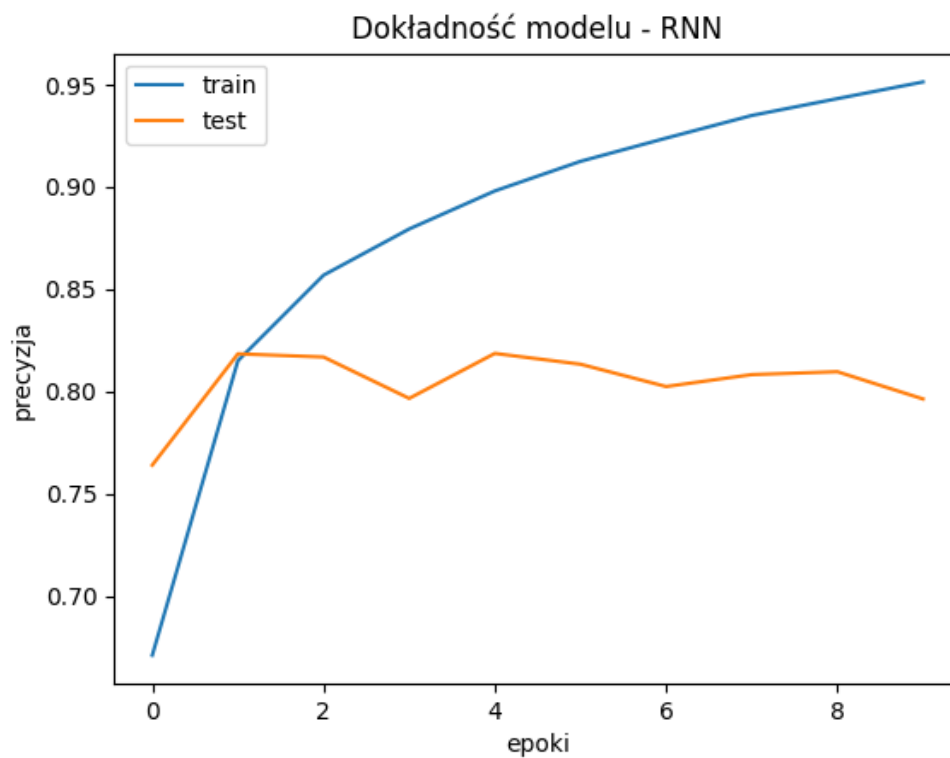
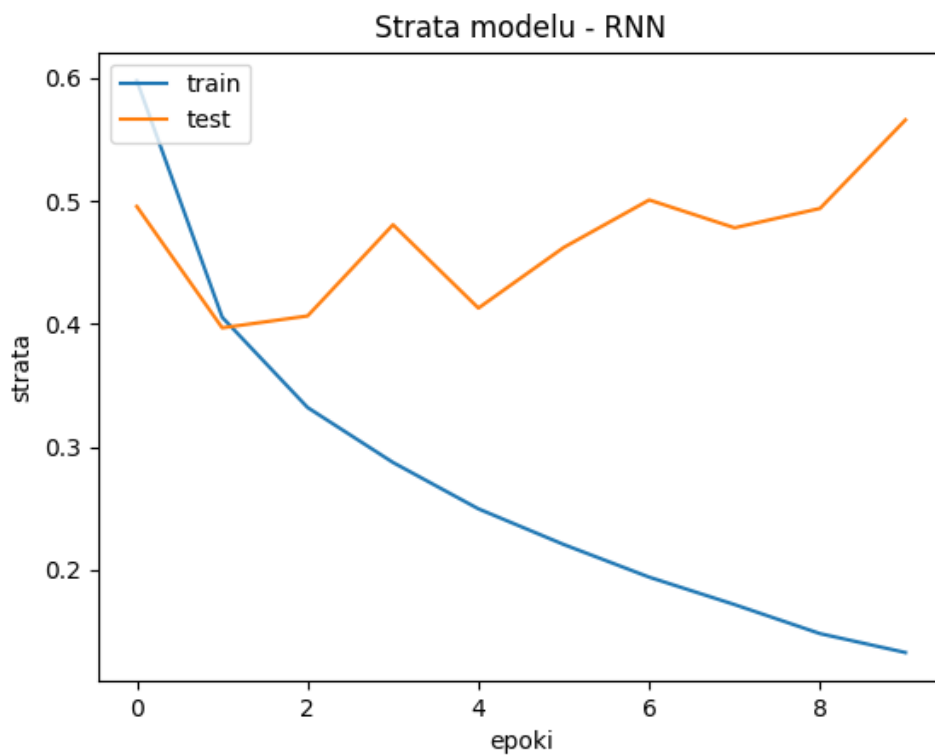
Dla funkcji aktywacji tangensa hiperbolicznego, oba modele niesamowicie szybko się przeuczają (co doprowadza do stanu gdzie po 1-2 epokach dokładność na zbiorze testowym w ogóle nie wzrasta).

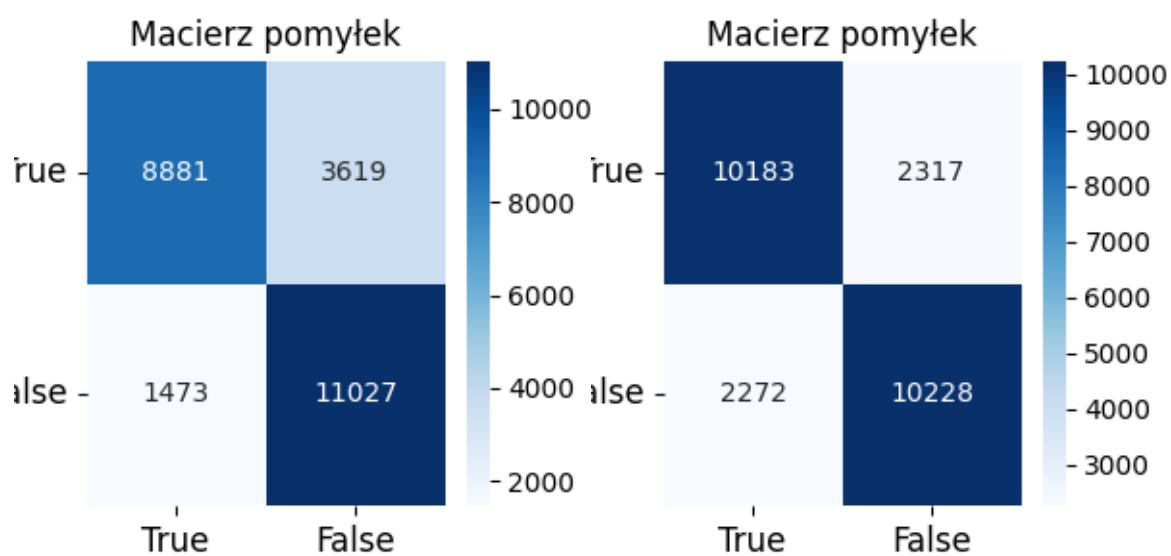
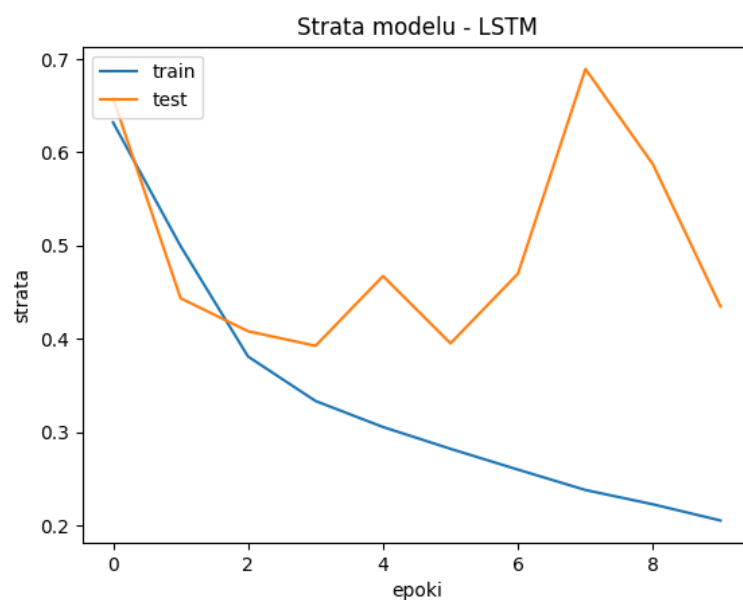
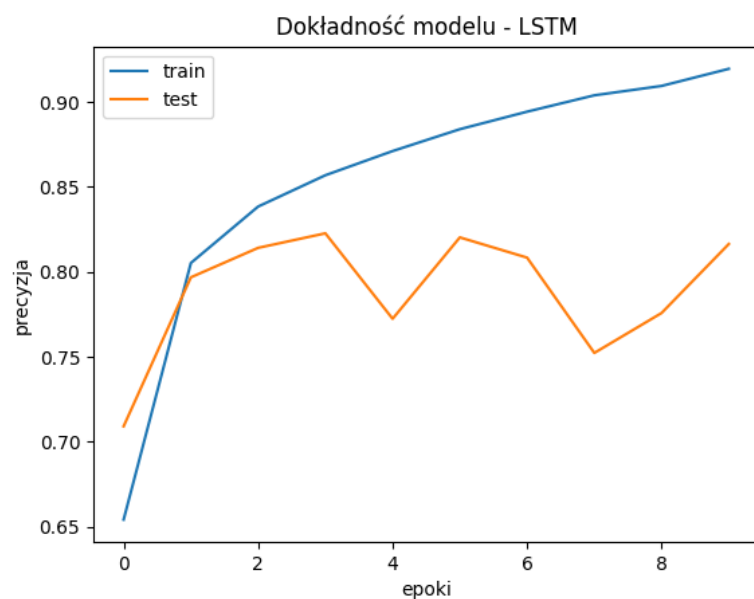
Dla funkcji sigmoid zaobserwować można mniej więcej jednostajny wzrost dokładności zarówno na zbiorze testowym jak i treningowym. Funkcja ReLU natomiast wydaje się osiągać pośredni efekt.

Efektywność paddingu

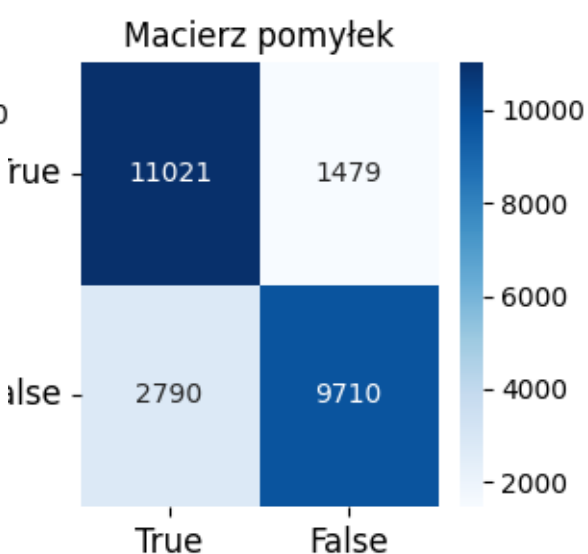
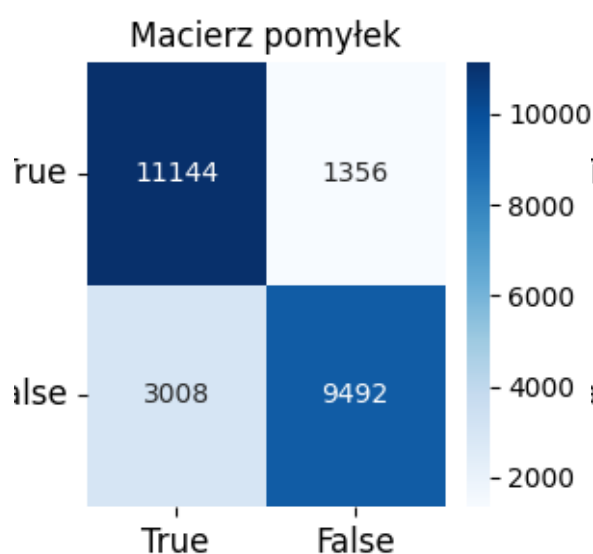
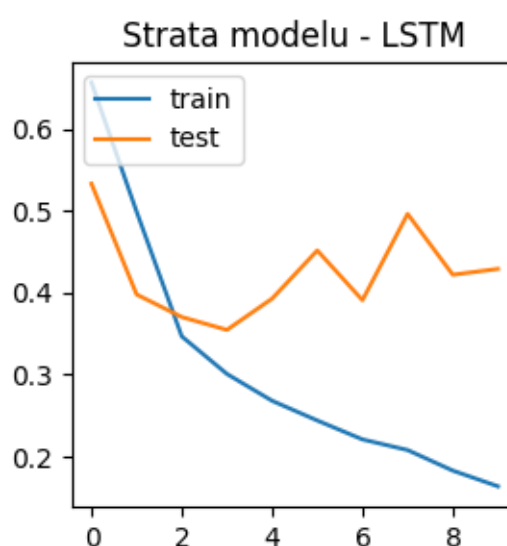
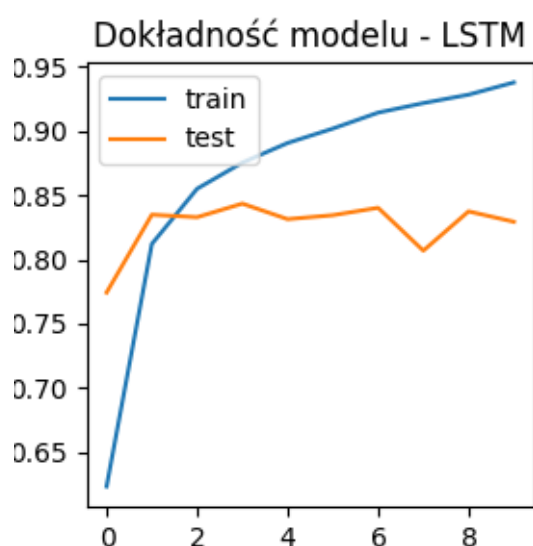
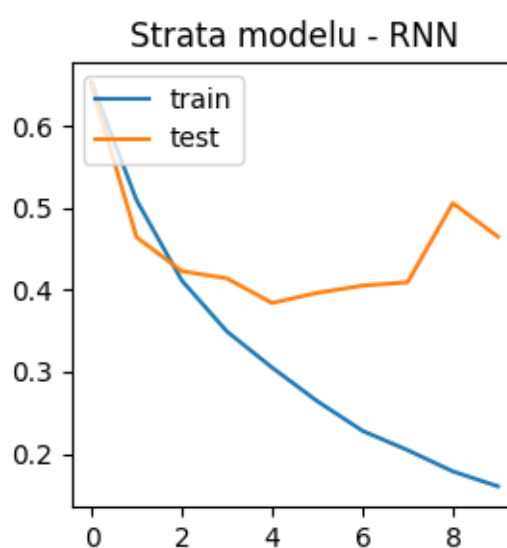
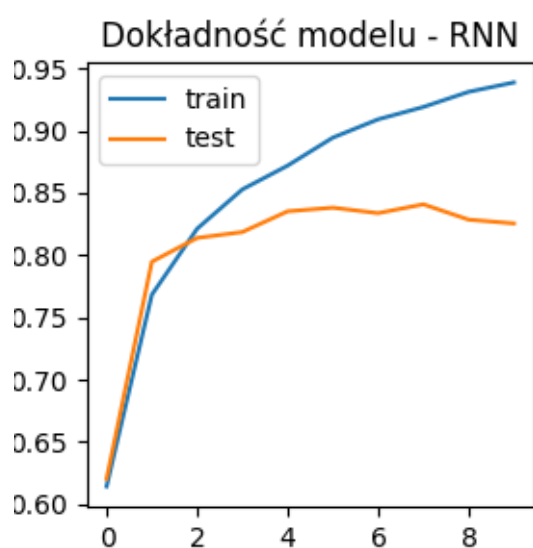
Rozmiar paddingu / długość maski

Rozmiar paddingu / długość maski – 50

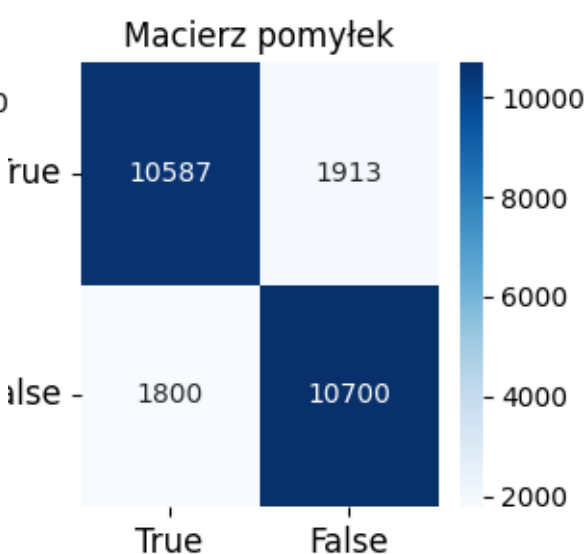
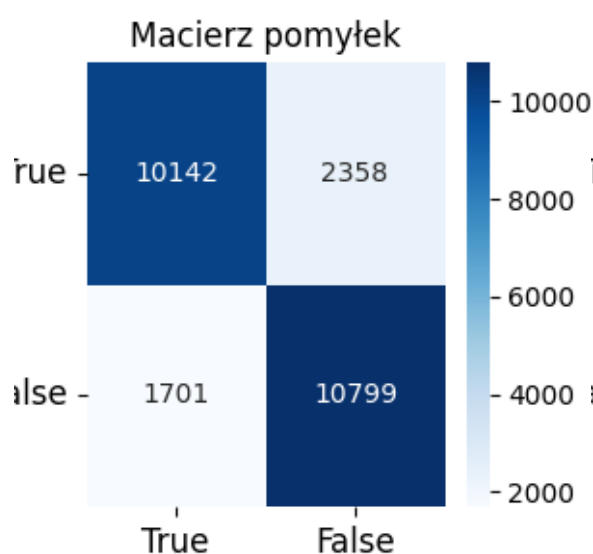
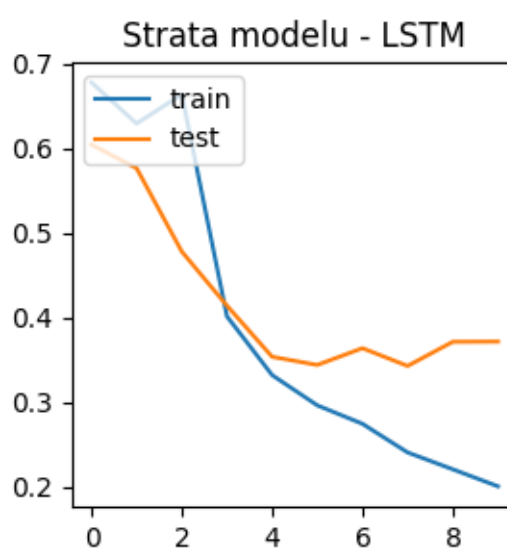
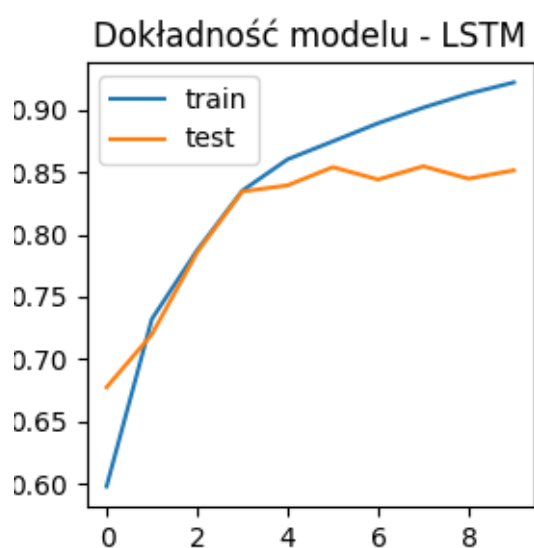
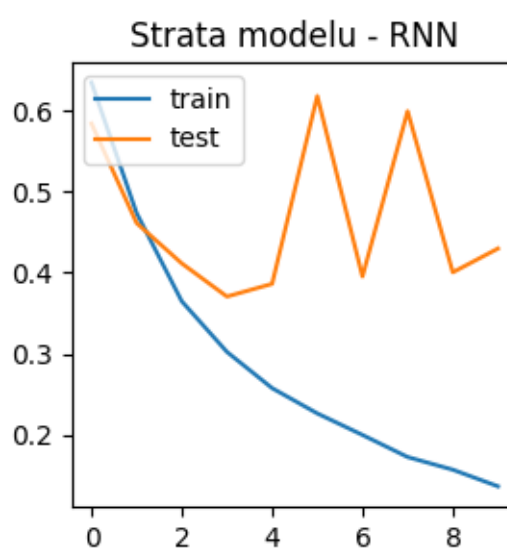
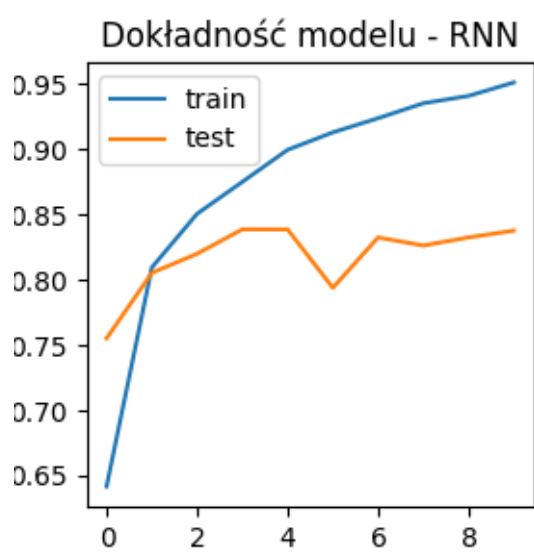




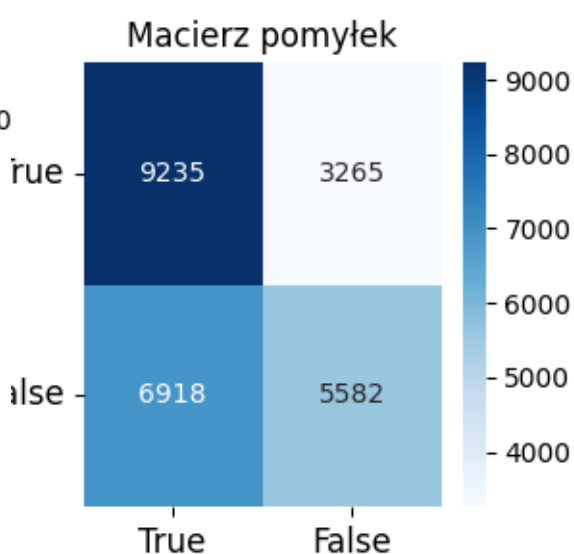
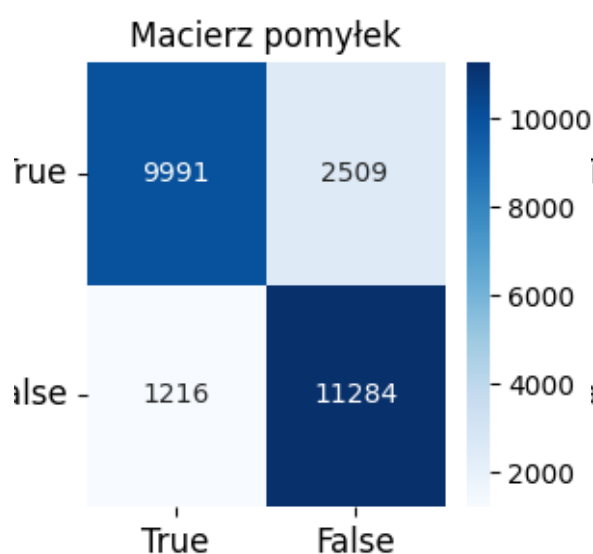
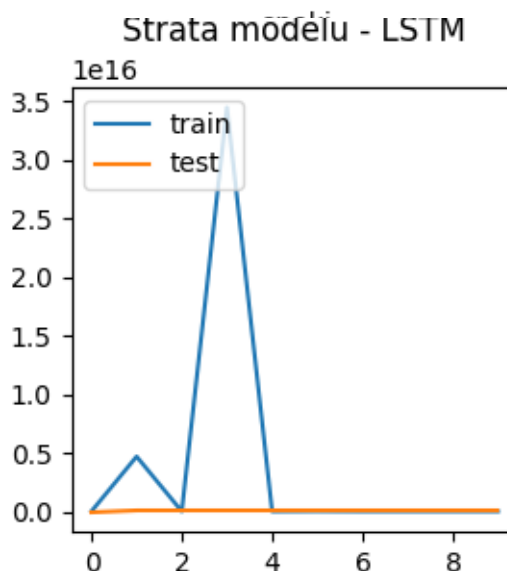
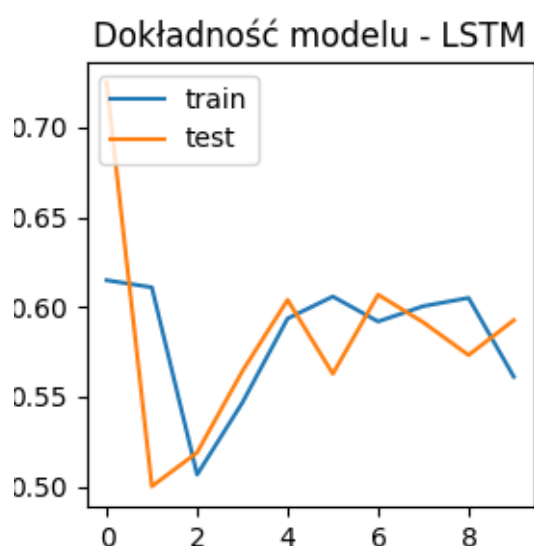
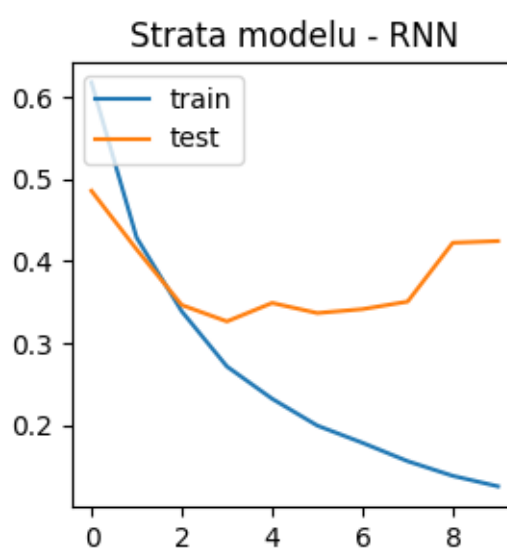
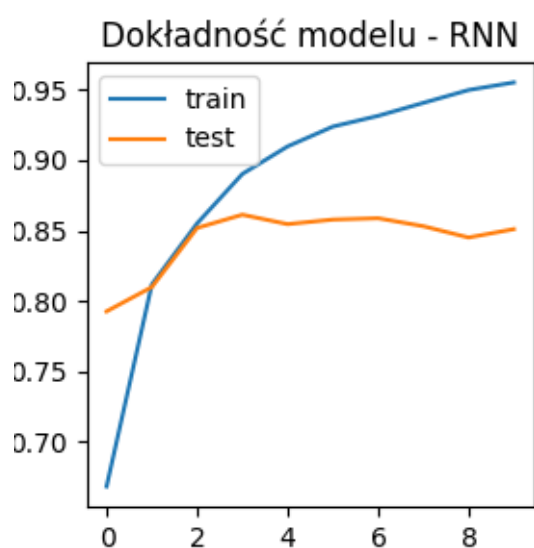
Rozmiar paddingu / długość maski – 80



Rozmiar paddingu / długość maski – 100

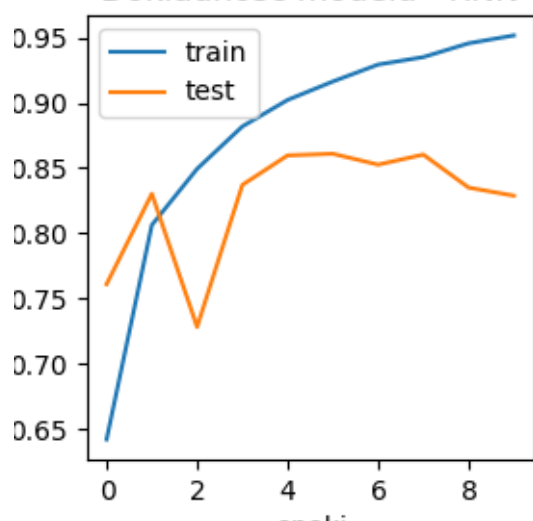


Rozmiar paddingu / długość maski – 130

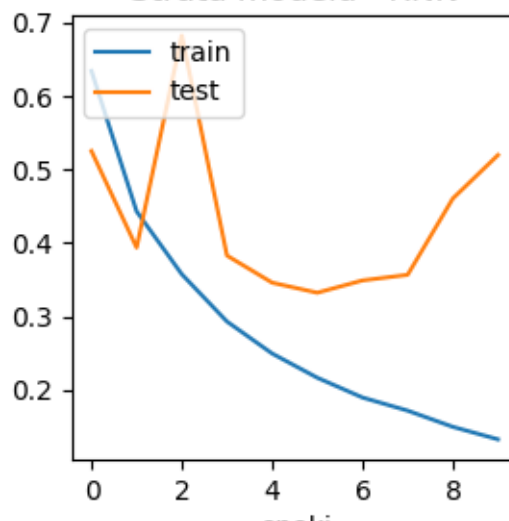


Rozmiar paddingu / długość maski - 150

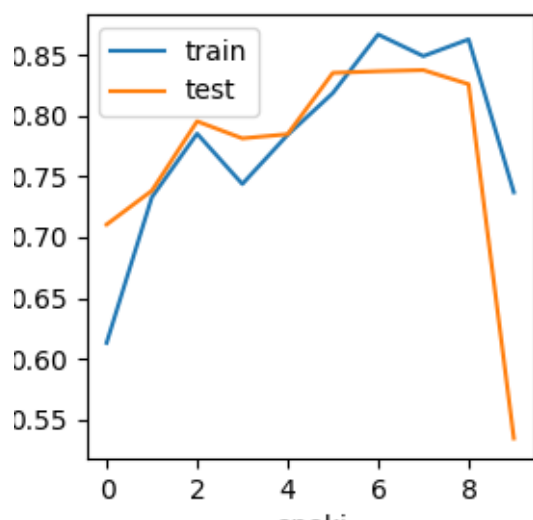
Dokładność modelu - RNN



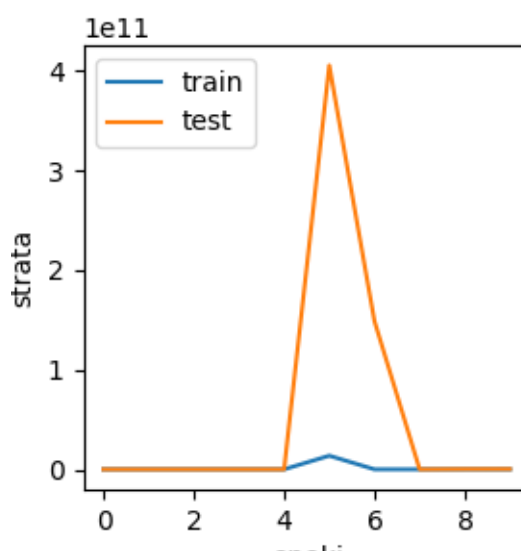
Strata modelu - RNN



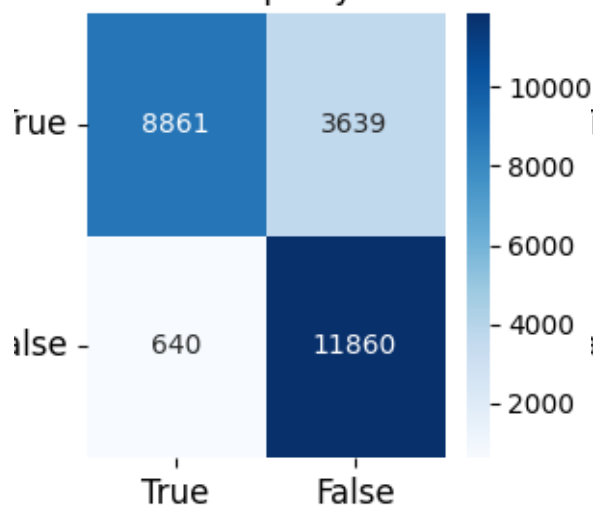
Dokładność modelu - LSTM



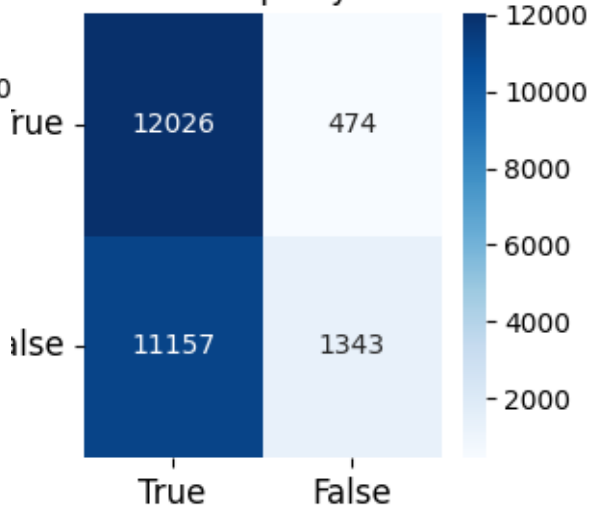
Strata modelu - LSTM



Macierz pomyłek



Macierz pomyłek



Wnioski

Długość maski zdecydowanie wpływa na ostateczne wyniki osiągane przez RNN i LSTM. Dla najmniejszego rozmiaru (50) widać, że osiągana jest dokładność około 80% na zbiorze testowym co oznacza spadek około pięciu punktów procentowych względem poprzednich eksperymentów. Z kolei dla maski o długości 130 czy 150 widać, że dokładność LSTM bardzo mocno oscyluje i potrafi być zdecydowanie gorsza (w przypadku RNN wydaje się osiągać najlepsze wyniki). W związku z tym mogę wywnioskować, że dobór odpowiedniej maski jest bardzo istotny i różnie wpływa na różne rodzaje komórek.

Wnioski

Po przeprowadzeniu eksperymentów mogę stwierdzić, że zarówno komórki RNN jak i LSTM wydają się osiągać podobne wyniki. Zakładam, że zależy to w zdecydowanej mierze od rozmiarów słownika, rozmiaru maski/paddingu, a przede wszystkim liczby epok i samej struktury/ilości danych. Dlatego w ramach moich doświadczeń, niezależnie od zastosowanych hiperparametrów czy funkcji aktywacji, obie komórki wydają się pracować na przyzwoitym poziomie.

Technika maskowania/paddingu wydaje się być kluczowym elementem w zadaniu klasyfikacji na podstawie sekwencji słów i od odpowiedniego jej dobrania może zależeć końcowa dokładność naszego modelu. Z własnych eksperymentów mogę stwierdzić, że nie widzę intuicyjnego sposobu na ustalanie tej maski od razu, dlatego raczej należałoby ją dostrajać jak pozostałe hiperparametry.